

The IBM logo consists of the letters "IBM" in a bold, white, sans-serif font, centered within a solid black square.

Systems Reference Library

IBM 9020D and 9020E System Principles of Operation

This manual is a comprehensive presentation of the characteristics, functions, and features of the IBM 9020 D/E System.

The manual defines 9020 D/E System operating principles, computing and Input/Output Control elements, instructions, System control and monitoring facilities, branching, status switching, interruption system, and input/output operations.

Descriptions of specific input/output devices used within the 9020 D/E System appear in separate publications.

For additional information concerning units attached to the 9020D or 9020E, reference should be made to specific interface control documents.

This information is furnished in accordance with the requirements of contract FA64WA-5223, and is subject to clause 24 thereof entitled Reproduction and use of Technical Data, which provides for its use, reproduction or disclosure by the government for governmental purposes.

This publication, form number A27-2734-2, obsoletes the previous version of the manual, form number A27-2734-1, and incorporates information contained in the following Technical Newsletters:

N27-2937	July 28, 1969
GN27-2942	November 13, 1970

This major revision to the publication includes the attachment of the Direct Access Storage Facility.

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained from an IBM 1403 Printer with a 120-character chain containing upper and lower case letters, special characters, and numerals.

Address comments concerning the contents of this publication to:
IBM Corporation, Federal Systems Division, FAA Project Office,
18100 Frederick Pike, Gaithersburg, Maryland 20760

Date: June 1, 1971

JANUARY 31, 1973

TNL: GN31-3005

Table of Contents

CHAPTER 1: INTRODUCTION TO 9020D/E SYSTEMS	1
1.1 INTRODUCTION	1
1.1.1 9020D SYSTEM	1
1.1.2 9020E SYSTEM	1
1.2 COMPATIBILITY	2
1.2.1 SYSTEM/360 COMPATIBILITY	2
1.2.2 9020A COMPATIBILITY	3
1.3 COMPUTING ELEMENT	3
1.4 STORAGE ELEMENT	3
1.5 DISPLAY ELEMENT	4
1.6 SYSTEM CONSOLE AND CONFIGURATION CONSOLE	4
1.6.1 9020D SYSTEM CONSOLE	4
1.6.2 9020E CONFIGURATION CONSOLE	4
1.7 INPUT/OUTPUT CONTROL ELEMENT	5
1.8 PERIPHERAL ADAPTER MODULE	5
1.9 TAPE CONTROL UNIT	6
1.10 STORAGE CONTROL UNIT	6
1.11 DATA ADAPTER UNIT	6
1.12 SYSTEM ALERTS	6
CHAPTER 2: SYSTEM STRUCTURE	8
2.1 MAIN STORAGE	8
2.1.1 INFORMATION FORMATS	11
2.1.2 ADDRESSING	11
2.1.3 INFORMATION POSITIONING	12
2.2 DISPLAY ELEMENT	13
2.2.1 INFORMATION FORMATS	13
2.2.2 ADDRESSING	13
2.3 COMPUTING ELEMENT	14
2.3.1 GENERAL REGISTERS	15
2.3.2 FLOATING-POINT REGISTERS	15
2.4 ARITHMETIC AND LOGICAL UNIT	16
2.4.1 FIXED-POINT ARITHMETIC	16
2.4.2 DECIMAL ARITHMETIC	17
2.4.3 FLOATING-POINT ARITHMETIC	18
2.4.4 LOGICAL OPERATION	19
2.5 PROGRAM EXECUTION	23
2.5.1 INSTRUCTION FORMAT	23
2.5.2 ADDRESS GENERATION	25
2.5.3 SEQUENTIAL INSTRUCTION EXECUTION	26
2.5.4 BRANCHING	26
2.5.5 PROGRAM STATUS WORD	27
2.5.6 INTERRUPTIONS	28
2.5.6.1 Input/Output Interruption	29
2.5.6.2 Program Interruption	30
2.5.6.3 Supervisor Call Interruption	30
2.5.6.4 External Interruption	30
2.5.6.5 Machine Check Interruption	31
2.5.6.6 Priority of Interruption	31
2.5.6.7 Program States	32
2.6 PROTECTION	33
2.7 TIMER	33
2.8 DIRECT CONTROL	34
2.9 MULTIPROCESSOR OPERATION	34
2.9.1 COMPUTING ELEMENT	34
2.9.2 IOCE-PROCESSOR OPERATION	34
2.10 DISPLAY PROCESSOR OPERATION	35
2.10.1 COMPUTING ELEMENT	35
2.11 INPUT/OUTPUT	35

Table of Contents, Cont.

2.11.1	INPUT/OUTPUT DEVICES AND CONTROL UNITS	35
2.11.2	INPUT/OUTPUT INTERFACE	35
2.11.3	CHANNELS	35
2.11.4	INPUT/OUTPUT INSTRUCTIONS	36
2.11.4.1	Start I/O	36
2.11.4.2	Test Channel	37
2.11.4.3	Test I/O	37
2.11.4.4	Halt I/O	37
2.11.4.5	Set PCI	37
2.11.5	INPUT/OUTPUT OPERATION INITIATION	37
2.11.5.1	Channel Address Word	37
2.11.5.2	Channel Command Word	37
2.11.6	INPUT/OUTPUT COMMANDS	38
2.11.6.1	Read	38
2.11.6.2	Write	38
2.11.6.3	Read Backward	38
2.11.6.4	Control	39
2.11.6.5	Sense	39
2.11.6.6	Transfer In Channel	39
2.11.6.7	Search	39
2.11.7	INPUT/OUTPUT TERMINATION	39
2.11.7.1	Channel Status Word	40
2.11.8	INPUT/OUTPUT INTERRUPTIONS	40
2.12	SYSTEM CONTROL FACILITIES	40
2.12.1	CONTROL PANEL FUNCTIONS	41
2.12.1.1	Store and Display	41
2.12.1.2	Initial Program Load	41
2.12.2	OPERATOR SECTION	41
2.12.3	MAINTENANCE SECTION	42
CHAPTER 3:	FIXED-POINT ARITHMETIC	43
3.1	DATA FORMAT	43
3.2	NUMBER REPRESENTATION	44
3.3	CONDITION CODE	44
3.4	INSTRUCTION FORMAT	45
3.5	INSTRUCTIONS	46
3.5.1	LOAD	48
3.5.2	LOAD HALFWORD	48
3.5.3	LOAD AND TEST	49
3.5.4	LOAD COMPLEMENT	49
3.5.5	LOAD POSITIVE	50
3.5.6	LOAD NEGATIVE	50
3.5.7	LOAD MULTIPLE	51
3.5.8	ADD	51
3.5.9	ADD HALFWORD	52
3.5.10	ADD LOGICAL	53
3.5.11	SUBTRACT	54
3.5.12	SUBTRACT HALFWORD	54
3.5.13	SUBTRACT LOGICAL	55
3.5.14	COMPARE	56
3.5.15	COMPARE HALFWORD	57
3.5.16	MULTIPLY	57
3.5.17	MULTIPLY HALFWORD	58
3.5.18	DIVIDE	59
3.5.19	CONVERT TO BINARY	59
3.5.20	CONVERT TO DECIMAL	60
3.5.21	STORE	61
3.5.22	STORE HALFWORD	61
3.5.23	STORE MULTIPLE	62
3.5.24	SHIFT LEFT SINGLE	62
3.5.25	SHIFT RIGHT SINGLE	63
3.5.26	SHIFT LEFT DOUBLE	64
3.5.27	SHIFT RIGHT DOUBLE	64
3.6	FIXED-POINT ARITHMETIC EXCEPTIONS	65

JANUARY 31, 1973

TNL: GN31-3005

Table of Contents

CHAPTER 1: INTRODUCTION TO 9020D/E SYSTEMS	1
1.1 INTRODUCTION	1
1.1.1 9020D SYSTEM	1
1.1.2 9020E SYSTEM	1
1.2 COMPATIBILITY	2
1.2.1 SYSTEM/360 COMPATIBILITY	2
1.2.2 9020A COMPATIBILITY	3
1.3 COMPUTING ELEMENT	3
1.4 STORAGE ELEMENT	3
1.5 DISPLAY ELEMENT	4
1.6 SYSTEM CONSOLE AND CONFIGURATION CONSOLE	4
1.6.1 9020D SYSTEM CONSOLE	4
1.6.2 9020E CONFIGURATION CONSOLE	4
1.7 INPUT/OUTPUT CONTROL ELEMENT	5
1.8 PERIPHERAL ADAPTER MODULE	5
1.9 TAPE CONTROL UNIT	6
1.10 STORAGE CONTROL UNIT	6
1.11 DATA ADAPTER UNIT	6
1.12 SYSTEM ALERTS	6
CHAPTER 2: SYSTEM STRUCTURE	8
2.1 MAIN STORAGE	8
2.1.1 INFORMATION FORMATS	11
2.1.2 ADDRESSING	11
2.1.3 INFORMATION POSITIONING	12
2.2 DISPLAY ELEMENT	13
2.2.1 INFORMATION FORMATS	13
2.2.2 ADDRESSING	13
2.3 COMPUTING ELEMENT	14
2.3.1 GENERAL REGISTERS	15
2.3.2 FLOATING-POINT REGISTERS	15
2.4 ARITHMETIC AND LOGICAL UNIT	16
2.4.1 FIXED-POINT ARITHMETIC	16
2.4.2 DECIMAL ARITHMETIC	17
2.4.3 FLOATING-POINT ARITHMETIC	18
2.4.4 LOGICAL OPERATION	19
2.5 PROGRAM EXECUTION	23
2.5.1 INSTRUCTION FORMAT	23
2.5.2 ADDRESS GENERATION	25
2.5.3 SEQUENTIAL INSTRUCTION EXECUTION	26
2.5.4 BRANCHING	26
2.5.5 PROGRAM STATUS WORD	27
2.5.6 INTERRUPTIONS	28
2.5.6.1 Input/Output Interruption	29
2.5.6.2 Program Interruption	30
2.5.6.3 Supervisor Call Interruption	30
2.5.6.4 External Interruption	30
2.5.6.5 Machine Check Interruption	31
2.5.6.6 Priority of Interruption	31
2.5.6.7 Program States	32
2.6 PROTECTION	33
2.7 TIMER	33
2.8 DIRECT CONTROL	34
2.9 MULTIPROCESSOR OPERATION	34
2.9.1 COMPUTING ELEMENT	34
2.9.2 IOCE-PROCESSOR OPERATION	34
2.10 DISPLAY PROCESSOR OPERATION	35
2.10.1 COMPUTING ELEMENT	35
2.11 INPUT/OUTPUT	35

Table of Contents, Cont.

2.11.1	INPUT/OUTPUT DEVICES AND CONTROL UNITS	35
2.11.2	INPUT/OUTPUT INTERFACE	35
2.11.3	CHANNELS	35
2.11.4	INPUT/OUTPUT INSTRUCTIONS	36
2.11.4.1	Start I/O	36
2.11.4.2	Test Channel	37
2.11.4.3	Test I/O	37
2.11.4.4	Halt I/O	37
2.11.4.5	Set PCI	37
2.11.5	INPUT/OUTPUT OPERATION INITIATION	37
2.11.5.1	Channel Address Word	37
2.11.5.2	Channel Command Word	37
2.11.6	INPUT/OUTPUT COMMANDS	38
2.11.6.1	Read	38
2.11.6.2	Write	38
2.11.6.3	Read Backward	38
2.11.6.4	Control	39
2.11.6.5	Sense	39
2.11.6.6	Transfer In Channel	39
2.11.6.7	Search	39
2.11.7	INPUT/OUTPUT TERMINATION	39
2.11.7.1	Channel Status Word	40
2.11.8	INPUT/OUTPUT INTERRUPTIONS	40
2.12	SYSTEM CONTROL FACILITIES	40
2.12.1	CONTROL PANEL FUNCTIONS	41
2.12.1.1	Store and Display	41
2.12.1.2	Initial Program Load	41
2.12.2	OPERATOR SECTION	41
2.12.3	MAINTENANCE SECTION	42
CHAPTER 3: FIXED-POINT ARITHMETIC		43
3.1	DATA FORMAT	43
3.2	NUMBER REPRESENTATION	44
3.3	CONDITION CODE	44
3.4	INSTRUCTION FORMAT	45
3.5	INSTRUCTIONS	46
3.5.1	LOAD	48
3.5.2	LOAD HALFWORD	48
3.5.3	LOAD AND TEST	49
3.5.4	LOAD COMPLEMENT	49
3.5.5	LOAD POSITIVE	50
3.5.6	LOAD NEGATIVE	50
3.5.7	LOAD MULTIPLE	51
3.5.8	ADD	51
3.5.9	ADD HALFWORD	52
3.5.10	ADD LOGICAL	53
3.5.11	SUBTRACT	54
3.5.12	SUBTRACT HALFWORD	54
3.5.13	SUBTRACT LOGICAL	55
3.5.14	COMPARE	56
3.5.15	COMPARE HALFWORD	57
3.5.16	MULTIPLY	57
3.5.17	MULTIPLY HALFWORD	58
3.5.18	DIVIDE	59
3.5.19	CONVERT TO BINARY	59
3.5.20	CONVERT TO DECIMAL	60
3.5.21	STORE	61
3.5.22	STORE HALFWORD	61
3.5.23	STORE MULTIPLE	62
3.5.24	SHIFT LEFT SINGLE	62
3.5.25	SHIFT RIGHT SINGLE	63
3.5.26	SHIFT LEFT DOUBLE	64
3.5.27	SHIFT RIGHT DOUBLE	64
3.6	FIXED-POINT ARITHMETIC EXCEPTIONS	65

JANUARY 31, 1973

TNL: GN31-3005

Table of Contents, Cont.

CHAPTER 4: DECIMAL ARITHMETIC	67
4.1 DATA FORMAT	67
4.2 NUMBER REPRESENTATION	68
4.3 CONDITION CODE	68
4.4 INSTRUCTION FORMAT	69
4.5 INSTRUCTIONS	69
4.5.1 ADD DECIMAL	70
4.5.2 SUBTRACT DECIMAL	71
4.5.3 ZERO AND ADD	72
4.5.4 COMPARE DECIMAL	72
4.5.5 MULTIPLY DECIMAL	73
4.5.6 DIVIDE DECIMAL	74
4.5.7 PACK	75
4.5.8 UNPACK	76
4.5.9 MOVE WITH OFFSET	76
4.6 DECIMAL ARITHMETIC EXCEPTIONS	77
CHAPTER 5: FLOATING-POINT ARITHMETIC	79
5.1 DATA FORMAT	79
5.2 NUMBER REPRESENTATION	80
5.3 NORMALIZATION	81
5.4 CONDITION CODE	81
5.5 INSTRUCTION FORMAT	82
5.6 INSTRUCTIONS	83
5.6.1 LOAD	85
5.6.2 LOAD AND TEST	86
5.6.3 LOAD COMPLEMENT	86
5.6.4 LOAD POSITIVE	87
5.6.5 LOAD NEGATIVE	88
5.6.6 ADD NORMALIZED	89
5.6.7 ADD UNNORMALIZED	91
5.6.8 SUBTRACT NORMALIZED	92
5.6.9 SUBTRACT UNNORMALIZED	93
5.6.10 COMPARE	94
5.6.11 HALVE	95
5.6.12 MULTIPLY	96
5.6.13 DIVIDE	97
5.6.14 STORE	98
5.7 FLOATING-POINT ARITHMETIC EXCEPTIONS	99
CHAPTER 6: LOGICAL OPERATIONS	101
6.1 DATA FORMAT	101
6.2 CONDITION CODE	102
6.3 INSTRUCTION FORMAT	104
6.4 INSTRUCTIONS	105
6.4.1 MOVE	107
6.4.2 MOVE NUMERICS	107
6.4.3 MOVE ZONES	108
6.4.4 COMPARE LOGICAL	109
6.4.5 AND	110
6.4.6 OR	111
6.4.7 EXCLUSIVE OR	112
6.4.8 TEST UNDER MASK	113
6.4.9 INSERT CHARACTER	113
6.4.10 STORE CHARACTER	114
6.4.11 LOAD ADDRESS	114
6.4.12 TRANSLATE	115
6.4.13 TRANSLATE AND TEST	115
6.4.14 EDIT	116
6.4.15 EDIT AND MARK	120
6.4.16 SHIFT LEFT SINGLE LOGICAL	121
6.4.17 SHIFT RIGHT SINGLE LOGICAL	122
6.4.18 SHIFT LEFT DOUBLE LOGICAL	122
6.4.19 SHIFT RIGHT DOUBLE LOGICAL	123

Table of Contents, Cont.

6.5 LOGICAL OPERATION EXCEPTIONS123
CHAPTER 7: BRANCHING125
7.1 NORMAL SEQUENTIAL OPERATION125
7.1.1 SEQUENTIAL OPERATION EXCEPTIONS126
7.2 DECISION-MAKING127
7.3 INSTRUCTION FORMATS128
7.4 INSTRUCTIONS129
7.4.1 BRANCH ON CONDITION130
7.4.2 BRANCH AND LINK131
7.4.3 BRANCH ON COUNT131
7.4.4 BRANCH ON INDEX HIGH132
7.4.5 BRANCH ON INDEX LOW OR EQUAL133
7.4.6 EXECUTE133
7.5 EXECUTE EXCEPTIONS134
CHAPTER 8: STATUS SWITCHING AND SPECIAL 9020D/E INSTRUCTIONS138
8.1 PROGRAM STATES138
8.1.1 PROBLEM STATE138
8.1.2 WAIT STATE139
8.1.3 MASKED STATES139
8.1.4 STOPPED STATE140
8.2 STORAGE PROTECTION141
8.2.1 AREA IDENTIFICATION141
8.2.2 PROTECTION ACTION141
8.2.3 LOCATIONS PROTECTED142
8.3 PROGRAM STATUS WORD142
8.4 MULTIPLE COMPUTING ELEMENT OPERATION144
8.4.1 IDENTIFICATION OF A COMPUTING ELEMENT144
8.4.2 SHARED MAIN STORAGE144
8.4.3 CONFIGURATION CONTROL145
8.4.4 INSTRUCTION FORMATS145
8.4.5 MULTIPLE COMPUTING ELEMENT INSTRUCTIONS146
8.4.5.1 Delay147
8.4.5.2 Load Identity147
8.4.5.3 Load Preferential-Storage Base Address148
8.4.5.4 Set Configuration151
8.4.5.5 Store Preferential-Storage Base Address162
8.4.5.6 Test And Set162
8.4.5.7 Set Address Translator163
8.4.5.8 Insert Address Translator170
8.4.5.9 Move Word170
8.4.5.10 Start I/O Processor171
8.4.6 MULTIPLE-OPERATION INSTRUCTION EXCEPTIONS173
8.5 STATUS-SWITCHING INSTRUCTION FORMAT174
8.5.1 STATUS-SWITCHING174
8.5.1.1 Load PSW175
8.5.1.2 Set Program Mask176
8.5.1.3 Set System Mask176
8.5.1.4 Supervisor Call177
8.5.1.5 Set Storage Key177
8.5.1.6 Insert Storage Key178
8.5.1.7 Write Direct178
8.5.1.8 Read Direct182
8.5.1.9 Diagnose183
8.5.2 STATUS-SWITCHING EXCEPTIONS188
8.6 DISPLAY INSTRUCTION FORMATS189
8.6.1 DISPLAY INSTRUCTIONS190
8.6.1.1 Load Chain190
8.6.1.2 Convert and Sort Symbols191
8.6.1.3 Convert Weather Lines207
8.6.1.4 Repack Symbols216
8.6.2 DISPLAY INSTRUCTION EXCEPTIONS240
CHAPTER 9: INTERRUPTIONS247

JANUARY 31, 1973

TNL: GN31-3005

Table of Contents, Cont.

9.1	INTERRUPTION ACTION	247
9.1.1	INSTRUCTION EXECUTION	247
9.1.2	SOURCE IDENTIFICATION	248
9.1.3	LOCATION DETERMINATION	249
9.2	INPUT/OUTPUT INTERRUPTION	250
9.3	PROGRAM INTERRUPTION	251
9.3.1	OPERATION EXCEPTION	252
9.3.2	PRIVILEGED-OPERATION EXCEPTION	252
9.3.3	EXECUTE EXCEPTION	252
9.3.4	PROTECTION EXCEPTION	252
9.3.5	ADDRESSING EXCEPTION	252
9.3.6	SPECIFICATION EXCEPTION	252
9.3.7	DATA EXCEPTION	254
9.3.8	FIXED-POINT-OVERFLOW EXCEPTION	254
9.3.9	FIXED-POINT-DIVIDE EXCEPTION	254
9.3.10	DECIMAL-OVERFLOW EXCEPTION	254
9.3.11	DECIMAL-DIVIDE EXCEPTION	255
9.3.12	EXPONENT-OVERFLOW EXCEPTION	255
9.3.13	EXPONENT-UNDERFLOW EXCEPTION	255
9.3.14	SIGNIFICANCE EXCEPTION	255
9.3.15	FLOATING-POINT-DIVIDE EXCEPTION	255
9.3.16	PREFERENTIAL-STORAGE AREA LOCKOUT EXCEPTION	255
9.3.17	SE/DE STOPPED EXCEPTION	256
9.4	SUPERVISOR-CALL INTERRUPTION	256
9.5	EXTERNAL INTERRUPTION	256
9.5.1	TIMER	257
9.5.2	INTERRUPT SWITCH	258
9.5.3	EXTERNAL SIGNAL	258
9.5.3.1	CE Direct Control Signals	258
9.5.3.2	Abnormal Condition Signals	258
9.5.3.3	IOCE Direct Control Signals	260
9.6	MACHINE-CHECK INTERRUPTION	260
9.6.1	CE MACHINE CHECK	260
9.6.2	IOCE MACHINE CHECK	261
9.7	PRIORITY OF INTERRUPTIONS	261
9.8	INTERRUPTION EXCEPTIONS	262
	CHAPTER 10: INPUT/OUTPUT OPERATIONS	267
10.1	ATTACHMENT OF INPUT/OUTPUT DEVICES	267
10.1.1	INPUT/OUTPUT DEVICES	267
10.1.2	CONTROL UNITS	267
10.1.3	CHANNELS	268
10.1.3.1	Modes of Operation	269
10.1.3.2	Types of Channels	269
10.1.4	SYSTEM OPERATION	270
10.1.5	COMPATIBILITY OF OPERATION	272
10.2	CONTROL OF INPUT/OUTPUT DEVICES	273
10.2.1	INPUT/OUTPUT DEVICE ADDRESSING	273
10.2.2	INSTRUCTION EXCEPTION HANDLING	274
10.2.3	STATES OF THE INPUT/OUTPUT SYSTEM	274
10.2.4	RESETTING OF THE INPUT/OUTPUT SYSTEM	277
10.2.4.1	Input/Output Control Element Reset	278
10.2.4.2	Malfunction Reset	278
10.2.5	CONDITION CODE	279
10.2.6	INSTRUCTION FORMAT	281
10.2.7	INSTRUCTIONS	282
10.2.7.1	Start I/O	283
10.2.7.2	Test I/O	284
10.2.7.3	Halt I/O	286
10.2.7.4	Test Channel	288
10.2.7.5	Set PCI	289
10.3	EXECUTION OF INPUT/OUTPUT OPERATIONS	290
10.3.1	BLOCKING OF DATA	290
10.3.2	CHANNEL ADDRESS WORD	291
10.3.3	CHANNEL COMMAND WORD	291

Table of Contents, Cont.

10.3.4	COMMAND CODE293
10.3.5	DEFINITION OF STORAGE AREA294
10.3.6	CHAINING295
10.3.6.1	Data Chaining297
10.3.6.2	Command Chaining299
10.3.7	SKIPPING299
10.3.8	PROGRAM-CONTROLLED INTERRUPTION300
10.3.9	COMMANDS301
10.3.9.1	Write302
10.3.9.2	Read302
10.3.9.3	Read Backwards303
10.3.9.4	Control303
10.3.9.5	Sense304
10.3.9.6	Transfer In Channel306
10.3.9.7	Search307
10.4	TERMINATION OF INPUT/OUTPUT OPERATIONS307
10.4.1	TYPES OF TERMINATION308
10.4.1.1	Termination at Operation Initiation308
10.4.1.2	Immediate Operations308
10.4.1.3	Termination of Data Transfer309
10.4.1.4	Termination by HALT I/O310
10.4.1.5	Termination Due to Equipment Malfunction312
10.4.2	INPUT/OUTPUT INTERRUPTIONS312
10.4.2.1	Interruption Conditions312
10.4.2.2	Priority of Interruptions313
10.4.2.3	Interruption Action314
10.4.3	CHANNEL STATUS WORD315
10.4.4	UNIT STATUS CONDITIONS316
10.4.4.1	Attention316
10.4.4.2	Status Modifier316
10.4.4.3	Control Unit End317
10.4.4.4	Busy318
10.4.4.5	Channel End319
10.4.4.6	Device End319
10.4.4.7	Unit Check320
10.4.4.8	Unit Exception321
10.4.5	CHANNEL STATUS CONDITIONS321
10.4.5.1	Program Controlled Interruption321
10.4.5.2	Incorrect Length321
10.4.5.3	Program Check322
10.4.5.4	Protection check323
10.4.5.5	Channel Data Check323
10.4.5.6	Channel Control Check323
10.4.5.7	Interface Control Check324
10.4.5.8	Chaining Check324
10.4.6	CONTENT OF CHANNEL STATUS WORD324
10.4.6.1	Information Provided by Channel Status Word325
10.4.6.2	Protection Key326
10.4.6.3	Command Address326
10.4.6.4	Status326
10.4.6.5	Count329

JANUARY 31, 1973

TNL: GN31-3005

Table of Contents, Cont.

CHAPTER 11: SYSTEM CONTROL FACILITIES	331
11.1 CONTROL PANEL FUNCTIONS	331
11.1.1 STORE AND DISPLAY	331
11.1.2 INITIAL PROGRAM LOADING	332
11.1.3 PROGRAM STATUS WORD RESTART	334
11.2 OPERATOR CONTROL SECTION	334
11.2.1 EMERGENCY PULL (SC,CC)	335
11.2.2 ELEMENT MASTER POWER OFF PULL SWITCH (CE,CC)	335
11.2.3 POWER ON/OFF SWITCH (CE, SC, CC)	335
11.2.4 POWER SEQUENCE COMPLETE (CE, SC, CC)	335
11.2.5 WAIT LIGHT (CE, SC, CC)	343
11.2.6 MANUAL LIGHT (CE, SC, CC)	343
11.2.7 SYSTEM LIGHT (CE)	343
11.2.8 TEST LIGHT (CE)	343
11.2.9 LOAD LIGHT (CE, SC, CC)	343
11.2.10 LOAD-UNIT SWITCHES (CE, SC, CC)	343
11.2.11 LOAD SWITCH (CE, SC, CC)	344
11.2.12 MAIN STORAGE SELECT SWITCH (CE, SC, CC)	344
11.2.13 SYSTEM INTERLOCK SWITCH (CE, SC, CC)	344
11.2.14 COMPUTING ELEMENT SELECT (SC, CC)	344
11.2.15 STOP SWITCH (CE, SC, CC)	346
11.2.16 RATE SWITCH (CE, SC, CC)	346
11.2.17 START SWITCH (CE, SC, CC)	347
11.2.18 STORAGE-SELECT SWITCH (CE, SC, CC)	347
11.2.19 ADDRESS SWITCHES (CE, SC, CC)	347
11.2.20 DATA SWITCHES (CE, SC, CC)	347
11.2.21 STORE SWITCH (CE, SC, CC)	348
11.2.22 DISPLAY SWITCH (CE, SC, CC)	348
11.2.23 SET INSTRUCTION COUNTER SWITCH (CE, SC, CC)	348
11.2.24 INTERRUPT SWITCH (CE, SC, CC)	348
11.2.25 PROGRAM STATUS WORD RESTART SWITCH (CE)	349
11.2.26 STATE THREE (CE, SC, CC)	349
11.2.27 STATE TWO (CE, SC, CC)	349
11.2.28 STATE ONE (CE, SC, CC)	349
11.2.29 STATE ZERO (CE, SC, CC)	349
11.2.30 LAMP TEST (CE,SC,CC)	349
11.2.31 CONTROL COMPUTING ELEMENT SWITCHES (SC, CC)	349
11.2.32 CONTROL COMPUTING ELEMENT ACTIVATE SWITCH (SC, CC)	349
11.2.33 ADDRESS-COMPARE SWITCHES (CE, SC, CC)	349
11.2.35 SENSE SWITCHES (SC, CC)	350
11.2.36 READER/PUNCH-PRINTER SELECT SWITCH (SC, CC)	350
11.2.37 READER/PUNCH-PRINTER ENABLE SWITCH-INDICATOR (SC, CC)	350
11.2.38 PRINTER KEYBOARD SELECT SWITCH (SC)	351
11.2.39 PRINTER KEYBOARD ENABLE SWITCH-INDICATOR (SC)	351
11.2.40 CONSOLE RESET SWITCH (SC, CC)	351
11.2.41 BELL RESET SWITCH (SC)	351
11.2.42 BUZZER RESET SWITCH (SC)	351
11.2.43 ALL STOP SWITCH (SC, CC)	351
11.2.44 INVALID SELECTION INDICATOR (SC, CC)	351
11.2.45 TEST MODE INDICATOR (SC, CC)	351
11.2.46 360-MODE SWITCH-INDICATOR (CE)	352
11.2.47 ENABLE SYSTEM IPL (CC)	352
11.2.48 I/O INTERFACE ENABLE/DISABLE SWITCH (SC,CC)	352
11.2.49 DISABLED INDICATOR (SC,CC)	352
11.2.50 FAULT RESET (CC)	352
11.2.51 BUZZER/BELL RESET (CC)	352
11.3 MAINTENANCE SECTION	352
CHAPTER 12 -- SYSTEM MONITORING	354

Table of Contents, Cont.

- 12.1 ELEMENT OPERATIONS 354
 - 12.1.1 CE OPERATION 354
 - 12.1.2 IOCE OPERATION 354
 - 12.1.2.1 Common Logic Unit (CLU) 354
 - 12.1.2.2 Maintenance and Channel Storage (MACH) 354
 - 12.1.2.3 Multiplexor Channel 355
 - 12.1.2.4 Selector Channel 355
 - 12.1.3 SE OPERATION 355
 - 12.1.4 DE OPERATION 355
 - 12.1.5 CHECK STOP 355
 - 12.1.6 RESETS 356
- 12.2 RECORDING AND HANDLING CHECK CONDITIONS 359
 - 12.2.1 RECORDING CHECK CONDITIONS 359
 - 12.2.2 SIGNALING CHECK CONDITIONS 359
 - 12.2.3 PRESERVATION OF CHECK DATA 362
- 12.3 ELEMENT CHECKING SUMMARY TABLES 364
- 12.4 COMPUTING ELEMENT CHECKING 373
 - 12.4.1 CE MACHINE-CHECK HANDLING 373
 - 12.4.2 INTERNAL MACHINE-CHECKS 374
 - 12.4.2.1 Out-of-Tolerance and On Battery Supply Checks 374
 - 12.4.2.2 CCR Parity Check 374
 - 12.4.2.3 Fetch Data Check 374
 - 12.4.2.4 Local Store Bus Check 374
 - 12.4.2.5 Log ROS Check 374
 - 12.4.2.6 Log ADR Check 375
 - 12.4.2.7 ATR Parity Check 375
 - 12.4.2.8 PSBAR Parity Check 375
 - 12.4.2.9 SAB Check 375
 - 12.4.2.10 SDBI Check 375
 - 12.4.3 EXTERNAL MACHINE-CHECKS 375
 - 12.4.3.1 Data Check 375
 - 12.4.3.2 Address Check 375
 - 12.4.3.3 Storage Timeout Check 376
 - 12.4.3.4 Read Direct Timeout 376
 - 12.4.3.5 IOCE Bus Check 376
 - 12.4.3.6 SE/DE Logout-Stop Check (LOS) 376
 - 12.4.3.8 PSBAR Not Configured 376
 - 12.4.3.9 PSBAR Alt. Check 377
 - 12.4.3.10 Split Logout 377
 - 12.4.3.11 Storage ID 377
- 12.5 INPUT-OUTPUT CONTROL ELEMENT CHECKING 377
 - 12.5.1 IOCE MACHINE-CHECK HANDLING 377
 - 12.5.2 INTERNAL MACHINE-CHECKS 378
 - 12.5.2.1 Out-of-Tolerance and On Battery Supply Checks 378
 - 12.5.2.2 CCR Parity Check 378
 - 12.5.2.3 Fetch Data Check 378
 - 12.5.2.4 Invalid ROS Address Check 378
 - 12.5.2.5 Log ROS Check 379
 - 12.5.2.6 Log ADR Check 379
 - 12.5.2.7 Multiplexor Interface Parity Check 379
 - 12.5.2.8 IOCE Bus Check 379
 - 12.5.2.9 ATR Parity Check 379
 - 12.5.3 EXTERNAL MACHINE-CHECKS 379
 - 12.5.3.1 Storage Check 379
 - 12.5.3.2 Storage Timeout Check 379
 - 12.5.3.3 SE Logout-Stop Check 380
 - 12.5.3.4 CE Log Request Check 380
 - 12.5.3.5 PSA Lockout Check 380
- 12.6 STORAGE ELEMENT CHECKING 383
 - 12.6.1 SE MACHINE-CHECK HANDLING 383
 - 12.6.2 SE LOGWORD 5 and LOGWORD 11 384
 - 12.6.2.1 Tag Mismatch 384
 - 12.6.2.2 Normal Op Check 384
 - 12.6.2.3 Storage Protect Address or Key Parity Check 384
 - 12.6.2.4 Address Parity Check 384

JANUARY 31, 1973

TNL: GN31-3005

Table of Contents, Cont.

12.6.2.5	Mark Parity Check384
12.6.2.6	Multi Accept Check384
12.6.2.7	Tag Parity385
12.6.2.8	Double Cycle Timeout385
12.6.2.9	IOCE SBI Check385
12.6.2.10	Data Parity Check385
12.6.2.11	Thermal Condition385
12.6.2.12	On-Battery Supply (OBS)385
12.6.2.13	CCR Parity Check385
12.6.2.14	Logout Stop Condition385
12.6.2.15	Power Off385
12.7	DISPLAY ELEMENT CHECKING385
12.7.1	DE MACHINE-CHECK HANDLING385
12.7.2	DE LOGWORD 3, LOGWORD 7, and LOGWORD 10386
12.7.2.1	Tag Mismatch386
12.7.2.2	Normal Op Check386
12.7.2.3	Storage Protect Address Parity386
12.7.2.4	Storage Protect Key Parity386
12.7.2.5	Address Parity387
12.7.2.6	Mark Parity387
12.7.2.7	Multi-Accept387
12.7.2.8	Tag Parity387
12.7.2.9	Data Parity387
12.7.2.10	Local Storage Parity388
12.7.2.11	DG Data Register Parity388
12.7.2.12	Thermal Condition388
12.7.2.13	On-Battery Supply388
12.7.2.14	CCR Parity388
12.7.2.15	Logout-Stop Check388
12.7.2.16	Power-Off Condition388
12.7.2.17	Invalid DG Configuration388
12.7.2.18	Refresh Parity Check389
12.8	DIAGNOSTIC LOGOUT OF THE CE, IOCE, SE AND DE389
12.8.1	CE Logout389
12.8.1.1	Logout Initiated by Another CE389
12.8.1.2	Logout Initiated by an Address or Data Check389
12.8.1.3	Logout Initiated by a Storage Timeout Check389
12.8.1.4	Logout Initiated by an Internal Check389
12.8.2	IOCE LOGOUT390
12.8.2.1	CLU Logout390
12.8.2.2	Check Handling During CLU Logout391
12.8.2.3	Selector Channel Logout391
12.8.2.4	Check Handling During Selector Channel Logout391
12.8.3	SE LOGOUT391
12.8.3.1	LOS from CE391
12.8.3.2	LOS from IOCE392
12.8.4	DE LOGOUT392
12.8.4.1	LOS from CE392
12.8.4.2	LOS from DG Access Parity Check393
12.9	COMPUTING ELEMENT LOGWORD FORMATS393
12.9.1	CHECK REGISTER LOGWORD FORMATS393
12.9.2	GENERAL LOGWORD FORMATS393
12.10	INPUT/OUTPUT CONTROL ELEMENT LOGWORD FORMATS403
12.10.1	CHECK-CONDITION LOGWORD FORMATS403
12.10.2	GENERAL LOGWORD FORMATS405
12.11	STORAGE ELEMENT LOGWORD FORMATS413
12.11.1	SE LOGWORDS413
12.12	DISPLAY STORAGE LOGWORD FORMATS416
12.12.1	DE LOGWORDS416
CHAPTER 13:	INSTRUCTION TIMING422
13.1	FIXED-POINT ARITHMETIC INSTRUCTIONS425
13.2	SHIFT INSTRUCTIONS426
13.3	FLOATING-POINT ARITHMETIC INSTRUCTIONS427
13.4	LOGICAL INSTRUCTIONS428

Table of Contents, Cont.

13.5 DECIMAL ARITHMETIC INSTRUCTIONS429
13.6 BRANCHING AND STATUS-SWITCHING INSTRUCTIONS430
13.7 DETAILED VARIABLE-FIELD-LENGTH INSTRUCTION TIMING430
13.7.1 ZERO AND ADD431
13.7.2 ADD DECIMAL AND SUBTRACT DECIMAL431
13.7.3 COMPARE DECIMAL432
13.7.4 MOVE WITH OFFSET432
13.7.5 PACK432
13.7.6 UNPACK432
13.7.7 MOVE433
13.7.8 MOVE NUMERIC, MOVE ZONE, AND, OR, AND EXCLUSIVE OR433
13.7.9 TRANSLATE433
13.7.10 TRANSLATE AND TEST434
13.7.11 EDIT AND EDIT AND MARK434
13.7.12 COMPARE LOGICAL434
13.8 INPUT/OUTPUT INSTRUCTIONS434
13.9 MULTIPLE COMPUTING ELEMENT INSTRUCTIONS435
13.10 DISPLAY INSTRUCTIONS435
13.11 MAINTENANCE INSTRUCTIONS437
13.12 IOCE-CE TO SE COMMUNICATIONS437
13.13 CE TO DE COMMUNICATIONS438
APPENDIX A. FORMATS AND TABLES440
APPENDIX B. INSTRUCTION USE EXAMPLES510
APPENDIX C. FIXED-POINT AND TWO'S-COMPLEMENT NOTATION526
APPENDIX D. FLOATING-POINT ARITHMETIC527
APPENDIX E. POWERS OF TWO TABLE529
APPENDIX F. HEXADECIMAL-DECIMAL CONVERSION TABLE530
APPENDIX G. SYSTEM/360 MODE OF OPERATION535
APPENDIX H: IOCE-PROCESSOR547
1.0 GENERAL CONCEPTS547
1.1 IOCE-CHANNEL CONTROLLER547
1.2 IOCE-PROCESSOR547
1.3 COMMON LOGIC UNIT (CLU)549
1.3.1 PRIORITY CONTROL549
1.3.2 STATUS CONTROL AND ERROR HANDLING550
1.4 OTHER FUNCTIONS550
1.5 MODE CONTROL550
2.0 CE OPERATION550
2.1 NEW OR EXTENDED CE INSTRUCTIONS550
2.1.1 WRITE DIRECT550
2.1.2 MOVE WORD550
2.1.3 START I/O PROCESSOR551
2.2 EXTERNAL INTERRUPTION ACTION551
2.3 DIAGNOSTIC KERNEL551
3.0 IOCE - PROCESSOR OPERATION551
3.1 ADDRESS GENERATION551
3.2 PROGRAM STATES IN THE IOCE-PROCESSOR552
3.2.1 PROBLEM/SUPERVISOR STATE552
3.2.2 STOPPED/OPERATING STATE553
3.2.3 WAIT/RUNNING STATE553
3.2.4 MASKED/INTERRUPTABLE STATE553
3.3 STORAGE PROTECTION554
3.4 PROGRAM STATUS WORD IN THE IOCE-PROCESSOR554
3.5 INTERRUPTIONS IN THE IOCE-PROCESSOR556
3.5.1 INTERRUPTION ACTION556
3.5.2 PROGRAM INTERRUPTIONS557
3.5.3 MACHINE-CHECK INTERRUPTION558

JANUARY 31, 1973

TNL: GN31-3005

Table of Contents, Cont.

3.5.4 EXTERNAL INTERRUPTIONS	558
3.5.4.1 Timer	558
3.5.4.2 Controlling CE	558
3.6 ADDITIONAL IOCE INSTRUCTIONS	558
3.6.1 MOVE WORD	558
3.6.2 LOAD DATA ADDRESS	558
3.6.3 LOAD IDENTITY	559
3.6.4 WRITE DIRECT	559
3.6.5 DELAY	560
3.6.6 TEST AND SET	560
4.0 IOCE-CHANNEL CONTROLLER OPERATION	561
4.1 CHANNEL ARCHITECTURE CHANGES	561
4.2 CHANNEL ADDRESS WORD	561
4.3 CHANNEL COMMAND WORD	561
4.4 CHANNEL STATUS WORD	562
5.0 SYSTEM MONITORING CONSIDERATIONS FOR IOCE-PROCESSOR	562
5.1 RESETS	562
5.2 RECORDING AND HANDLING CHECK CONDITIONS	563
5.2.1 RECORDING CHECK CONDITIONS	563
5.2.2 SIGNALING CHECK CONDITIONS	563
5.2.3 IOCE MACHINE-CHECK HANDLING	563
5.3 PRESERVATION OF CHECK DATA	563
5.3.1 CE LOGWORD FORMATS	563
5.3.2 IOCE LOGWORD FORMATS	563
6.0 SYSTEM/360 MODE OF OPERATION	564
6.1 CE INSTRUCTION REPERTOIRE	564
6.2 IOCE-OPERATION	564
 APPENDIX I. IBM 7289-04 DISPLAY ELEMENT	 579
1.1 Introduction	579
2.1 System Characteristics	579
3.1 Data Organization	580
3.1.1 Data Management	580
3.1.2 Display Image	581
3.1.3 Data Formats	582
3.1.4 Commands	582
3.1.4.1 Start of Display Command	582
3.1.4.2 End of Display Command	583
3.1.4.3 Transfer Command	583
3.1.5 Address Wrap-Around	584
3.1.6 PVD Test Word	584
4.1 Operational Characteristics	584
4.1.1 Image Initialization	586
4.1.2 Test and Set	587
4.1.3 Regeneration Control	587
4.1.4 Regeneration Cycle	588
 APPENDIX J: CHANNEL-TO-CHANNEL ADAPTER	 589
1.1 Introduction	589
2.1 General Operation	589
2.1.1 Commands	590
2.1.2 Adapter Ready Condition	591
2.1.3 Status Indications	591
2.1.4 Sense Indications	593
2.1.5 Unusual Conditions	593
3.1 Operations	594
3.1.1 Channel Command Word	594
3.1.2 Commands	595
3.1.2.1 Control	595
3.1.2.2 Sense Command Byte	595
3.1.2.3 Sense Adapter	601
3.1.2.4 Read and Read Backward	601
3.1.2.5 Write	606
3.1.2.6 Write End of File	608
3.1.3 Effect of Test I/O and Halt I/O	610

Table of Contents, Cont.

- 3.1.3.1 Test I/O610
- 3.1.3.2 Halt I/O610
- 3.1.4 Selective Reset (Malfunction Reset)612
- 3.1.5 System Reset612

- APPENDIX K. IBM 2701-01 DATA ADAPTER UNIT614
- 1.1 Introduction614
- 2.1 System Characteristics614
- 2.1.1 System Configuration614
- 2.1.2 DAU Communications614
- 2.1.2.1 Commands614
- 2.1.2.2 Data616
- 2.1.2.3 Chaining616
- 2.1.2.4 Mode of Transfer616
- 3.1 Operational Characteristics616
- 3.1.1 Addressing617
- 3.1.2 Commands617
- 3.1.2.1 Read Type Commands617
- 3.1.2.2 Write Type Commands617
- 3.1.2.3 Control Type Commands618
- 3.1.2.4 Status Seeking Commands618
- 3.1.2.5 Interface Control Commands618
- 3.1.2.6 Command Formatting618
- 3.1.3 Status618
- 3.1.3.1 RKM Status618
- 3.1.3.2 DAU Status623
- 3.1.4 Sense624
- 3.1.4.1 RKM Sense Bits624
- 3.1.4.2 DAU Sense Bits624
- 3.1.5 Interrupts625
- 3.1.5.1 Interrupt In A625
- 3.1.5.2 Interrupt In B625
- 4.1 Reconfiguration625
- 5.1 Program Maintenance Aids626
- 5.1.1 Command Checking626

JANUARY 31, 1973

TNL: GN31-3005

Abbreviated Table of Contents

CHAPTER 1: INTRODUCTION TO 9020D/E SYSTEMS	1
1.1 INTRODUCTION	1
1.2 COMPATIBILITY	2
1.3 COMPUTING ELEMENT	3
1.4 STORAGE ELEMENT	3
1.5 DISPLAY ELEMENT	4
1.6 SYSTEM CONSOLE AND CONFIGURATION CONSOLE	4
1.7 INPUT/OUTPUT CONTROL ELEMENT	5
1.8 PERIPHERAL ADAPTER MODULE	5
1.9 TAPE CONTROL UNIT	6
1.10 STORAGE CONTROL UNIT	6
1.11 DATA ADAPTER UNIT	6
1.12 SYSTEM ALERTS	6
CHAPTER 2: SYSTEM STRUCTURE	8
2.1 MAIN STORAGE	8
2.2 DISPLAY ELEMENT	13
2.3 COMPUTING ELEMENT	14
2.4 ARITHMETIC AND LOGICAL UNIT	16
2.5 PROGRAM EXECUTION	23
2.6 PROTECTION	33
2.7 TIMER	33
2.8 DIRECT CONTROL	34
2.9 MULTIPROCESSOR OPERATION	34
2.10 DISPLAY PROCESSOR OPERATION	35
2.11 INPUT/OUTPUT	35
2.12 SYSTEM CONTROL FACILITIES	40
CHAPTER 3: FIXED-POINT ARITHMETIC	43
3.1 DATA FORMAT	43
3.2 NUMBER REPRESENTATION	44
3.3 CONDITION CODE	44
3.4 INSTRUCTION FORMAT	45
3.5 INSTRUCTIONS	46
3.6 FIXED-POINT ARITHMETIC EXCEPTIONS	65
CHAPTER 4: DECIMAL ARITHMETIC	67
4.1 DATA FORMAT	67
4.2 NUMBER REPRESENTATION	68
4.3 CONDITION CODE	68
4.4 INSTRUCTION FORMAT	69
4.5 INSTRUCTIONS	69
4.6 DECIMAL ARITHMETIC EXCEPTIONS	77
CHAPTER 5: FLOATING-POINT ARITHMETIC	79
5.1 DATA FORMAT	79
5.2 NUMBER REPRESENTATION	80
5.3 NORMALIZATION	81
5.4 CONDITION CODE	81
5.5 INSTRUCTION FORMAT	82
5.6 INSTRUCTIONS	83
5.7 FLOATING-POINT ARITHMETIC EXCEPTIONS	99
CHAPTER 6: LOGICAL OPERATIONS	101
6.1 DATA FORMAT	101
6.2 CONDITION CODE	102
6.3 INSTRUCTION FORMAT	104
6.4 INSTRUCTIONS	105
6.5 LOGICAL OPERATION EXCEPTIONS	123

Abbreviated Table of Contents, Cont.

CHAPTER 7: BRANCHING125
7.1 NORMAL SEQUENTIAL OPERATION125
7.2 DECISION-MAKING127
7.3 INSTRUCTION FORMATS128
7.4 INSTRUCTIONS129
7.5 EXECUTE EXCEPTIONS134

CHAPTER 8: STATUS SWITCHING AND SPECIAL 9020D/E INSTRUCTIONS138
8.1 PROGRAM STATES138
8.2 STORAGE PROTECTION141
8.3 PROGRAM STATUS WORD142
8.4 MULTIPLE COMPUTING ELEMENT OPERATION144
8.5 STATUS-SWITCHING INSTRUCTION FORMAT174
8.6 DISPLAY INSTRUCTION FORMATS189

CHAPTER 9: INTERRUPTIONS247
9.1 INTERRUPTION ACTION247
9.2 INPUT/OUTPUT INTERRUPTION250
9.3 PROGRAM INTERRUPTION251
9.4 SUPERVISOR-CALL INTERRUPTION256
9.5 EXTERNAL INTERRUPTION256
9.6 MACHINE-CHECK INTERRUPTION260
9.7 PRIORITY OF INTERRUPTIONS261
9.8 INTERRUPTION EXCEPTIONS262

CHAPTER 10: INPUT/OUTPUT OPERATIONS267
10.1 ATTACHMENT OF INPUT/OUTPUT DEVICES267
10.2 CONTROL OF INPUT/OUTPUT DEVICES273
10.3 EXECUTION OF INPUT/OUTPUT OPERATIONS290
10.4 TERMINATION OF INPUT/OUTPUT OPERATIONS307

JANUARY 31, 1973

TNL: GN31-3005

Abbreviated Table of Contents, Cont.

CHAPTER 11: SYSTEM CONTROL FACILITIES	331
11.1 CONTROL PANEL FUNCTIONS	331
11.2 OPERATOR CONTROL SECTION	334
11.3 MAINTENANCE SECTION	352
CHAPTER 12 -- SYSTEM MONITORING	354
12.1 ELEMENT OPERATIONS	354
12.2 RECORDING AND HANDLING CHECK CONDITIONS	359
12.3 ELEMENT CHECKING SUMMARY TABLES	364
12.4 COMPUTING ELEMENT CHECKING	373
12.5 INPUT-OUTPUT CONTROL ELEMENT CHECKING	377
12.6 STORAGE ELEMENT CHECKING	383
12.7 DISPLAY ELEMENT CHECKING	385
12.8 DIAGNOSTIC LOGOUT OF THE CE, IOCE, SE AND DE	389
12.9 COMPUTING ELEMENT LOGWORD FORMATS	393
12.10 INPUT/OUTPUT CONTROL ELEMENT LOGWORD FORMATS	403
12.11 STORAGE ELEMENT LOGWORD FORMATS	413
12.12 DISPLAY STORAGE LOGWORD FORMATS	416
CHAPTER 13: INSTRUCTION TIMING	422
13.1 FIXED-POINT ARITHMETIC INSTRUCTIONS	425
13.2 SHIFT INSTRUCTIONS	426
13.3 FLOATING-POINT ARITHMETIC INSTRUCTIONS	427
13.4 LOGICAL INSTRUCTIONS	428
13.5 DECIMAL ARITHMETIC INSTRUCTIONS	429
13.6 BRANCHING AND STATUS-SWITCHING INSTRUCTIONS	430
13.7 DETAILED VARIABLE-FIELD-LENGTH INSTRUCTION TIMING	430
13.8 INPUT/OUTPUT INSTRUCTIONS	434
13.9 MULTIPLE COMPUTING ELEMENT INSTRUCTIONS	435
13.10 DISPLAY INSTRUCTIONS	435
13.11 MAINTENANCE INSTRUCTIONS	437
13.12 IOCE-CE TO SE COMMUNICATIONS	437
13.13 CE TO DE COMMUNICATIONS	438
APPENDIX A. FORMATS AND TABLES	440
APPENDIX B. INSTRUCTION USE EXAMPLES	510
APPENDIX C. FIXED-POINT AND TWO'S-COMPLEMENT NOTATION	526
APPENDIX D. FLOATING-POINT ARITHMETIC	527
APPENDIX E. POWERS OF TWO TABLE	529
APPENDIX F. HEXADECIMAL-DECIMAL CONVERSION TABLE	530
APPENDIX G. SYSTEM/360 MODE OF OPERATION	535
APPENDIX H: IOCE-PROCESSOR	547
1.0 GENERAL CONCEPTS	547
1.1 IOCE-CHANNEL CONTROLLER	547
1.2 IOCE-PROCESSOR	547
1.3 COMMON LOGIC UNIT (CLU)	549
1.4 OTHER FUNCTIONS	550
1.5 MODE CONTROL	550
2.0 CE OPERATION	550
2.1 NEW OR EXTENDED CE INSTRUCTIONS	550
2.2 EXTERNAL INTERRUPTION ACTION	551
2.3 DIAGNOSTIC KERNEL	551

Abbreviated Table of Contents, Cont.

3.0 IOCE - PROCESSOR OPERATION551

3.1 ADDRESS GENERATION551

3.2 PROGRAM STATES IN THE IOCE-PROCESSOR552

3.3 STORAGE PROTECTION554

3.4 PROGRAM STATUS WORD IN THE IOCE-PROCESSOR554

3.5 INTERRUPTIONS IN THE IOCE-PROCESSOR556

3.6 ADDITIONAL IOCE INSTRUCTIONS558

4.0 IOCE-CHANNEL CONTROLLER OPERATION561

4.1 CHANNEL ARCHITECTURE CHANGES561

4.2 CHANNEL ADDRESS WORD561

4.3 CHANNEL COMMAND WORD561

4.4 CHANNEL STATUS WORD562

5.0 SYSTEM MONITORING CONSIDERATIONS FOR IOCE-PROCESSOR562

5.1 RESETS562

5.2 RECORDING AND HANDLING CHECK CONDITIONS563

5.3 PRESERVATION OF CHECK DATA563

6.0 SYSTEM/360 MODE OF OPERATION564

6.1 CE INSTRUCTION REPERTOIRE564

6.2 IOCE-OPERATION564

APPENDIX I. IBM 7289-04 DISPLAY ELEMENT579

1.1 Introduction579

2.1 System Characteristics579

3.1 Data Organization580

4.1 Operational Characteristics584

APPENDIX J: CHANNEL-TO-CHANNEL ADAPTER589

1.1 Introduction589

2.1 General Operation589

3.1 Operations594

APPENDIX K. IBM 2701-01 DATA ADAPTER UNIT614

1.1 Introduction614

2.1 System Characteristics614

3.1 Operational Characteristics616

4.1 Reconfiguration625

5.1 Program Maintenance Aids626

JANUARY 31, 1973

TNL: GN31-3005

Location of Figures

Figure 2-1 IBM 9020D CCC System Basic Logical Structure	9
Figure 2-2 IBM 9020E DCP System Basic Logical Structure	10
Figure 2-3 Sample Information Formats	12
Figure 2-4 Integral Boundaries for Halfwords, Words, and Double Words	14
Figure 2-5 Computing Element	15
Figure 2-6 General and Floating-Point Registers	16
Figure 2-7 Fixed-Point Number Formats	17
Figure 2-8 Bit Codes for Digits and Signs	18
Figure 2-9 Packed Decimal Number Format	18
Figure 2-10 Zoned Decimal Number Format	18
Figure 2-11 Short and Long Floating-Point Number Formats	19
Figure 2-12 Fixed-Length Logical Information	20
Figure 2-13 Variable-Length Logical Information	20
Figure 2-14 Extended Binary Coded Decimal Interchange Code	21
Figure 2-15 United States of America Standard Code for Information Interchange	22
Figure 2-16 Five Basic Instruction Formats	24
Figure 2-17 Program Status Word Format	28
Figure 2-18 Preferential Storage Assignments	29
Figure 2-19 Interruption Code for Program Interruption	30
Figure 2-20 Interruption Code for External Interruption	31
Figure 2-21 Interruption Code for Machine Check Interruption	31
Figure 2-22 Channel Address Word Format	37
Figure 2-23 Channel Command Word Format	38
Figure 2-24 Relationship of I/O Instructions, Commands, and Orders	39
Figure 2-25 Channel Status Word Format	40
Figure 8-1 Load Preferential Storage Base Address Register	149
Figure 8-2 9020D/E Configuration and Selection Mask Formats and Configuration Control Register Contents	153
Figure 8-3 A 9020D Three-Subsystem Configuration	159
Figure 8-4 9020D/E Address Translation	165
Figure 8-5 9020D/E Maintenance Control Word (MCW)	184
Figure 8-6 Input Formats for Primary Radar/Single Symbols and Beacon	193
Figure 8-7 Output Format for CSS	193
Figure 8-8 Input Operand Register Assignment for Primary Radar/Single Symbol Data in CSS	195
Figure 8-9 Input Operand Register Assignment for Beacon Data in CSS	196
Figure 8-10 Bin Addressing	203
Figure 8-11 Input and Output Formats for Weather Line Data	207
Figure 8-12 Input Operand Register Assignment for CVWL	208
Figure 8-13 Example of a descriptor table for PVD Y sort bin X	217
Figure 8-14 Input Operand and Address Assignment for RPSB	218
Figure 8-15 Work Control Table	225
Figure 8-16 Major Position and Single Symbol Output Format	234
Figure 8-17 CSS Operation Flowchart	241
Figure 8-18 CVWL Operation Flowchart	243
Figure 8-19 RPSB Operation Flowchart	245

Location of Figures, Cont.

Figure 11-1 Computing Element (CE) Control Panel (1 of 1)337

Figure 11-2 9020D System Console (SC) Panel341

Figure 11-3 9020E Configuration Console (CC) Panel342

Figure 12-1 IBM 9020D/E System Reset Schematic358

Figure 12-2 IBM 9020D/E Subsystem Reset Schematic359

Figure 12-3 CE Logword Formats397

Figure 12-4 IOCE Logword Formats407

Figure 12-5 SE Logword Format417

Figure 12-6 DE Logword Format419

Figure 13-1 Physical Placement - Cable Delay Relationship423

Figure H-1 IOCE Block Diagram548

FIGURE H-2 -- IOCE-Processor PSW Format (Operational Mode) . . .555

Figure I-1 Display Element Configuration580

Figure I-2 Display Element Organization581

Figure I-3 Display Image582

Figure I-4 Display Element Control Area585

Figure I-5 Use of Image Initialization Area586

Figure I-6 Test and Set Byte587

Figure I-7 Regeneration Cycle588

Figure J-1 Data Flow Through a Channel-to-Channel Adapter590

Figure J-2 No Operation Command599

Figure J-3 Control Command600

Figure J-4 Sense Command Byte602

Figure J-5 Sense Adapter Command603

Figure J-6 Read or Read Backward Command605

Figure J-7 Write Command607

Figure J-8 Write End of File Command609

Figure J-9 Test I/O Instruction611

Figure J-10 Halt I/O, Selective Reset, and System Reset613

Figure K-1 DAU-RKM System Configuration615

JANUARY 31, 1973

TNL: GN31-3005

Location of Tables

TABLE 2-I	INSTRUCTION LENGTH RECORDING	24
TABLE 3-I	CONDITION CODE SETTING FOR FIXED-POINT ARITHMETIC	45
TABLE 3-II	FIXED-POINT INSTRUCTIONS	47
TABLE 4-I	CONDITION CODE SETTING FOR DECIMAL ARITHMETIC	69
TABLE 4-II	DECIMAL INSTRUCTIONS	70
TABLE 5-I	CONDITION CODE SETTING FOR FLOATING-POINT ARITHMETIC	82
TABLE 5-II	FLOATING-POINT INSTRUCTIONS	84
TABLE 6-I	CONDITION CODE SETTING FOR LOGICAL OPERATIONS	103
TABLE 6-II	LOGICAL INSTRUCTIONS	106
TABLE 6-III	EDITING	120
TABLE 7-I	BRANCHING INSTRUCTIONS	129
TABLE 7-II	COMPLETE TABLE OF CONDITION CODE SETTINGS	135
TABLE 8-I	SYSTEM MASK ASSIGNMENT	143
TABLE 8-II	PROGRAM MASK ASSIGNMENT	144
TABLE 8-III	MULTIPLE COMPUTING ELEMENT INSTRUCTIONS	147
TABLE 8-IV	SUMMARY OF FUNCTIONAL STATE DEFINITIONS FOR A COMPUTING ELEMENT	156
TABLE 8-V	RANGE OF ADDRESSES	166
TABLE 8-VI	ADDRESS TRANSLATION REGISTER SLOT SELECTION	167
TABLE 8-VII	STORAGE/DISPLAY MODULE IDENTIFIERS	168
TABLE 8-VIII	STATUS-SWITCHING INSTRUCTIONS	175
TABLE 8-IX	WRITE DIRECT - IMMEDIATE FIELD	179
TABLE 8-X	9020D/E CE DIAGNOSE KERNELS (OPERATIONAL - ANY STATE)	186
TABLE 8-XI	DISPLAY INSTRUCTIONS	190
TABLE 8-XII	CONVERSION MULTIPLIERS	199
TABLE 9-I	PREFERENTIAL/STORAGE ASSIGNMENTS	249
TABLE 9-II	INSTRUCTION LENGTH RECORDING	250
TABLE 9-III	INTERRUPTION ACTION	263
TABLE 9-IV	ABNORMAL CONDITION INTERRUPTION ACTION	265
TABLE 10-I	INPUT/OUTPUT ADDRESS ASSIGNMENTS	273
TABLE 10-II	INPUT/OUTPUT STATES	275
TABLE 10-III	CONDITION CODE SETTING FOR INPUT/OUTPUT INSTRUCTIONS	280
TABLE 10-IV	INPUT/OUTPUT INSTRUCTIONS	282
TABLE 10-V	COMMAND-CODE ASSIGNMENT	294
TABLE 10-VI	ACTION IN CHANNEL UPON EXHAUSTION OF COUNT OR RECEIPT OF CHANNEL END	296
TABLE 10-VII	INPUT/OUTPUT COMMANDS	301
TABLE 10-VIII	SENSE BIT DESIGNATION	305
TABLE 10-IX	STATUS BIT DESIGNATION	316
TABLE 10-X	INDICATION OF THE BUSY CONDITION IN CHANNEL STATUS WORD	318
TABLE 10-XI	CONTENT OF CHANNEL STATUS WORD ADDRESS FIELD	327
TABLE 10-XII	TIME AND METHOD OF CREATING AND STORING STATUS INFORMATION	328
TABLE 10-XIII	CONTENT OF CHANNEL STATUS WORD COUNT FIELD	329

Location of Tables, Cont.

TABLE 11-I CONTROLS AND LIGHTS ON OPERATOR CONTROL SECTION336
TABLE 11-II STATES OF WAIT, MANUAL, AND SYSTEM LIGHTS343
TABLE 11-III STATE DEPENDENT OPERATOR SWITCHES345
TABLE 12-I ABNORMAL-CONDITION SIGNALS SUMMARY361
TABLE 12-II IBM 9020 SYSTEM PREFERENTIAL-STORAGE AREA ASSIGNMENT363
TABLE 12-IIIa COMPUTING ELEMENT CHECKING SUMMARY (CHECK REGISTER 1)365
TABLE 12-IIIb COMPUTING ELEMENT CHECKING SUMMARY (CHECK REGISTER 2)367
TABLE 12-IIIc COMPUTING ELEMENT CHECKING SUMMARY (DAR)369
TABLE 12-IVa INPUT/OUTPUT CONTROL ELEMENT CHECKING SUMMARY (CHECK REGISTER 1)370
TABLE 12-IVb INPUT/OUTPUT CONTROL ELEMENT CHECKING SUMMARY (CHECK REGISTER 2)372
TABLE 12-IVc INPUT/OUTPUT CONTROL ELEMENT CHECKING SUMMARY (DAR)	.373
TABLE 12-V STORAGE ELEMENT CHECKING SUMMARY381
TABLE 12-VI DISPLAY ELEMENT CHECKING SUMMARY382
TABLE 13-I INSTRUCTION TIME AVERAGE CABLE DELAYS424
TABLE 13-II FIXED-POINT ARITHMETIC INSTRUCTIONS425
TABLE 13-III SHIFT INSTRUCTIONS426
TABLE 13-IV FLOATING-POINT ARITHMETIC INSTRUCTIONS427
TABLE 13-V LOGICAL INSTRUCTIONS428
TABLE 13-VI DECIMAL ARITHMETIC INSTRUCTIONS429
TABLE 13-VII BRANCHING AND STATUS SWITCHING INSTRUCTIONS430
TABLE 13-VIII MULTIPLE COMPUTING ELEMENT INSTRUCTIONS435
TABLE 13-IX DISPLAY INSTRUCTIONS435
TABLE H-1 -- INTERRUPTION ACTION IOCE-PROCESSOR557
TABLE H-2 -- ALPHABETIC LIST OF IOCE-PROCESSOR INSTRUCTIONS565
TABLE H-3 -- PROGRAMMING EXCEPTIONS IN THE IOCE-PROCESSOR568
TABLE H-4 IOCE PROCESSOR INSTRUCTION TIMINGS573
TABLE J-I COMMANDS USED BY THE CHANNEL-TO-CHANNEL ADAPTER591
TABLE J-II SUMMARY OF STATUS INDICATIONS597
TABLE K-I DAU ADDRESSING STRUCTURE617
TABLE K-II RKM-DAU COMMANDS AND FORMATS620
TABLE K-III IOCE/DAU EXECUTABLE COMMANDS621

JUNE 1, 1971

CHAPTER 1: INTRODUCTION TO 9020D/E SYSTEMS

1.1 INTRODUCTION

The IBM 9020D and 9020E Systems are solid-state data processing systems which provide the speed, precision, and data manipulating versatility demanded by the challenge of Air Traffic Control. The IBM 9020D/E Systems, with advanced logical design implemented by microminiature technology, provide a new dimension of performance, flexibility, and reliability. This dimension makes possible a new more efficient systems approach to information processing, with economy of implementation and ease of use.

The logical design of the IBM 9020D/E Systems permit efficient use at several levels of performance. Extremely high performance and reliability requirements are met by the modularity inherent in the basic logical structure of both the 9020D and 9020E Systems.

Differentiation between a 9020D and 9020E System is primarily one of system configuration and utilization. The 9020D System is designed to provide increased compute capability and increased high speed core storage in the Central Computer Complex (CCC) task. The IBM 9020E System, through system configuration and the utilization of unique task oriented micro-programmed instructions, provides both high performance and reliability in the Display Channel Processor (DCP) environment.

The 9020D and 9020E Systems utilize the same computing elements, storage elements and input-output control elements. The remaining major system elements in each system are configuration dependent.

1.1.1 9020D SYSTEM

Individual 9020D Systems, designed for the Central Computer Complex (CCC) environment, are composed of the following types of major self-contained, self-powered elements and units:

Computing Element	(7201-02)
Storage Element	(7251-09)
System Console	(7265-02)
Peripheral Adapter Module	(7289-02)
Tape Control Unit	(2803-01)
Storage Control Unit	(2314-A1)

The IBM 9020D System design incorporates the following features:

- Any Computing Element can access any storage element.
- Any Input/Output Control Element can access any storage element.
- Each Peripheral Adapter Module is connected to the system through two input/output control elements.
- Each Input/Output line is connected to the system through a subchannel in each of two peripheral adapter modules.

1.1.2 9020E SYSTEM

Individual 9020E Systems, designed for the Display Channel Processor (DCP) environment, are composed of the following types of major self-

JUNE 1, 1971

contained, self-powered elements and units:

Computing Element	(7201-02)
Storage Element	(7251-09)
I/O Control Element	(7231-02)
Display Element	(7289-04)
Configuration Console	(7265-03)
Tape Control Unit	(2803-01)
Data Adapter Unit	(2701-01)

The IBM 9020E System design incorporates the following features:

- Any Computing Element can access any storage element or any display element.
- Any Input/Output Control Element can access any storage element.
- Each Data Adapter Unit is connected to the system through two input/output control elements.
- Each 9020A or 9020D IOCE communication path is connected to the 9020E System through two input/output control elements.

Other units which are not part of the 9020E, but which are nevertheless discussed in this publication for completeness, are the Display Generator (DG), Radar Keyboard Multiplexor (RKM), System Maintenance Monitor Console (SMMC), and Plan View Display (PVD). This publication does not apply to these units except where explicitly stated.

1.2 COMPATIBILITY

1.2.1 SYSTEM/360 COMPATIBILITY

The IBM 9020D/E Systems are compatible with the IBM System/360 when operating in 360-mode. Compatibility allows for ease in systems growth, convenience in systems backup, and simplicity in education.

The compatibility rule has limitations:

- 1) The systems facilities used by a program should be the same in each case. Thus, the optional computing-element features and the storage capacity, as well as the quantity, type, and priority of input/output equipment, should be equivalent. In particular, the 9020D/E multiprocessor and display instructions are not available in the System/360 mode.
- 2) The program should be independent of the relation of instruction execution time and of input/output data rates, access times, and command execution times.
- 3) The compatibility rule does not apply to detail functions for which neither frequency of occurrence nor usefulness of result warrants identical action in all models of System/360. These functions, all explicitly identified in the System/360 Principles of Operation manual, are concerned with the handling of invalid programs and machine malfunctions.

An example of this variance is the DIAGNOSE instruction. The 9020D and 9020E Systems use a Maintenance Control Word in conjunction with the DIAGNOSE while the 9020A System does not. This reflects different diagnostic approaches for various models of IBM System/360.

JUNE 1, 1971

- 4) Programs not specifically designed to run on both System/360 and 9020D/E require use of the lowest numbered storage and will be unable to address some input/output channels when not run on their home system.

1.2.2 9020A COMPATIBILITY

The 9020D and 9020E Systems are designed to be compatible with the 9020A System. Modifications in some areas were necessary to provide increased storage element capacity, display processing, and functional duality for both 9020D and 9020E utilization.

The major differences in the 9020A and 9020D/E hardware result from the following modifications:

- | | |
|-------------------------|--------------------------------------------------------------|
| 1. SE logout format | 5. SATR instruction |
| 2. CE logout format | 6. SCON instruction |
| 3. SE storage capacity | 7. Operator panels |
| 4. DIAGNOSE instruction | 8. Inclusion of four special display processor instructions. |

1.3 COMPUTING ELEMENT

The Computing Element of the IBM 9020D/E contains the logic to perform arithmetic, logical, and input/output instructions and initiates storage selection, input/output data transfers, IOCE-processor operations, and intercomputing-element data transfers.

Data are processed as a 36-bit word made up of four bytes of information (byte = eight data bits plus parity bit). Data are always fetched from storage in doublewords (eight bytes) but may be stored by byte. The computing element has a local store containing 16 general-purpose registers, four floating-point registers, adder logic, high-speed transistor registers, and an internal read-only storage used for control. All data transfers into, out of, and within the computing element are parity checked on a byte basis.

The Computing Element (CE) operates with a 200-nanosecond machine cycle. Storage address capability accommodates up to 16,777,216 storage byte locations.

The Computing Element may also house an optional 1052 Keyboard Printer Adapter within the 9020E System.

1.4 STORAGE ELEMENT

Storage Elements (SE), in the form of 65,536 doubleword (doubleword = eight bytes) modules, have an 0.80-microsecond doubleword cycle (read and restore). Two-way interleaving is provided to produce an effective 0.40 microsecond doubleword cycle for alternating accesses to even and odd addresses. Each storage element is a completely self-contained unit, with its own registers and power supplies. Each storage element has seven "tails", which allow up to four CE and three input/output control elements to access each element. Priority logic contained in each storage element determines which of the seven elements accesses storage when more than one simultaneous access is attempted. The input/output control elements have priority over CE's, with each IOCE contending for priority at the point in its cycle when a storage cycle is required. Logic will break ties if two or more elements of the same type attempt simultaneous access to one storage element.

1.5 DISPLAY ELEMENT

Display storage is provided within a 9020E System by Display Elements (DE). These synchronous two-way interleaved elements, which interface only computing elements and display generators, provide 32,768 doublewords of display storage with an 0.80 microsecond doubleword cycle time. Each DE is a stand-alone unit containing its own power, registers, and tails. Up to four computing elements and eight display generators may be attached to each DE; of the eight display generator interfaces only four may be active at one time. A fixed, 'slotted', priority scheme is provided in the DE, allocating one of nine doubleword cycles to the CES as a minimum. Those storage cycles normally allocated for the display generators will be made available to the CES if not required by the display generators.

1.6 SYSTEM CONSOLE AND CONFIGURATION CONSOLE

1.6.1 9020D SYSTEM CONSOLE

A valuable tool to trace down system errors -- equipment or programming -- is the system console. The system console of the IBM 9020D provides the following seven types of indicators and controls:

- 1) Element error and status indicators
- 2) Manual controls and switches for the selection of major elements and certain peripheral units
- 3) Data and sense switches
- 4) Computing element register display
- 5) Power indicators
- 6) Initial program loading and other unit-control switches
- 7) An interface to SMMC for subsystem configuration indicators and check signals.

The system console contains those indicators and controls necessary for the effective monitoring and operation of the IBM 9020D System. Equipment out of service for data reduction, maintenance, or diagnostic tests is so indicated on the system console. Operation of such units for maintenance purposes will be performed from unit test panels or, in the case of a subsystem configuration, from the individual CE control panel.

1.6.2 9020E CONFIGURATION CONSOLE

The Configuration Console (CC) provides three basic functions for the 9020E System:

1. System Console
2. Reconfiguration Unit for DAU, DG and RKM units.
3. Status Reporting Unit for DG and RKM units.

The System Console portion of the CC contains those indicators and controls necessary for the effective monitoring and operation of the 9020E System. These include:

1. Element error and status indicators
2. Manual controls and switches for the selection of major elements and certain peripheral units.
3. Data and Sense switches.
4. Computing Element register display
5. Power indicators.
6. Initial program loading controls.
7. In addition a programmable data path to the SMMC is provided.

JUNE 1, 1971

The Reconfiguration Unit portion of the CC provides duplex paths for configuring and monitoring of DG and RKM units and the 9020E 2701 data adapter units.

Those indicators necessary for effective system monitoring of the DG and RKM units are provided on the CC panel. These include:

1. Unit Status Indicators
2. Power Status
3. Check Indicators

1.7 INPUT/OUTPUT CONTROL ELEMENT

The input/output control element (IOCE) is a stand-alone device, made up of selector channels, one multiplexor channel, and a processor.

From the user's point of view, the IOCE consists of two distinct and independent functional units sharing a common logic and internal storage unit (i.e., Maintenance and Channel (MACH) Storage): IOCE - channel controller and IOCE-processor. Since the IOCE-channel controller and IOCE-processor are functionally independent, the IOCE can perform I/O operations and processing operations concurrently.

The IOCE-channel controller receives a basic input/output command from a selecting CE, energizes the appropriate device, controls the specified operation, and responds to the CE with interruption information. The data path for the IOCE is directly to and from any of the main-storage elements, or to and from the IOCE's internal (MACH) storage and is under program control. The storage areas are utilized by the IOCE as a data-buffer and instruction areas. Each IOCE contains its own control storage.

The IOCE has the capability of working with a device on each selector channel and all the devices on the multiplexor channel simultaneously.

Only one CE at a time can control an IOCE. By masking unused channels, the input/output system program will insure that no interruptions are accepted from the IOCE channels not used by the CE.

The base address that a CE is using to determine its preferential storage location is supplied to the IOCE(s) with which it is working. Using the base address, the IOCE routes interruption words to and fetches certain control words from the proper locations.

The IOCE-processor operation (i.e., instruction execution) is initiated by a controlling CE. The IOCE-processor can execute a subset of the IBM 9020 System instruction set with a few special instructions added. The IOCE-processor has access to all main storage elements and its own internal MACH storage for instruction execution and data sets. The IOCE-processor cannot initiate I/O operations. (See Appendix H).

NOTE: The CE does not have access to MACH storage.

The IOCE may house up to two Channel-to-Channel Adapters.

1.8 PERIPHERAL ADAPTER MODULE

The peripheral adapter module satisfies two basic requirements for real-time processing:

1. Subchannels to provide entry points for multiple devices and a wide range of data rates and message lengths.

JUNE 1, 1971

2. Priority control of data transfers and program interruptions as appropriate for individual system configuration.

1.9 TAPE CONTROL UNIT

The magnetic tape units are connected to the I/O control element via tape control units. The tape control units provide data byte assembly and disassembly as well as the control functions for the tape units. Only one tape drive may be transmitting data to, or accepting data from a particular tape control unit at a time; however, up to eight tape drives may be attached to each tape control unit.

The tape control unit is capable of being controlled from two selector channels on a first-come-first-served basis. In any event, the TCU must be configured to communicate with either one or both IOCEs. If one channel attempts to use a tape control unit already operating under control of another channel, a busy I/O indication will occur.

1.10 STORAGE CONTROL UNIT

The disk storage units are connected to the I/O control element via storage control units. The storage control units provide data byte assembly and disassembly as well as the control functions for the disk storage units. Only one disk storage unit may be transmitting data to, or accepting data from a particular storage control unit at a time; however, up to five disk storage units (maximum configuration is five 2312s or four 2318s and a 2312) may be attached to each storage control unit for a maximum of eight active and one spare disk module.

The storage control unit is capable of being controlled from two selector channels on a first-come-first-served basis. In any event, the SCU must be configured to communicate with either one or both IOCEs. If one channel attempts to use a storage control unit already operating under control of another channel, a busy I/O indication will occur.

1.11 DATA ADAPTER UNIT

The Radar Keyboard Multiplexers (RKM) associated with the 9020E System are interfaced to the I/O control elements via the data adapter units. The data adapter units provide buffering as well as control functions for the RKM's. Only one RKM may be transmitting data to/or accepting data from a data adapter unit; however, up to five RKM's may be attached to each data adapter unit.

1.12 SYSTEM ALERTS

The interruption system permits a CE to automatically change state as a result of conditions arising outside of the system, in input/output (I/O) units, or in a CE itself. Interruption switches the CE from one program to another by changing not only the instruction address, but all of the essential machine-status information.

A storage-protection feature permits one program to be preserved when another program erroneously attempts to store or fetch information in the area assigned to the first program. Protection does not cause any loss of performance. Storage operations initiated from a CE, as well as those initiated from the IOCE-processor or an IOCE channel, are subject to the protection procedure.

Programs are checked for correct instructions and data as they are executed. This policing action identifies and separates program errors

JUNE 1, 1971

and machine errors. Thus, program errors do not create machine checks since each type of error causes a unique interruption. In addition to an interruption due to machine malfunction, the information necessary to identify the error is recorded automatically in a predetermined storage location. This procedure appreciably reduces the mean-time to repair a machine fault. Moreover, operator errors are reduced by minimizing the active manual controls. To reduce accidental operator errors, operator consoles are I/O devices and function under control of the system program.

JUNE 1, 1971

CHAPTER 2: SYSTEM STRUCTURE

The basic structure of an IBM 9020D system consists of main-storage, computing elements (CE), input/output control elements (IOCE) containing the selector and multiplexor channels and channel-to-channel adapters, and the input/output devices attached to the channels through control units. It is possible for subsystems to communicate with each other by means of shared input/output devices (I/O), shared storage, or direct control. The basic organization of an IBM 9020D system is shown in Figure 2-1.

The basic structure of an IBM 9020E System consists of main storage, display storage, computing elements (CE), input/output control elements (IOCE) containing the selector and multiplexor channels and channel-to-channel adapters, and the input/output devices attached to the channels through control units. It is possible for subsystems to communicate with each other by means of shared input/output devices, shared storage, or direct control. The basic organization of an IBM 9020E System is shown in Figure 2-2.

2.1 MAIN STORAGE

Storage units are self-contained and self-powered units. The storage cycle is not directly related to the internal cycling of the CE, thus permitting selection of optimum storage speed. Each storage unit provides two-way interleaving for consecutive odd/even doubleword accesses.

The fetching and storing of data by the CE is not affected by any concurrent IOCE operation (i.e., I/O operation or IOCE-processor operation). When an IOCE is requesting data (fetch operation) from the same SE as a CE operation, the IOCE access is honored first, followed by the CE. When the IOCE is storing data, the IOCE access is acknowledged first, however, due to the basic difference between the machine speeds of the IOCE and CE, the storage unit will allow the CE to receive the first storage cycle followed by the IOCE storage cycle. Concurrent IOCE and CE references to the same storage element do not cause a machine-check indication.

Instructions that involve fetching and subsequent storing of data do not necessarily take the storage cycles contiguously, and it is possible for any combination of CEs and/or IOCEs to take one or more intervening cycles. When any combination of CEs and/or IOCEs concurrently cause the contents of the same location to be updated, such interleaving may cause the information stored in one of the accesses to be lost or the results to be meaningless.

For example, if one CE attempts to update a storage location by executing an instruction, or instructions, that fetch data, update it, and store it back into the original location, it is possible for another CE, or an IOCE, to fetch or store data at that same location between the fetch and store cycles of the first CE. This would result in loss of the data stored by the second CE, or IOCE.

JUNE 1, 1971

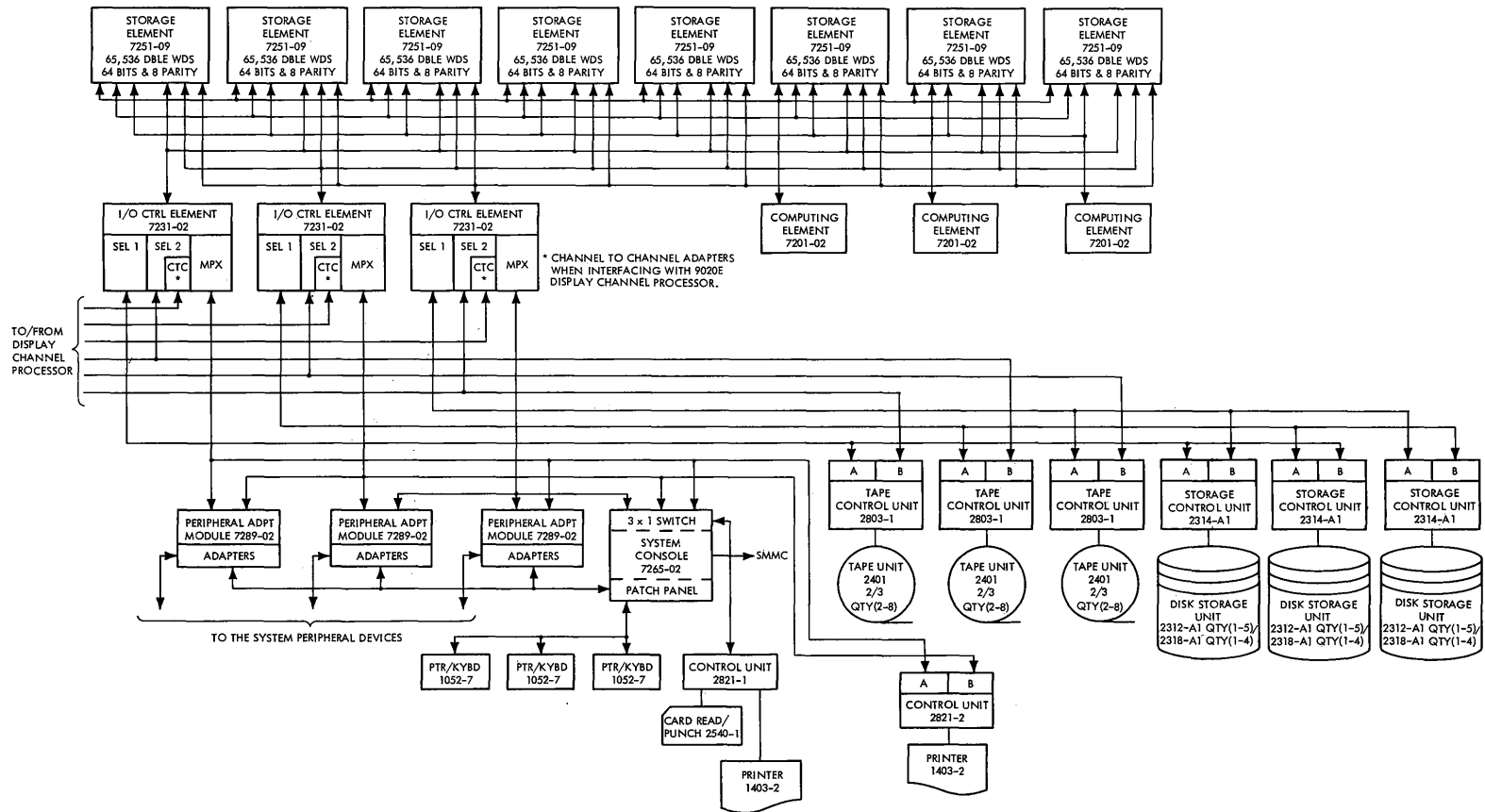


Figure 2-1 IBM 9020D CCC System Basic Logical Structure

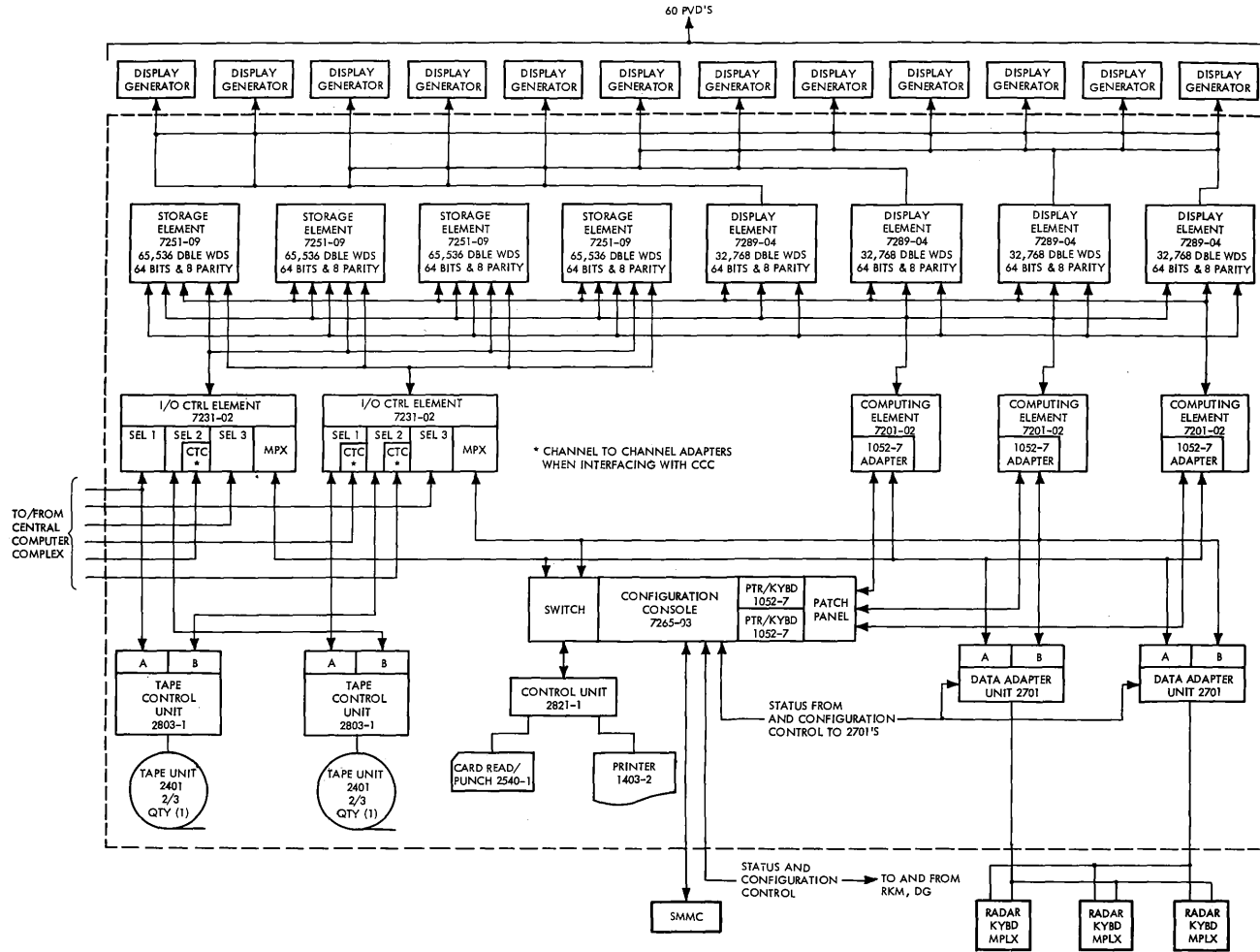


Figure 2-2 IBM 9020E DCP System Basic Logical Structure

JUNE 1, 1971

JUNE 1, 1971

TEST AND SET and the immediate instructions (AND, OR, and EXCLUSIVE OR) avoid this problem by forcing the storage to give the executing CE two consecutive cycles before allowing any other unit access to that Storage Element.

2.1.1 INFORMATION FORMATS

The system transmits information between main storage and a CE, eight bytes at a time. Four bytes at a time are transferred between main storage and an IOCE. An eight-bit unit of information is called a byte -- the basic building block of all formats. With each byte a ninth bit, the parity or check bit, is transmitted. It carries the parity of the byte. The parity bit cannot be affected by the program. The effect of improper parity is to cause a machine-check interruption. References to the size of data fields and registers in this manual generally exclude the associated parity bits.

Bytes may be handled separately or grouped together in fields. A halfword is a group of two consecutive bytes and is the basic building block of instructions. A word is a group of four consecutive bytes; a double word is a field consisting of two words (Figure 2-3). The location of any field or group of bytes is specified by the address of its leftmost byte.

The length of fields is either implied by the operation to be performed or stated explicitly as part of the instruction. When the length is implied, the information is said to have a fixed length, which can be either one, two, four, or eight bytes.

When the length of a field is not implied by the operation code, but is stated explicitly, the information is said to have variable field length. Variable-length operands are variable in length by increments of one byte.

Within any program format or any fixed-length operand format, the bits making up the format are consecutively numbered from left to right starting with the number 0.

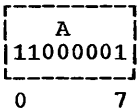
2.1.2 ADDRESSING

Byte locations in storage are consecutively numbered starting with "logical" 0; each number is considered the address of the corresponding byte. A group of bytes in storage is addressed by the leftmost byte of the group. The number of bytes in the group is either implied or explicitly defined by the operation. The addressing arrangement uses a 24-bit binary address to accommodate a maximum of 16,777,216 byte addresses. This set of main-storage addresses includes some locations reserved for special purposes.

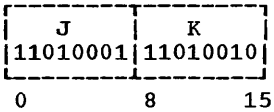
An addressing exception is recognized when any part of an operand is located beyond the available storage capacity of an installation, or located outside the configured storage, or the storage assigned by the storage address translator for a particular CE or IOCE. The addressing exception is only recognized when the address is actually used to attempt an access, and not during an address manipulation operation prior to its use.

In the IBM 9020D/E Systems main storage may be shared by more than one CE and/or IOCE. The address of a byte location is determined by the content of a ten-position storage address translator (ATR) in each accessing element. Addresses generated by the CE and IOCE may be

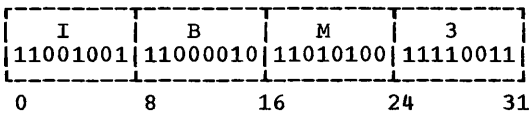
Byte



Halfword



Word



Doubleword

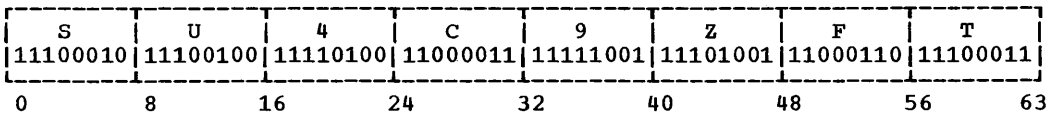


Figure 2-3 Sample Information Formats

relocated using the current value of the preferential-storage base address (Chapter 8). When the preferential-storage base address is other than zero, CE accesses to the first 4096-byte block of main storage are relocated, and therefore are not available to the programmer. This block is available, however, to the IOCE-processor and IOCE Channel Controller.

PROGRAMMING NOTE

The IBM 9020D System uses Storage Elements (SE) in modules of 131,072 words (or 524,288 bytes) each. The 9020D configuration provides 4,194,304 bytes of storage, expandable to 5,242,880 bytes, which represents eight and ten modules, respectively. Byte addresses extend from 0 to 4,194,303 or 5,242,879 inclusive for main storage in a 9020D System.

2.1.3 INFORMATION POSITIONING

Fixed-length fields, such as halfwords and double words, must be located in main storage on an integral boundary for that unit of information. A boundary is called integral for a unit of information when its storage address is a multiple of the length of the unit in bytes. For example, words (four bytes) must be located in storage so that their address is a multiple of the number 4. A halfword (two bytes) must have an address that is a multiple of the number 2, and double words (eight bytes) must have an address that is a multiple of the number 8.

Storage addresses are expressed in binary form. In binary, integral boundaries for halfwords, words, and double words can be specified only by the binary addresses in which one, two, or three of the low-order bits, are zero respectively (Figure 2-4). For example, the integral

JUNE 1, 1971

boundary for a word is a binary address in which the two low-order positions are zero.

Variable-length fields are not limited to integral boundaries, but may start on any byte location.

2.2 DISPLAY ELEMENT

Note: This section is only intended as an introduction to the display element (DE). Due to the task-oriented architecture of this element, Appendix I should be referenced for a more detailed discussion of its system functions and utilization.

Display elements are constructed as self-contained and self-powered units for utilization within a 9020E system. These synchronous units provide a data path from a 9020E system to the display subsystem as well as providing data storage and some control functions for the display subsystem.

The DE storage cycle for these units is not directly related to the internal cycling of the attached units, thus permitting an optimum storage speed. Each DE provides two-way interleaving for two consecutive doubleword accesses.

Display elements are accessed by up to four computing elements (CE) and eight display generators (DG) (only four DGs being active at a time). Unlike SEs the display elements are synchronous or "slotted" devices in order to provide a synchronous environment to the display subsystem. Storage access availability is likewise a predetermined "slotted" scheme which allocates accesses to specific display generator access requests and to computing elements (as a group). One out of every nine accesses is allocated to CEs, the remaining eight being allocated to specific DG access requests. Those accesses specifically allocated to DGs will, if not required by a specific DG, be made available to the computing elements.

2.2.1 INFORMATION FORMATS

The CEs transmit information between themselves and DEs eight bytes at a time (doubleword). Transmission of data from a DE to a display generator is performed two bytes at a time (halfword). The actual storage fetch for a DG however, is a quadword (sixteen byte) fetch which is implemented as two consecutive interleaved odd-even or even-odd storage cycles. The quadword fetched is stored in a buffer within the DE for subsequent transmission to a DG as eight halfword data transfers.

Data within the display element is rigidly formatted for subsequent transmission to the display subsystem. This formatted data is basically of a doubleword form. Consequently, there is a minimal requirement for handling display element data in other than doubleword units of information.

2.2.2 ADDRESSING

The DEs are externally addressed by the CEs in the same manner as an SE, including address translation via the CE resident ATRs. There is no external addressing of a DE by a display generator, such addressing being determined internally by the DE itself (initially specified by the programmer). Additionally there is no DE selection addressing from display generators since each display generator can be attached to only one DE at a time.

Byte locations within each DE are consecutively numbered starting with 0, each number being considered the address of the corresponding

byte. Each DE provides 262,144 bytes (65,536 words) of storage. The first 98 doublewords of each DE are not normally available for data storage as they are used by the DE logic for address determination relative to display generator requests.

Display storage addressing from a system standpoint (multiple DEs) is disjunctive by DE, there being system address "gaps" between each DE. storage capacity of 262,144 bytes. Additionally, display storage always begins at physical system byte address 2,621,440 (See Table 8-VI).

PROGRAMMING NOTE

The IBM 9020E System uses Storage Elements (SE) in modules of 131,072 words (524,288 bytes) each and Display Elements (DE) in modules of 65,536 words (262,144 bytes) each. The 9020E configuration provides 2,097,152 bytes of main storage plus 1,048,576 bytes of display storage, expandable to 2,621,440 and 1,310,720 bytes respectively. This storage capacity represents four SEs and four DEs expandable to five SEs and five DEs. Main storage byte addresses range contiguously from 0 to either 2,097,151 or 2,621,439. Display storage byte addresses range disjunctively from 2,621,440 through 4,456,447 or 4,980,735. Display elements do not provide for address wrap-around within the unit nor are preferential storage areas (either primary or alternate) permitted within these elements (reference the discussion of Preferential Storage Base Address Register in Chapter 8).

2.3 COMPUTING ELEMENT

Each computing element (CE) contains the facilities for addressing main storage and display storage, for fetching or storing information, for arithmetic and logical processing of data, for sequencing instructions in the desired order, for initiating IOCE-processor operations, and for initiating the communication between storage and external devices.

The system control section provides the normal CE control that guides the CE through the operations necessary to execute the various instructions.

The CE provides 16 general registers for fixed-point operands and four floating-point registers for floating-point operands.

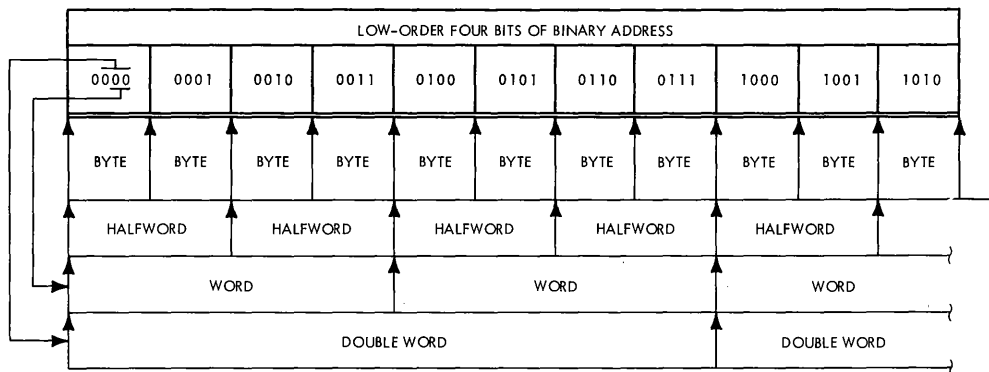


Figure 2-4 Integral Boundaries for Halfwords, Words, and Double Words

JUNE 1, 1971

2.3.1 GENERAL REGISTERS

The CE can address information in 16 general registers. The general registers can be used as index registers, in address arithmetic and indexing, and as accumulators in fixed-point arithmetic and logical operations. The registers have a capacity of one word (32 bits). The general registers are identified by numbers 0-15 and are selected by a four-bit R field in an instruction (Figure 2-6).

For some operations, two adjacent general registers are coupled together, providing a two-word capacity. In these operations, the addressed register contains the high-order operand bits and must have an even address, and the implied register, containing the low-order operand bits, has the next higher address.

2.3.2 FLOATING-POINT REGISTERS

Four floating-point registers are available for floating-point operations. They are identified by the numbers 0, 2, 4, and 6 (Figure 2-6). These floating-point registers, are two words (64 bits) in length and can contain either a short (one word) or a long (two words) floating-point operand. A short operand occupies the high-order bits of a floating-point register. The low-order portion of the register is ignored and remains unchanged in short-precision arithmetic. The instruction operation code determines which type of register (general or floating-point) is to be used in an operation.

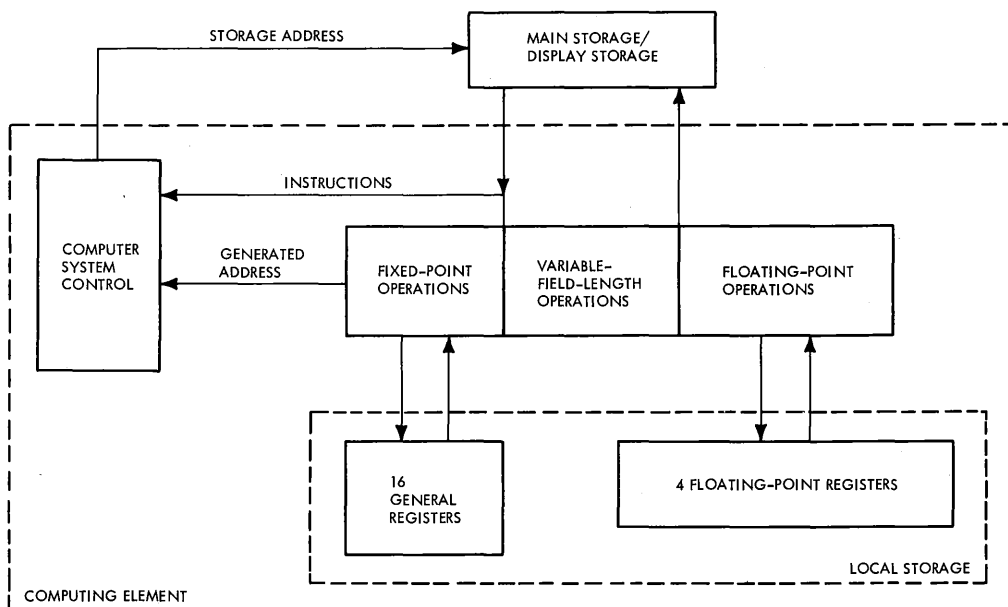


Figure 2-5 Computing Element

R FIELD	REG NO.	GENERAL REGISTERS	FLOATING-POINT REGISTERS
0000	0	32 BITS	64 BITS
0001	1		
0010	2		
0011	3		
0100	4		
0101	5		
0110	6		
0111	7		
1000	8		
1001	9		
1010	10		
1011	11		
1100	12		
1101	13		
1110	14		
1111	15		

Figure 2-6 General and Floating-Point Registers

2.4 ARITHMETIC AND LOGICAL UNIT

The arithmetic and logical unit can process binary integers and floating-point fractions of fixed length, decimal integers of variable length, and logical information of either fixed or variable length.

Arithmetic and logical operations performed by the CE fall into four classes: fixed-point arithmetic, decimal arithmetic, floating-point arithmetic, and logical operations. These classes differ in the data formats used, the registers involved, the operations provided, and the way the field length is stated.

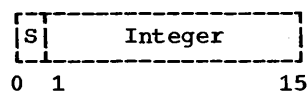
2.4.1 FIXED-POINT ARITHMETIC

The basic arithmetic operand is the 32-bit fixed-point binary word. Sixteen-bit halfword operands may be specified in most operations for improved storage utilization. To preserve precision, some products and all dividends are 64 bits in length (chapter 3).

Because the 32-bit word size conveniently accommodates a 24-bit address, fixed-point arithmetic can be used both for integer operand arithmetic and for address arithmetic. This combined usage provides economy of implementation and permits the entire fixed-point instruction set and several logical operations to be used in address computation. Thus, multiplication, shifting, and logical manipulation of address components are provided.

JUNE 1, 1971

Halfword



Full Word

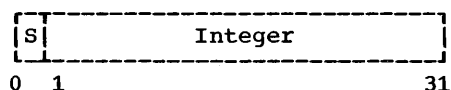


Figure 2-7 Fixed-Point Number Formats

Additions, subtractions, multiplications, divisions, and comparisons are performed upon an operand in a register and another operand either in a register or from storage. Multiple-precision operation is made convenient by the twos-complement notation and by recognition of the carry from one word to another. A word in one register or a double word in a pair of adjacent registers may be shifted left or right. A pair of conversion instructions -- CONVERT TO BINARY and CONVERT TO DECIMAL -- provides transition between decimal and binary radix (number base) without the use of tables. Multiple-register loading and storing instructions facilitate subroutine switching.

2.4.2 DECIMAL ARITHMETIC

Decimal arithmetic lends itself to data-processing procedures that require few computational steps between the source input and the documented output. This type of processing is frequently found in commercial applications, particularly when use is made of problem-oriented languages. Because of the limited number of arithmetic operations performed on each item of data, radix conversion from decimal to binary and back to decimal is not justified, and the use of registers for intermediate results yields no advantage over storage-to-storage processing. Hence, decimal arithmetic is provided, and both operands and results are located in storage. Decimal arithmetic includes addition, subtraction, multiplication, division, and comparison.

Decimal numbers are treated as signed integers with a variable-field-length format from one to 16 bytes in length. Negative numbers are carried in true form.

The decimal digits 0-9 are represented in the four-bit binary-coded-decimal form by 0000-1001, respectively (Figure 2-8). The codes 1010-1111 are not valid as digits and are reserved for sign codes; 1011 and 1101 represent a minus; the other four codes are interpreted as plus. The sign codes generated in decimal arithmetic depend upon the character set preferred (Figure 2-8). When the extended binary coded decimal interchange code (EBCDIC) is preferred, the codes are 1100 and 1101. When the ASCII set, expanded to eight bits, is preferred, the codes are 1010 and 1011. The choice between the two code sets is determined by a mode bit in the PSW.

	Digit Code	Sign Code
0	0000	+ 1010
1	0001	- 1011
2	0010	+ 1100
3	0011	- 1101
4	0100	+ 1110
5	0101	+ 1111
6	0110	
7	0111	
8	1000	
9	1001	

Figure 2-8 Bit Codes for Digits and Signs

Decimal operands and results are represented by four-bit binary-coded-decimal digits packed two to a byte. They appear in fields of variable length and are accompanied by a sign in the rightmost four bits of the low-order byte. Operand fields may be located on any byte boundary, and may have length up to 31 digits and sign. Operands participating in an operation may have different lengths. Packing of digits within a byte (Figure 2-9) and of variable-length fields within storage results in efficient use of storage, in increased arithmetic performance, and in an improved rate of data transmission between storage and files.

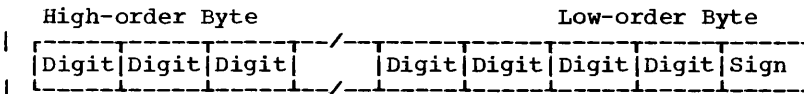


Figure 2-9 Packed Decimal Number Format

Decimal numbers may also appear in a zoned format as a subset of the eight-bit alphanumeric character set (Figure 2-10). This representation is required for character-set sensitive I/O devices. A zoned format number carries its sign in the leftmost four bits of the low-order byte. The zoned format is not used in decimal arithmetic operations. Instructions are provided for packing and unpacking decimal numbers so that they may be changed from the zoned to the packed format and vice versa.

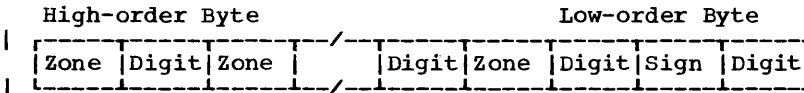


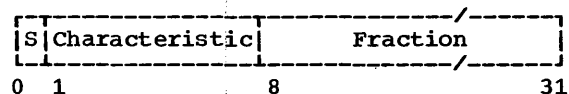
Figure 2-10 Zoned Decimal Number Format

2.4.3 FLOATING-POINT ARITHMETIC

Floating-point numbers occur in either of two fixed-length formats -- short or long. These formats differ only in the length of the fractions (Figure 2-11). (See Chapter 5)

JUNE 1, 1971

Short Floating-Point Number (One Word)



Long Floating-Point Number (Double Word)

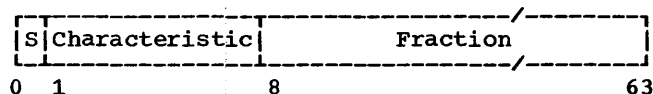


Figure 2-11 Short and Long Floating-Point Number Formats

Floating-point operands are either 32 or 64 bits long. The short length, equivalent to seven decimal places of precision, permits a maximum number of operands to be placed in storage and gives the shortest execution times. The long length, used when higher precision is desired, gives up to 17 decimal places of precision, thus eliminating most requirements for double-precision arithmetic.

The operand lengths, being powers of 2, permit maximum efficiency in the use of binary addressing. Floating-point arithmetic is designed to allow easy transition between two formats.

The fraction of a floating-point number is expressed in hexadecimal (base 16) digits, each consisting of four binary bits and having the values 0-15. In the short format, the fraction consists of six hexadecimal digits occupying bits 8-31. In the long format the fraction has 14 hexadecimal digits occupying bits 8-63.

The radix point of the fraction is assumed to be immediately to the left of the high-order fraction digit. To provide the proper magnitude for the floating-point number, the fraction is considered to be multiplied by a power of 16. The characteristic portion, bits 1-7 of both formats, is used to indicate this power. The characteristic is treated as an excess-64 number with a range from -64 through +63, and permits representation of decimal numbers with magnitudes in the range of 10^{-78} , to 10^{75} .

Bit position 0 in either format is the sign (S) of the fraction. The fraction of negative numbers is carried in true form.

Four 64-bit floating-point registers are provided. Arithmetic operations are performed with one operand in a register and another either in a register or from storage. The result, developed in a register, is generally of the same length as the operands. The availability of several floating-point registers eliminates much storing and loading of intermediate results.

2.4.4 LOGICAL OPERATION

Logical information is handled as fixed- or variable-length data. It is subject to such operations as comparison, translation, editing, bit testing, and bit setting. (Chapter 6)

When used as a fixed-length operand, logical information can consist of either one, four, or eight bytes and is processed in the general registers (Figure 2-12).

A large portion of logical information consists of alphabetic or numeric character codes, called alphanumeric data, and is used for communication with character-set sensitive I/O devices. This information has the variable-field-length format and can consist of up to 256 bytes (Figure 2-13). It is processed storage to storage, left to right, an eight-bit byte at a time.

Fixed-Length Logical Operand (One, Four, or Eight Bytes)

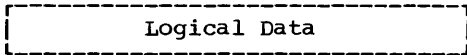


Figure 2-12 Fixed-Length Logical Information

Variable-Length Logical Operand (Up to 256 Bytes)

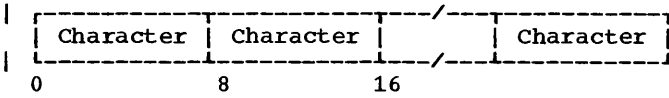


Figure 2-13 Variable-Length Logical Information

The CE can handle any eight-bit character set, although certain restrictions are assumed in the decimal arithmetic and editing operations. However, all character-set sensitive I/O equipment will assume either the Extended Binary-Coded-Decimal Interchange Code (EBCDIC) [Figure 2-14] or the United States of America Standard Code for Information Interchange (USASCII-8) (Figure 2-15).

The numbering convention for bit positions within a character differ for each code. The conventions are as follows:

	Bit Positions
EBCDIC	01234567
USASCII-8	87654321

The preferred codes do not have a graphic defined for all 256 eight-bit codes. When it is desirable to represent all possible bit patterns, a hexadecimal representation may be used instead of the preferred eight-bit code. The hexadecimal representation uses one graphic for a four-bit code, and therefore, two graphics for a eight-bit byte. The graphics 0-9 are used for codes 0000-1001; the graphics A-F are used for codes 1010-1111.

JUNE 1, 1971

Bit Positions 4,5,6,7 Second Hexadecimal Digit Digit Punctures		00				01				10				11				Bit Positions 0,1	
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	Bit Positions 2,3	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	First Hexadecimal Digit	
		12	11	0	9	12	11	0	9	12	11	0	9	12	11	0	9	Zone Punctures	
0000	0	8-1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
		NUL	DLE	DS	SP	&	-												
0001	1	1	1						13	a	i			A	J	14	1		
0010	2	2	2	3	4					b	k	s		B	K	S	2		
0011	3	3	3							c	l	t		C	L	T	3		
0100	4	4	4	5	6					d	m	u		D	M	U	4		
0101	5	5	5	6	7					e	n	v		E	N	V	5		
0110	6	6	6	7	8					f	o	w		F	O	W	6		
0111	7	7	7	8	9					g	p	x		G	P	X	7		
1000	8	8	8							h	q	y		H	Q	Y	8		
1001	9	8-1								i	r	z		I	R	Z	9		
1010	A	8-2	SMM	CC	SM	ç	l	15	:										
1011	B	8-3	VT	CU1	CU2	CU3	\$,	#										
1100	C	8-4	FF	IFS		DC4	<	*	%	@									
1101	D	8-5	CR	IGS	ENQ	NAK	()	-	'									
1110	E	8-6	SO	IRS	ACK		+	;	>	=									
1111	F	8-7	SI	IUS	BEL	SUB		¬	?	"									
		12				11				10				09				Zone Punctures	
		11				10				09				08				Digit Punctures	
		0				9				8				7					

Card Hole Patterns

- ① 12-0-9-8-1 ⑤ No Punctures ⑨ 12-0 ⑬ 0-1
- ② 12-11-9-8-1 ⑥ 12 ⑩ 11-0 ⑭ 11-0-9-1
- ③ 11-0-9-8-1 ⑦ 11 ⑪ 0-8-2 ⑮ 12-11
- ④ 12-11-0-9-8-1 ⑧ 12-11-0 ⑫ 0

Control Character Representations

ACK	Acknowledge	EOT	End of Transmission
BEL	Bell	ESC	Escape
BS	Backspace	ETB	End of Transmission Block
BYP	Bypass	ETX	End of Text
CAN	Cancel	FF	Form Feed
CC	Cursor Control	FS	Field Separator
CR	Carriage Return	HT	Horizontal Tab
CU1	Customer Use 1	IFS	Interchange File Separator
CU2	Customer Use 2	IGS	Interchange Group Separator
CU3	Customer Use 3	IL	Idle
DC1	Device Control 1	IRS	Interchange Record Separator
DC2	Device Control 2	IUS	Interchange Unit Separator
DC4	Device Control 4	LC	Lower Case
DEL	Delete	LF	Line Feed
DLE	Data Link Escape	NAK	Negative Acknowledge
DS	Digit Select	NL	New Line
EM	End of Medium	NUL	Null
ENQ	Enquiry		

Special Graphic Characters

ç	Cent Sign	-	Minus Sign, Hyphen
.	Period, Decimal Point	/	Slash
<	Less-than Sign	,	Comma
(Left Parenthesis	%	Percent
+	Plus Sign	_	Underscore
	Logical OR	>	Greater-than Sign
&	Ampersand	?	Question Mark
!	Exclamation Point	:	Colon
\$	Dollar Sign	#	Number Sign
*	Asterisk	@	At Sign
)	Right Parenthesis	'	Prime, Apostrophe
;	Semicolon	=	Equal Sign
¬	Logical NOT	"	Quotation Mark

Figure 2-14 Extended Binary Coded Decimal Interchange Code

JUNE 1, 1971

Bit Positions 4,3,2,1 Second Hexadecimal Digit		00				01				10				11				Bit Positions 8,7 Bit Positions 6,5 First Hexadecimal Digit	
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11		
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
0000	0	NUL	DLE			SP	0				@	P			~	p			
0001	1	SOH	DC1			!①	1				A	Q			a	q			
0010	2	STX	DC2			"	2				B	R			b	r			
0011	3	ETX	DC3			#	3				C	S			c	s			
0100	4	EOT	DC4			\$	4				D	T			d	t			
0101	5	ENQ	NAK			%	5				E	U			e	u			
0110	6	ACK	SYN			&	6				F	V			f	v			
0111	7	BEL	ETB			'	7				G	W			g	w			
1000	8	BS	CAN			(8				H	X			h	x			
1001	9	HT	EM)	9				I	Y			i	y			
1010	A	LF	SUB			*	:				J	Z			j	z			
1011	B	VT	ESC			+	;				K	[k	{			
1100	C	FF	FS			,	<				L	\			l				
1101	D	CR	GS			-	=				M]			m	}			
1110	E	SO	RS			.	>				N	^②			n	~			
1111	F	SI	US			/	?				O	_			o	DEL			

① If IBM equipment implementing USASCII-8 is provided, the graphic | (Logical OR) will be used instead of ! (Exclamation Point).

② If IBM equipment implementing USASCII-8 is provided, the graphic ^ (Logical NOT) will be used instead of ^ (Circumflex).

NOTE: Current activities in committees under the auspices of the United States of America Standards Institute may result in changes to the characters and/or structure of the eight-bit representation of USASCII devised by the Institute. Such changes may cause the eight-bit representation of USASCII implemented in System/360 (USASCII-8) to be different from a future USA Standard. Since a difference of this nature may eventually lead to a modification of System/360, it is recommended that users avoid: (1) operation with PSW bit 12 set to 1, and (2) the use of any sign codes in decimal data other than those preferred for EBCDIC.

Control Character Representations

NUL	Null	SOH	Start of Heading (CC)	STX	Start of Text (CC)	ETX	End of Text (CC)	EOT	End of Transmission (CC)	ENQ	Enquiry (CC)	ACK	Acknowledge (CC)	BEL	Bell	BS	Backspace (FE)	HT	Horizontal Tabulation (FE)	LF	Line Feed (FE)	VT	Vertical Tabulation (FE)	FF	Form Feed (FE)	CR	Carriage Return (FE)	SO	Shift Out	SI	Shift In
(CC)	Communication Control	(FE)	Format Effector	(IS)	Information Separator																										

Special Graphic Characters

SP	Space	<	Less Than
!	Exclamation Point	=	Equals
	Logical OR	>	Greater Than
"	Quotation Marks	?	Question Mark
#	Number Sign	@	Commercial At
\$	Dollar Sign	[Opening Bracket
%	Percent	\	Reverse Slant
&	Ampersand]	Closing Bracket
'	Apostrophe	^	Circumflex
(Opening Parenthesis	~	Logical NOT
)	Closing Parenthesis	~	Underline
*	Asterisk	~	Grave Accent
+	Plus	~	Opening Brace
,	Comma		Vertical Line (This graphic is stylized to distinguish it from Logical OR)
-	Hyphen (Minus)	}	Closing Brace
.	Period (Decimal Point)	~	Tilde
/	Slant (Decimal Point)		
:	Colon		
;	Semicolon		

Figure 2-15 United States of America Standard Code for Information Interchange

JUNE 1, 1971

2.5 PROGRAM EXECUTION

Each CE program consists of instructions, index words, and control words specifying the operations to be performed. This information resides in main storage and general registers, and may be operated upon as data.

2.5.1 INSTRUCTION FORMAT

The length of an instruction format can be one, two, or three halfwords. It is related to the number of storage addresses necessary for the operation. An instruction consisting of only one halfword causes no reference to main storage. A two-halfword instruction provides one storage address specification; a three-halfword instruction provide two storage address specifications. All instructions must be located in storage on integral boundaries for halfwords. Figure 2-16 shows five basic instruction formats.

The five basic instruction formats are denoted by the format codes RR, RX, RS, SI, and SS. The format codes express, in general terms, the operation to be performed. RR denotes a register-to-register operation; RX, a register-to-indexed-storage operation; RS, a register-to-storage operation; SI, a storage and immediate-operand operation; and SS, a storage-to-storage operation. An immediate operand is one contained within the instruction.

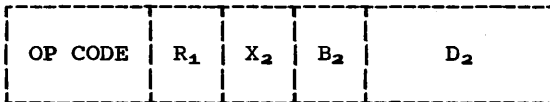
For purposes of describing the execution of instructions, operands are designated as first and second operands and, in the case of load or store multiple, load chain and branch-on-index instructions, third operands. These names refer to the manner in which the operands participate. The operand to which a field in an instruction format applies is generally denoted by the number following the code name of the field, for example, R₁, B₁, L₂, D₂.

In each format, the first instruction halfword consists of two parts. The first byte contains the operation code (op code). The length and format of an instruction are specified by the first two bits of the operation code.

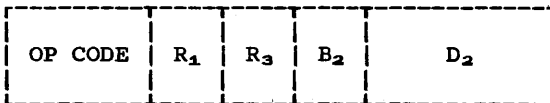
RR Format



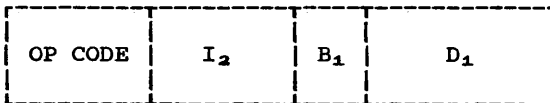
RX Format



RS Format



SI Format



SS Format

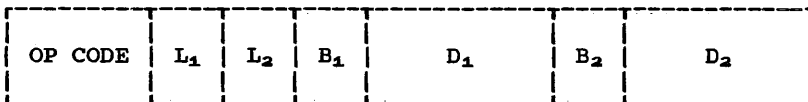


Figure 2-16 Five Basic Instruction Formats

TABLE 2-I INSTRUCTION LENGTH RECORDING

BIT POSITIONS (0-1)	INSTRUCTION LENGTH	INSTRUCTION FORMAT
00	One halfword	RR
01	Two halfwords	RX
10	Two halfwords	RS or SI
11	Three halfwords	SS

The second byte is used either as two 4-bit fields or as a single eight-bit field. This byte can contain the following information:

JUNE 1, 1971

Four-bit operand register specification (R_1 , R_2 , or R_3)
Four-bit index register specification (X_2)
Four-bit mask (M_1)
Four-bit operand length specification (L_1 or L_2)
Eight-bit operand length specification (L)
Eight-bit byte of immediate data (I_2)

In some instructions a four-bit field or the whole second byte of the first halfword is ignored.

The second and third halfwords always have the same format:

Four-bit base register designator (B_1 or B_2), followed by a 12-bit displacement (D_1 or D_2).

2.5.2 ADDRESS GENERATION

For addressing purposes, operands can be grouped in three classes: explicitly addressed operands in main storage, immediate operands placed as part of the instruction stream in main storage, and operands located in the general or floating-point registers.

To permit the ready relocation of program segments and to provide for the flexible specifications of input, output, and working areas, all instructions referring to main storage are given the capacity of employing a full address.

The address used to refer to main storage is generated from the following three binary numbers:

Base Address (B) is a 24-bit number contained in a general register specified by the program in the B field of the instruction. The B field is included in every address specification. The base address can be used as a means of static relocation of programs and data. In array-type calculations, it can specify the location of an array and, in record-type processing, it can identify the record. The base address provides for addressing the entire main storage. The base address may also be used for indexing purposes.

Index (X) is a 24-bit number contained in a general register specified by the program in the X field of the instruction. It is included only in the address specified by the RX instruction format. The RX format instructions permit double indexing; *i.e.*, the index can be used to provide the address of an element within an array.

Displacement (D) is a 12-bit number contained in the instruction format. It is included in every address computation. The displacement provides for relative addressing up to 4095 bytes beyond the element or base address. In array-type calculations the displacement can be used to specify one of many items associated with an element. In the processing of records, the displacement can be used to identify items within a record.

In forming the address, the base address and index are treated as unsigned 24-bit positive binary integers. The displacement is similarly treated as a 12-bit positive binary integer. The three are added as 24-bit binary numbers, ignoring overflow. Since every address includes a base, the sum is always 24 bits long. The address bits are numbered 8-31 corresponding to the numbering of the base address and index bits in the general register.

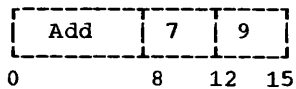
The program may have zeros in the base address, index, or displacement fields. A zero is used to indicate the absence of the corresponding address component. A base or index of zero implies that a zero

JUNE 1, 1971

quantity is to be used in forming the address, regardless of the contents of general register 0. A displacement of zero has no special significance. Initialization, modification, and testing of base addresses and indexes can be carried out by fixed-point instructions, or by BRANCH AND LINK, BRANCH ON COUNT, or BRANCH-ON-INDEX instructions.

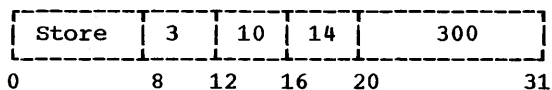
As an aid in describing the logic of the instruction format, examples of two instructions and their related instruction formats follow.

RR Format



Execution of the ADD instruction adds the contents of general register 9 to the contents of general register 7 and the sum of the addition is placed in general register 7.

RX Format



Execution of the STORE instruction stores the contents of general register 3 at a main-storage location addressed by the sum of 300 and the low-order 24 bits of general registers 14 and 10.

2.5.3 SEQUENTIAL INSTRUCTION EXECUTION

Normally, the operation of the CE is controlled by instructions taken in sequence. An instruction is fetched from a location specified by the instruction address in the current program status word (PSW). The instruction address is then increased by the number of bytes in the instruction fetched to address the next instruction in sequence. The instruction is then executed and the same steps are repeated using the new value of the instruction address.

A change from sequential operation may be caused by branching, status switching, interruptions, certain external signals, or manual intervention.

2.5.4 BRANCHING

The normal sequential execution of instructions is changed when reference is made to a subroutine, or when a two-way choice is encountered, or when a segment of coding, such as a loop, is to be repeated. All these tasks can be accomplished with branching instructions (Chapter 7). Subroutine linkage permits not only the introduction of a new instruction address but also the preservation of the return address and associated information.

Decision-making is generally and symmetrically provided by the BRANCH ON CONDITION instruction. This instruction inspects a two-bit condition code that reflects the result of a majority of the arithmetic, logical, and I/O operations. Each of these operations can set the code in any one of four states, and the conditional branch can specify any selection of these four states as the criterion for branching. For example, the

JUNE 1, 1971

condition code reflects such conditions as nonzero, first operand high, equal, overflow, channel busy, zero, etc. Once set, the condition code remains unchanged until modified by an instruction that reflects a different condition code.

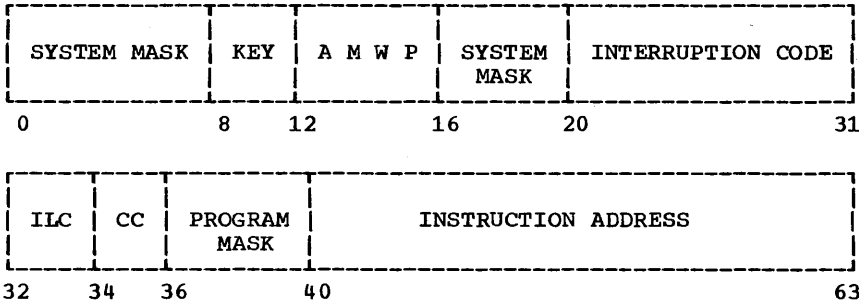
The two bits of the condition code provide for four possible condition-code settings: 0, 1, 2, and 3. The specific meaning of any setting is significant only to the operation setting the condition code.

Loop control can be performed by the conditional branch when it tests the outcome of address arithmetic and counting operations. For some particularly frequent combinations of arithmetic and tests, the instructions BRANCH ON COUNT and BRANCH ON INDEX instructions are provided. These branches, being specialized, provide increased performance for these tasks.

2.5.5 PROGRAM STATUS WORD

A double word, the program status word (PSW), contains the information required for proper program execution. The PSW includes the instruction address, condition code and other fields to be discussed. In general, the PSW is used to control instruction sequencing and to hold and indicate the status of the system in relation to the program currently being executed. The active or controlling PSW is called the "current PSW". By storing the current PSW during an interruption, the status of the CE can be preserved for subsequent inspection. By loading a new PSW or part of a PSW, the state of the CE can be initialized or changed. Figure 2-17 shows the PSW format. (See Chapter 8)

JUNE 1, 1971



- 0 - 7 System mask
 - 0 Multiplexor channel A mask
 - 1 Selector channel 1A mask
 - 2 Selector channel 2A mask
 - 3 Selector channel 3A mask
 - 4 Multiplexor channel B mask
 - 5 Selector channel 1B mask
 - 6 Selector channel 2B mask
 - 7 External mask
- 8 - 11 Protection key
 - 12 USASCII-8 Mode (A)
 - 13 Machine Check Mask (M)
 - 14 Wait State (W)
 - 15 Problem state (P)
- 16 - 19 System mask
 - 16 Selector channel 3B mask
 - 17 Multiplexor channel C mask
 - 18 Selector channel 1C mask
 - 19 Selector channel 2C mask
- 20 - 31 Interruption code
- 32 - 33 Instruction length code (ILC)
- 34 - 35 Condition code (CC)
- 36 - 39 Program mask
 - 36 Fixed-point overflow mask
 - 37 Decimal overflow mask
 - 38 Exponent underflow mask
 - 39 Significance mask
- 40 - 63 Instruction address

Figure 2-17 Program Status Word Format

2.5.6 INTERRUPTIONS

The interruption system permits the CE to change state as a result of conditions external to the system, in input/output (I/O) units or in the CE itself. Five classes of interruption conditions are possible: I/O, program, supervisor call, external, and machine check. (Chapter 9)

Each class has two related PSWs called "old" and "new" in unique preferential main-storage locations (Figure 2-18). The location of the preferential-storage area is determined by a base address in each CE, thus allowing different areas to be assigned for each CE. All addresses generated automatically by the CE are constructed using the current value of the preferential-storage base address. In all classes, an interruption involves merely storing the current PSW in its "old" position and making the PSW at the "new" position the current PSW. The "old" PSW holds all necessary status information of the system existing at the time of the interruption. If, at the conclusion of the

JUNE 1, 1971

interruption routine, there is an instruction to make the old PSW the current PSW, the system is restored to the state prior to the interruption and the interrupted routine continues. The following table lists the main-storage location in the preferential-storage area. The addresses shown are relative to the preferential-storage base address.

PERMANENT STORAGE ASSIGNMENT			
	ADDRESS	LENGTH	PURPOSE
0	0000 0000	double word	Initial program loading PSW
8	0000 1000	double word	Initial program loading CCW1
16	0001 0000	double word	Initial program loading CCW2
24	0001 1000	double word	External old PSW
32	0010 0000	double word	Supervisor call old PSW
40	0010 1000	double word	Program old PSW
48	0011 0000	double word	Machine check old PSW
56	0011 1000	double word	Input/output old PSW
64	0100 0000	double word	Channel status word
72	0100 1000	word	Channel address word
76	0100 1100	word	Unused
80	0101 0000	word	Timer
84	0101 0100	word	Unused
88	0101 1000	double word	External new PSW
96	0110 0000	double word	Supervisor call new PSW
104	0110 1000	double word	Program new PSW
112	0111 0000	double word	Machine check new PSW
120	0111 1000	double word	Input/output new PSW
128	1000 0000		Diagnostic logout area*

* The diagnostic logout area extends through byte location 511.

Figure 2-18 Preferential Storage Assignments

Interruptions are taken only when the CE is interruptable for the interruption source. The system mask, program mask, and machine check mask bits in the PSW may be used to mask certain interruptions. When masked off, an interruption either remains pending or is ignored. The system mask may keep I/O and external interruptions pending, the program mask may cause four of the program interruptions to be ignored, and the machine-check mask may cause machine-check interruptions to remain pending. Other interruptions cannot be masked off. With the exception of CONVERT and SORT SYMBOLS, CONVERT WEATHER LINES, REPACK SYMBOLS, and DELAY, an interruption always takes place after one instruction interpretation is finished and before a new instruction execution is started. However, the occurrence of an interruption may affect the execution of the current instruction. To permit proper programmed action following an interruption, the cause of the interruption is identified and provision is made to locate the last executed instruction.

2.5.6.1 Input/Output Interruption

An I/O interruption provides a means by which the CE responds to conditions in the IOCE channels, to which it is configured, and I/O units.

An I/O interruption can occur only when the related channel is not masked. The address of the channel and I/O unit involved are recorded in the old PSW. Further information concerning the I/O action is preserved in the channel status word (CSW) which is stored in location 64 of the preferential-storage area during the interruption.

2.5.6.2 Program Interruption

Unusual conditions encountered in a program create program interruptions. They include incorrect operands and operand specifications, as well as exceptional results. Failure of an IOCE to gain access to a preferential storage area (PSA) creates a PSA lockout exception which is also recognized as a program interruption. The interruption code identifies the interruption cause. The different causes that may occur are shown in Figure 2-19.

Interruption Code	Program Interruption Cause	Mask Bit
1 00000001	Operation	
2 00000010	Privileged operation	
3 00000011	Execute	
4 00000100	Protection	
5 00000101	Addressing	
6 00000110	Specification	
7 00000111	Data	
8 00001000	Fixed-point overflow	36
9 00001001	Fixed-point divide	
10 00001010	Decimal overflow	37
11 00001011	Decimal divide	
12 00001100	Exponent overflow	
13 00001101	Exponent underflow	38
14 00001110	Significance	39
15 00001111	Floating-point divide	
16 00010000	IOCE-3 PSA lockout	
32 00100000	IOCE-2 PSA lockout	
64 01000000	IOCE-1 PSA lockout	
128 10000000	SE stopped	

Figure 2-19 Interruption Code for Program Interruption

2.5.6.3 Supervisor Call Interruption

This interruption occurs as a result of execution of the instruction, SUPERVISOR CALL. Eight bits from the instruction format are placed in the interruption code of the old PSW, permitting an identification to be associated with the interruptions. A major use for the instruction SUPERVISOR CALL is to switch from the problem-state to the supervisor state. This interruption may also be used for other modes of status-switching. (Chapter 8)

2.5.6.4 External Interruption

The external interruption provides the means by which the CE responds to signals from the interrupt switch on the system console, configuration console or CE control panel, the timer, the external signals of Direct Control (CE or IOCE-processor), and the diagnose accessible register (DAR). (Chapter 9)

With the exception of abnormal condition signals, which are also masked by the DAR mask and may be subject to configuration control, an external interruption can occur only when the system mask bit 7 is one.

The source of the interruption is identified by the interruption code in bits 20-31 of the PSW. (Figure 2-20)

JUNE 1, 1971

Interruption Code Bit	External Interruption Cause	Mask Bit
20	CE1 Read direct	7
21	CE1 Write direct	7
22	CE2 Read direct	7
23	CE2 Write direct	7
24	Timer	7
25	Interrupt switch	7
26	CE3 Read direct	7
27	CE3 Write direct	7
28	CE4 Read direct	7
29	CE4 Write direct	7
30	Processor Interruption Register	7
31	Diagnose accessible register	7

Figure 2-20 Interruption Code for External Interruption

2.5.6.5 Machine Check Interruption

The occurrence of a machine check (CE malfunction) terminates the current instruction, initiates a CE diagnostic procedure, and subsequently causes the machine-check interruption. The occurrence of a machine check (IOCE malfunction) terminates the current instruction if an I/O instruction, initiates an IOCE diagnostic procedure, and subsequently causes the machine-check interruption. The occurrence of a machine check (Read Direct Timeout) terminates the current instruction and subsequently causes the machine check interruption; no diagnostic procedure is performed. A machine-check interruption can occur only when PSW mask bit 13 is one. A machine check cannot be caused by erroneous data or instructions. The diagnostic scan is performed into the logout area which starts at location 128 and extends through location 511 in each of the preferential-storage areas. Proper execution of these steps depends on the nature of the machine check (Chapter 9). The interruption code identifies the interruption cause. (Figure 2-21)

Interruption Code	Machine Check Interruption Cause	Mask Bit
0	CE Malfunction	13
1	IOCE-1 Malfunction	13
2	IOCE-2 Malfunction	13
3	IOCE-3 Malfunction	13
4	Read Direct Timeout	13

Figure 2-21 Interruption Code for Machine Check Interruption

2.5.6.6 Priority of Interruption

During execution of an instruction, several interruption requests may occur. Simultaneous interruption requests are honored in the following predetermined order:

- Machine Check
- Program or Supervisor Call

JUNE 1, 1971

External Input/Output

The program and supervisor-call interruptions are mutually exclusive and cannot occur at the same time.

When more than one interruption cause requests service, the action consists of storing the old PSW and fetching the new PSW belonging to the interruption which is taken first. This new PSW subsequently is stored without any instruction execution and the next interruption PSW is fetched. This process continues until no more interruptions are to be serviced. When the last interruption request has been serviced, instruction execution is resumed using the PSW last fetched. The order of execution of the interruption subroutines is, therefore, the reverse of the order in which the interruption requests are serviced.

Thus, the most important interruption subroutines -- I/O, external, program or supervisor call -- are actually serviced first. Machine check, when it occurs, does not allow any other interruptions to be taken.

2.5.6.7 Program States

The overall status of each CE is determined by several program-state alternatives, each of which can be changed independently to its opposite and most of which are indicated by a bit or bits in the PSW. The four types of program-state alternatives, which determine the over-all status, are named stopped or operating, running or waiting, masked or interruptable, and supervisor or problem state. These states differ in the way they affect the CE functions, and the manner in which their status is indicated and switched. All program states are independent of each other in their function, indication, and status-switching (Chapter 8).

Stopped or Operating States: The stopped state is entered and left by either programmed or manual procedures. Instructions are not executed, interruptions are not accepted, and the timer is not updated. In the operating state, the CE is capable of executing instructions, being interrupted, and the timer is allowed to update.

Running or Waiting State: In the running state, instruction fetching and execution proceed in the normal manner. The wait state is normally entered by the program to await an interruption, for example, an I/O interruption or operator intervention from the console. In the wait state, no instructions are processed, the timer is updated, and I/O and external interruptions are accepted, unless masked. Running or waiting state is determined by the setting of bit 14 in the PSW.

Masked or Interruptable State: The CE may be interruptable or masked for the system, program, and machine-check interruptions. When the CE is interruptible for a class of interruptions, these interruptions are accepted. When the CE is masked, the system and machine-check interruptions remain pending, whereas program interruptions are ignored. The interruptable states of the CE are changed by changing the mask bits of the PSW.

Supervisor or Problem State: In the problem state, all I/O instructions and a group of control instructions are invalid. In the supervisor state, all instructions are valid. The choice of problem or supervisor state is determined by bit 15 of the PSW.

JUNE 1, 1971

2.6 PROTECTION

Store protection is provided to protect the contents of certain areas of main storage or display storage from destruction due to storing of erroneous information during the execution of a program. Similarly, fetch protection is provided to protect a program from making erroneous use of the contents of certain areas of storage as data or instructions. This protection is achieved by identifying blocks of storage with a storage key and comparing this key with a protection key supplied with the address of the location to be accessed. The detection of a mismatch results in a protection interruption (Chapter 8).

For protection purposes main storage and display storage are divided into blocks of 2048 bytes. A five-bit storage key is associated with each block. When data are stored in a storage block, the four high-order bits of the storage key are compared with the protection key. When data are fetched, the fetch-protection (fifth, i.e., low-order) bit is inspected. When the fetch-protection bit is one, the four high-order bits of the storage key are compared with the protection key. The protection key of the current PSW is used as the comparand when a storage access is specified by an instruction. When a main storage access is specified by a channel operation, a protection key supplied by the channel is used as the comparand. The keys are said to match when they are equal or when either one is zero.

The storage key is not part of addressable storage. The key is changed by SET STORAGE KEY and is inspected by INSERT STORAGE KEY. The protection key in the PSW occupies bits 8-11 of that control word. The protection key of a channel is recorded in bits 0-3 of the CSW, which is stored as a result of the channel operation. When a protection mismatch due to an instruction is detected, the execution of this instruction is suppressed or terminated, and the program execution is altered by an interruption. The content of the protected storage location always remains unchanged on a store-protection violation, and is never loaded into an addressable register or moved to another storage location on a fetch-protection violation.

Protection mismatch due to an I/O operation causes the data transmission to be terminated in such a manner that the content of the protected main or display storage location remains unchanged on a store-protection violation, and is not recorded on an output medium on a fetch-protection violation. The mismatch is indicated in the CSW stored as a result of the operation.

Storage protection applies to main storage (SE) and display storage (DE) references only. This protection is effective in main storage for all CE and IOCE accesses and in display storage for CE accesses only. Storage protection is not provided for IOCE MACH storage access (i.e., during I/O operations or IOCE-processor operation), nor is it provided within display storage for display generator accesses.

2.7 TIMER

The timer is provided as an interval timer and may be programmed to maintain the time of day. The timer consists of a full word in preferential-main-storage location 80. The timer word is counted down at a rate of 60 cycles per second. The timer word is treated as a signed integer following the rules of fixed-point arithmetic. An external interruption condition is signaled when the value of the timer word goes from positive to negative. The full cycle time of the timer is 15.5 hours. (Chapter 9)

The timer is updated automatically at the end of each instruction execution but is not updated in the stopped state. The timer is changed

JUNE 1, 1971

by addressing preferential-storage location 80. As an interval timer, the timer is used to measure elapsed time over relatively short intervals. It can be set to any value at any time.

The timer is not updated whenever the CE is in state zero with the Disable Timer Switch on.

2.8 DIRECT CONTROL

CE Direct Control provides two instructions -- READ DIRECT and WRITE DIRECT -- and external interruption lines. The instructions provide for the transfer of a single byte of information between computing elements. WRITE DIRECT may also be used to cause the external start or external stop of another CE, the logout of another CE or an IOCE, or the start, stop, or interruption of an IOCE-processor. (See Chapter 9.)

IOCE-processor Direct Control provides one instruction -- WRITE DIRECT -- and an external interruption line to computing elements. This instruction is used by an IOCE-processor to request an external interruption in the controlling CE.

Each of the external signal lines, when pulsed, sets up conditions for an external interruption.

2.9 MULTIPROCESSOR OPERATION

2.9.1 COMPUTING ELEMENT

The design of the IBM 9020D and 9020E Systems permit communication between individual CE's at several transmission rates. The communication is possible through shared control units, through CE Direct Control and through shared storage. CE Direct Control, described in the previous section, can be used to signal from one CE to another and from a CE to an IOCE. Multiprocessor operation provides direct address relocation, malfunction indication, and electronic CE initialization. (See Chapter 8.)

The relocation procedure applies to the first 4,096 bytes of storage. This area contains all permanent storage assignments and, generally, has special significance to supervisory programs. The relocation is accomplished by a preferential-storage base address register that determines the location of this storage area. Instructions are provided for loading and storing this base address.

To alert one CE to the possible malfunction of another CE, an element check (ELC) out-signal is provided, which can serve as an external interruption to another CE.

Finally, provision is made for an external start or external stop initiated by a signal from another CE.

2.9.2 IOCE-PROCESSOR OPERATION

The design of the IBM 9020 System permits the IOCE to perform processing in addition to controlling I/O operations. From the user's viewpoint, the IOCE-processor operation and I/O operations are independent of each other and can be performed concurrently.

The IOCE-processor is under control of a CE. Processing (i.e., instruction execution) is initiated by a controlling CE. As is true for I/O operations, once initiated, IOCE-processing is independent of the CE. The IOCE-processor can execute a subset of the IBM 9020 System

JUNE 1, 1971

instruction set with a few special instructions added. The IOCE-processor has access to main storage (as well as the internal (MACH) storage in the IOCE) for instruction execution and data sets. An IOCE Direct Control facility is provided to allow the IOCE-processor to signal external interruption requests to its controlling CE. (See Section 2.8). The IOCE-processor cannot initiate I/O operations. (See Appendix H for more details on the IOCE-processor.)

2.10 DISPLAY PROCESSOR OPERATION

2.10.1 COMPUTING ELEMENT

The design of the IBM 9020D and 9020E Systems computing element provides several task-oriented instructions for utilization in the 9020E Display Channel Processor environment. These instructions provide for storage paging, processing of radar and weather line input data, and for data management of display storage. These instructions are described in Chapter 8.

2.11 INPUT/OUTPUT

The following information is introductory in nature. For thorough definition of the input/output system, see "Input/Output Operations". (Chapter 10)

2.11.1 INPUT/OUTPUT DEVICES AND CONTROL UNITS

Input/output operations involve the transfer of information to or from storage and an I/O device. Input/output devices include such equipment as card readers and punches, magnetic tape units, printer-keyboard devices, page printers, peripheral adapter modules (PAM), data adapter units, channel-to-channel adapters, storage control units, etc.

Many I/O devices function with an external document, such as a punched card, disk pack or a reel of magnetic tape. Some I/O devices handle only electrical signals, such as those found in printer-keyboard devices. In either case, I/O device operation is regulated by a control unit. The control-unit function may be housed with the I/O device, the channels, or a separate control unit may be used. In all cases, the control-unit function provides the logical and buffering capabilities necessary to operate the associated I/O device. From the programming point of view, most control-unit functions merge with I/O device functions.

Each control unit functions only with the I/O device for which it is designed, but each control unit has standard-signal connections with regard to the channel to which it is attached.

2.11.2 INPUT/OUTPUT INTERFACE

To allow the IOCE to control a wide variety of I/O devices, all control units are designed to respond to a standard set of signals from the channel. This control-unit-to-channel connection is called the I/O Interface Channel to Device Control Unit. It enables the CE to handle all I/O operation with only five instructions.

2.11.3 CHANNELS

The IOCE channels connect with the CE and main storage, and, by means of the I/O interface, with control units or other 9020 Systems. Each

JUNE 1, 1971

channel has facilities for:

- Accepting I/O instructions,
- Addressing devices specified by I/O instructions,
- Fetching channel control information from storage,
- Decoding control information,
- Testing control information for validity,
- Executing control information,
- Providing control signals to the I/O interface,
- Accepting control-response signals from the interface,
- Buffering data transfers,
- Checking parity of bytes transferred,
- Counting the number of bytes transferred,
- Accepting status information from I/O devices,
- Maintaining channel-status information,
- Sending requested status information to main storage,
- Sequencing interruption requests from I/O devices, and
- Signaling interruptions to the CE.

The IBM 9020 System IOCE has two types of channels: multiplexor and selector. The channel facility necessary to sustain an operation with an I/O device is called a subchannel. The selector channel has one subchannel; the multiplexor channel has multiple subchannels.

Channels have two modes of operation: burst and multiplex.

In the burst mode, the data-transfer facilities of the channel are monopolized for the duration of transfer of a burst of data. Other devices attached to the channel cannot transfer data until the burst ceases. The selector channel functions only in the burst mode.

The multiplexor channel functions in either the burst mode or in the multiplex mode. In the multiplex mode, the multiplexor channel can sustain concurrent I/O operations on several subchannels. Bytes of data associated with different I/O devices are interleaved and routed to or from the desired locations in storage. The I/O interface is time-shared by a number of concurrently operating I/O devices, each of which uses its own subchannel.

Some I/O devices can operate only in burst mode. Other I/O devices have a manual switch in the control unit that may be set to a burst-mode or to a multiplex-mode position, when attached to a multiplexor channel. When attached to a selector channel, an I/O device can operate only in burst mode.

2.11.4 INPUT/OUTPUT INSTRUCTIONS

The IBM 9020D and 9020E Systems use only five I/O instructions:

- START I/O
- TEST I/O
- HALT I/O
- TEST CHANNEL
- SET PCI

Input/output instructions can be executed only while the CE is in the supervisor state.

2.11.4.1 Start I/O

The START I/O instruction is used to initiate an I/O operation. The address part of the instruction specifies the channel and I/O device.

JUNE 1, 1971

2.11.4.2 Test Channel

The TEST CHANNEL sets the condition code in the PSW to indicate the state of the channel addressed by the instruction. The condition code then indicates channel available, interruption condition in channel, channel working, or channel not operational.

2.11.4.3 Test I/O

The TEST I/O instruction sets the condition code in the PSW to indicate the state of the addressed channel, subchannel, and I/O device, and may cause a CSW to be stored. The instruction may be used to clear I/O interruption conditions, selectively by device.

2.11.4.4 Halt I/O

The HALT I/O instruction terminates a channel operation.

2.11.4.5 Set PCI

This instruction requests an interruption from the channel or subchannel controlling the addressed device. The requested interruption if permitted, which normally will occur after this or the following instruction, will not affect channel and subchannel operation.

2.11.5 INPUT/OUTPUT OPERATION INITIATION

All I/O operations are initiated by a START I/O instruction. If the IOCE channel facilities are free, START I/O is accepted and the CE continues its program. The channel independently selects the I/O device specified by the instruction.

2.11.5.1 Channel Address Word

Successful execution of START I/O causes the channel to fetch a channel address word (CAW) from the CE preferential main-storage location 72. The CAW specifies the byte location in storage where the channel program begins.

Figure 2-22 shows the format for the CAW. Bits 0-3 specify the storage-protection key that will govern the I/O operation. Bits 4-7 must contain zeros. Bits 8-31 specify the location of the first channel command word (CCW).

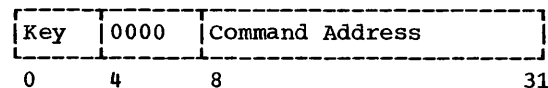


Figure 2-22 Channel Address Word Format

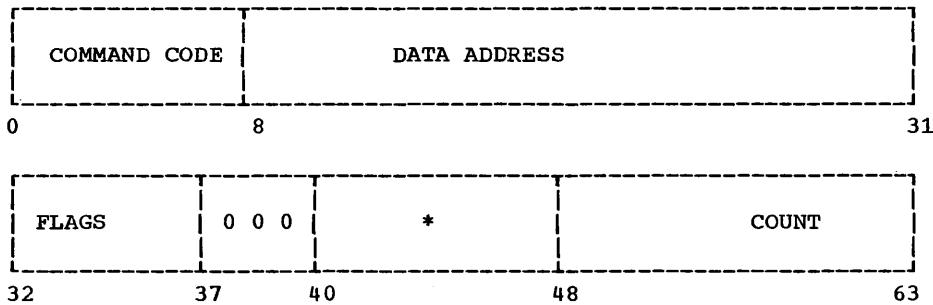
2.11.5.2 Channel Command Word

The byte location specified by the CAW is the first of eight bytes of information that the channel fetches from storage. These 64 bits of information are called a channel command word (CCW). Only the START I/O instruction may cause the channel to fetch CCWs. (Figure 2-23)

One or more CCWs make up the channel program that directs channel operations. If more than one CCW is to be used, each CCW points to the next CCW to be fetched, except for the last CCW in the chain, which identifies itself as the last in the chain. A channel command word can specify one of seven commands:

JUNE 1, 1971

Read
Write
Read Backward
Control
Sense
Transfer In Channel
Search



Bits 0-7 specify the command code.
Bits 8-31 specify the location of a byte in storage.
Bits 32-36 are flag bits.

Bit 32 causes the address portion of the next CCW to be used.

Bit 33 causes the command code and data address in the next CCW to be used.

Bit 34 causes a possible incorrect length indication to be suppressed.

Bit 35 suppresses the transfer of information to storage.

Bit 36 causes an interruption as soon as it is decoded.

Bits 37-39 must contain zeros.

Bits 40-47 are ignored

Bits 48-63 specify the number of bytes involved in the operation.

Figure 2-23 Channel Command Word Format

2.11.6 INPUT/OUTPUT COMMANDS

2.11.6.1 Read

The Read command causes data to be read from the selected I/O device and defines the area in storage to be used.

2.11.6.2 Write

The Write command causes a write operation on the selected I/O device and defines the data in storage to be written.

2.11.6.3 Read Backward

The Read Backward command causes a read operation in which the characters are read from the external document in reverse order by the I/O device. Bytes read backward are placed in descending storage locations.

JUNE 1, 1971

2.11.6.4 Control

The control command contains information used to control the selected I/O device. This control information is called an order. Orders are peculiar to the particular I/O device in use; orders can specify such functions as rewinding a tape unit, or line skipping on a printer. A control command may cause mechanical motion by an I/O device, or it may specify a function altogether electronic in nature, such as setting the recording density for a tape unit operation.

The general relationship of I/O instructions, commands, and orders is shown in Figure 2-24.



Figure 2-24 Relationship of I/O Instructions, Commands, and Orders

2.11.6.5 Sense

The Sense command specifies the beginning storage location to which sense information is transferred from the selected control unit. One or more bytes of sense data may be specified, depending upon the type of I/O device. The sense data provides detailed information concerning the selected I/O device, such as a stacker-full condition of a card reader or a file-protected condition of a reel of magnetic tape on a tape unit. Sense data have significance peculiar to the type of I/O device involved.

2.11.6.6 Transfer In Channel

The Transfer In Channel (TIC) command specifies the location of the next CCW to be fetched and used by the channel. The TIC command is used whenever the programmer wants to specify a CCW that is not located at the next higher double-word location in storage. The TIC command permits a programmer to cause execution of any CCW, including a CCW immediately preceding a TIC command, except that the channel will not permit a TIC command to specify execution of another TIC command. Also, the CAW may not address a TIC command.

External documents, such as punched cards, disk packs or magnetic tape, may carry CCWs that the channel can use to govern reading of the external document being read.

2.11.6.7 Search

The Search command specifies a comparison between data from a storage location and the specified area (Identifier, Home Address, or Key) on the Disk Pack in the Direct Access Storage Facility (DASF). The search terminates when the specified condition has been satisfied or when the end of the Search occurs. The Search does not cause any transfer of data. The Search command is normally followed by a Read or Write command which performs the data transfer.

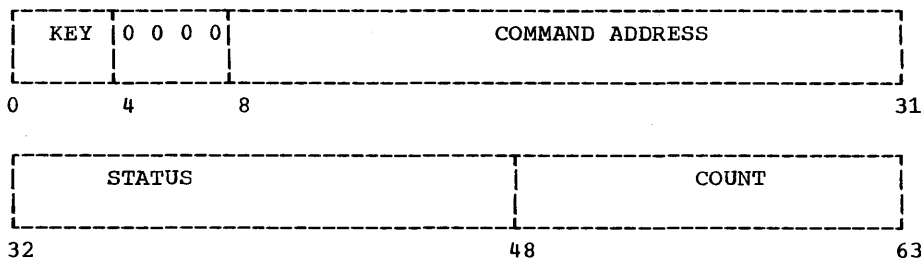
2.11.7 INPUT/OUTPUT TERMINATION

Input/output operations terminate with the device and channel signaling end of operation and a request for an I/O interruption to the CE.

A command can be rejected during an attempt to execute a START I/O by a busy condition, by a channel programming error, etc. The condition code set in the PSW by an unsuccessful START I/O instruction will indicate one of the following: that a channel status word (CSW) has been stored to detail the conditions that precluded initiation of the I/O operation, that the equipment is busy, or that the addressed equipment is not operational.

2.11.7.1 Channel Status Word

The channel status word (CSW) provides information about the termination of an I/O operation. It can be formed or reformed by START I/O, SPCI, TEST I/O, HALT I/O, or by an I/O interruption. The instruction TEST CHANNEL does not affect the CSW. Figure 2-25 shows the CSW format.



Bits 0-3 contains the storage-protection key used in the operation.
 Bits 4-7 contain zeros.
 Bits 8-31 specify the location of the last CCW used.
 Bits 32-47 contain an I/O device-status byte and a channel-status byte. The status bytes provide such information as data check, chaining check, control-unit end, etc.
 Bits 48-63 contain the residual count of the last CCW used.

| Figure 2-25 Channel Status Word Format

2.11.8 INPUT/OUTPUT INTERRUPTIONS

Input/output interruptions are caused by termination of an I/O operation or by operator intervention at the I/O device. Input/output interruptions enable the CE to provide appropriate programmed response to conditions which occur in I/O devices or channels.

Input/output interruptions have priority sequences, one for the I/O devices attached to a channel, and another sequence for channel interruptions. A channel establishes interruption priority for its associated I/O devices before initiating an I/O interruption signal to the CE. Conditions responsible for I/O interruption requests are preserved in the I/O devices or channels until they are accepted by the CE.

2.12 SYSTEM CONTROL FACILITIES

Each CE control panel provides the switches, keys, and lights necessary to operate and control a subsystem, while the system console (9020D system) or configuration console (9020E system) provides these facilities for their respective systems. The need for operator manipulation of manual controls is held to a minimum by the system design and

JUNE 1, 1971

the governing supervisory program. The result is fewer and less serious operator errors.

2.12.1 CONTROL PANEL FUNCTIONS

The main functions provided by the CE control panel, the configuration console, and the system console are the ability to: store and display information in main storage, in registers, and in the PSW; and load initial program information. The CE and system console controls are divided into two sections -- operator control and intervention, and maintenance control. The configuration console, in addition to the above functions, provides a communications path for configuring and monitoring the status of the DG and RKM units and the data adapter units.

2.12.1.1 Store and Display

The store-and-display function permits manual intervention in the progress of a program. The function may be provided by a supervisory program in conjunction with proper I/O equipment and the interrupt switch. Or, the system-control-panel facilities may be used to place the CE in the stopped state, and then to store and display information in main storage, in general and floating-point registers, and in the instruction-address portion of the PSW.

2.12.1.2 Initial Program Load

The initial-program-loading (IPL) procedure is used to begin or renew system or subsystem operation. The load switch is pressed after first selecting a storage element and an input device with the main storage select switches and the load-unit switches. This causes a read operation at the selected input device. Six words of information are read into main storage and are used as channel control words and as a PSW that controls subsequent operations.

2.12.2 OPERATOR SECTION

The main functions provided are the control and indication of power, the indication of system status, and operator-to-machine communication (see Chapter 11). These are implemented as follows:

JUNE 1, 1971

Element master power-off pull switch **
| Power-on/off switch*
| Power sequence complete light*
360 mode switch-indicator
Interrupt switch*
Wait light*
Manual light*
System light
Test light
Load light*
Load-unit switches*
Load switch*
Main storage select switch*
System interlock key-operated switch*
PSW restart switch
State Three light*
State Two light*
State One light*
State Zero light*
| Lamp test switch*
Stop switch*
Start switch*
Rate switch*
Storage-select switches*
Address switches (24)*
Data switches (64)* (32 on system and configuration consoles)
Store switch*
Display switch*
Set IC switch*
Address-compare switch***
Address-Compare enable switch***
Computing Element Select switch***
Control CE switch***
Control CE activate switch***
Sense switches***
Reader/Punch-Printer select switch***
Reader/Punch-Printer enable switch-indicator***
Printer keyboard select switch****
Printer keyboard enable switch-indicator****
Console reset switch***
| Bell reset switch****
| Buzzer/Bell reset switch*****
| Buzzer reset switch****
All-Stop switch***
Invalid selection light***
Test mode light****
| Fault reset switch*****

* These functions are also provided on the System Console and the Configuration Console
** System Emergency Power Off switch provided on System Console and Configuration Console only
| *** Provided on System Console and Configuration Console only
| **** Provided on System Console only
| ***** Provided on Configuration Console only

2.12.3 MAINTENANCE SECTION

| This section of the CE control panel provides the controls intended
| only for maintenance use. Maintenance controls may also be available on
| individual storage, channel, and control-unit equipment.

JUNE 1, 1971

CHAPTER 3: FIXED-POINT ARITHMETIC

The fixed-point instruction set performs binary arithmetic on operands serving as addresses, index quantities, and counts, as well as fixed-point data. In general, both operands are signed and 32 bits long. Negative quantities are held in two's-complement form. One operand is always in one of the 16 general registers; the other operand may be in main storage or in a general register.

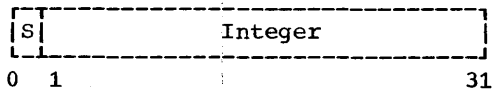
The instruction set provides for loading, adding, subtracting, comparing, multiplying, dividing, and storing, as well as for the sign control, radix conversion, and shifting of fixed-point operands.

The condition code is set as a result of all sign-control, add, subtract, compare, and shift operations.

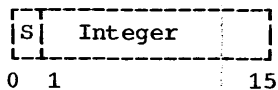
3.1 DATA FORMAT

Fixed-point numbers occupy a fixed-length format consisting of a one-bit sign followed by the integer field. When held in one of the general registers, a fixed-point quantity has a 31-bit integer field and occupies all 32 bits of the register. Some multiply, divide, and shift operations use an operand consisting of 64 bits with a 63-bit integer field. These operands are located in a pair of adjacent general registers and are addressed by an even address referring to the left-most register of the pair. The sign-bit position of the rightmost register contains part of the integer. In register-to-register operations the same register may be specified for both operand locations.

Fullword Fixed-Point Number



Halfword Fixed-Point Number



Fixed-point data in main storage occupy a 32-bit word or a 16-bit halfword, with a binary integer field of 31 or 15 bits, respectively. The conversion instructions use a 64-bit decimal field. These data must be located on integral storage boundaries for these units of information; i.e., double-word, fullword, or halfword operands must be addressed with three, two, or one low-order address bit(s) set to zero.

A halfword operand in main storage is extended to a fullword as the operand is fetched from storage. Subsequently, the operand participates as a fullword operand.

In all discussions of fixed-point numbers in this publication, the expression "32-bit signed integer" denotes a 31-bit integer with a sign bit, and the expression "64-bit signed integer" denotes a 63-bit integer with a sign bit.

3.2 NUMBER REPRESENTATION

All fixed-point operands are treated as signed integers. Positive numbers are represented in true binary notation with the sign bit set to zero. Negative numbers are represented in two's-complement notation with a one in the sign bit. The two's-complement of a number is obtained by inverting each bit of the number and adding a one in the low-order bit position.

This type of number representation can be considered the low-order portion of an infinitely long representation of the number. When the number is positive, all bits to the left of the most significant bit of the number, including the sign bit, are zeros. When the number is negative, all these bits, including the sign bit, are ones. Therefore, when an operand must be extended with high-order bits, the expansion is achieved by prefixing a field in which each bit is set equal to the high-order bit of the operand.

Two's-complement notation does not include a negative zero. It has a number range in which the set of negative numbers is one larger than the set of positive numbers. The maximum positive number consists of an all-one integer field with a sign bit of zero, whereas the maximum negative number (the negative number with the greatest absolute value) consists of an all-zero integer field with a one-bit for sign.

The CE cannot represent the complement of the maximum negative number. When an operation, such as a subtraction from zero, produces the complement of the maximum negative number, the number remains unchanged, and a fixed-point overflow exception is recognized. An overflow does not result, however, when the number is complemented and the final result is within the representable range. An example of this case is a subtraction from minus one. The product of two maximum negative numbers is representable as a double-length positive number.

The sign bit is leftmost in a number. In an arithmetic operation, a carry out of the integer field changes the sign. However, in algebraic shifting the sign bit does not change even if significant high-order bits are shifted out.

PROGRAMMING NOTE

Two's-complement notation is particularly suited to address computation and multiple-precision arithmetic.

The two's-complement representation of a negative number may be considered the sum of the integer part of the field, taken as a positive number, and the maximum negative number. Hence, in multiple-precision arithmetic the low-order fields should be treated as positive numbers. Also, when negative numbers are shifted to the right, the resulting rounding, if any, is toward minus infinity and not toward zero.

3.3 CONDITION CODE

The results of fixed-point sign-control, add, subtract, compare, and shift operations are used to set the condition code in the program status word (PSW). All other fixed-point operations leave this code undisturbed. The condition code can be used for decision-making by subsequent branch-on-condition instructions.

The condition code can be set to reflect three types of results for fixed-point arithmetic. For most operations the states 0, 1, or 2 indicate a zero, less than zero, or greater than zero content of the result register, while the state 3 is used when the result overflows.

JUNE 1, 1971

For a comparison, the states 0, 1, or 2 indicate that the first operand is equal, low, or high.

For ADD LOGICAL and SUBTRACT LOGICAL, the codes 0 and 1 indicate a zero or nonzero result register content in the absence of a logical carry out of the sign position; the codes 2 and 3 indicate a zero or nonzero result register content with a logical carry out of the sign position.

TABLE 3-I CONDITION CODE SETTING FOR FIXED-POINT ARITHMETIC

	0	1	2	3
ADD (halfword/fullword)	zero	< zero	> zero	overflow
ADD LOGICAL	zero, no carry	not zero, no carry	zero, carry	not zero, carry
COMPARE (halfword/fullword)	equal	low	high	--
LOAD AND TEST	zero	< zero	> zero	--
LOAD COMPLEMENT	zero	< zero	> zero	overflow
LOAD NEGATIVE	zero	< zero	--	--
LOAD POSITIVE	zero	--	> zero	overflow
SHIFT LEFT DOUBLE	zero	< zero	> zero	overflow
SHIFT LEFT SINGLE	zero	< zero	> zero	overflow
SHIFT RIGHT DOUBLE	zero	< zero	> zero	--
SHIFT RIGHT SINGLE	zero	< zero	> zero	--
SUBTRACT (halfword/fullword)	zero	< zero	> zero	overflow
SUBTRACT LOGICAL	--	not zero, no carry	zero, carry	not zero, carry

3.4 INSTRUCTION FORMAT

Fixed-point instructions use the following three formats:

RR Format

OP	R ₁	R ₂
0	8	12 15

RX Format

Op Code	R ₁	X ₂	B ₂	D ₂
0	8	12	16	20 31

RS Format

Op Code	R ₁	R ₃	B ₂	D ₂
0	8	12	16	20 31

JUNE 1, 1971

In these formats, R_1 specifies the address of the general register containing the first operand. The second operand location, if any, is defined differently for each format.

In the RR format, the R_2 field specifies the address of the general register containing the second operand. The same register may be specified for the first and second operand.

In the RX format, the contents of the general registers specified by the X_2 and B_2 fields are added to the content of the D_2 field to form an address designating the storage location of the second operand.

In the RS format, the content of the general register specified by the B_2 field is added to the content of the D_2 field to form an address. This designates the storage location of the second operand in LOAD MULTIPLE and STORE MULTIPLE. In the shift operations, the address specifies the number of bits of the shift. The R_3 field specifies the address of a general register in LOAD MULTIPLE and STORE MULTIPLE and is ignored in the shift operations.

A zero in an X_2 or B_2 field indicates the absence of the corresponding address component.

An instruction can specify the same general register both for address modification and for operand location. Address modification is always completed prior to operation execution.

Results replace the first operand, except for STORE and CONVERT TO DECIMAL, where the result replaces the second operand.

The contents of all general registers and storage locations participating in the addressing or execution part of an operation remain unchanged, except for the storing of the final result.

3.5 INSTRUCTIONS

The fixed-point arithmetic instructions and their mnemonics, formats, and operation codes are listed in the following table. The table also indicates when the condition code is set and the exceptional conditions which cause a program interruption.

JUNE 1, 1971

TABLE 3-II FIXED-POINT INSTRUCTIONS

NAME	MNEMONIC	TYPE	CC	EXCEPTIONS	CODE	SECTION
ADD	AR	RR	C	IF	1A	3.5.8
ADD	A	RX	C	P,A,S IF	5A	3.5.8
ADD HALFWORD	AH	RX	C	P,A,S IF	4A	3.5.9
ADD LOGICAL	ALR	RR	C		1E	3.5.10
ADD LOGICAL	AL	RX	C	P,A,S	5E	3.5.10
COMPARE	CR	RR	C		19	3.5.14
COMPARE	C	RX	C	P,A,S	59	3.5.14
COMPARE HALFWORD	CH	RX	C	P,A,S	49	3.5.15
CONVERT TO BINARY	CVB	RX		P,A,S,D,IK	4F	3.5.19
CONVERT TO DECIMAL	CVD	RX		P,A,S	4E	3.5.20
DIVIDE	DR	RR		S IK	1D	3.5.18
DIVIDE	D	RX		P,A,S IK	5D	3.5.18
LOAD	LR	RR			18	3.5.1
LOAD	L	RX		P,A,S	58	3.5.1
LOAD AND TEST	LTR	RR	C		12	3.5.3
LOAD COMPLEMENT	LCR	RR	C	IF	13	3.5.4
LOAD HALFWORD	LH	RX		P,A,S	48	3.5.2
LOAD MULTIPLE	LM	RS		P,A,S	98	3.5.7
LOAD NEGATIVE	LNR	RR	C		11	3.5.6
LOAD POSITIVE	LPR	RR	C	IF	10	3.5.5
MULTIPLY	MR	RR		S	1C	3.5.16
MULTIPLY	M	RX		P,A,S	5C	3.5.16
MULTIPLY HALFWORD	MH	RX		P,A,S	4C	3.5.17
SHIFT LEFT DOUBLE	SLDA	RS	C	S IF	8F	3.5.26
SHIFT LEFT SINGLE	SLA	RS	C	IF	8B	3.5.24
SHIFT RIGHT DOUBLE	SRDA	RS	C	S	8E	3.5.27
SHIFT RIGHT SINGLE	SRA	RS	C		8A	3.5.25
SUBTRACT	SR	RR	C	IF	1B	3.5.11
SUBTRACT	S	RX	C	P,A,S IF	5B	3.5.11
SUBTRACT HALFWORD	SH	RX	C	P,A,S IF	4B	3.5.12
SUBTRACT LOGICAL	SLR	RR	C		1F	3.5.13
SUBTRACT LOGICAL	SL	RX	C	P,A,S	5F	3.5.13
STORE	ST	RX		P,A,S	50	3.5.21
STORE HALFWORD	STH	RX		P,A,S	40	3.5.22
STORE MULTIPLE	STM	RS		P,A,S	90	3.5.23

Legend

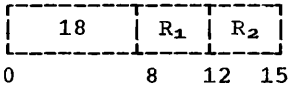
A Addressing exception
 C Condition code is set
 D Data exception
 IF Fixed-point-overflow exception
 IK Fixed-point-divide exception
 P Protection exception
 S Specification exception

PROGRAMMING NOTE

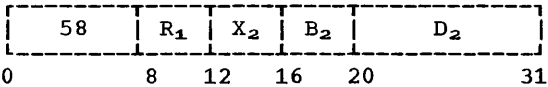
The logical comparisons, shifts, and connectives, as well as LOAD ADDRESS, BRANCH ON COUNT, BRANCH ON INDEX HIGH, and BRANCH ON INDEX LOW OR EQUAL, also may be used in fixed-point calculations.

3.5.1 LOAD

LR R_1, R_2 [RR]



L $R_1, D_2(X_2, B_2)$ [RX]



The second operand is placed in the first operand location. The second operand is not changed.

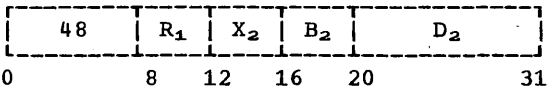
Condition Code: the code remains unchanged.

Program Interruptions:

- Protection (fetch violation by L only)
- Addressing (L only)
- Specification (L only)

3.5.2 LOAD HALFWORD

LH $R_1, D_2(X_2, B_2)$ [RX]



The halfword second operand is placed in the first operand location.

The halfword second operand is expanded to a fullword by propagating the sign-bit value through the 16 high-order bit positions. Expansion occurs after the operand is obtained from storage and prior to insertion in the register.

Condition Code: the code remains unchanged.

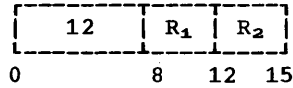
Program Interruptions:

- Protection (fetch violation)
- Addressing
- Specification

JUNE 1, 1971

3.5.3 LOAD AND TEST

LTR R_1, R_2 [RR]



The second operand is placed in the first operand location, and the sign and magnitude of the second operand determine the condition code. The second operand is not changed.

Condition Code:

0	Result is zero
1	Result is less than zero
2	Result is greater than zero
3	--

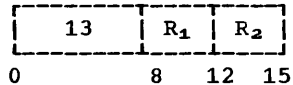
Program Interruptions: None.

PROGRAMMING NOTE

When the same register is specified as first and second operand location, the operation is equivalent to a test without data movement.

3.5.4 LOAD COMPLEMENT

LCR R_1, R_2 [RR]



The two's complement of the second operand is placed in the first operand location.

An overflow condition occurs when the maximum negative number is complemented; the number remains unchanged. The overflow causes a program interruption when the fixed-point overflow mask bit is one.

Condition Code:

0	Result is zero
1	Result is less than zero
2	Result is greater than zero
3	Overflow

Program Interruptions:

Fixed-Point Overflow

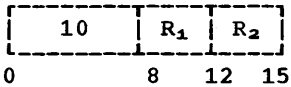
PROGRAMMING NOTE

Zero remains invariant under complementation.

JUNE 1, 1971

3.5.5 LOAD POSITIVE

LPR R_1, R_2 [RR]



The absolute value of the second operand is placed in the first operand location.

The operation includes complementation of negative numbers; positive numbers remain unchanged.

An overflow condition occurs when the maximum negative number is complemented; the number remains unchanged. The overflow causes a program interruption when the fixed-point overflow mask bit is one.

Condition Code:

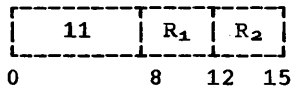
0	Result is zero
1	--
2	Result is greater than zero
3	Overflow

Program Interruptions:

Fixed-Point Overflow

3.5.6 LOAD NEGATIVE

LNR R_1, R_2 [RR]



The two's complement of the absolute value of the second operand is placed in the first operand location.

The operation complements positive numbers; negative numbers remain unchanged. The number zero remains unchanged with positive sign.

Condition Code:

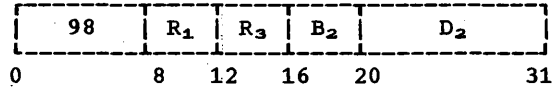
0	Result is zero
1	Result is less than zero
2	--
3	--

JUNE 1, 1971

Program Interruptions: None.

3.5.7 LOAD MULTIPLE

LM $R_1, R_3, D_2(B_2)$ [RS]



The set of general registers starting with the register specified by R_1 and ending with the register specified by R_3 is loaded from the locations designated by the second operand address.

The storage area from which the contents of the general registers are obtained starts at the location designated by the second operand address and continues through as many words as needed. The general registers are loaded in the ascending order of their addresses, starting with the register specified by R_1 and continuing up to and including the register specified by R_3 , with register 0 following register 15.

The second operand remains unchanged.

Condition Code: the code remains unchanged.

Program Interruptions:

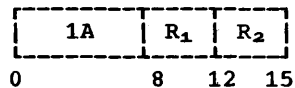
Protection (fetch violation)
Addressing
Specification

PROGRAMMING NOTE

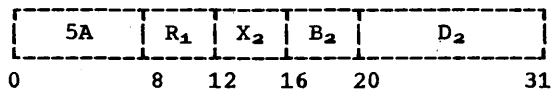
All combinations of register addresses specified by R_1 and R_3 are valid. When the register addresses are equal, only one word is transmitted. When the address specified by R_3 is less than the address specified by R_1 , the register addresses wrap around from 15 to 0.

3.5.8 ADD

AR R_1, R_2 [RR]



A $R_1, D_2(X_2, B_2)$ [RX]



The second operand is added to the first operand, and the sum is placed in the first operand location.

JUNE 1, 1971

Addition is performed by adding all 32 bits of both operands. If the carries out of the sign-bit position and the high-order numeric bit position agree, the sum is satisfactory; if they disagree, an overflow occurs. The sign bit is not changed after the overflow. A positive overflow yields a negative final sum, and a negative overflow results in a positive sum. The overflow causes a program interruption when the fixed-point overflow mask bit is one.

Condition Code:

- 0 Sum is zero
- 1 Sum is less than zero
- 2 Sum is greater than zero
- 3 Overflow

Program Interruptions:

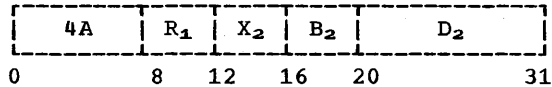
Protection (fetch violation by A only)
Addressing (A only)
Specification (A only)
Fixed-Point Overflow

PROGRAMMING NOTE

In two's-complement notation a zero result is always positive.

3.5.9 ADD HALFWORD

AH $R_1, D_2(X_2, B_2)$ [RX]



The halfword second operand is added to the first operand and the sum is placed in the first operand location.

The halfword second operand is expanded to a fullword prior to the addition by propagating the sign-bit value through the 16 high-order bit positions.

Addition is performed by adding all 32 bits of both operands. If the carries out of the sign-bit position and the high-order numeric bit position agree, the sum is satisfactory; if they disagree, an overflow occurs. The sign bit is not changed after the overflow. A positive overflow yields a negative final sum, and a negative overflow results in a positive sum. The overflow causes a program interruption when the fixed-point overflow mask bit is one.

Condition Code:

- 0 Sum is zero
- 1 Sum is less than zero
- 2 Sum is greater than zero
- 3 Overflow

Program Interruptions:

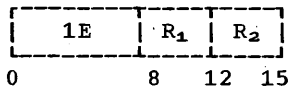
Protection (fetch violation)
Addressing

JUNE 1, 1971

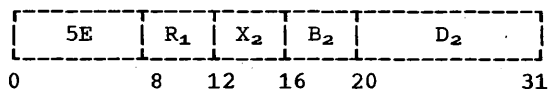
Specification
Fixed-Point Overflow

3.5.10 ADD LOGICAL

ALR R_1, R_2 [RR]



AL $R_1, D_2(X_2, B_2)$ [RX]



The second operand is added to the first operand, and the sum is placed in the first operand location. The occurrence of a carry out of the sign position is recorded in the condition code.

Logical addition is performed by adding all 32 bits of both operands without further change to the resulting sign bit. The instruction differs from ADD in the meaning of the condition code and in the absence of the interruption for overflow.

If a carry out of the sign position occurs, the left-most bit of the condition code (PSW bit 34) is made one. In the absence of a carry, bit 34 is made zero. When the sum is zero, the right-most bit of the condition code (PSW bit 35) is made zero. A nonzero sum is indicated by a one in bit 35.

Condition Code:

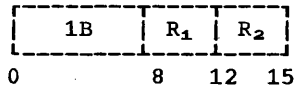
- 0 Sum is zero (no carry)
- 1 Sum is not zero (no carry)
- 2 Sum is zero (carry)
- 3 Sum is not zero (carry)

Program Interruptions:

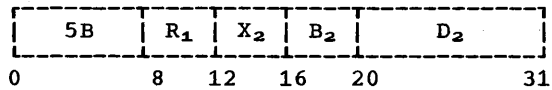
- Protection (fetch violation by AL only)
- Addressing (AL only)
- Specification (AL only)

3.5.11 SUBTRACT

SR R_1, R_2 [RR]



S $R_1, D_2(X_2, B_2)$ [RX]



The second operand is subtracted from the first operand, and the difference is placed in the first operand location.

Subtraction is performed by adding the one's complement of the second operand and a low order one to the first operand. All 32 bits of both operands participate, as in ADD. If the carries out of the sign-bit position and the high-order numeric bit position agree, the difference is satisfactory; if they disagree, an overflow occurs. The overflow causes a program interruption when the fixed-point overflow mask bit is one.

Condition Code:

- 0 Difference is zero
- 1 Difference is less than zero
- 2 Difference is greater than zero
- 3 Overflow

Program Interruptions:

- Protection (fetch violation by S only)
- Addressing (S only)
- Specification (S only)
- Fixed-Point Overflow

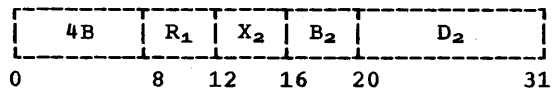
PROGRAMMING NOTE

When the same register is specified as first and second operand location, subtracting is equivalent to clearing the register.

Subtracting a maximum negative number from another maximum negative number gives a zero result and no overflow.

3.5.12 SUBTRACT HALFWORD

SH $R_1, D_2(X_2, B_2)$ [RX]



The halfword second operand is subtracted from the first operand, and the difference is placed in first operand location.

JUNE 1, 1971

The halfword second operand is expanded to a fullword prior to the subtraction by propagating the sign-bit value through the 16 high-order bit positions.

Subtraction is performed by adding the one's complement of the expanded second operand and a low order one to the first operand. All 32 bits of both operands participate, as in ADD. If the carries out of the sign-bit position and the high-order numeric bit position agree, the difference is satisfactory; if they disagree, an overflow occurs. The overflow causes a program interruption when the fixed-point overflow mask bit is one.

Condition Code:

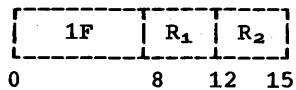
- 0 Difference is zero
- 1 Difference is less than zero
- 2 Difference is greater than zero
- 3 Overflow

Program Interruptions:

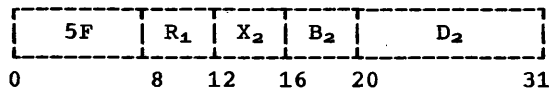
- Protection (fetch violation)
- Addressing
- Specification
- Fixed-Point Overflow

3.5.13 SUBTRACT LOGICAL

SIR R_1, R_2 [RR]



SL $R_1, D_2(X_2, B_2)$ [RX]



The second operand is subtracted from the first operand, and the difference is placed in the first operand location. The occurrence of a carry out of the sign position is recorded in the condition code.

Logical subtraction is performed by adding the one's complement of the second operand and a low order one to the first operand. All 32 bits of both operands participate, without further change to the resulting sign bit. The instruction differs from SUBTRACT in the meaning of the condition code and in the absence of the interruption for overflow.

If a carry out of the sign position occurs, the left-most bit of the condition code (PSW bit 34) is made one. In the absence of a carry, bit 34 is made zero. When the sum is zero, the right-most bit of the condition code (PSW bit 35) is made zero. A nonzero sum is indicated by a one in bit 35.

JUNE 1, 1971

Condition Code:

- 0 --
- 1 Difference is not zero (no carry)
- 2 Difference is zero (carry)
- 3 Difference is not zero (carry)

Program Interruptions:

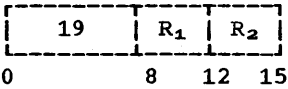
Protection (fetch violation by SL only)
Addressing (SL only)
Specification (SL only)

PROGRAMMING NOTE

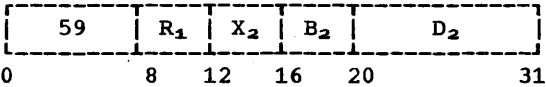
A zero difference cannot be obtained without a carry out of the sign position.

3.5.14 COMPARE

CR R_1, R_2 [RR]



C $R_1, D_2(X_2, B_2)$ [RX]



The first operand is compared with the second operand, and the result determines the setting of the condition code.

Comparison is algebraic, treating both comparands as 32-bit signed integers. Operands in registers or storage are not changed.

Condition Code:

- 0 Operands are equal
- 1 First operand is low
- 2 First operand is high
- 3 --

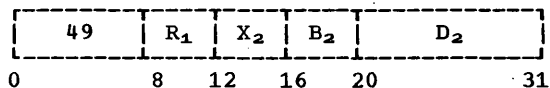
Program Interruptions:

Protection (fetch violation by C only)
Addressing (C only)
Specification (C only)

JUNE 1, 1971

3.5.15 COMPARE HALFWORD

CH $R_1, D_2(X_2, B_2)$ [RX]



The first operand is compared with the halfword second operand, and the result determines the setting of the condition code.

The halfword second operand is expanded to a fullword prior to the comparison by propagating the sign-bit value through the 16 high-order bit positions.

Comparison is algebraic, treating both comparands as 32-bit signed integers. Operands in registers or storage are not changed.

Condition Code:

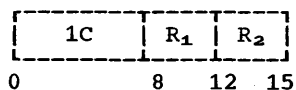
0	Operands are equal
1	First operand is low
2	First operand is high
3	--

Program Interruptions:

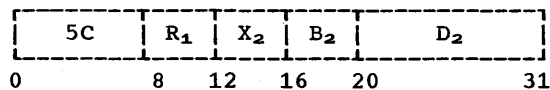
Protection (fetch violation)
Addressing
Specification

3.5.16 MULTIPLY

MR R_1, R_2 [RR]



M $R_1, D_2(X_2, B_2)$ [RX]



The product of the multiplier (the second operand) and the multiplicand (the first operand) replaces the multiplicand.

Both multiplier and multiplicand are 32-bit signed integers. The product is always a 64-bit signed integer and occupies an even/odd register pair. Because the multiplicand is replaced by the product, the R_1 field of the instruction must refer to an even-numbered register. A specification exception occurs when R_1 is odd. The multiplicand is taken from the odd register of the pair. The content of the even-numbered register replaced by the product is ignored, unless the register contains the multiplier. An overflow cannot occur.

JUNE 1, 1971.

The sign of the product is determined by the rules of algebra from the multiplier and multiplicand sign, except that a zero result is always positive.

Condition Code: The code remains unchanged.

Program Interruptions:

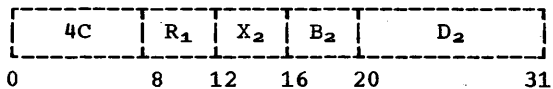
Protection (fetch violation by M only)
Addressing (M only)
Specification

PROGRAMMING NOTE

The significant part of the product usually occupies 62 bits or fewer. Only when two maximum negative numbers are multiplied are 63 significant product bits formed. Since two's-complement notation is used, the sign bit is extended right until the first significant product digit is encountered.

3.5.17 MULTIPLY HALFWORD

MH $R_1, D_2(X_2, B_2)$ [RX]



The product of the halfword multiplier (second operand) and multiplicand (first operand) replaces the multiplicand.

Both multiplicand and product are 32-bit signed integers and may be located in any general register. The halfword multiplier is expanded to a fullword prior to multiplication by propagating the sign-bit value through the 16 high-order bit positions. The multiplicand is replaced by the low-order part of the product. The bits to the left of the 32 low-order bits are not tested for significance; no overflow indication is given.

The sign of the product is determined by the rules of algebra from the multiplier and multiplicand sign, except that a zero result is always positive.

Condition Code: the code remains unchanged.

Program Interruptions:

Protection (fetch violation)
Addressing
Specification

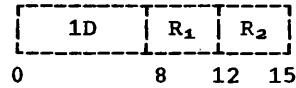
PROGRAMMING NOTE

The significant part of the product usually occupies 46 bits or fewer, the exception being 47 bits when both operands are maximum negative. Since the low-order 32 bits of the product are stored unchanged, ignoring all bits to the left, the sign bit of the result may differ from the true sign of the product in the case of overflow.

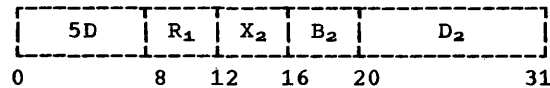
JUNE 1, 1971

3.5.18 DIVIDE

DR R_1, R_2 [RR]



D $R_1, D_2 (X_2, B_2)$ [RX]



The dividend (first operand) is divided by the divisor (second operand), and replaced by quotient and remainder.

The dividend is a 64-bit signed integer and occupies the even/odd pair of registers specified by the R_1 field of the instruction. A specification exception occurs when R_1 is odd. A 32-bit signed remainder and a 32-bit signed quotient replace the dividend in the even-numbered and odd-numbered registers, respectively. The divisor is a 32-bit signed integer.

The sign of the quotient is determined by the rules of algebra. The remainder has the same sign as the dividend, except that a zero quotient or a zero remainder is always positive. All operands and results are treated as signed integers. When the relative magnitude of dividend and divisor is such that the quotient cannot be expressed by a 31-bit signed integer, a fixed-point divide exception is recognized. A program interruption occurs, no division takes place, and the dividend remains unchanged in the general registers.

Condition Code: the code remains unchanged.

Program Interruptions:

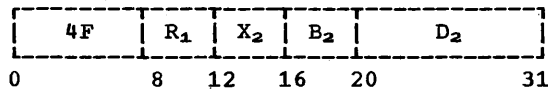
Protection (fetch violation by D only)
Addressing (D only)
Specification
Fixed-Point Divide

PROGRAMMING NOTE

Division applies to fullword operands in storage only.

3.5.19 CONVERT TO BINARY

CVB $R_1, D_2 (X_2, B_2)$ [RX]



The radix of the second operand is changed from decimal to binary, and the result is placed in the first operand location. The number is

JUNE 1, 1971

treated as a right-aligned signed integer both before and after conversion.

The second operand has the packed decimal data format, and is checked for valid sign and digit codes. Improper codes are a data exception and cause a program interruption. The decimal operand occupies a double-word storage field, which must be located on an integral boundary. The low-order four bits of the field represent the sign. The remaining 60 bits contain 15 binary-coded-decimal digits in true notation. The packed decimal data format is described under "Decimal Arithmetic".

The result of the conversion is placed in the general register specified by R_1 . The maximum number that can be converted and still be contained in a 32-bit register is 2,147,483,647; the minimum number is -2,147,483,648. For any decimal number outside this range, the operation is completed by placing the 32 low-order binary bits in the register; a fixed-point divide exception exists, and a program interruption follows. In the case of a negative second operand, the low-order part is in two's-complement notation.

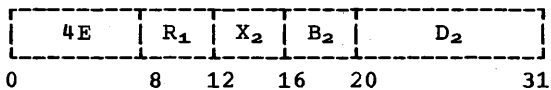
Condition Code: the code remains unchanged.

Program Interruptions:

Protection (fetch violation)
Addressing
Specification
Data
Fixed-Point Divide

3.5.20 CONVERT TO DECIMAL

CVD $R_1, D_2(X_2, B_2)$ [RX]



The radix of the first operand is changed from binary to decimal, and the result is stored in the second operand location. The number is treated as a right-aligned signed integer both prior to and after conversion.

The result is placed in the storage location designated by the second operand and has the packed decimal format, as described in "Decimal Arithmetic". The result occupies a double-word in storage and must be located on an integral boundary. The low-order four bits of the field represent the sign. A positive sign is encoded as 1100 or 1010; a negative sign is encoded as 1101 or 1011. The choice between the two sign representations is determined by the state of PSW bit 12. When PSW bit 12 (USASCII-8 Mode) is on, the second alternative listed is used. The remaining 60 bits contain 15 binary-coded-decimal digits in true notation.

The number to be converted is obtained as a 32-bit signed integer from a general register. Since 15 decimal digits are available for the decimal equivalent of 31 bits, an overflow cannot occur.

Condition Code: the code remains unchanged.

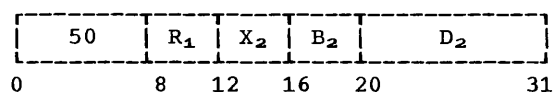
JUNE 1, 1971

Program Interruptions:

Protection (store violation)
Addressing
Specification

3.5.21 STORE

ST $R_1, D_2(X_2, B_2)$ [RX]



The first operand is stored at the second operand location.

The 32 bits in the general register are placed unchanged at the second operand location.

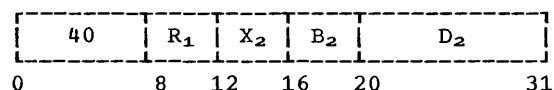
Condition Code: the code remains unchanged.

Program Interruptions:

Protection (store violation)
Addressing
Specification

3.5.22 STORE HALFWORD

STH $R_1, D_2(X_2, B_2)$ [RX]



The first operand is stored at the halfword second operand location.

The 16 low-order bits in the general register are placed unchanged at the second operand location. The 16 high-order bits of the first operand do not participate and are not tested.

Condition Code: the code remains unchanged.

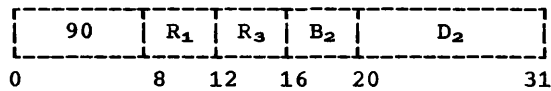
Program Interruptions:

Protection (store violation)
Addressing
Specification

JUNE 1, 1971

3.5.23 STORE MULTIPLE

STM $R_1, R_3, D_2(B_2)$ [RS]



The set of general registers starting with the register specified by R_1 and ending with the register specified by R_3 is stored at the locations designated by the second operand address.

The storage area where the contents of the general registers are placed starts at the location designated by the second operand address and continues through as many words as needed. The general registers are stored in the ascending order of their addresses, starting with the register specified by R_1 and continuing up to and including the register specified by R_3 , with register 0 following register 15. The first operands remain unchanged.

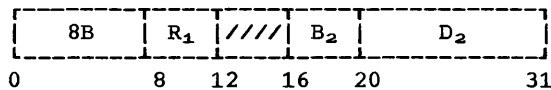
Condition Code: the code remains unchanged.

Program Interruptions:

Protection (store violation)
Addressing
Specification

3.5.24 SHIFT LEFT SINGLE

SLA $R_1, D_2(B_2)$ [RS]



The integer part of the first operand is shifted left the number of bits specified by the second operand address.

The second operand address is not used to address data; its low-order six bits indicate the number of bit positions to be shifted. The remainder of the address is ignored.

The sign of the first operand remains unchanged. All 31 integer bits of the operand participate in the left shift. Zeros are supplied to the vacated low-order register positions.

If a bit unlike the sign bit is shifted out of position 1, an overflow occurs. The overflow causes a program interruption when the fixed-point overflow mask bit is one.

Condition Code:

0 Result is zero
1 Result is less than zero
2 Result is greater than zero
3 Overflow

JUNE 1, 1971

Program Interruptions:

Fixed-Point Overflow

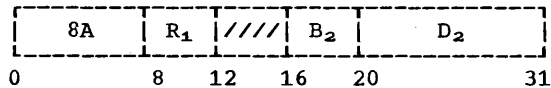
PROGRAMMING NOTE

For numbers with an absolute value of less than 2^{30} , a left shift of one bit position is equivalent to multiplying the number by two.

The base register participating in the generation of the second operand address permits indirect specification of the shift amount. A zero in the B_2 field indicates the absence of indirect shift specification.

3.5.25 SHIFT RIGHT SINGLE

SRA $R_1, D_2(B_2)$ [RS]



The integer part of the first operand is shifted right the number of bits specified by the second operand address.

The second operand address is not used to address data; its low-order six bits indicate the number of bit positions to be shifted. The remainder of the address is ignored.

The sign of the first operand remains unchanged. All 31 integer bits of the operand participate in the right shift. Bits equal to the sign are supplied to the vacated high-order bit positions. Low-order bits are shifted out without inspection and are lost.

Condition Code:

- 0 Result is zero
- 1 Result is less than zero
- 2 Result is greater than zero
- 3 --

Program Interruptions: None.

PROGRAMMING NOTE

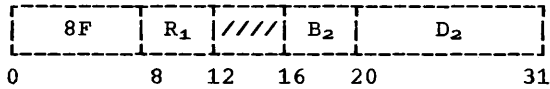
Right-shifting is similar to division by powers of two and to low-order truncation. Since negative numbers are kept in two's-complement notation, truncation is in the negative direction for both positive and negative numbers, rather than toward zero as in decimal arithmetic.

Shift amounts from 31-63 cause the entire integer to be shifted out of the register. When the entire integer field of a positive number has been shifted out, the register contains a value of zero. For a negative number, the register contains a value of -1.

The base register participating in the generation of the second operand address permits indirect specification of the shift amount. A zero in the B_2 field indicates the absence of indirect shift specification.

3.5.26 SHIFT LEFT DOUBLE

SLDA $R_1, D_2(B_2)$ [RS]



The double-length integer part of the first operand is shifted left the number of bits specified by the second operand address.

The R_1 field of the instruction specifies an even/odd pair of registers and must contain an even register address. A specification exception occurs when R_1 is odd.

The second operand address is not used to address data; its low-order six bits indicate the number of bit positions to be shifted. The remainder of the address is ignored.

The operand is treated as a number with 63 integer bits and a sign in the sign position of the even register. The sign remains unchanged. The high-order position of the odd register contains an integer bit, and the content of the odd register participates in the shift in the same manner as the other integer bits. Zeros are supplied to the vacated positions of the registers.

If a bit unlike the sign bit is shifted out of bit position 1 of the even register, an overflow occurs. The overflow causes a program interruption when the fixed-point overflow mask bit is one.

Condition Code:

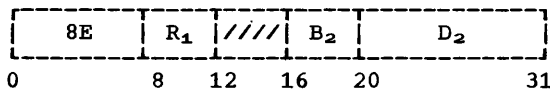
- 0 Result is zero
- 1 Result is less than zero
- 2 Result is greater than zero
- 3 Overflow

Program Interruptions:

Specification
Fixed-Point Overflow

3.5.27 SHIFT RIGHT DOUBLE

SRDA $R_1, D_2(B_2)$ [RS]



The double-length integer part of the first operand is shifted right the number of places specified by the second operand address.

The R_1 field of the instruction specifies an even/odd pair of registers and must contain an even register address. A specification exception occurs when R_1 is odd.

JUNE 1, 1971

The second operand address is not used to address data; its low-order six bits indicate the number of bit positions to be shifted. The remainder of the address is ignored.

The operand is treated as a number with 63 integer bits and a sign in the sign position of the even register. The sign remains unchanged. The high-order position of the odd register contains an integer bit, and the content of the odd register participates in the shift in the same manner as the other integer bits. The low-order bits are shifted out without inspection and are lost. Bits equal to the sign are supplied to the vacated positions of the registers.

Condition Code:

0	Result is zero
1	Result is less than zero
2	Result is greater than zero
3	--

Program Interruptions:

Specification

PROGRAMMING NOTE

A zero shift amount in the double-shift operations provides a double-length sign and magnitude test.

3.6 FIXED-POINT ARITHMETIC EXCEPTIONS

Exceptional instructions, data, or results cause a program interruption. When a program interruption occurs, the current PSW is stored as an old PSW, and a new PSW is obtained. The interruption code in the old PSW identifies the cause of the interruption. The following exceptions cause a program interruption in fixed-point arithmetic.

Protection: The storage key of an accessed location does not match the protection key in the PSW.

The operation is suppressed for a store violation. Therefore, the condition code and data in registers and storage remain unchanged. The only exception is STORE MULTIPLE, which is terminated; the amount of data stored is unpredictable and should not be used for further computation. The operation is terminated on any fetch protection violation.

Addressing: An address designates a location outside the available storage for a particular installation, or outside the configured storage, or storage assigned by the storage address translator for a particular computing element.

The operation is terminated. Therefore the result data are unpredictable and should not be used for further computation.

Operand addresses are tested only when used to address storage. Addresses used as a shift amount are not tested. The address restrictions do not apply to the components from which an address is generated -- the content of the D_2 field, and the contents of the registers specified by X_2 and B_2 .

Specification: A double-word operand is not located on a 64-bit boundary, a fullword operand is not located on a 32-bit boundary, a halfword operand is not located on a 16-bit boundary, or an instruction

JUNE 1, 1971

specifies an odd register address for a pair of general registers containing a 64-bit operand.

The operation is suppressed. Therefore, the condition code and data in registers and storage remain unchanged.

Data: A sign or a digit code of the decimal operand in CONVERT TO BINARY is incorrect. The operation is suppressed. Therefore, the condition code and data in registers and storage remain unchanged.

Fixed-Point Overflow: The result of a sign-control, add, subtract, or shift operation overflows. The interruption occurs only when the fixed-point overflow mask bit is one. The operation is completed by placing the truncated low-order result in the register and setting the condition code to 3. The overflow bits are lost. In add-type operations the sign stored in the register is the opposite of the sign of the sum or difference. In shift operations the sign of the shifted number remains unchanged. The state of the mask bit does not affect the result.

Fixed-Point Divide: The quotient of a division exceeds the register size, including division by zero, or the result in CONVERT TO BINARY exceeds 31 bits.

Division is suppressed. Therefore, data in the registers remain unchanged. The conversion is completed by recording the truncated low-order result in the register.

JUNE 1, 1971

CHAPTER 4: DECIMAL ARITHMETIC

Decimal arithmetic operates on data in the packed format. In this format two decimal digits are placed in one 8-bit byte. Since data are often communicated to or from external devices in the zoned format (which has one digit in an eight-bit byte) the necessary format-conversion operations are also provided in this instruction group.

Data are interpreted as integers, right-aligned in their fields. They are kept in true notation with a sign in the low-order eight-bit byte.

Processing takes place right to left between main-storage locations. All decimal arithmetic instructions use a two-address format. Each address specifies the leftmost byte of an operand. Associated with this address is a length field, indicating the number of additional bytes that the operand extends beyond the first byte.

The decimal arithmetic instruction set provides for adding, subtracting, comparing, multiplying, and dividing, as well as the format conversion of variable-length operands. All of the instructions discussed in this section except PACK, UNPACK, and MOVE WITH OFFSET are part of the decimal feature.

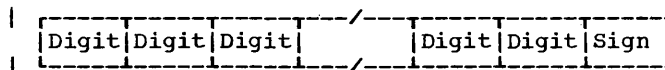
The condition code is set as a result of all add-type and comparison operations.

4.1 DATA FORMAT

Decimal operands reside in main storage only. They occupy fields that may start at any byte address and are composed of one to sixteen 8-bit bytes.

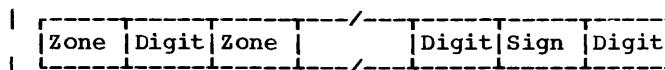
Lengths of the two operands specified in an instruction need not be the same. If necessary they are considered to be extended with zeros to the left of the high-order digits. Results never exceed the limits set by address and length specification. Lost carries or lost digits from arithmetic operations are signaled as a decimal overflow exception. Although decimal arithmetic is performed on data in the packed format, decimal operands may be either in the packed or zoned format.

Packed Decimal Number



In the packed format, two decimal digits normally are placed adjacent in a byte, except for the rightmost byte of the field. In the rightmost byte a sign is placed to the right of the decimal digit. Both digits and a sign are encoded and occupy four bits each.

Zoned Decimal Number



JUNE 1, 1971

In the zoned format the low-order four bits of a byte, the numeric, are normally occupied by a decimal digit. The four high-order bits of a byte are called the zone, except for the rightmost byte of the field, where normally the sign occupies the zone position.

Arithmetic is performed with operands in the packed format and results in the packed format. In the zoned format, the digits are represented as part of an alphanumeric character set. A PACK instruction is provided to transform zoned data into packed data, and an UNPACK instruction performs the reverse transformation. Moreover, the editing instructions may be used to change data from packed to zoned. (Chapter 6)

The fields specified in decimal arithmetic other than in PACK, UNPACK, and MOVE WITH OFFSET either should not overlap at all or should have coincident rightmost bytes. In ZERO AND ADD, the destination field may also overlap to the right of the source field. Because the code configurations for digits and sign are verified during arithmetic, improper overlapping fields are recognized as data exceptions. In move-type operations, the operand digits and signs are not checked, and the operand fields may overlap without any restrictions.

The rules for overlapped fields are established for the case where operands are fetched right to left from storage, eight bits at a time, just before they are processed. Similarly, the results are placed in storage, eight bits at a time, as soon as they are generated. Actual processing procedure may be considerably different because of the use of high-speed local storage for intermediate results. Nevertheless the same rules are observed.

4.2 NUMBER REPRESENTATION

Numbers are represented as right-aligned true integers with a plus or minus sign.

The digits 0-9 have the binary encoding 0000-1001. The codes 1010-1111 are invalid as digits. This set of codes is interpreted as sign codes, with 1010, 1100, 1110, and 1111 recognized as plus and with 1011 and 1101 recognized as minus. The codes 0000-1001 are invalid as sign codes. The zones are not tested for valid codes inasmuch as they are eliminated in changing data from the zoned to the packed format.

The sign and zone codes generated for all decimal arithmetic results differ for the Extended Binary-Coded-Decimal Interchange Code (EBCDIC) and the United States of America Standard Code for Information Interchange (USASCII-8). The choice between the two codes is determined by bit 12 of the PSW. When bit 12 is zero, the preferred EBCDIC codes are generated; these are plus, 1100; minus, 1101; and zone 1111. When bit 12 is one, the preferred USASCII-8 codes are generated; these are plus, 1010; minus, 1011; and zone, 0101.

4.3 CONDITION CODE

The results of all add-type and comparison operations are used to set the condition code. All other decimal arithmetic operations leave the code unchanged. The condition code can be used for decision-making by subsequent branch-on-condition instructions.

The condition code can be set to reflect two types of results for decimal arithmetic. For most operations the states 0, 1, and 2 indicate a zero, less than zero, and greater than zero content of the result field; the state 3 is used when the result of the operation overflows.

JUNE 1, 1971

For the comparison operation the states 0, 1, and 2 indicate that the first operand compared equal, low, or high.

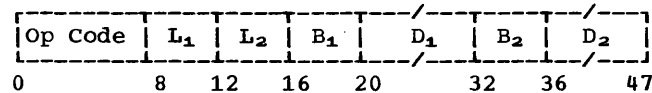
TABLE 4-I CONDITION CODE SETTING FOR DECIMAL ARITHMETIC

	0	1	2	3
ADD DECIMAL	zero	< zero	> zero	overflow
COMPARE DECIMAL	equal	low	high	--
SUBTRACT DECIMAL	zero	< zero	> zero	overflow
ZERO AND ADD	zero	< zero	> zero	overflow

4.4 INSTRUCTION FORMAT

Decimal instructions use the following format:

SS Format



For this format, the content of the general register specified by B₁ is added to the content of the D₁ field to form an address. This address specifies the leftmost byte of the first operand field. The number of operand bytes to the right of this byte is specified by the L₁ field of the instruction. Therefore the length in bytes of the first operand field is 1-16, corresponding to a length code in L₁ of 0000-1111. The second operand field is specified similarly by the L₂, B₂, and D₂ instruction fields.

A zero in the B₁ or B₂ field indicates the absence of the corresponding address component.

Results of operations are always placed in the first operand field. The result is never stored outside the field specified by the address and length. In the event the first operand is longer than the second, the second operand is extended with high-order zeros up to the length of the first operand. Such extension never modifies storage. The second operand field and the contents of all general registers remain unchanged.

4.5 INSTRUCTIONS

The decimal arithmetic instructions, their mnemonics, and operation codes follow. All instructions use the SS format, and assume packed operands and results. The only exceptions are PACK, which has a zoned operand, and UNPACK, which has a zoned result. The table indicates when the condition code is set and the exceptions that cause a program interruption.

TABLE 4-II DECIMAL INSTRUCTIONS

NAME	MNEMONIC	TYPE	CC	EXCEPTIONS	CODE	SECTION
ADD DECIMAL	AP	SS	C	P,A D,DF	FA	4.5.1
COMPARE DECIMAL	CP	SS	C	P,A D	F9	4.5.4
DIVIDE DECIMAL	DP	SS		P,A,S,D,DK	FD	4.5.6
MOVE WITH OFFSET	MVO	SS		P,A	F1	4.5.9
MULTIPLY DECIMAL	MP	SS		P,A,S,D	FC	4.5.5
SUBTRACT DECIMAL	SP	SS	C	P,A D,DF	FB	4.5.2
PACK	PACK	SS		P,A	F2	4.5.7
UNPACK	UNPK	SS		P,A	F3	4.5.8
ZERO AND ADD	ZAP	SS	C	P,A D,DF	F8	4.5.3

Legend

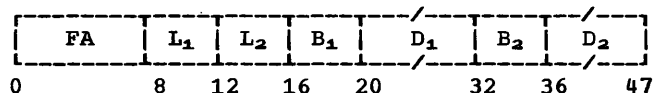
A Addressing exception
 C Condition code is set
 D Data exception
 DF Decimal-overflow exception
 DK Decimal-divide exception
 P Protection exception
 S Specification exception

PROGRAMMING NOTE

The moving, editing, and logical comparing instructions may also be used in decimal calculations.

4.5.1 ADD DECIMAL

AP $D_1(L_1, B_1), D_2(L_2, B_2)$ [SS]



The second operand is added to the first operand, and the sum is placed in the first operand location.

Addition is algebraic, taking into account sign and all digits of both operands. All signs and digits are checked for validity. If necessary, high-order zeros are supplied for either operand. When the first operand field is too short to contain all significant digits of the sum, a decimal overflow occurs, and a program interruption is taken, provided that the corresponding mask bit is one.

Overflow has two possible causes. The first is the loss of a carry out of the high-order digit position of the result field. The second cause is an oversized result, which occurs when the second operand

JUNE 1, 1971

field is larger than the first operand field and significant result digits are lost. The field sizes alone are not an indication of overflow.

The first and second operand fields may overlap when their low-order bytes coincide; therefore it is possible to add a number to itself.

The sign of the sum is determined by the rules of algebra. When the operation is completed without an overflow, a zero sum has a positive sign, but when high-order digits are lost because of an overflow, a zero sum may be either positive or negative, as determined by what the sign of the correct sum would have been.

Condition Code:

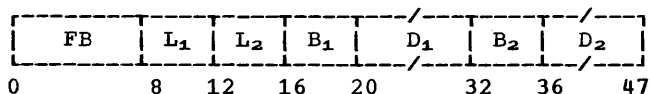
- 0 Sum is zero
- 1 Sum is less than zero
- 2 Sum is greater than zero
- 3 Overflow

Program Interruptions:

Protection (store or fetch violation)
Addressing
Data
Decimal Overflow

4.5.2 SUBTRACT DECIMAL

SP $D_1(L_1, B_1), D_2(L_2, B_2)$ [SS]



The second operand is subtracted from the first operand, and the difference is placed in the first operand location.

Subtraction is algebraic, taking into account sign and all digits of both operands. The SUBTRACT DECIMAL is similar to ADD DECIMAL, except that the sign of the second operand is changed from positive to negative or from negative to positive after the operand is obtained from storage and before the arithmetic.

The sign of the difference is determined by the rules of algebra. When the operation is completed without an overflow, a zero difference has a positive sign, but when high-order digits are lost because of an overflow, a zero difference may be either positive or negative, as determined by what the sign of the correct difference would have been.

Condition Code:

- 0 Difference is zero
- 1 Difference is less than zero
- 2 Difference is greater than zero
- 3 Overflow

Program Interruptions:

Protection (store or fetch violation)
Addressing

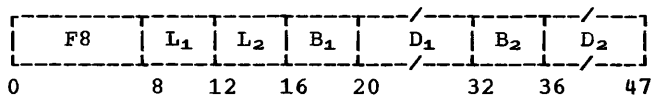
Data
 Decimal Overflow

PROGRAMMING NOTE

The operands of SUBTRACT DECIMAL may overlap when their low-order bytes coincide, even when their lengths are unequal. This property may be used to set to zero an entire field or the low-order part of a field.

4.5.3 ZERO AND ADD

ZAP $D_1(L_1, B_1), D_2(L_2, B_2)$ [SS]



The second operand is placed in the first operand location. The operation is equivalent to an addition to zero. A zero result is positive. When high-order digits are lost because of overflow, a zero result has the sign of the second operand.

Only the second operand is checked for valid sign and digit codes. Extra high-order zeros are supplied if needed. When the first operand field is too short to contain all significant digits of the second operand, a decimal overflow occurs and results in a program interruption, provided that the decimal overflow mask bit is one. The first and second operand fields may overlap when the rightmost byte of the first operand field is coincident with or to the right of the rightmost byte of the second operand.

Condition Code:

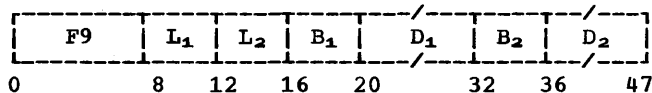
- 0 Result is zero
- 1 Result is less than zero
- 2 Result is greater than zero
- 3 Overflow

Program Interruptions:

Protection (store or fetch violation)
 Addressing
 Data
 Decimal Overflow

4.5.4 COMPARE DECIMAL

CP $D_1(L_1, B_1), D_2(L_2, B_2)$ [SS]



The first operand is compared with the second, and the condition code indicates the comparison result.

JUNE 1, 1971

Comparison is right to left, taking into account the sign and all digits of both operands. All signs and digits are checked for validity, and any valid plus or minus sign is considered equal to any other valid plus or minus sign, respectively. If the fields are unequal in length, the shorter is extended with high-order zeros. A field with a zero value and positive sign is considered equal to a field with a zero value but negative sign. Neither operand is changed as a result of the operation. Overflow cannot occur in this operation.

The first and second fields may overlap when their low-order bytes coincide. It is possible, therefore, to compare a number to itself.

Condition Code:

0	Operands equal
1	First operand is low
2	First operand is high
3	--

Program Interruptions:

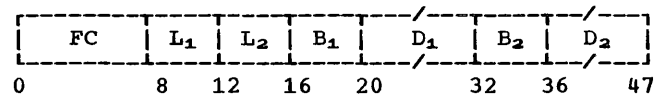
Protection (fetch violation)
Addressing
Data

PROGRAMMING NOTE

The COMPARE DECIMAL is unique in processing from right to left; taking signs, zeros, and invalid characters into account; and extending variable-length fields when they are unequal in length.

4.5.5 MULTIPLY DECIMAL

MP $D_1(L_1, B_1), D_2(L_2, B_2)$ [SS]



The product of the multiplier (the second operand) and the multiplicand (the first operand) replaces the multiplicand.

The multiplier size is limited to fifteen digits and sign and must be less than the multiplicand size. Length code L₂, larger than seven, or larger than or equal to the length code L₁, is recognized as a specification exception. The operation is suppressed and a program interruption occurs.

Since the number of digits in the product is the sum of the number of digits in the operands, the multiplicand must have high-order zero digits for at least a field size that equals the multiplier field size; otherwise a data exception is recognized and a program interruption occurs. This definition of the multiplicand field insures that no product overflow can occur. The maximum product size is 31 digits. At least one high-order digit of the product field is zero.

All operands and results are treated as signed integers, right-aligned in their field. The sign of the product is determined by the rules of algebra from the multiplier and multiplicand signs, even if one or both operands are zero.

JUNE 1, 1971

The multiplier and product fields may overlap when their low-order bytes coincide.

Condition Code: the code remains unchanged.

Program Interruptions:

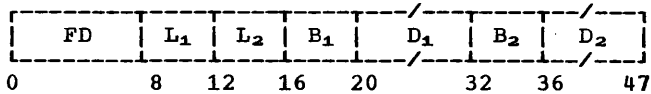
Protection (store or fetch violation)
Addressing
Specification
Data

PROGRAMMING NOTE

When the multiplicand does not have the desired number of leading zeros, multiplication may be preceded by a ZERO AND ADD into a larger field.

4.5.6 DIVIDE DECIMAL

DP $D_1(L_1, B_1), D_2(L_2, B_2)$ [SS]



The dividend (The first operand) is divided by the divisor (second operand), and replaced by the quotient and remainder.

The quotient field is placed leftmost in the first operand field. The remainder field is placed rightmost in the first operand field and has a size equal to the divisor size. Together, the quotient and remainder occupy the entire dividend field; therefore, the address of the quotient field is the address of the first operand. The size of the quotient field in eight-bit bytes is $L_1 - L_2$, and the length code for this field is one less ($L_1 - L_2 - 1$). When the divisor length code is larger than seven (15 digits and sign) or larger than or equal to the dividend length code, a specification exception is recognized. The operation is suppressed, and a program interruption occurs.

The dividend, divisor, quotient, and remainder are all signed integers, right-aligned in their fields. The sign of the quotient is determined by the rules of algebra from dividend and divisor signs. The sign of the remainder has the same value as the dividend sign. These rules are true even when quotient or remainder is zero.

Overflow cannot occur. A quotient larger than the number of digits allowed is recognized as a decimal-divide exception. The operation is suppressed, and a program interruption occurs. Divisor and dividend remain unchanged in their storage locations.

The divisor and dividend fields may overlap only if their low-order bytes coincide.

Condition Code: the code remains unchanged.

Program Interruptions:

Protection (store or fetch violation)
Addressing
Specification

JUNE 1, 1971

Data
Decimal Divide

PROGRAMMING NOTE

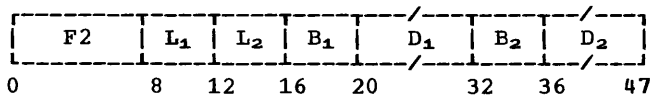
The maximum dividend size is 31 digits and sign. Since the smallest remainder size is one digit and sign, the maximum quotient size is 29 digits and sign.

The condition for a divide exception can be determined by a trial subtraction. The leftmost digit of the divisor field is aligned with the leftmost-less-one digit of the dividend field. When the divisor, so aligned, is less than or equal to the dividend, a divide exception is indicated.

A decimal-divide exception occurs if the dividend does not have at least one leading zero.

4.5.7 PACK

PACK $D_1(L_1, B_1), D_2(L_2, B_2)$ [SS]



The format of the second operand is changed from zoned to packed, and the result is placed in the first operand location.

The second operand is assumed to have the zoned format. All zones are ignored, except the zone over the low-order digit, which is assumed to represent a sign. The sign is placed in the right four bits of the low-order byte, and the digits are placed adjacent to the sign and to each other in the remainder of the result field. The sign and digits are moved unchanged to the first operand field, and are not checked for valid codes.

The fields are processed right to left. If necessary, the second operand is extended with high-order zeros. If the first operand field is too short to contain all significant digits of the second operand field, the remaining high-order digits are ignored. Overlapping fields may occur and are processed by storing each result byte immediately after the necessary operand bytes are fetched. Except for the rightmost byte of the result field, which is stored immediately upon fetching the first operand byte, two operand bytes are needed for each result byte.

Condition Code: the code remains unchanged.

Program Interruptions:

Protection (store or fetch violation)
Addressing

PROGRAMMING NOTE

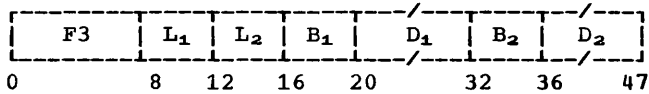
The PACK instruction may be used to switch the two digits in one byte by specifying a length of one and the same address for both operands.

To remove the zones of all bytes of a field, including the low-order byte, both operands must be extended with a dummy byte in the low-order position, which subsequently is ignored in the result field.

JUNE 1, 1971

4.5.8 UNPACK

UNPK $D_1(L_1, B_1), D_2(L_2, B_2)$ [SS]



The format of the second operand is changed from packed to zoned, and the result is placed in the first operand location.

The digits and sign of the packed operand are placed unchanged in the first operand location, using the zoned format. Zones with coding 1111 in EBCDIC and coding 0101 in USASCII-8 are supplied for all bytes, except the low-order byte, which receives the sign of the packed operand. The operand sign and digits are not checked for valid codes.

The fields are processed right to left. The second operand is extended with zero digits before unpacking, if necessary. If the first operand field is too short to contain all significant digits of the second operand, the remaining high-order digits are ignored. The first and second operand fields may overlap and are processed by storing the first result byte immediately after the rightmost operand byte is fetched; for the remaining operand bytes, two result bytes are stored immediately after one byte is fetched.

Condition Code: the code remains unchanged.

Program Interruptions:

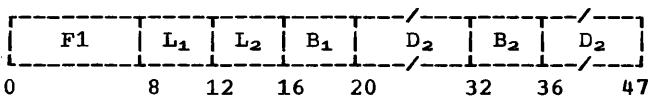
Addressing
Protection (store or fetch violation)

PROGRAMMING NOTE

A field that is to be unpacked can be destroyed by improper overlapping. If it is desired to save storage space for unpacking by overlapping the operand fields, the low-order position of the first operand must be to the right of the low-order position of the second operand by the number of bytes in the second operand minus two. If only one or two bytes are to be unpacked, the low-order positions of the two operands may coincide.

4.5.9 MOVE WITH OFFSET

MVO $D_1(L_1, B_1), D_2(L_2, B_2)$ [SS]



The second operand is placed to the left of and adjacent to the low-order four bits of the first operand.

The low-order four bits of the first operand are attached as low-order bits to the second operand, the second operand bits are offset by four bit positions, and the result is placed in the first operand

JUNE 1, 1971

location. The first and second operand bytes are not checked for valid codes.

The fields are processed right to left. If necessary, the second operand is extended with high-order zeros. If the first operand field is too short to contain all bytes of the second operand, the remaining information is ignored. Overlapping fields may occur and are processed by storing a result byte as soon as the necessary operand bytes are fetched.

Condition Code: the code remains unchanged.

Program Interruptions:

Protection (store or fetch violation)
Addressing

PROGRAMMING NOTE

The instruction set for decimal arithmetic includes no shift instructions since the equivalent of a shift can be obtained by programming. Programs for right or left shift and for an even or odd shift amount may be written with MOVE WITH OFFSET, and the logical move instructions.

4.6 DECIMAL ARITHMETIC EXCEPTIONS

Exceptional instructions, data, or results cause a program interruption. When the interruption occurs the current PSW is stored as an old PSW and a new PSW is obtained. The interruption code in the old PSW identifies the cause of the interruption. The following exceptions cause a program interruption in decimal arithmetic.

Protection: The storage key of an accessed location does not match the protection key in the PSW.

The operation is terminated for either a store or a fetch violation by a decimal instruction; the result data and condition code are unpredictable.

Addressing: An address designates a location outside the available storage for the installed system, or outside the configured storage, or storage assigned by the storage address translator for a particular computing element.

For this exception the operation is terminated. The result data and the condition code are unpredictable, and should not be used for further computation.

These address exceptions do not apply to the components from which an address is generated -- the contents of the D_1 and D_2 fields and the contents of the registers specified by B_1 and B_2 .

Specification: A multiplier or a divisor size exceeds 15 digits and sign, or exceeds the multiplicand or dividend size. The instruction is suppressed; therefore, the condition code and data in storage and registers remain unchanged.

Data: A sign or digit code of an operand in ADD DECIMAL, SUBTRACT DECIMAL, ZERO AND ADD, COMPARE DECIMAL, MULTIPLY DECIMAL, or DIVIDE DECIMAL is incorrect, a multiplicand has insufficient high-order zeros, or the operand fields in these operations overlap incorrectly. The operation is terminated. The result data and the condition code are unpredictable and should not be used for further computation.

JUNE 1, 1971

Decimal Overflow: The result of ADD DECIMAL, SUBTRACT DECIMAL, or ZERO AND ADD overflows. The program interruption occurs only when the decimal-overflow mask bit is one. The operation is completed by placing the truncated low-order result in the result field and setting the condition code to 3. The sign and low-order digits contained in the result field are the same as they would have been for an infinitely long result field.

Decimal Divide Check: The quotient exceeds the specified data field, including division by zero. Division is suppressed. Therefore, the dividend and divisor remain unchanged in storage.

JUNE 1, 1971

CHAPTER 5: FLOATING-POINT ARITHMETIC

The floating-point instruction set is used to perform calculations on operands with a wide range of magnitude and yielding results scaled to preserve precision.

A floating-point number consists of a signed exponent and a signed fraction. The quantity expressed by this number is the product of the fraction and the number sixteen raised to the power of the exponent. The exponent is expressed in excess 64 binary notation; the fraction is expressed as a hexadecimal number having a radix point to the left of the high-order digit.

To avoid unnecessary storing and loading operations for results and operands, four floating-point registers are provided. The floating-point instruction set provides for the loading, adding, subtracting, comparing, multiplying, dividing, halving, and storing, as well as the sign control of short or long operands. Short operands generally provide faster processing and require less storage than long operands. On the other hand, long operands provide greater preciseness in computation. Operations may be either register to register or storage to register.

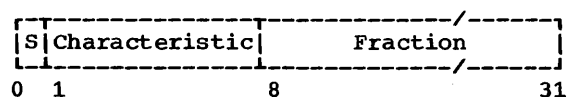
Maximum precision is preserved in addition, subtraction, multiplication, halving, and division by producing normalized results. For addition and subtraction, instructions are also provided that generate unnormalized results. Normalized and unnormalized operands may be used in any floating-point operation.

The condition code is set as a result of all sign control, add, subtract, and compare operations.

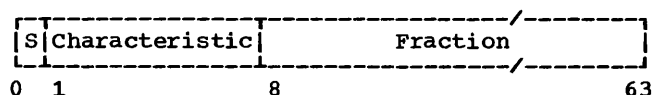
5.1 DATA FORMAT

Floating-point data occupy a fixed-length format, which may be either a fullword short format or a double-word long format. Both formats may be used in main storage and in the floating-point registers. The floating-point registers are numbered 0, 2, 4, and 6.

Short Floating-Point Number



Long Floating-Point Number



The first bit in either format is the sign bit (S). The subsequent seven bit positions are occupied by the characteristic. The fraction field may have either six or fourteen hexadecimal digits.

JUNE 1, 1971

The entire set of floating-point instructions is available for both short and long operands. When short-precision is specified, all operands and results are thirty-two-bit floating-point words, and the rightmost thirty-two bits of the floating-point registers do not participate in the operations and remain unchanged. An exception is the product in MULTIPLY, which is sixty-four bits in length and occupies a full register. When long-precision is specified, all operands and results have the sixty-four-bit floating-point format.

Although final results have 6 fraction digits in short precision and 14 fraction digits in long precision, results in ADD NORMALIZED, SUBTRACT NORMALIZED, ADD UNNORMALIZED, SUBTRACT UNNORMALIZED, COMPARE, HALVE, and MULTIPLY may have one additional low-order digit. This low-order digit, the guard digit, increases the precision of the final result.

5.2 NUMBER REPRESENTATION

The fraction of a floating-point number is expressed in hexadecimal digits. The radix point of the fraction is assumed to be immediately to the left of the high-order fraction digit. To provide the proper magnitude for the floating-point number, the fraction is considered to be multiplied by a power of sixteen. The characteristic portion, bits 1-7 of both floating-point formats, indicates this power. The bits within the characteristic field can represent numbers from 0 through 127. To accommodate large and small magnitudes, the characteristic is formed by adding 64 to the actual exponent. The range of the exponent is thus -64 through +63. This technique produces a characteristic in excess-64 notation.

Both positive and negative quantities have a true fraction, the difference in sign being indicated by the sign bit. The number is positive or negative accordingly as the sign bit is zero or one.

The range covered by the magnitude (M) of a normalized floating-point number is

in short precision $16^{-65} \leq M \leq (1-16^{-6}) \cdot 16^{63}$, and

in long precision $16^{-65} \leq M \leq (1-16^{-14}) \cdot 16^{63}$, or approximately $5.4 \cdot 10^{-79} \leq M \leq 7.2 \cdot 10^{75}$ in either precision.

A number with zero characteristic, zero fraction, and plus sign is called a true zero. A true zero may arise as the result of an arithmetic operation because of the particular magnitude of the operands. A result is forced to be true zero when (1) an exponent underflow occurs and the exponent-underflow mask (PSW bit 38) is zero, (2) a result fraction of an addition or subtraction operation is zero and the significance mask (PSW bit 39) is zero, or (3) the operand of HALVE, one or both operands of MULTIPLY, or the dividend in DIVIDE has a zero fraction. When a program interruption due to exponent underflow occurs, a true zero fraction is not forced; instead, the fraction and sign remain correct. Whenever a result has a zero fraction, the exponent overflow and underflow exceptions do not cause a program interruption. When a divisor has a zero fraction, division is suppressed, a floating-point divide exception exists, and a program interruption occurs. In addition and subtraction, an operand with a zero fraction or characteristic participates as a normal number.

The sign of a sum, difference, product, or quotient with zero fraction is positive. The sign of a zero fraction resulting from other operations is established by the rules of algebra from the operand signs.

JUNE 1, 1971

5.3 NORMALIZATION

A quantity can be represented with the greatest precision by a floating-point number of given fraction length when that number is normalized. A normalized floating-point number has a nonzero high-order hexadecimal fraction digit. If one or more high-order fraction digits are zero, the number is said to be unnormalized. The process of normalization consists of shifting the fraction left until the high-order hexadecimal digit is nonzero, and reducing the characteristic by the number of hexadecimal digits shifted. A zero fraction can not be normalized, and its associated characteristic therefore remains unchanged when normalization is called for.

Normalization usually takes place when the intermediate arithmetic result is changed to the final result. This function is called postnormalization. In performing multiplication and division, the operands are normalized before the arithmetic process. This function is called prenormalization.

Floating-point operations may be performed with or without normalization. Most operations are performed in only one of these two ways. Addition and subtraction may be specified either way.

When an operation is performed without normalization, high-order zeros in the result fraction are not eliminated. The result may or may not be normalized, depending upon the original operands.

In both normalized and unnormalized operations, the initial operands need not be in normalized form. Also, intermediate fraction results are shifted right when an overflow occurs, and the intermediate fraction result is truncated to the final result length after the shifting, if any.

PROGRAMMING NOTE

Since normalization applies to hexadecimal digits, the three high-order bits of a normalized fraction may be zero.

5.4 CONDITION CODE

The results of floating-point sign-control, add, subtract, and compare operations are used to set the condition code. Multiplication, division, loading, halving, and storing leave the code unchanged. The condition code can be used for decision-making by subsequent branch-on-condition instructions.

The condition code can be set to reflect two types of results for floating-point arithmetic. For most operations, the codes 0, 1, or 2 indicate that the the result is zero, less than zero, or greater than zero. A zero result is indicated whenever the result fraction is zero, including a forced zero. Code 3 is never set by a floating-point operation.

For comparison, the codes 0, 1, or 2 indicate that the first operand is equal, low, or high.

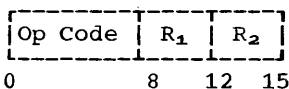
TABLE 5-I CONDITION CODE SETTING FOR FLOATING-POINT ARITHMETIC

	0	1	2	3
ADD NORMALIZED (short/long)	zero	< zero	> zero	--
ADD UNNORMALIZED (short/long)	zero	< zero	> zero	--
COMPARE (short/long)	equal	low	high	--
LOAD AND TEST (short/long)	zero	< zero	> zero	--
LOAD COMPLEMENT (short/long)	zero	< zero	> zero	--
LOAD NEGATIVE (short/long)	zero	< zero	--	--
LOAD POSITIVE (short/long)	zero	--	> zero	--
SUBTRACT NORMALIZED (short/long)	zero	< zero	> zero	--
SUBTRACT UNNORMALIZED (short/long)	zero	< zero	> zero	--

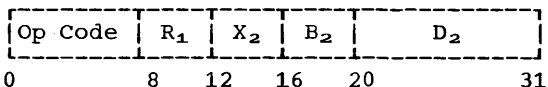
5.5 INSTRUCTION FORMAT

Floating-point instructions use the following two formats:

RR Format



RX Format



In these formats, R₁ designates the address of a floating-point register. The contents of this register will be called the first operand. The second operand location is defined differently for each format.

In the RR format, the R₂ field specifies the address of a floating-point register containing the second operand. The same register may be specified for the first and second operand.

In the RX format, the contents of the general register specified by X₂ and B₂ are added to the content of the D₂ field to form an address designating the location of the second operand.

A zero in an X₂ or B₂ field indicates the absence of the corresponding address component.

The register address specified by the R₁ and R₂ fields should be 0, 2, 4, or 6. Otherwise, a specification exception is recognized, and a program interruption is caused.

The storage address of the second operand should designate word boundaries for short operands and double-word boundaries for long operands. Otherwise, a specification exception is recognized, and a program interruption is caused.

JUNE 1, 1971

Results replace the first operand, except for the storing operations, where the second operand is replaced.

Except for the storing of the final result, the contents of all floating-point or general registers and storage locations participating in the addressing or execution part of an operation remain unchanged.

5.6 INSTRUCTIONS

The floating-point arithmetic instructions and their mnemonics, formats, and operation codes follow. All operations can be specified in short and long precision. The following table indicates when normalization occurs, when the condition code is set, and the exceptions in operand designations, data, or results that cause a program interruption.

JUNE 1, 1971

TABLE 5-II FLOATING-POINT INSTRUCTIONS

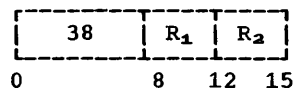
NAME		MNEMONIC	TYPE	CC	EXCEPTIONS	CODE	SECTION
ADD NORMALIZED (1)	N	ADR	RR	C	S, U, E, LS	2A	5.6.6
ADD NORMALIZED (1)	N	AD	RX	C	P, A, S, U, E, LS	6A	5.6.6
ADD NORMALIZED (s)	N	AER	RR	C	S, U, E, LS	3A	5.6.6
ADD NORMALIZED (s)	N	AE	RX	C	P, A, S, U, E, LS	7A	5.6.6
ADD UNNORMALIZED (1)		AWR	RR	C	S E, LS	2E	5.6.7
ADD UNNORMALIZED (1)		AW	RX	C	P, A, S E, LS	6E	5.6.7
ADD UNNORMALIZED (s)		AUR	RR	C	S E, LS	3E	5.6.7
ADD UNNORMALIZED (s)		AU	RX	C	P, A, S E, LS	7E	5.6.7
COMPARE (1)		CDR	RR	C	S	29	5.6.10
COMPARE (1)		CD	RX	C	P, A, S	69	5.6.10
COMPARE (s)		CER	RR	C	S	39	5.6.10
COMPARE (s)		CE	RX	C	P, A, S	79	5.6.10
DIVIDE (1)	N	DDR	RR		S, U, E, FK	2D	5.6.13
DIVIDE (1)	N	DD	RX		P, A, S, U, E, FK	6D	5.6.13
DIVIDE (s)	N	DER	RR		S, U, E, FK	3D	5.6.13
DIVIDE (s)	N	DE	RX		P, A, S, U, E, FK	7D	5.6.13
HALVE (1)	N	HDR	RR		S, U	24	5.6.11
HALVE (s)	N	HER	RR		S, U	34	5.6.11
LOAD (1)		LDR	RR		S	28	5.6.1
LOAD (1)		LD	RX		P, A, S	68	5.6.1
LOAD (s)		LER	RR		S	38	5.6.1
LOAD (s)		LE	RX		P, A, S	78	5.6.1
LOAD AND TEST (1)		LTDR	RR	C	S	22	5.6.2
LOAD AND TEST (s)		LTER	RR	C	S	32	5.6.2
LOAD COMPLEMENT (1)		LCDR	RR	C	S	23	5.6.3
LOAD COMPLEMENT (s)		LCER	RR	C	S	33	5.6.3
LOAD NEGATIVE (1)		LNDR	RR	C	S	21	5.6.5
LOAD NEGATIVE (s)		LNER	RR	C	S	31	5.6.5
LOAD POSITIVE (1)		LPDR	RR	C	S	20	5.6.4
LOAD POSITIVE (s)		LPER	RR	C	S	30	5.6.4
MULTIPLY (1)	N	MDR	RR		S, U, E	2C	5.6.12
MULTIPLY (1)	N	MD	RX		P, A, S, U, E	6C	5.6.12
MULTIPLY (s)	N	MER	RR		S, U, E	3C	5.6.12
MULTIPLY (s)	N	ME	RX		P, A, S, U, E	7C	5.6.12
STORE (1)		STD	RX		P, A, S	60	5.6.14
STORE (s)		STE	RX		P, A, S	70	5.6.14
SUBTRACT NORMALIZED (1)	N	SDR	RR	C	S, U, E, LS	2B	5.6.8
SUBTRACT NORMALIZED (1)	N	SD	RX	C	P, A, S, U, E, LS	6B	5.6.8
SUBTRACT NORMALIZED (s)	N	SER	RR	C	S, U, E, LS	3B	5.6.8
SUBTRACT NORMALIZED (s)	N	SE	RX	C	P, A, S, U, E, LS	7B	5.6.8
SUBTRACT UNNORMALIZED (1)		SWR	RR	C	S E, LS	2F	5.6.9
SUBTRACT UNNORMALIZED (1)		SW	RX	C	P, A, S E, LS	6F	5.6.9
SUBTRACT UNNORMALIZED (s)		SUR	RR	C	S E, LS	3F	5.6.9
SUBTRACT UNNORMALIZED (s)		SU	RX	C	P, A, S E, LS	7F	5.6.9

JUNE 1, 1971

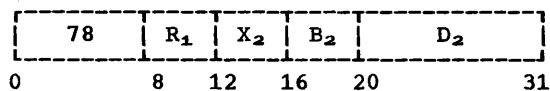
Floating-Point Instructions (continued)	
<u>Legend</u>	
A	Addressing exception
C	Condition code is set
FK	Floating-point divide exception
E	Exponent overflow exception
LS	Significance exception
N	Normalized operation
P	Protection exception
S	Specification exception
U	Exponent underflow exception
s	short
l	long

5.6.1 LOAD

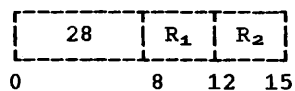
LER R_1, R_2 [RR, Short Operands]



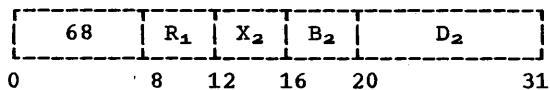
LE $R_1, D_2(X_2, B_2)$ [RX, Short Operands]



LDR R_1, R_2 [RR, Long Operand]



LD $R_1, D_2(X_2, B_2)$ [RX, Long Operands]



The second operand is placed in the first operand location. The second operand is not changed. In short-precision the low-order half of the result register remains unchanged. Exponent overflow, exponent underflow, or lost significance cannot occur.

Condition Code: the code remains unchanged.

Program Interruptions:

Protection (fetch violation by LE and LD only)
Addressing (LE, LD only)

JUNE 1, 1971

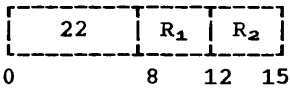
Specification

5.6.2 LOAD AND TEST

LTER R_1, R_2 [RR, Short Operands]



LTDR R_1, R_2 [RR, Long Operands]



The second operand is placed in the first operand location, and its sign and magnitude determine the condition code.

The second operand is not changed. In short-precision the low-order half of the result register remains unchanged and is not tested.

Condition Code:

- 0 Result fraction is zero
- 1 Result is less than zero
- 2 Result is greater than zero
- 3 --

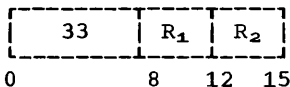
Program Interruptions: Specification

PROGRAMMING NOTE

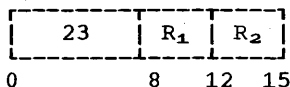
When the same register is specified as first and second operand location, the operation is equivalent to a test without data movement.

5.6.3 LOAD COMPLEMENT

LCER R_1, R_2 [RR, Short Operands]



LCDR R_1, R_2 [RR, Long Operands]



The second operand is placed in the first operand location with the sign changed to the opposite value.

JUNE 1, 1971

The sign bit of the second operand is inverted, while characteristic and fraction are not changed. In short-precision, the low-order half of the result register remains unchanged and is not tested.

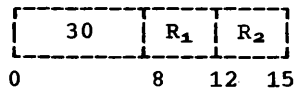
Condition Code:

0	Result fraction is zero
1	Result is less than zero
2	Result is greater than zero
3	--

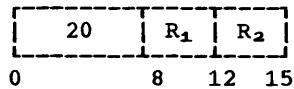
Program Interruptions:
Specification

5.6.4 LOAD POSITIVE

LPER R_1, R_2 [RR, Short Operands]



LPDR R_1, R_2 [RR, Long Operands]



The second operand is placed in the first operand location with the sign made plus.

The sign bit of the second operand is made zero, while characteristic and fraction are not changed. In short-precision, the low-order half of the result register remains unchanged and is not tested.

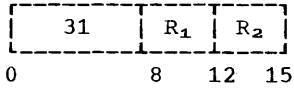
Resulting Condition Code:

0	Result fraction is zero
1	--
2	Result is greater than zero
3	--

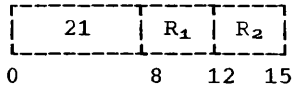
Program Interruptions:
Specification

5.6.5 LOAD NEGATIVE

LNER R_1, R_2 [RR, Short Operands]



LNDR R_1, R_2 [RR, Long Operands]



The second operand is placed in the first operand location with the sign made minus.

The sign bit of the second operand is made one, even if the fraction is zero. Characteristic and fraction are not changed. In short-precision, the low-order half of the result register remains unchanged and is not tested.

Condition Code:

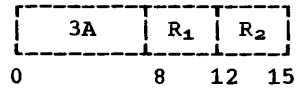
- 0 Result fraction is zero
- 1 Result is less than zero
- 2 --
- 3 --

Program Interruptions:
Specification

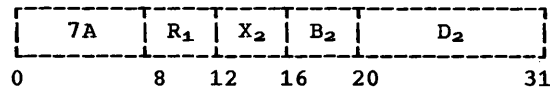
JUNE 1, 1971

5.6.6 ADD NORMALIZED

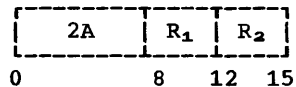
AER R_1, R_2 [RR, Short Operands]



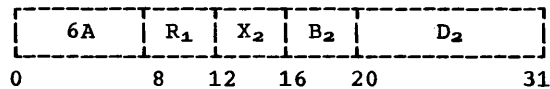
AE $R_1, D_2(X_2, B_2)$ [RX, Short Operands]



ADR R_1, R_2 [RR, Long Operands]



AD $R_1, D_2(X_2, B_2)$ [RX, Long Operands]



The second operand is added to the first operand, and the normalized sum is placed in the first operand location.

In short-precision, the low-order halves of the floating-point registers are ignored and remain unchanged.

Addition of two floating-point numbers consists of a characteristic comparison and a fraction addition. The characteristics of the two operands are compared, and the fraction with the smaller characteristic is right-shifted; its characteristic is increased by one for each hexadecimal digit of shift, until the two characteristics agree. The fractions are then added algebraically to form an intermediate sum. If an overflow carry occurs, the intermediate sum is right-shifted one digit, and the characteristic is increased by one. If this increase causes a characteristic overflow, an exponent-overflow exception is signaled, and a program interruption occurs. The fraction is normalized and correct, the sign is correct, and the characteristic is 128 smaller than the correct characteristic.

The short intermediate sum consists of 7 hexadecimal digits and a possible carry. The long intermediate sum consists of 15 hexadecimal digits and a possible carry. The low-order digit is a guard digit obtained from the fraction which is shifted right. Only one guard digit position participates in the fraction addition. The guard digit is zero if no shift occurs.

After the addition, the intermediate sum is left-shifted as necessary to form a normalized fraction, vacated low-order digit positions filled with zeros and the characteristic is reduced by the amount of shift.

If normalization causes the characteristic to underflow, and if the corresponding mask bit is one, a program interruption occurs. In this case the fraction is correct and normalized, the sign is correct, and

JUNE 1, 1971

the characteristic is 128 larger than the correct one. If the corresponding mask bit is zero, the result is made a true zero. If no left shift takes place the intermediate sum is truncated to the proper fraction length.

When the intermediate sum is zero and the significance mask bit is one, a significance exception exists, and a program interruption takes place. No normalization occurs; the intermediate sum characteristic remains unchanged. When the intermediate sum is zero and the significance mask bit is zero, the program interruption for the significance exception does not occur; rather, the characteristic is made zero, yielding a true zero result. Exponent underflow does not occur for a zero fraction.

The sign of the sum is derived by the rules of algebra. The sign of a sum with zero result fraction is always positive.

Condition Code:

0	Result fraction is zero
1	Result is less than zero
2	Result is greater than zero
3	--

Program Interruptions:

Protection (fetch violation by AE and AD only)
Addressing (AE and AD only)
Specification
Significance
Exponent Overflow
Exponent Underflow

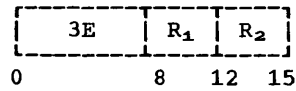
PROGRAMMING NOTE

Interchanging the two operands in a floating-point addition does not affect the value of the sum.

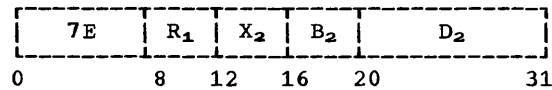
JUNE 1, 1971

5.6.7 ADD UNNORMALIZED

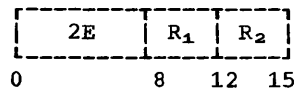
AUR R_1, R_2 [RR, Short Operands]



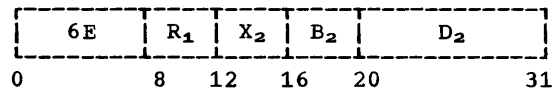
AU $R_1, D_2(X_2, B_2)$ [RX, Short Operands]



AWR R_1, R_2 [RR, Long Operands]



AW $R_1, D_2(X_2, B_2)$ [RX, Long Operands]



The second operand is added to the first operand, and the unnormalized sum is placed in the first operand location.

In short-precision, the low-order halves of the floating-point registers are ignored and remain unchanged.

After the addition the intermediate sum is truncated to the proper fraction length.

When the resulting fraction is zero and the significance mask bit is one, a significance exception exists and a program interruption takes place. When the resulting fraction is zero and the significance mask bit is zero, the program interruption for the significance exception does not occur; rather, the characteristic is made zero, yielding a true zero result.

Leading zeros in the result are not eliminated by normalization, and an exponent underflow cannot occur.

The sign of the sum is derived by the rules of algebra. The sign of a sum with zero result fraction is always positive.

Condition Code:

0	Result fraction is zero
1	Result is less than zero
2	Result is greater than zero
3	--

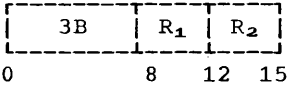
Program Interruptions:

Protection (fetch violation by AU and AW only)
Addressing (AU and AW only)
Specification

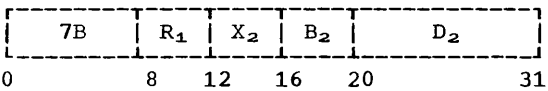
Significance
Exponent Overflow

5.6.8 SUBTRACT NORMALIZED

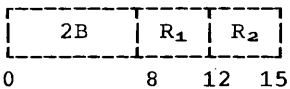
SER R_1, R_2 [RR, Short Operands]



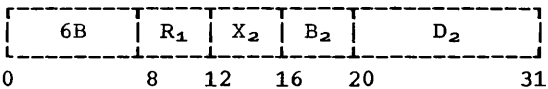
SE $R_1, D_2(X_2, B_2)$ [RX, Short Operands]



SDR R_1, R_2 [RR, Long Operands]



SD $R_1, D_2(X_2, B_2)$ [RX, Long Operands]



The second operand is subtracted from the first operand, and the normalized difference is placed in the first operand location.

In short-precision, the low-order halves of the floating-point registers are ignored and remain unchanged.

The SUBTRACT NORMALIZED is similar to ADD NORMALIZED, except that the sign of the second operand is inverted before addition.

The sign of the difference is derived by the rules of algebra. The sign of a difference with zero result fraction is always positive.

Condition Code:

- 0 Result fraction is zero
- 1 Result is less than zero
- 2 Result is greater than zero
- 3 --

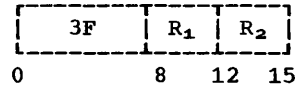
Program Interruptions:

- Protection (fetch violation by SE and SD only)
- Addressing (SD and SE only)
- Specification
- Significance
- Exponent Overflow
- Exponent Underflow

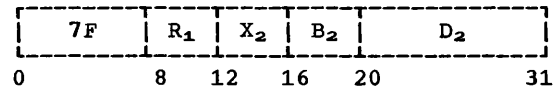
JUNE 1, 1971

5.6.9 SUBTRACT UNNORMALIZED

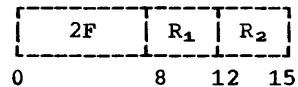
SUR R_1, R_2 [RR, Short Operands]



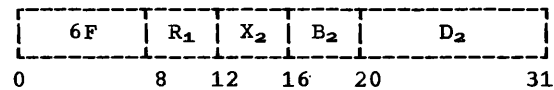
SU $R_1, D_2(X_2, B_2)$ [RX, Short Operands]



SWR R_1, R_2 [RR, Long Operands]



SW $R_1, D_2(X_2, B_2)$ [RX, Long Operands]



The second operand is subtracted from the first operand, and the unnormalized difference is placed in the first operand location.

In short-precision, the low-order halves of the floating-point registers are ignored and remain unchanged.

The SUBTRACT UNNORMALIZED is similar to ADD UNNORMALIZED, except for the inversion of the sign of the second operand before addition.

The sign of the difference is derived by the rules of algebra. The sign of a difference with zero result fraction is always positive.

Condition Code:

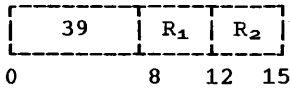
- 0 Result fraction is zero
- 1 Result is less than zero
- 2 Result is greater than zero
- 3 --

Program Interruptions:

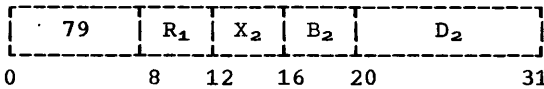
- Protection (fetch violation by SU and SW only)
- Addressing (SW and SU only)
- Specification
- Significance
- Exponent Overflow

5.6.10 COMPARE

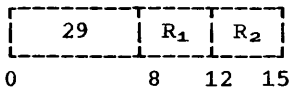
CER R_1, R_2 [RR, Short Operands]



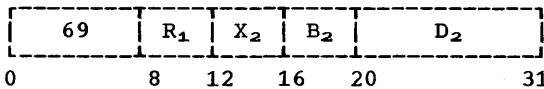
CE $R_1, D_2(X_2, B_2)$ [RX, Short Operands]



CDR R_1, R_2 [RR, Long Operands]



CD $R_1, D_2(X_2, B_2)$ [RX, Long Operands]



The first operand is compared with the second operand, and the condition code indicates the result.

In short-precision, the low-order halves of the floating-point registers are ignored.

Comparison is algebraic, taking into account the sign, fraction, and exponent of each number. An exponent inequality is not decisive for magnitude determination since the fractions may have different numbers of leading zeros. An equality is established by following the rules for normalized floating-point subtraction. When the intermediate sum, including the guard digit, is zero, the operands are equal. Neither operand is changed as a result of the operation.

Exponent-overflow, exponent-underflow, or lost-significance exception cannot occur.

Condition Code:

- 0 Operands are equal
- 1 First operand is low
- 2 First operand is high
- 3 --

Program Interruptions:

- Protection (fetch violation by CE and CD only)
- Addressing (CD and CE only)
- Specification

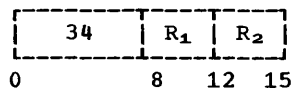
PROGRAMMING NOTE

Numbers with zero fraction compare equal even when they differ in sign or characteristic.

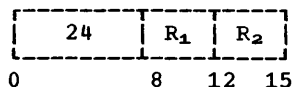
JUNE 1, 1971

5.6.11 HALVE

HER R_1, R_2 [RR, Short Operands]



HDR R_1, R_2 [RR, Long Operands]



The second operand is divided by two, and the normalized quotient is placed in the first operand location. The second operand remains unchanged.

In short-precision, the low-order half of the floating-point registers remains unchanged.

The fraction of the second operand is shifted right one bit position, placing the contents of the low-order bit position into the high-order bit position of the guard digit and introducing a zero into the high-order bit position of the fraction. The intermediate result is subsequently normalized, and the normalized quotient is placed in the first-operand location. The guard digit participates in the normalization.

When normalization causes the characteristic to become less than zero, exponent-underflow occurs. If the exponent-underflow mask is zero, the sign, characteristic, and fraction are set to zero, thus making the result a true zero. If the exponent-underflow mask is one, a program interruption occurs. The result is normalized, its sign and fraction remain correct, and the characteristic is made 128 larger than the correct characteristic.

When the fraction of the second operand is zero, the sign, characteristic, and fraction of the result are made zero. No normalization is attempted, and a significance exception is not recognized.

Condition Code: the code remains unchanged.

Program Interruptions:

Specification
Exponent Underflow

PROGRAMMING NOTE

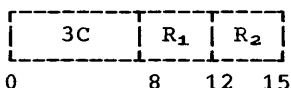
In short- and long-precision, the halve operation is identical to a divide operation with the number two as divisor. In long-precision, the halve operation is identical to a multiply operation with one-half as a multiplier. In short-precision, HALVE differs from multiplication with one-half as the multiplier to the extent that halving preserves the contents of the low-order half of the register.

The result of HALVE is replaced by a true zero only when the second-operand fraction is zero, or when exponent underflow occurs with the exponent-underflow mask set to zero. When the fraction of the second operand is zero, except for the low-order bit position, the

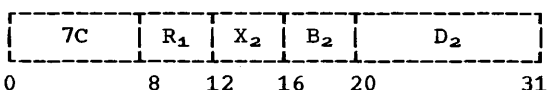
low-order one is shifted into the guard digit position and participates in the postnormalization.

5.6.12 MULTIPLY

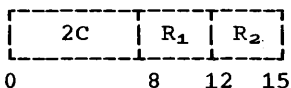
MER R_1, R_2 [RR, Short Operands]



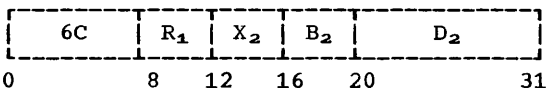
ME $R_1, D_2(X_2, B_2)$ [RX, Short Operands]



MDR R_1, R_2 [RR, Long Operands]



MD $R_1, D_2(X_2, B_2)$ [RX, Long Operands]



The normalized product of multiplier (the second operand) and multiplicand (the first operand) replaces the multiplicand.

The multiplication of two floating-point numbers consists of a characteristic addition and a fraction multiplication. The sum of the characteristics less 64 is used as the characteristic of an intermediate product. The sign of the product is determined by the rules of algebra.

The product fraction is normalized by prenormalizing the operands and postnormalizing the intermediate product, if necessary. The intermediate product characteristic is reduced by the number of left-shifts. For long operands, the intermediate product fraction is truncated to 15 digits before the left-shifting, if any. For short operands (six-digit fractions), the product fraction has the full 14 digits of the long format, and the two low-order fraction digits are accordingly always zero.

Exponent overflow occurs if the final product characteristic exceeds 127. The operation is completed and a program interruption occurs. The fraction is normalized and correct, the sign is correct, and the characteristic is 128 smaller than the correct characteristic. The overflow exception does not occur for an intermediate product characteristic exceeding 127 when the final characteristic is brought within range because of normalization.

Exponent underflow occurs if the final product characteristic is less than zero. If the corresponding mask bit is one, a program interruption occurs. The fraction is normalized and correct, the sign is correct, and the characteristic is 128 larger than the correct characteristic.

JUNE 1, 1971

If the corresponding mask bit is not one, the result is made a true zero. Underflow is not signaled when an operand's characteristic becomes less than zero during prenormalization, and the correct characteristic and fraction value are used in the multiplication.

When all 15 digits of the intermediate product fraction are zero, the product sign and characteristic are made zero, yielding a true zero result. No interruption for exponent underflow and exponent overflow can occur when the result fraction is zero. The program interruption for lost significance is never taken for multiplication.

Condition Code: the code remains unchanged.

Program Interruptions:

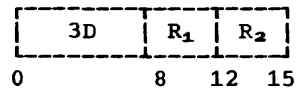
Protection (fetch violation by ME and MD only)
Addressing (MD and ME only)
Specification
Exponent Overflow
Exponent Underflow

PROGRAMMING NOTE

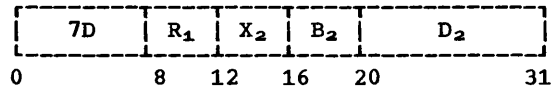
Interchanging the two operands in a floating-point multiplication does not affect the value of the product.

5.6.13 DIVIDE

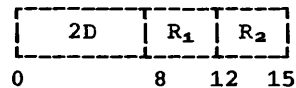
DER R_1, R_2 [RR, Short Operands]



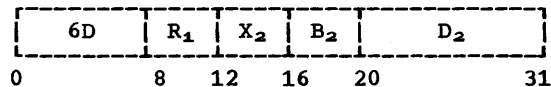
DE $R_1, D_2(X_2, B_2)$ [RX, Short Operands]



DDR R_1, R_2 [RR, Long Operands]



DD $R_1, D_2(X_2, B_2)$ [RX, Long Operands]



The dividend (the first operand) is divided by the divisor (the second operand) and replaced by the quotient. No remainder is preserved.

In short-precision, the low-order halves of the floating-point registers are ignored and remain unchanged.

JUNE 1, 1971

A floating-point division consists of a characteristic subtraction and a fraction division. The difference between the dividend and divisor characteristics plus 64 is used as an intermediate quotient characteristic. The sign of the quotient is determined by the rules of algebra.

The quotient fraction is normalized by prenormalizing the operands. Postnormalizing the intermediate quotient is never necessary, but a right-shift may be called for. The intermediate-quotient characteristic is adjusted for the shifts. All dividend fraction digits participate in forming the quotient, even if the normalized dividend fraction is larger than the normalized divisor fraction. The quotient fraction is truncated to the desired number of digits.

A program interruption for exponent overflow occurs when the final-quotient characteristic exceeds 127. The operation is completed. The fraction is correct and normalized, the sign is correct, and the characteristic is 128 smaller than the correct characteristic.

If the final quotient characteristic is less than zero and the mask bit is one, a program interruption for exponent underflow occurs. The fraction is correct and normalized, the sign is correct, and the characteristic is 128 larger than the correct characteristic. If the corresponding mask bit is not one, the result is made a true zero. Underflow is not signaled for the intermediate quotient or for the operand characteristics during prenormalization.

When division by a divisor with zero fraction is attempted, the operation is suppressed. The dividend remains unchanged and a program interruption for floating-point divide occurs. When the dividend fraction is zero the quotient fraction will be zero. The quotient sign and characteristic are made zero, yielding a true zero result without taking the program interruption for exponent underflow and exponent overflow. The program interruption for lost significance is never taken for division.

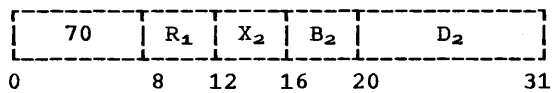
Condition Code: the code remains unchanged.

Program Interruptions:

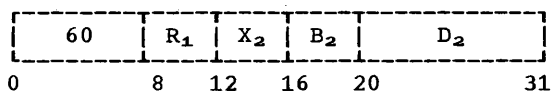
- Protection (fetch violation by DE and DD only)
- Addressing (DD and DE only)
- Specification
- Exponent Overflow
- Exponent Underflow
- Floating-Point Divide

5.6.14 STORE

STE $R_1, D_2(X_2, B_2)$ [RX, Short Operands]



STD $R_1, D_2(X_2, B_2)$ [RX, Long Operands]



JUNE 1, 1971

The first operand is stored at the second operand location.

In short-precision, the low-order half of the first operand register is ignored. The first operand remains unchanged.

Condition Code: the code remains unchanged.

Program Interruptions:

Addressing
Protection (store violation)
Specification

5.7 FLOATING-POINT ARITHMETIC EXCEPTIONS

Exceptional operation codes, operand designations, data, or results cause a program interruption. When the interruption occurs the current PSW is stored as an old PSW and a new PSW is obtained. The interruption code in the old PSW identifies the cause of the interruption. The following exceptions cause a program interruption in floating-point arithmetic.

Protection: The key of an operand in storage does not match the protection key in the PSW. The operation is suppressed on a store violation. Therefore, the condition code and data in registers remain unchanged. On a fetch violation, the operation is terminated; result data and the condition code are unpredictable.

Addressing: An address designates a location outside the available storage for the installed system, or outside the configured storage, or storage assigned by the storage address translator for a particular computing element.

The operation is terminated. The result data and the condition code, if affected, are unpredictable, and should not be used for further computation.

Specification: A short operand is not located on a 32-bit boundary, or a long operand is not located on a 64-bit boundary; or, a floating-point register address other than 0, 2, 4, or 6 is specified.

The instruction is suppressed. Therefore, the condition code, and data in registers and storage remain unchanged.

The address restrictions do not apply to the components from which an address is generated -- the content of the D_2 field and the contents of the registers specified by X_2 and B_2 .

Exponent Overflow: The result characteristic of an addition, subtraction, multiplication, or division exceeds 127, and the result fraction is not zero.

The operation is completed, and a program interruption occurs. The fraction is normalized, and the sign and fraction of the result remain correct. The result characteristic is made 128 smaller than the correct characteristic. For addition and subtraction, the condition code is set to 1 when the result is less than zero, and the condition code is set to 2 when the result is greater than zero. For multiplication and division, the condition code remains unchanged.

Exponent Underflow: The result characteristic of an addition, subtraction, multiplication, halving, or division is less than zero, and the result fraction is not zero.

JUNE 1, 1971

The operation is completed, and a program interruption occurs, if the exponent-underflow mask bit (PSW bit 38) is one.

The setting of the exponent-underflow mask also affects the result of the operation. When the mask bit is zero, the sign, characteristic, and fraction are set to zero, thus making the result a true zero. When the mask bit is one, the fraction is normalized, the characteristic is made 128 larger than the correct characteristic, and the sign and fraction remain correct.

For addition and subtraction, the condition code is set to 0 when the exponent-underflow mask bit is zero. With the mask bit one, the condition code for addition and subtraction is set to 1 when the result is less than zero, and the condition code is set to 2 when the result is greater than zero. For multiplication, halving, and division, the condition code is left unchanged.

Significance: The result fraction of an addition or subtraction is zero. A program interruption occurs if the significance mask bit (PSW bit 39) is one. The mask bit affects also the result of the operation.

When the significance mask bit is a zero, the operation is completed by replacing the result with a true zero. When the significance mask bit is one, the operation is completed without further change to the characteristic of the result. In either case the condition code is set to 0.

Floating-Point Divide: Division by a number with zero fraction is attempted.

The division is suppressed; therefore, the condition code and data in registers and storage remain unchanged.

JUNE 1, 1971

CHAPTER 6: LOGICAL OPERATIONS

A set of instructions is provided for the logical manipulation of data. Generally, the operands are treated as eight-bit bytes. In a few cases the left or right four bits of a byte are treated separately, or operands are shifted a bit at a time. The operands are either in storage or in the general registers. Some operands are introduced from the instruction stream.

Processing of data in storage proceeds left to right through fields which may start at any byte position. In the general registers the processing, as a rule, involves the entire register contents.

Except for the editing instructions, data are not treated as numbers. Editing provides a transformation from packed decimal digits to alphanumeric characters.

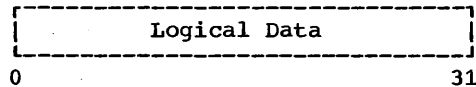
The set of logical operations includes moving, comparing, bit connecting, bit testing, translating, editing, and shift operations.

The condition code is set as a result of all logical comparing, connecting, testing, and editing operations.

6.1 DATA FORMAT

Data reside in general registers or in storage or are introduced from the instruction stream. The data size may be a single or double word, a single character, or variable in length. When two operands participate they have equal length, except in the editing instructions.

Fixed-Length Logical Information

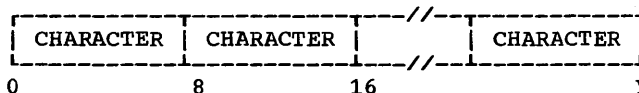


Data in general registers normally occupy all 32 bits. Bits are treated uniformly, and no distinction is made between sign and numeric bits. In a few operations only the low-order eight bits of a register participate, leaving the remaining 24 bits unchanged. In some shift operations 64 bits of an even/odd pair of registers participate.

The LOAD ADDRESS introduces a 24-bit address into a general register. The high-order 8 bits of the register are made zero.

In storage-to-register operations, the storage data occupy either a word of 32 bits or a byte of 8 bits. The word must be located on word boundaries; i.e., its address must have the two low-order bits zero.

Variable-Length Logical Information



JUNE 1, 1971

In storage-to-storage operations, data have a variable field-length format, starting at any byte address and continuing for up to a total of 256 bytes. Processing is left to right.

Operations introducing data from the instruction stream into storage, as immediate data, are restricted to an eight-bit byte. Only one byte is introduced from the instruction stream and only one byte in storage participates.

Use of general register 1 is implied in TRANSLATE AND TEST and EDIT AND MARK. A twenty-four bit address may be placed in this register during these operations. The TRANSLATE AND TEST also implies general register 2. The low-order eight bits of register 2 may be replaced by a function byte during a translate-and-test operation.

Editing requires a packed decimal field and generates zoned decimal digits. The digits, signs, and zones are recognized and generated as for decimal arithmetic. Otherwise, no internal data structure is required, and all bit configurations are considered valid.

The translating operations use a list of arbitrary values. A list provides a relation between an argument (the quantity used to reference the list) and the function (the content of the location related to the argument). The purpose of the translation may be to convert data from one code to another code or to perform a control function.

A list is specified by an initial address -- the address designating the leftmost byte location of the list. The byte from the operand to be translated is the argument. The actual address used to address the list is obtained by adding the argument to the low-order positions of the initial address. As a consequence, the list contains 256 eight-bit function bytes. In cases where it is known that not all eight-bit argument values will occur, it may be possible to reduce the size of the list.

In a storage-to-storage operation the operand fields may be defined in such a way that they overlap. The effect of this overlap depends upon the operation. When the operands remain unchanged, as in COMPARE or TRANSLATE AND TEST, overlapping does not affect the execution of the operation. In the case of MOVE, EDIT, and TRANSLATE, one operand is replaced by new data, and the execution of the operation may be affected by the amount of overlap and the manner in which data are fetched or stored. For purposes of evaluating the effect of overlapped operands, consider that data are handled one eight-bit byte at a time. All overlapping fields are considered valid, but in editing overlapping fields give unpredictable results.

6.2 CONDITION CODE

The results of most logical operations are used to set the condition code in the PSW. The LOAD ADDRESS, INSERT CHARACTER, STORE CHARACTER, TRANSLATE, and the moving and shift operations leave this code unchanged. The condition code can be used for decision-making by subsequent branch-on-condition instructions.

The condition code can be set to reflect five types of results for logical operations:

- | For COMPARE LOGICAL the codes 0, 1, or 2 indicate that the first operand is equal, low, or high.
- | For the logical-connectives the codes 0 and 1 indicate a zero or nonzero result field.

JUNE 1, 1971

| For TEST UNDER MASK the codes 0, 1, or 3 indicate that the selected bits are all-zero, mixed zero and one, or all-one.

| For TRANSLATE AND TEST the codes 0, 1, or 2 indicate an all-zero function byte, a nonzero function byte with the operand incompletely tested, or a last function byte nonzero.

| For editing the codes 0, 1, or 2 indicate a zero, less than zero, or greater than zero content of the last result field.

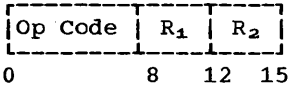
TABLE 6-I CONDITION CODE SETTING FOR LOGICAL OPERATIONS

	0	1	2	3
AND	zero	not zero	--	--
COMPARE LOGICAL	equal	low	high	--
EDIT	zero	< zero	> zero	--
EDIT AND MARK	zero	< zero	> zero	--
EXCLUSIVE OR	zero	not zero	--	--
OR	zero	not zero	--	--
TEST UNDER MASK	zero	mixed	--	one
TRANSLATE AND TEST	zero	incomplete	complete	--

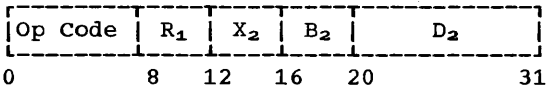
6.3 INSTRUCTION FORMAT

Logical instructions use the following five formats:

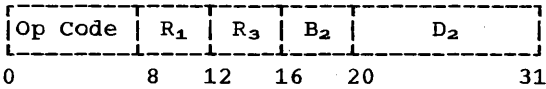
RR Format



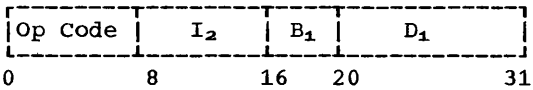
RX Format



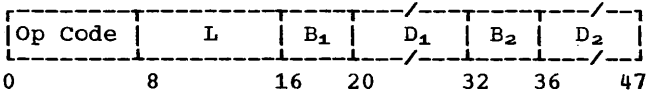
RS Format



SI Format



SS Format



In the RR, RX, and RS formats, the content of the register specified by R₁ is called the first operand.

In the SI and SS formats the content of the general register specified by B₁ is added to the content of the D₁ field to form an address. This address designates the leftmost byte of the first operand field. The number of bytes to the right of this first byte is specified by the L field in the SS format. In the SI format the operand size is one byte.

In the RR format, the R₂ field specifies the register containing the second operand. The same register may be specified for the first and second operand.

In the RX format, the contents of the general registers specified by the X₂ and B₂ fields are added to the content of the D₂ field to form the address of the second operand.

In the RS format, used for shift operations, the content of the general register specified by the B₂ field is added to the content of the D₂ field. This sum is not used as an address, but specifies the number of bits of the shift. The R₃ field is ignored in the shift operations.

JUNE 1, 1971

In the SI format, the second operand is the eight-bit immediate data field, I_2 , of the instruction.

In the SS format, the content of the general register specified by B_2 is added to the content of the D_2 field to form the address of the second operand. The second operand field has the same length as the first operand field.

A zero in any of the X_2 , B_1 , or B_2 fields indicates the absence of the corresponding address or shift-amount component. An instruction can specify the same general register both for address modification and for operand location. Address modification is always completed prior to operation execution.

Results replace the first operand, except in STORE CHARACTER, where the result replaces the second operand. A variable-length result is never stored outside the field specified by the address and length.

The contents of all general registers and storage locations participating in the addressing or execution of an operation generally remain unchanged. Exceptions are the result locations, general register 1 in EDIT AND MARK, and general registers 1 and 2 in TRANSLATE AND TEST.

6.4 INSTRUCTIONS

The logical instructions, their mnemonics, formats, and operation codes follow. The table also indicates when the condition code is set and the exceptions which cause a program interruption.

JUNE 1, 1971

TABLE 6-II LOGICAL INSTRUCTIONS

NAME	MNEMONIC	TYPE	CC	EXCEPTIONS	CODE	SECTION
AND	NR	RR	C		14	6.4.5
AND	N	RX	C	P,A,S	54	6.4.5
AND	NI	SI	C	P,A	94	6.4.5
AND	NC	SS	C	P,A	D4	6.4.5
COMPARE LOGICAL	CLR	RR	C		15	6.4.4
COMPARE LOGICAL	CL	RX	C	P,A,S	55	6.4.4
COMPARE LOGICAL	CLI	SI	C	P,A	95	6.4.4
COMPARE LOGICAL	CLC	SS	C	P,A	D5	6.4.4
EDIT	ED	SS	C	P,A,D	DE	6.4.14
EDIT AND MARK	EDMK	SS	C	P,A,D	DF	6.4.15
EXCLUSIVE OR	XR	RR	C		17	6.4.7
EXCLUSIVE OR	X	RX	C	P,A,S	57	6.4.7
EXCLUSIVE OR	XI	SI	C	P,A	97	6.4.7
EXCLUSIVE OR	XC	SS	C	P,A	D7	6.4.7
INSERT CHARACTER	IC	RX		P,A	43	6.4.9
LOAD ADDRESS	LA	RX			41	6.4.11
MOVE (character)	MVC	SS		P,A	D2	6.4.1
MOVE (immediate)	MVI	SI		P,A	92	6.4.1
MOVE NUMERICS	MVN	SS		P,A	D1	6.4.2
MOVE ZONES	MVZ	SS		P,A	D3	6.4.3
OR	OR	RR	C		16	6.4.6
OR	O	RX	C	P,A,S	56	6.4.6
OR	OI	SI	C	P,A	96	6.4.6
OR	OC	SS	C	P,A	D6	6.4.6
SHIFT RIGHT DOUBLE LOGICAL	SRDL	RS		S	8C	6.4.19
SHIFT RIGHT SINGLE LOGICAL	SRL	RS			88	6.4.17
SHIFT LEFT DOUBLE LOGICAL	SIDL	RS		S	8D	6.4.18
SHIFT LEFT SINGLE LOGICAL	SLL	RS			89	6.4.16
STORE CHARACTER	STC	RX		P,A	42	6.4.10
TEST UNDER MASK	TM	SI	C	P,A	91	6.4.8
TRANSLATE	TR	SS		P,A	DC	6.4.12
TRANSLATE AND TEST	TRT	SS	C	P,A	DD	6.4.13

Legend

A Addressing exception
 C Condition code is set
 D Data exception
 P Protection exception
 S Specification exception

PROGRAMMING NOTE

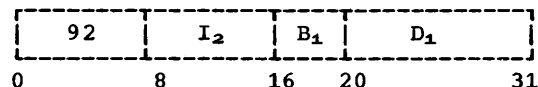
The fixed-point loading and storing instructions also may be used for logical operations. (See Chapter 3)

JANUARY 31, 1973

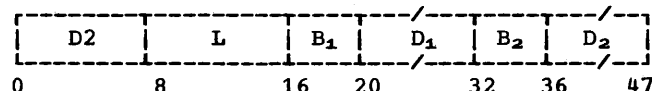
TNL: GN31-3005

6.4.1 MOVE

MVI D₁(B₁), I₂ [SI]



MVC D₁(L, B₁), D₂(B₂) [SS]



The second operand is placed in the first operand location.

The SS format is used for a storage-to-storage move. The SI format introduces one eight-bit byte from the instruction stream.

In storage-to-storage movement the fields may overlap in any desired way. Movement is left to right through each field a byte at a time.

The bytes to be moved are not changed or inspected.

Condition Code: the code remains unchanged.

Program Interruptions:

Protection (store violation for MVI; store or fetch violation for MVC)
Addressing

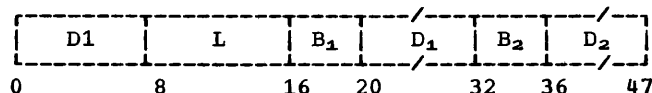
PROGRAMMING NOTES

It is possible to propagate one character through an entire field by having the first operand field start one character to the right of the second operand field.

This instruction does a Common I-Fetch into the destination SE. Therefore, it is advisable to do a Store Multiple into the first doubleword to prevent a possible Fetch Check.

6.4.2 MOVE NUMERICS

MVN D₁(L, B₁), D₂(B₂) [SS]



The low-order four bits of each byte in the second operand field, the numerics, are placed in the low-order bit positions of the corresponding bytes in the first operand fields.

The instruction is storage to storage. Movement is left to right through each field one byte at a time, and the fields may overlap in any desired way.

The numerics are not changed or checked for validity. The high-order four bits of each byte, the zones, remain unchanged in both operand fields.

Condition Code: the code remains unchanged.

Program Interruptions:

Protection (store or fetch violation)
Addressing

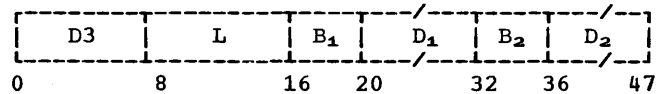
PROGRAMMING NOTES

It is possible to propagate one character through an entire field by having the first operand field start one character to the right of the second operand field.

This instruction does a Common I-Fetch into the destination SE. Therefore it is advisable to do a Store Multiple into the first doubleword to prevent a possible Fetch Check.

6.4.3 MOVE ZONES

MVZ $D_1(L, B_1), D_2(B_2)$ [SS]



The high-order four bits of each byte in the second operand field, the zones, are placed in the high-order four bit positions of the corresponding bytes in the first operand field.

The instruction is storage to storage. Movement is left to right through each field one byte at a time, and the fields may overlap in any desired way.

The zones are not changed or checked for validity. The low-order four bits of each byte, the numerics, remain unchanged in both operand fields.

Condition Code: the code remains unchanged.

Program Interruptions:

Protection (store or fetch violation)
Addressing

PROGRAMMING NOTES

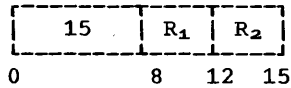
It is possible to propagate one character through an entire field by having the first operand field start one character to the right of the second operand field.

This instruction does a Common I-Fetch into the destination SE. Therefore it is advisable to do a Store Multiple into the first doubleword to prevent a possible Fetch Check.

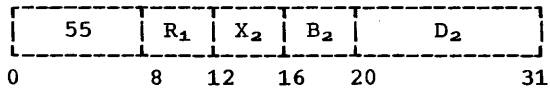
JUNE 1, 1971

6.4.4 COMPARE LOGICAL

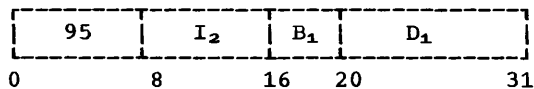
CLR R₁, R₂ [RR]



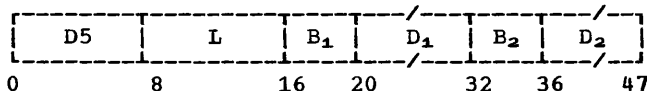
CL R₁, D₂(X₂, B₂) [RX]



CLI D₁(B₁), I₂ [SI]



CLC D₁(L, B₁), D₂(B₂) [SS]



The first operand is compared with the second operand, and the result is indicated in the condition code.

The instructions allow comparisons that are register to register, storage to register, instruction to storage, and storage to storage.

Comparison is binary and all codes are valid. The operation proceeds left to right and ends as soon as an inequality is found or an end of the fields is reached. However, when part of an operand in CLC is specified in an unavailable location, the operation may be terminated by the addressing exception, even though an inequality could have been found in a comparison of the available operand parts.

Condition Code:

0	Operands are equal
1	First operand is low
2	First operand is high
3	--

Program Interruptions:

Protection (fetch violation for CL, CLI, and CLC only)
Addressing (CL, CLI, and CLC only)
Specification (CL only)

PROGRAMMING NOTE

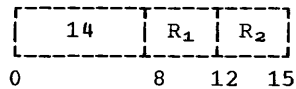
The COMPARE LOGICAL is unique in treating all bits alike as part of an unsigned binary quantity. In variable-length operation, comparison is left to right and may extend to field lengths of 256 bytes. The operation may be used to compare unsigned packed decimal fields or alphanumeric information in any code that has a collating sequence

JUNE 1, 1971

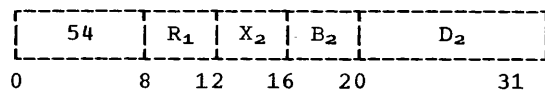
based on ascending or descending binary values. For example, EBCDIC has a collating sequence based on ascending binary values.

6.4.5 AND

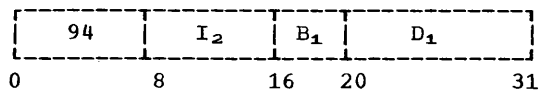
NR R_1, R_2 [RR]



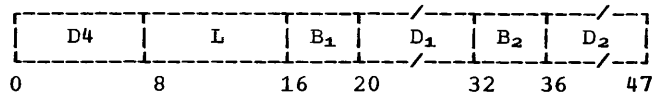
N $R_1, D_2(X_2, B_2)$ [RX]



NI $D_1(B_1), I_2$ [SI]



NC $D_1(L, B_1), D_2(B_2)$ [SS]



The logical product (AND) of the bits of the first and second operand is placed in the first operand location.

Operands are treated as unstructured logical quantities, and the connective AND is applied bit by bit. A bit position in the result is set to one if the corresponding bit positions in both operands contain a one; otherwise, the result bit is set to zero. All operands and results are valid.

For the NI instruction, the byte in storage, if in a Storage Element (SE), is fetched and the logical operation is performed. No other access to this half (ODD or EVEN) of the SE is permitted between the moment of fetching and the moment of storing the result byte.

Condition Code:

- 0 Result is zero
- 1 Result not zero
- 2 --
- 3 --

Program Interruptions:

- Protection (fetch violation only for N; store violation only for NI; store or fetch violation for NC)
- Addressing (N, NI, NC only)
- Specification (N only)

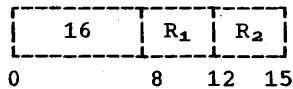
JUNE 1, 1971

PROGRAMMING NOTE

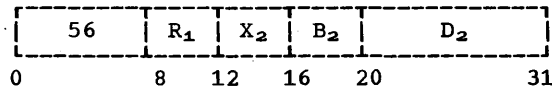
The AND may be used to set a bit to zero.

6.4.6 OR

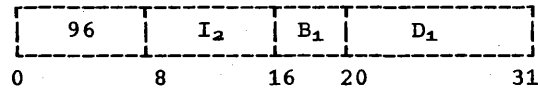
OR R_1, R_2 [RR]



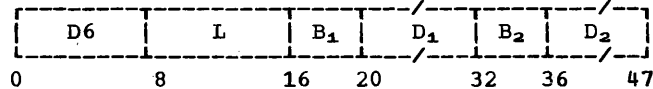
O $R_1, D_2(X_2, B_2)$ [RX]



OI $D_1, (B_1), I_2$ [SI]



OC $D_1(L, B_1), D_2(B_2)$ [SS]



The logical sum (OR) of the bits of the first and second operand is placed in the first operand location.

Operands are treated as unstructured logical quantities, and the connective inclusive OR is applied bit by bit. A bit position in the result is set to one if the corresponding bit position in one or both operands contains a one; otherwise, the result bit is set to zero. All operands and results are valid.

For the OI instruction, the byte in storage, if in a Storage Element (SE), is fetched and the logical operation is performed. No other access to this half (ODD or EVEN) of the SE is permitted between the moment of fetching and the moment of storing the result byte.

Condition Code:

- 0 Result is zero
- 1 Result not zero
- 2 --
- 3 --

Program Interruptions:

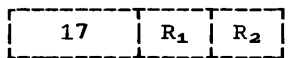
Protection (fetch violation only for O; store violation only for OI; store or fetch violation for OC)
Addressing (O, OI, OC only)
Specification (O only)

PROGRAMMING NOTE

The OR may be used to set a bit to one.

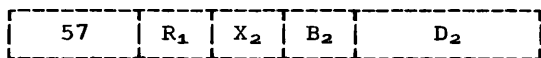
6.4.7 EXCLUSIVE OR

XR R₁,R₂ [RR]



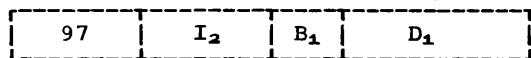
0 8 12 15

X R₁,D₂(X₂,B₂) [RX]



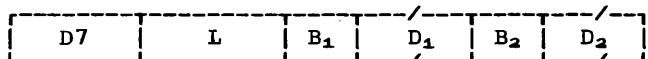
0 8 12 16 20 31

XI D₁, (B₁), I₂ [SI]



0 8 16 20 31

XC D₁(L, B₁), D₂(B₂) [SS]



0 8 16 20 32 36 47

The modulo-two sum (exclusive OR) of the bits of the first and second operand is placed in the first operand location.

Operands are treated as unstructured logical quantities, and the connective exclusive OR is applied bit by bit. A bit position in the result is set to one if the corresponding bit positions in the two operands are unlike; otherwise, the result bit is set to zero.

The instruction differs from AND and OR only in the connective applied.

For the XI instruction, the byte in storage, if in a Storage Element (SE), is fetched and the logical operation is performed. No other access to this half (ODD or EVEN) of the SE is permitted between the moment of fetching and the moment of storing the result byte.

Condition Code:

- 0 Result is zero
- 1 Result not zero
- 2 --
- 3 --

Program Interruptions:

Protection (fetch violation only for X; store violation only for XI; store or fetch violation for XC)

JUNE 1, 1971

Addressing (X, XI, XC only)
Specification (X only)

PROGRAMMING NOTE

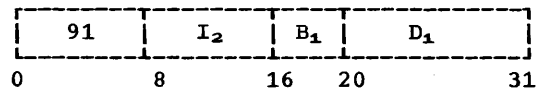
The EXCLUSIVE OR may be used to invert a bit, an operation particularly useful in testing and setting programmed binary bit switches.

Any field EXCLUSIVE OR'ed with itself becomes all zeros.

The sequence A EXCLUSIVE OR'ed B, B EXCLUSIVE OR'ed A, A EXCLUSIVE OR'ed B results in the exchange of the contents of A and B without the use of an auxiliary buffer area.

6.4.8 TEST UNDER MASK

TM $D_1(B_1), I_2$ [SI]



The state of the first operand bits selected by a mask is used to set the condition code.

The byte of immediate data, I_2 , is used as an eight-bit mask. The bits of the mask are made to correspond one for one with the bits of the character in storage specified by the first operand address.

A mask bit of one indicates that the storage bit is to be tested. When the mask bit is zero, the storage bit is ignored. When all storage bits thus selected are zero, the condition code is made 0. The code is also made 0 when the mask is all-zero. When the selected bits are all-one, the code is made 3; otherwise, the code is made 1. The character in storage is not changed.

Condition Code:

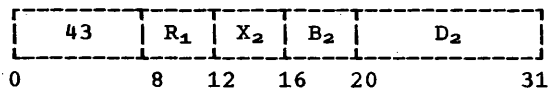
- 0 Selected bits all-zero; mask is all-zero
- 1 Selected bits mixed zero and one
- 2 --
- 3 Selected bits all-one

Program Interruptions:

Protection (fetch violation)
Addressing

6.4.9 INSERT CHARACTER

IC $R_1, D_2(X_2, B_2)$ [RX]



JUNE 1, 1971

The eight-bit character at the second operand address is inserted into bit positions 24-31 of the register specified as the first operand location. The remaining bits of the register remain unchanged.

The instruction is storage to general register. The byte to be inserted is not changed or inspected.

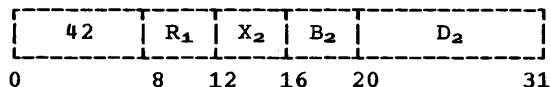
Condition Code: the code remains unchanged.

Program Interruptions:

Protection (fetch violation)
Addressing

6.4.10 STORE CHARACTER

STC $R_1, D_2(X_2, B_2)$ [RX]



Bit positions 24-31 of the register designated as the first operand are placed at the second operand address.

The instruction is general register to storage. The byte to be stored is not changed or inspected.

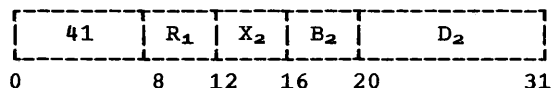
Condition Code: the code remains unchanged.

Program Interruptions:

Protection (store violation)
Addressing

6.4.11 LOAD ADDRESS

LA $R_1, D_2(X_2, B_2)$ [RX]



The address of the second operand is inserted in the low-order 24 bits of the general register specified by R_1 . The remaining bits of the general register are made zero. No storage references for operands take place.

The address specified by the X_2 , B_2 , and D_2 fields is inserted in bits 8-31 of the general register specified by R_1 . Bits 0-7 are set to zero. The address is not inspected for availability, protection, or resolution.

The address computation follows the rules for address arithmetic. Any carries beyond the 24th bit are ignored.

JUNE 1, 1971

Condition Code: the code remains unchanged.

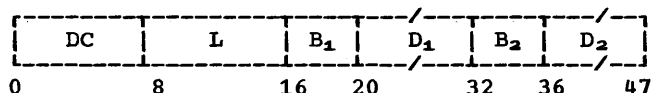
Program Interruptions: none.

PROGRAMMING NOTE

The same general register may be specified by the R_1 , X_2 , and B_2 instruction field, except that general register 0 can be specified only by the R_1 field. In this manner it is possible to increment the low-order 24 bits of a general register, other than 0, by the contents of the D_2 field of the instruction. The register to be incremented should be specified by R_1 and by either X_2 (with B_2 set to zero) or B_2 (with X_2 set to zero).

6.4.12 TRANSLATE

TR $D_1(L, B_1), D_2(B_2)$ [SS]



The eight-bit bytes of the first operand are used as arguments to reference the list designated by the second operand address. Each eight-bit function byte selected from the list replaces the corresponding argument in the first operand.

The bytes of the first operand are selected one by one for translation, proceeding left to right. Each argument byte is added to the entire initial address, the second operand address, in the low-order bit positions. The sum is used as the address of the function byte, which then replaces the original argument byte.

All data are valid. The operation proceeds until the first operand field is exhausted. The list is not altered unless an overlap occurs.

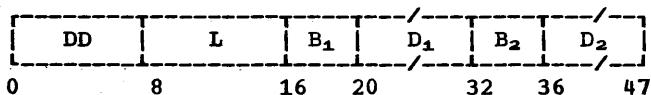
Condition Code: the code remains unchanged.

Program Interruptions:

Protection (store or fetch violation)
Addressing

6.4.13 TRANSLATE AND TEST

TRT $D_1(L, B_1), D_2(B_2)$ [SS]



The eight-bit bytes of the first operand are used as arguments to reference the list designated by the second operand address. Each eight-bit function byte thus selected from the list is used to determine the continuation of the operation. When the function byte is a zero, the operation proceeds by fetching and translating the next argument byte. When the function byte is nonzero, the operation is completed by

JUNE 1, 1971

inserting the related argument address in general register 1, and by inserting the function byte in general register 2.

The bytes of the first operand are selected one by one for translation, proceeding from left to right. The first operand remains unchanged in storage. Fetching of the function byte from the list is performed as in TRANSLATE. The function byte retrieved from the list is inspected for the all-zero combination.

When the function byte is zero, the operation proceeds with the next operand byte. When the first operand field is exhausted before a nonzero function byte is encountered, the operation is completed by setting the condition code to 0. The contents of general registers 1 and 2 remain unchanged.

When the function byte is nonzero, the related argument address is inserted in the low-order twenty-four bits of general register 1. This address points to the argument last translated. The high-order eight bits of register 1 remain unchanged. The function byte is inserted in the low-order eight bits of general register 2. Bits 0-23 of register 2 remain unchanged. The condition code is set to 1 when one or more argument bytes have not been translated. The condition code is set to 2 if the last function byte is nonzero.

Condition Code:

- 0 All function bytes are zero.
- 1 Nonzero function byte before the first operand field is exhausted.
- 2 The last function byte is nonzero.
- 3 --

Program Interruptions:

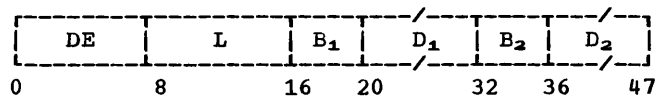
Protection (fetch violation)
Addressing

PROGRAMMING NOTE

The TRANSLATE AND TEST is useful for scanning an input stream and locating delimiters. The stream can thus be rapidly broken into statements or data fields for further processing.

6.4.14 EDIT

ED $D_1(L, B_1), D_2(B_2)$ [SS]



The format of the source (the second operand) is changed from packed and zoned, and is modified under control of the pattern (the first operand). The edited result replaces the pattern.

Editing includes sign and punctuation control, and the suppressing and protecting of leading zeros. It also facilitates programmed blanking of all-zero fields. Several fields may be edited in one operation, and numeric information may be combined with text.

JUNE 1, 1971

The length field applies to the pattern (the first operand). The pattern has the zoned format and may contain any character. The source (the second operand) has the packed format and must contain valid decimal digit and sign codes. The leftmost four bits of a source byte must contain only the codes 0000-1001; the codes 1010-1111 are recognized as a data exception and cause a program interruption. The rightmost four bits are recognized as either a sign or a decimal digit.

Both operands are processed left to right one byte at a time. Overlapping pattern and source fields give unpredictable results.

During the editing process, each character of the pattern is affected in one of three ways:

1. It is left unchanged.
2. It is replaced by a source digit expanded to zoned format.
3. It is replaced by the first character in the pattern, called the fill character.

Which of the three actions takes place is determined by one or more of the following: the type of the pattern character, the state of the significance indicator, and whether the source digit examined is zero.

Pattern Characters: There are four types of pattern characters: digit selector, significance starter, field separator, and message character. Their coding is as follows:

NAME	CODE
Digit selector	0010 0000
Significance starter	0010 0001
Field separator	0010 0010
Message character	Any other

The detection of either a digit selector or a significance starter in the pattern causes an examination to be made of the significance indicator and of a source digit. As a result, either the expanded source digit or the fill character, as appropriate, is selected to replace the pattern character. Additionally, encountering a digit selector or a significance starter may cause the significance indicator to be changed.

The field separator identifies individual fields in a multiple-field editing operation. It is always replaced in the result by the fill character, and the significance indicator is always off after the field separator is encountered.

Message characters in the pattern are either replaced by the fill character or remain unchanged in the result, depending on the state of the significance indicator. They may thus be used for padding, punctuation, or text in the significant portion of a field or for the insertion of sign-dependent symbols.

Fill Character: The fill character is obtained from the pattern as part of the editing operation. The first character of the pattern is used as the fill character. The choice of the fill character is not dependent on the code of the first pattern character and on the editing function, if any, initiated upon recognition of the code. If this character is a digit selector or significance starter, the indicated editing action is taken after the code has been assigned to the fill character.

JUNE 1, 1971

Source Digits: Each time a digit selector or significance starter is encountered in the pattern, a new source digit is examined for placement in the pattern field. The source digit either is zoned and replaces the pattern character or is disregarded. When a sign code is detected in the four high-order bit positions, the operation is terminated.

The source digits are selected one byte at a time, and a source byte is fetched for inspection only once during an editing operation. Each source digit is examined only once for a zero value. The leftmost four bits of each byte are examined first, and the rightmost four bits, when they represent a decimal-digit code, remain available for the next character that calls for a digit examination. At the time the high-order digit of a source byte is examined, the low-order four bits are checked for the existence of a sign code. When a sign code is encountered in the four rightmost bit positions, these bits are not treated as a decimal-digit code, and a new source byte is fetched from storage for the next pattern character that calls for a source-digit examination.

When the source digit is stored in the result, its code is expanded from the packed to the zoned format by attaching a zone. When PSW bit 12 is zero, the preferred EBCDIC zone code 1111 is generated. When PSW bit 12 is one, the preferred USASCII-8 zone code 0101 is generated.

Significance Indicator: The significance indicator, by its on or off state, indicates the significance or nonsignificance, respectively, of subsequent source digits or message characters. Significant source digits replace their corresponding digit selectors or significance starters in the result. Significant message characters remain unchanged in the result.

The significance indicator, by its on or off state, indicates also the negative or positive value, respectively, of the source and is used as one factor in the setting of the condition code.

The indicator is set to the off state, if not already so set, at the start of the editing operation, after a field separator is encountered, or after a source byte is examined that has a plus code in the four low-order bit positions. Any of the codes 1010, 1100, 1110, and 1111 is considered a plus code.

The indicator is set to the on state, if not already so set, when a significance starter is encountered whose source digit is a valid decimal digit, or when a digit selector is encountered whose source digit is a nonzero decimal digit, and if in both instances the source byte does not have a plus code in the four low-order bit positions.

In all other situations, the indicator is not changed. A minus sign code has no effect on the significance indicator.

Result Characters: The field resulting from an editing operation replaces and is equal in length to the pattern. It is composed from pattern characters, fill characters, and zoned source digits.

If the pattern character is a message character and the significance indicator is on, the message character remains unchanged in the result. If the pattern character is a field separator or if the significance indicator is off when a message character is encountered in the pattern, the fill character replaces the pattern character in the result.

If the digit selector or significance starter is encountered in the pattern with the significance indicator off and the source digit zero, the source digit is considered nonsignificant, and the fill character replaces the pattern character. If a digit selector or significance starter is encountered with either the significance indicator on or

JUNE 1, 1971

with a nonzero decimal source digit, the source digit is considered significant, is zoned, and replaces the pattern character in the result.

Result Condition: All digits examined are tested for the code 0000. The sign of the last field edited and whether all source digits in the field contain zeros are recorded in the condition code at the completion of the editing operation.

The condition code is made 0 when all source digits examined in the last field are zeros. When the pattern has no digit selectors or significance starters, the source is not examined, and the condition code is made 0. Similarly, the condition code is made 0 when the last character in the pattern is a field separator or when no digit selector or significance starter is encountered beyond the last field separator.

When the last field edited is nonzero and the significance indicator is on, the condition code is made 1 to indicate a result field less than zero.

When the last field edited is nonzero and the significance indicator is off, the condition code is made 2 to indicate a result field is greater than zero.

For multiple-field editing operations the condition code reflects the sign and value only of the field following the last field separator.

Summary: The following table summarizes the functions of the editing operation. The leftmost four columns list all the significant combinations of the four conditions that can be encountered in the execution of an editing operation. The rightmost two columns list the action taken for each case -- the type of character placed in the result field and the new setting of the significance indicator. See Appendix A for an instruction-use example of EDIT.

Condition Code:

- 0 Source inspected for last field is zero
- 1 Source inspected for last field is less than zero
- 2 Source inspected for last field is greater than zero
- 3 --

Program Interruptions:

Protection (store or fetch violation)
Addressing
Data

Programming Notes:

As a rule, the source is shorter than the pattern because for each source digit a zone and numeric are inserted in the result.

The total number of digit selectors and significance starters in the pattern must equal the number of source digits to be edited.

If the fill character is a blank, if no significance starter appears in the pattern, and if the source is all zeros, the editing operation blanks the result field.

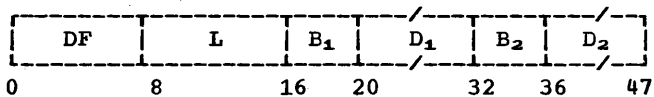
TABLE 6-III EDITING

CONDITIONS				RESULTS	
PATTERN CHARACTER	PREVIOUS STATE OF SIGNIFICANCE INDICATOR	SOURCE DIGIT	LOW-ORDER SOURCE DIGIT IS A PLUS SIGN	RESULT CHARACTER	STATE OF SIGNIFICANCE INDICATOR AT END OF DIGIT EXAMINATION
Digit selector	off	0	*	fill character	off
		1-9	no	source digit	on
		1-9	yes	source digit	off
Significance starter	off	0	no	fill character	on
		0	yes	fill character	off
	on	1-9	no	source digit	on
		1-9	yes	source digit	off
Field separator	*	0-9	no	source digit	on
		0-9	yes	source digit	off
Message character	off	**	**	fill character	off
	on	**	**	message character	on

Legend
 * No effect on result character and new state of significance indicator
 ** Not applicable because source digit not examined

6.4.15 EDIT AND MARK

EDMK D₁(L, B₁), D₂(B₂) [SS]



The format of the source (the second operand) is changed from packed to zoned and is edited under control of the pattern (the first operand). The address of each first significant result digit is recorded in general register 1. The edited result replaces the pattern.

The operation is identical to EDIT, except for the additional function of inserting a byte address in general register 1.

The use of general register 1 is implied. The byte address is inserted in bits 8-31 of this register. The byte address is inserted each time the significance indicator is in the off state and a nonzero digit is inserted in the result field. The address is not inserted when significance is

JUNE 1, 1971

forced by the significance-start character of the pattern while inserting a fill character in the result field. Bits 0-7 are not changed.

Condition Code:

0 Source inspected for last field is zero
1 Source inspected for last field is less than zero
2 Source inspected for last field is greater than zero
3 --

Program Interruptions:

Protection (store or fetch violation)
Addressing
Data

PROGRAMMING NOTE

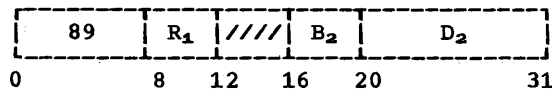
The EDIT AND MARK facilitates the programming of floating currency-symbol insertion. The character address inserted in general register 1 is one more than the address where a floating currency-sign would be inserted. The BRANCH ON COUNT, with zero in the R_2 field, may be used to reduce the inserted address by one.

The character address is not stored when significance is forced. To ensure that general register 1 contains a valid address when significance is forced, it is necessary to place into the register beforehand the address of the pattern character that immediately follows the significance starter.

When a single instruction is used to edit several fields, the address of the first significant result character of each field is inserted into bit positions 8-31 of general register 1. Only the address of the first significant character of the last field is available after the instruction is completed.

6.4.16 SHIFT LEFT SINGLE LOGICAL

SLL $R_1, D_2(B_2)$ [RS]



The first operand is shifted left the number of bits specified by the second operand address.

The second operand address is not used to address data; its low-order six bits indicate the number of bit positions to be shifted. The remainder of the address is ignored.

All 32 bits of the general register specified by R_1 participate in the shift. High-order bits are shifted out without inspection and are lost. Zeros are supplied to the vacated low-order register positions.

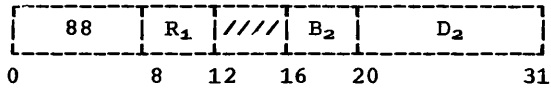
Condition Code: the code remains unchanged.

Program Interruptions: none.

JUNE 1, 1971

6.4.17 SHIFT RIGHT SINGLE LOGICAL

SRL $R_1, D_2(B_2)$ [RS]



The first operand is shifted right the number of bits specified by the second operand address.

The second operand address is not used to address data; its low-order six bits indicate the number of bit positions to be shifted. The remainder of the address is ignored.

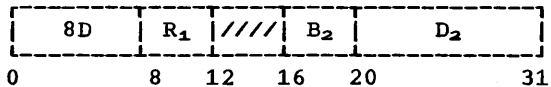
All 32 bits of the general register specified by R_1 participate in the shift. Low-order bits are shifted out without inspection and are lost. Zeros are supplied to the vacated high-order register positions.

Condition Code: the code remains unchanged.

Program Interruptions: none.

6.4.18 SHIFT LEFT DOUBLE LOGICAL

SLDL $R_1, D_2(B_2)$ [RS]



The double-length first operand is shifted left the number of bits specified by the second operand address.

The R_1 field of the instruction specifies an even/odd pair of registers and must contain an even register address. An odd value for R_1 is a specification exception and causes a program interruption. The second operand address is not used to address data; its low-order six bits indicate the number of bit positions to be shifted. The remainder of the address is ignored.

All 64 bits of the even/odd register pair specified by R_1 participate in the shift. High-order bits are shifted out of the even-numbered register without inspection and are lost. Zeros are supplied to the vacated positions of the registers.

Condition Code: the code remains unchanged.

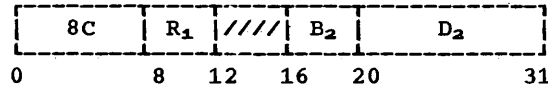
Program Interruptions:

Specification

JUNE 1, 1971

6.4.19 SHIFT RIGHT DOUBLE LOGICAL

SRDL $R_1, D_2(B_2)$ [RS]



The double-length first operand is shifted right the number of bits specified by the second operand address.

The R_1 field of the instruction specifies an even/odd pair of registers and must contain an even register address. An odd value for R_1 is a specification exception and causes a program interruption. The second operand address is not used to address data; its low-order six bits indicate the number of bit positions to be shifted. The remainder of the address is ignored.

All 64 bits of the even/odd register pair specified by R_1 participate in the shift. Low-order bits are shifted out of the odd-numbered register without inspection and are lost. Zeros are supplied to the vacated high order positions of the registers.

Condition Code: the code remains unchanged.

Program Interruptions:

Specification

PROGRAMMING NOTE

The logical shifts differ from the arithmetic shifts in that the high-order bit participates in the shift and is not propagated, the condition code is not changed, and no overflow occurs.

6.5 LOGICAL OPERATION EXCEPTIONS

Exceptional instructions, data or results cause a program interruption. When the interruption occurs the current PSW is stored as an old PSW and a new PSW is obtained. The interruption code in the old PSW identifies the cause of the interruption. The following exceptions cause a program interruption in logical operations.

Protection: The storage key of an accessed location in storage does not match the protection key in the PSW. The operation is suppressed on a store-protection violation. Therefore, the condition code and data in registers and storage remain unchanged. The only exceptions are the variable-length, storage-to-storage operations, which are terminated. The operation is terminated on any fetch-protection violation. For terminated operations, the result data and condition code, if affected, are unpredictable and should not be used for further computation.

Addressing: An address designates a location outside the available storage for the installed system, or outside the configured storage, or storage assigned by the ATR for a particular computing element.

The operation is terminated. The result data and the condition code, if affected, are unpredictable and should not be used for further computation.

JUNE 1, 1971

Specification: A fullword operand in a storage-to-register operation is not located on a 32-bit boundary or an odd register address is specified for a pair of general registers containing a 64-bit operand.

The operation is suppressed. Therefore, the condition code and data in registers and storage remain unchanged.

Data: A digit code of the second operand in EDIT or EDIT AND MARK is invalid. The operation is terminated. The result data and the condition code are unpredictable and should not be used for further computation.

Operand addresses are tested only when used to address storage. Addresses used as a shift amount are not tested. Similarly the address generated by the use of LOAD ADDRESS is not tested. The address restrictions do not apply to the components from which an address is generated -- the contents of the D_1 and D_2 fields, and the contents of the registers specified by X_2 , B_1 , and B_2 .

JUNE 1, 1971

CHAPTER 7: BRANCHING

Instructions are performed by the CE primarily in the sequential order of their locations. A departure from this normal sequential operation may occur when branching is performed. The branching instructions provide a means for making a two-way choice, to refer to a subroutine, or to repeat a segment of coding, such as a loop.

Branching is performed by introducing a branch address as a new instruction address.

The branch address may be obtained from one of the general registers or it may be the address specified by the instruction. The branch address is independent of the updated instruction address.

The detailed operation of branching is determined by the condition code which is part of the program status word (PSW) or by the results in the general registers which are specified in the loop-closing operations.

During a branching operation, the rightmost half of the PSW, including the updated instruction address, may be stored before the instruction address is replaced by the branch address. The stored information may be used to link the new instruction sequence with the preceding sequence.

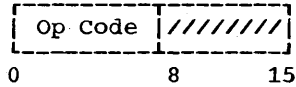
The instruction EXECUTE is grouped with the branching instructions. The branch address of EXECUTE designates a single instruction to be inserted in the instruction sequence. The updated instruction address normally is not changed in this operation, and only the instruction located at the branch address is executed.

7.1 NORMAL SEQUENTIAL OPERATION

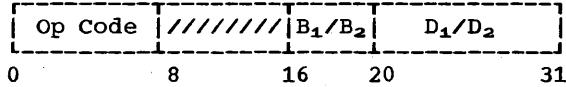
Normally, operation of the CE is controlled by instructions taken in sequence. An instruction is fetched from a location specified by the instruction-address field of the PSW. The instruction address is increased by the number of bytes of the instruction so as to address the next instruction in sequence. This new instruction-address value, called the updated instruction address, replaces the previous contents of the instruction-address field in the PSW. The current instruction is executed, and the same steps are repeated, using the updated instruction address to fetch the next instruction.

Instructions occupy a halfword or a multiple thereof. An instruction may have up to three halfwords. The number of halfwords in an instruction is specified by the first two instruction bits. A 00 code indicates a halfword instruction, codes 01 and 10 indicate a two-halfword instruction, and code 11 indicates a three-halfword instruction.

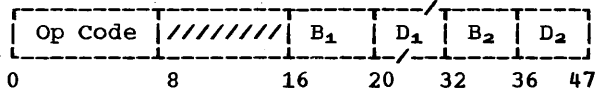
Halfword Format



Two-Halfword Format



Three-Halfword Format



Storage wraps around from the maximum addressable storage location, byte location 16,777,215, to byte location 0. An instruction having its last halfword at the maximum storage location is followed by the instruction at address 0. Also, a multiple-halfword instruction may straddle the upper storage boundary; no special indication is given in these cases.

Conceptually, an instruction is fetched from storage after the preceding operation is completed and prior to execution of the current operation, even though physical storage word size and overlap of instruction execution with storage access may cause actual instruction fetching to be different.

A change in the sequential operation may be caused by branching, status-switching (Chapter 8), interruption (Chapter 9), or manual intervention (Chapter 11). Sequential operation is initiated and terminated from the system control panel.

PROGRAMMING NOTE

It is possible to modify an instruction in storage by the immediately preceding instruction.

7.1.1 SEQUENTIAL OPERATION EXCEPTIONS

Exceptional instruction addresses or operation codes cause a program interruption. When the interruption occurs, the current PSW is stored as an old PSW, and a new PSW is obtained. The interruption code in the old PSW identifies the cause of the interruption. Part of the description of each class of instructions is a list of the program interruptions that may occur for these instructions. The new PSW is not checked for exceptions when it becomes current. These checks occur when the next instruction is executed. The following program interruptions may occur in normal instruction sequencing, independently of the instruction performed.

Operation: An operation exception occurs when the CE attempts to decode an operation code that is not assigned.

Protection: A protection exception occurs when an attempt is made to fetch an instruction halfword from a fetch-protected location. This error can occur when normal instruction sequencing goes from an unprotected region into a protected region, or following a branching operation, load-PSW operation, or an interruption.

JUNE 1, 1971

Addressing: An instruction halfword is located outside the available storage for a particular installation, or outside the configured storage, or storage assigned by the storage address translator for a particular computing element.

Specification: The low-order bit of the instruction address is one.

In each case, the operation is suppressed; therefore, the condition code and data in storage and registers remain unchanged. The instruction address stored as part of the old PSW has been updated by the number of halfwords indicated by the instruction-length code in the old PSW.

PROGRAMMING NOTE

An unavailable instruction address may occur when normal instruction sequencing proceeds from a valid storage region into an unavailable region, or following a branching or status-switching operation.

The odd instruction address can occur only following branching or status-switching operations.

When the last location in available storage contains an instruction which again introduces a valid instruction address, no program interruption is caused, even though the updated instruction address designates an unavailable location.

The main-storage or register address specification of an instruction with unassigned operation code may cause an addressing or specification interruption when the requirements for the particular instruction class are not met.

7.2 DECISION-MAKING

Branching may be conditional or unconditional. Unconditional branches replace the updated instruction address with the branch address. Conditional branches may use the branch address or may leave the updated instruction address unchanged. When branching takes place the instruction is called successful; otherwise it is called unsuccessful.

Whether or not a conditional branch is successful depends on the result of operations concurrent with the branch or preceding the branch. The former case is represented by BRANCH ON COUNT and the branch-on-index instructions. The latter case is represented by BRANCH ON CONDITION, which inspects the condition code in which is reflected the result of a previous arithmetic, logical, multiple computing element, status-switching, display, or I/O operation.

The condition code provides a means for data-dependent decision-making. The code is inspected to qualify the execution of the conditional-branch instructions. The code is set by some operations to reflect the result of the operation, independently of the previous setting of the code. The code remains unchanged for all other operations.

The condition code occupies bit positions 34 and 35 of the PSW. When the PSW is stored during status switching, the condition code is preserved as part of the PSW. Similarly, the condition code is stored as part of the rightmost half of the PSW in a branch-and-link operation. A new condition code is obtained by a LOAD PSW or SET PROGRAM MASK or by the new PSW loaded as a result of an interruption.

The condition code indicates the outcome of some of the arithmetic, logical, multiple computing element, status-switching, display, or I/O operations. It is not changed for any branching operation, except for EXECUTE. In the case of EXECUTE, the condition code is set or left

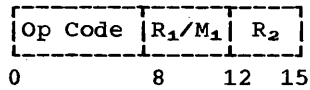
unchanged by the subject instruction, as would have been the case had the subject instruction been in the normal instruction stream.

The table at the end of this chapter lists all instructions capable of altering the condition code and the meaning of the codes for these instructions.

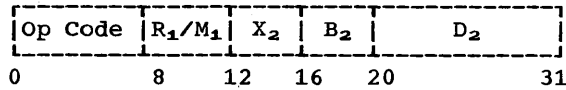
7.3 INSTRUCTION FORMATS

Branching instructions use the following three formats:

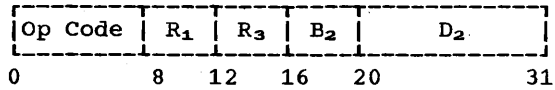
RR Format



RX Format



RS Format



In these formats R₁ specifies the address of a general register. In BRANCH ON CONDITION a mask field (M₁) is used to identify the bit values of the condition code. The branch address is defined differently for the three formats.

In the RR format, the R₂ field specifies the address of a general register containing the branch address, except when R₂ is zero, which indicates no branching. The same register may be specified by R₁ and R₂.

In the RX format, the contents of the general registers specified by the X₂ and B₂ fields are added to the content of the D₂ field to form the branch address.

In the RS format, the content of the general register specified by the B₂ field is added to the content of the D₂ field to form the branch address. The R₃ field in this format specifies the location of the second operand and implies the location of the third operand. The first operand is specified by the R₁ field. The third operand location is always odd. If the R₃ field specifies an even register, the third operand is obtained from the next higher addressed register. If the R₃ field specifies an odd register, the third operand location coincides with the second operand location.

A zero in a B₂ or X₂ field indicates the absence of the corresponding address component.

An instruction can specify the same general register for both address modification and operand location. The order in which the contents of the general registers are used for the different parts of an operation is:

JUNE 1, 1971

- 1) Address computation.
- 2) Arithmetic or link information storage.
- 3) Replacement of the instruction address by the branch address obtained under step 1.

Results are placed in the general register specified by R_1 . Except for the storing of the final results, the contents of all general registers and storage locations participating in the addressing or execution part of an operation remain unchanged.

PROGRAMMING NOTE

In several instructions the branch address may be specified in two ways: in the RX format, the branch address is the address specified by X_2 , B_2 , and D_2 ; in the RR format, the branch address is in the low-order 24 bits of the register specified by R_2 . Note that the relation of the two formats in branch-address specification is not the same as in operand-address specification. For operands the address specified by X_2 , B_2 , and D_2 is the operand address, but the register specified by R_2 contains the operand itself.

7.4 INSTRUCTIONS

The branching instructions and their mnemonics, formats, and operation codes follow. The table also shows the exceptions which cause a program interruption. The subject instruction of EXECUTE follows its own rules for interruptions. The condition code is never changed for branching instructions.

TABLE 7-I BRANCHING INSTRUCTIONS

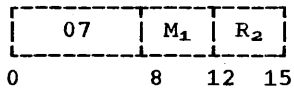
NAME	MNEMONIC	TYPE	CC	EXCEPTIONS	CODE	SECTION
BRANCH AND LINK	BALR	RR			05	7.4.2
BRANCH AND LINK	BAL	RX			45	7.4.2
BRANCH ON CONDITION	BCR	RR			07	7.4.1
BRANCH ON CONDITION	BC	RX			47	7.4.1
BRANCH ON COUNT	BCTR	RR			06	7.4.3
BRANCH ON COUNT	BCT	RX			46	7.4.3
BRANCH ON INDEX HIGH	BXH	RS			86	7.4.4
BRANCH ON INDEX LOW OR EQUAL	BXLE	RS			87	7.4.5
EXECUTE	EX	RX		P, A, S EX	44	7.4.6

Legend

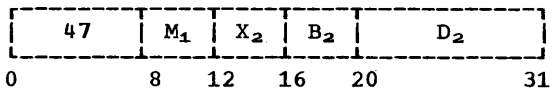
- A Addressing exception
- EX Execute exception
- S Specification exception
- P Protection exception

7.4.1 BRANCH ON CONDITION

BCR M_1, R_2 [RR]



BC $M_1, D_2(X_2, B_2)$ [RX]



The updated instruction address is replaced by the branch address if the state of the condition code is as specified by M_1 ; otherwise, normal instruction sequencing proceeds with the updated instruction address.

The M_1 field is used as a four-bit mask. The four bits of the mask correspond, left to right, with the four condition codes (0, 1, 2, and 3) as shown in the following table.

INSTRUCTION BIT	MASK POSITION VALUE	CONDITION CODE
8	8	0
9	4	1
10	2	2
11	1	3

The branch is successful whenever the condition code has a corresponding mask bit of one.

Condition Code: the code remains unchanged.

Program Interruptions: none.

PROGRAMMING NOTE

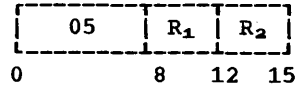
When a branch is to be made on more than one condition code, the pertinent condition codes are specified in the mask as the sum of their mask position values. A mask of 12, for example, specifies that a branch is to be made on condition codes 0 and 1.

When all four mask bits are ones, that is, the mask position value is 15, the branch is unconditional. When all four mask bits are zero or when the R_2 field in the RR format contains zero, the branch instruction is equivalent to a no-operation.

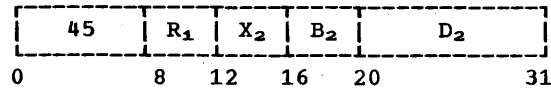
JUNE 1, 1971

7.4.2 BRANCH AND LINK

BALR R_1, R_2 [RR]



BAL $R_1, D_2(X_2, B_2)$ [RX]



The right-most 32 bits of the PSW, including the updated instruction address, are stored as link information in the general register specified by R_1 . Subsequently, the instruction address is replaced by the branch address.

The branch address is determined prior to the storing of the link information. The link information contains the instruction-length code, the condition code, and the program mask bits, as well as the updated instruction address. The instruction-length code is 1 or 2, depending on the format of the BRANCH AND LINK.

Condition Code: the code remains unchanged.

Program Interruptions: none.

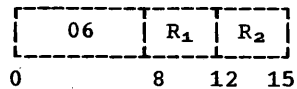
PROGRAMMING NOTE

The link information is stored without branching when in the RR format the R_2 field contains zero.

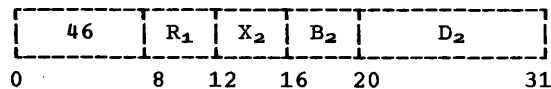
When BRANCH AND LINK is the subject instruction of EXECUTE, the instruction-length code is 2.

7.4.3 BRANCH ON COUNT

BCTR R_1, R_2 [RR]



BCT $R_1, D_2(X_2, B_2)$ [RX]



The content of the general register specified by R_1 is algebraically reduced by one. When the result is zero, normal instruction sequencing proceeds with the updated instruction address. When the result is not zero, the instruction address is replaced by the branch address.

JUNE 1, 1971

The branch address is determined prior to the counting operation. Counting does not change the condition code. The overflow occurring on transition from the maximum negative number to the maximum positive number is ignored. Otherwise the subtraction proceeds as in fixed-point arithmetic, and all 32 bits of the general register participate in the operation.

Condition Code: the code remains unchanged.

Program Interruptions: none.

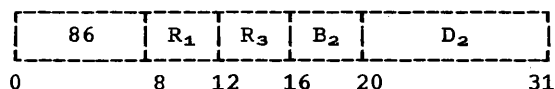
PROGRAMMING NOTE

An initial count of zero results in minus one and causes branching to be executed.

Counting is performed without branching when the R_2 field in the RR format contains zero.

7.4.4 BRANCH ON INDEX HIGH

BXH $R_1, R_3, D_2(B_2)$ [RS]



An increment is added to the first operand, and the sum is compared algebraically with a comparand. Subsequently, the sum is placed in the first operand location, regardless of whether the branch is taken. When the sum is high, the instruction address is replaced by the branch address. When the sum is low or equal, instruction sequencing proceeds with the updated instruction address.

The first operand and the increment are in the registers specified by R_1 and R_3 . The comparand register address is odd and is either one larger than R_3 or equal to R_3 . The branch address is determined prior to the addition and comparison.

Overflow caused by the addition is ignored and does not affect the comparison. Otherwise, the addition and comparison proceed as in fixed-point arithmetic. All 32 bits of the general registers participate in the operations, and negative quantities are expressed in two's-complement notation. When the first operand and comparand locations coincide, the original register contents are used as the comparand.

Condition Code: the code remains unchanged.

Program Interruptions: none.

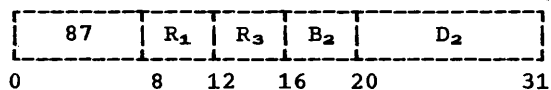
PROGRAMMING NOTE

The name "branch on index high" indicates that one of the major purposes of this instruction is the incrementing and testing of an index value. The increment may be algebraic and of any magnitude.

JUNE 1, 1971

7.4.5 BRANCH ON INDEX LOW OR EQUAL

BXLE $R_1, R_3, D_2(B_2)$ [RS]



An increment is added to the first operand, and the sum is compared algebraically with a comparand. Subsequently, the sum is placed in the first operand location, regardless of whether the branch is taken. When the sum is low or equal, the instruction address is replaced by the branch address. When the sum is high, normal instruction sequencing proceeds with the updated instruction address.

The first operand and the increment are in the registers specified by R_1 and R_3 . The comparand register address is odd and is either one larger than R_3 or equal to R_3 . The branch address is determined prior to the addition and comparison.

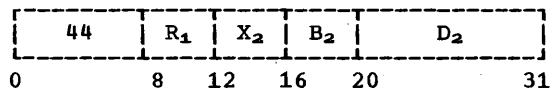
This instruction is similar to BRANCH ON INDEX HIGH, except that the branch is successful when the sum is low or equal compared to the comparand.

Condition Code: the code remains unchanged.

Program Interruptions: none.

7.4.6 EXECUTE

EX $R_1, D_2(X_2, B_2)$ [RX]



The single instruction at the branch address is modified by the content of the general register specified by R_1 , and the resulting subject instruction is executed.

Bits 8-15 of the instruction designated by the branch address are OR'ed with bits 24-31 of the register specified by R_1 , except when register 0 is specified, which indicates that no modification takes place. The subject instruction may be 16, 32, or 48 bits in length. The OR'ing does not change either the content of the register specified by R_1 or the instruction in storage, and is effective only for the interpretation of the instruction to be executed.

The execution and exception handling of the subject instruction are exactly as if the subject instruction were obtained in normal sequential operation, except for instruction address and instruction-length recording.

The instruction address of the PSW is increased by the length of EXECUTE. This updated address and the length code (2) of EXECUTE are stored in the PSW in the event of a branch-and-link subject instruction, or in the event of an interruption.

JUNE 1, 1971

When the subject instruction is a successful branching instruction, the updated instruction address of the PSW is replaced by the branch address of the subject instruction. When the subject instruction in turn is an EXECUTE, an execute exception occurs and results in a program interruption. The effective address of EXECUTE must be even; if not, a specification exception will cause a program interruption.

Condition Code: the code may be set by the subject instruction.

Program Interruptions:

Execute
Protection (fetch violation)
Addressing
Specification

PROGRAMMING NOTE

The OR'ing of eight bits from the general register with the designated instruction permits indirect length, index, mask, immediate data, and arithmetic-register specification.

If the subject instruction is a successful branch, the length code still stands at 2.

An addressing or specification exception may be caused by EXECUTE or by the subject instruction.

7.5 EXECUTE EXCEPTIONS

Exceptional operand designations or a subject-instruction operation code specifying EXECUTE cause a program interruption. When the interruption occurs, the current PSW is stored as an old PSW, and a new PSW is obtained. The interruption code in the old PSW identifies the cause. Exceptions that cause a program interruption in the use of EXECUTE are:

Execute: An EXECUTE instruction has as its subject instruction another EXECUTE.

Protection: An EXECUTE instruction accesses a fetch-protected location whose storage key does not match the protection key in the PSW.

Addressing: The branch address of EXECUTE designates an instruction-halfword location outside the available storage for the particular installation, or outside the configured storage, or storage assigned by the storage address translator for a particular computing element.

Specification: The branch address of EXECUTE is odd.

Branching exceptions occur only for EXECUTE. The instruction is suppressed on an execute exception. Therefore the condition code and data in registers and storage remain unchanged. The instruction is terminated on a fetch-protection violation.

Exceptions arising for the subject instruction of EXECUTE are the same as would have arisen had the subject instruction been in the normal instruction stream. However, the instruction address stored in the old PSW is the address of the instruction following EXECUTE. Similarly, the instruction-length code in the old PSW is the instruction-length code (2) of EXECUTE.

The address restrictions do not apply to the components from which an address is generated -- the content of the D₁ field and the content of the register specified by B₁.

JUNE 1, 1971

PROGRAMMING NOTE

An unavailable or odd branch address of a successful branch is detected during the execution of the next instruction and not as part of the branch.

TABLE 7-II COMPLETE TABLE OF CONDITION CODE SETTINGS

	0	1	2	3
<u>Fixed-Point Arithmetic</u>				
ADD (halfword/fullword)	zero	< zero	> zero	overflow
ADD LOGICAL	zero, no carry	not zero, no carry	zero, carry	not zero, carry
COMPARE (halfword/fullword)	equal	low	high	--
LOAD AND TEST	zero	< zero	> zero	--
LOAD COMPLEMENT	zero	< zero	> zero	overflow
LOAD NEGATIVE	zero	< zero	--	--
LOAD POSITIVE	zero	--	> zero	overflow
SHIFT LEFT DOUBLE	zero	< zero	> zero	overflow
SHIFT LEFT SINGLE	zero	< zero	> zero	overflow
SHIFT RIGHT DOUBLE	zero	< zero	> zero	--
SHIFT RIGHT SINGLE	zero	< zero	> zero	--
SUBTRACT (halfword/fullword)	zero	< zero	> zero	overflow
SUBTRACT LOGICAL	--	not zero, no carry	zero, carry	not zero, carry
<u>Decimal Arithmetic</u>				
ADD DECIMAL	zero	< zero	> zero	overflow
COMPARE DECIMAL	equal	low	high	--
SUBTRACT DECIMAL	zero	< zero	> zero	overflow
ZERO AND ADD	zero	< zero	> zero	overflow
<u>Floating-Point Arithmetic</u>				
ADD NORMALIZED (short/long)	zero	< zero	> zero	--
ADD UNNORMALIZED (short/long)	zero	< zero	> zero	--
COMPARE (short/long)	equal	low	high	--
LOAD AND TEST (short/long)	zero	< zero	> zero	--
LOAD COMPLEMENT (short/long)	zero	< zero	> zero	--
LOAD NEGATIVE (short/long)	zero	< zero	--	--
LOAD POSITIVE (short/long)	zero	--	> zero	--
SUBTRACT NORMALIZED (short/long)	zero	< zero	> zero	--
SUBTRACT UNNORMALIZED (short/long)	zero	< zero	> zero	--
<u>Status-Switching</u>				
WRITE DIRECT	oper	wait	redundant	timeout

JUNE 1, 1971

Condition-Code Setting (continued)				
	0	1	2	3
<u>Multiple Computing Element</u>				
TEST AND SET	zero	one	--	--
SET ADDRESS TRANSLATOR	accepted	rejected	not zero	zero
SET CONFIGURATION	zero	--	> zero	--
START I/O PROCESSOR	proceeding	invalid	--	not oper
<u>Display</u>				
LOAD CHAIN	zero	--	--	one
CONVERT AND SORT SYMBOLS	finished, no data stored	finished, data stored	boundary	inv form
REPACK SYMBOLS	finished	--	boundary	--
<u>Logical</u>				
AND	zero	not zero	--	--
COMPARE LOGICAL	equal	low	high	--
EDIT	zero	< zero	> zero	--
EDIT AND MARK	zero	< zero	> zero	--
EXCLUSIVE OR	zero	not zero	--	--
OR	zero	not zero	--	--
TEST UNDER MASK	zero	mixed	--	one
TRANSLATE AND TEST	zero	incomplete	complete	--
<u>Input/Output</u>				
HALT I/O	not working	halted	stopped	not oper
START I/O	available	CSW stored	busy	not oper
TEST CHANNEL	available	CSW ready	working	not oper
TEST I/O	available	CSW stored	working	not oper
SET PCI	available	CSW stored	set	not oper

JUNE 1, 1971

Condition-Code Setting (continued)

Legend

accepted	Storage element assignment mask accepted
available	Unit and channel available
busy	Unit or channel busy
carry	A carryout of the sign position occurs
complete	Last result byte nonzero
CSW ready	Channel status word ready for test or interruption
CSW stored	Channel status word stored
equal	Operands compare equal
>zero	Result is greater than zero
halted	Data transmission stopped. Unit in halt-reset mode
high	First operand compares high
incomplete	Nonzero result byte; not last
<zero	Result is less than zero
low	First operand compares low
mixed	Selected bits are both zero and one
no carry	No carry occurs
not working	Unit or channel not working
not zero	Result is not all zero
one	Selected bits are one
overflow	Result overflows
rejected	Storage element assignment mask rejected because of configuration error
set	Program-controlled interruption flag set
stopped	Data transmission stopped
working	Unit or channel working
zero	Result or selected bits are zero
invalid	Invalid PSW loaded into IOCE-processor
not oper	Unit or channel (or IOCE-processor) not operational
proceeding	Successful loading of PSW in IOCE-processor
redundant	Redundant operation
timeout	IOCE operation not completed; timeout occurred
wait	IOCE-processor start/stop operation completed (to/from Wait state)
oper	IOCE-processor start/stop operation or logout completed
finished	Processing complete
boundary	Page boundary encountered in the sort bin
inv form	Invalid beacon format

NOTE: The condition code may also be changed by LOAD PSW, SET PROGRAM MASK and DIAGNOSE, and by an interruption.

JUNE 1, 1971

CHAPTER 8: STATUS SWITCHING AND SPECIAL 9020D/E INSTRUCTIONS

A set of operations is provided to switch the status of the CE, of storage, and of communication between elements of the system.

The over-all CE status is determined by several program-state alternatives, each of which can be changed independently to its opposite and most of which are indicated by a bit in the program status word (PSW). The CE status is further defined by the instruction address, the condition code, the instruction-length code, the storage-protection key, and the interruption code. These all occupy fields in the PSW.

Protection of main storage and display storage is achieved by matching a key in storage with a protection key in the PSW or in a channel. The protection status of storage may be changed by introducing new storage keys, using SET STORAGE KEY. The storage keys may be inspected by using INSERT STORAGE KEY.

8.1 PROGRAM STATES

The four types of program-state alternatives, which determine the over-all CE status, are named Problem/Supervisor, Wait/Running, Masked/Interruptable, and Stopped/Operating. These states differ in the way they affect the CE functions and in the way their status is indicated and switched. The masked states have several alternatives; all other states have only one alternative.

All program states are independent of each other in their function, indication, and status-switching. Status-switching does not affect the contents of the arithmetic registers or the execution of I/O operations, but may affect the timer operation.

8.1.1 PROBLEM STATE

The choice between Supervisor and Problem state determines whether or not the full set of instructions is valid. The names of these states reflect their normal use.

In the Problem state all I/O, protection, and direct-control instructions are invalid, as well as LOAD PS BASE, STORE PS BASE, SET ADDRESS TRANSLATOR, SET CONFIGURATION, LOAD PSW, SET SYSTEM MASK, START I/O PROCESSOR, and DIAGNOSE. These are called privileged instructions. A privileged instruction encountered in the Problem state constitutes a privileged-operation exception and causes a program interruption. All instructions are valid in the supervisor state.

The CE is in the Supervisor state when bit 15 of the PSW is zero. When bit 15 is one, the CE is in the Problem state. The Supervisor state is not indicated on the operator sections of the CE control panel.

The CE is switched between Problem and Supervisor state by changing bit 15 of the PSW. This bit can be changed only by introducing a new PSW. Thus status-switching may be performed by LOAD PSW, using a new PSW with the desired value for bit 15. Since LOAD PSW is a privileged instruction, the CE must be in the Supervisor state before the switch. A new PSW is also introduced when the CE is interrupted. The SUPERVISOR CALL causes an interruption and thus may change the CE state. Similarly, initial program loading introduces a new PSW and with it a new CE state. The new PSW may introduce the Problem or Supervisor state regardless of the preceding state. No explicit operator control is provided for changing the Supervisor state.

JUNE 1, 1971

Timer updating is not affected by the choice between Supervisor and Problem state.

PROGRAMMING NOTE

To allow return from an interruption-handling routine to a preceding program by a LOAD PSW, the PSW for the interruption routine should specify the Supervisor state.

8.1.2 WAIT STATE

In the Wait state, no instructions are processed and storage is not addressed repeatedly for this purpose; whereas in the Running state, instruction fetching and execution proceed in the normal manner.

When bit 14 of the PSW is one, the CE is waiting. When bit 14 is zero, the CE is in the Running state. The Wait state is indicated on the operator control section of the CE control panel by the WAIT light.

The CE is switched between Wait and Running state by changing bit 14 of the PSW. This bit can be changed only by introducing an entire new PSW, as is the case with the problem-state bit. Thus, switching from the Running state may be achieved by the privileged instruction LOAD PSW, by an interruption such as for SUPERVISOR CALL, or by initial program loading. Switching from the Wait state may be achieved by an I/O or external interruption or, again, by initial program loading. The new PSW may introduce the Wait or Running state regardless of the preceding state. No explicit operator control is provided for changing the Wait state.

Timer updating is not affected by the choice between Running and Wait state.

PROGRAMMING NOTE

To leave the Wait state without manual intervention, the CE should remain interruptable for some active I/O or external interruption source.

8.1.3 MASKED STATES

The CE may be masked or interruptable for all system and machine-interruptions, and for some program interruptions. When the CE is interruptable for a class of interruptions, these interruptions are accepted. When the CE is masked the program interruptions are ignored, while the system and machine-check interruptions remain pending.

The system mask bits (PSW bits 0-7 and 16-19), the program mask bits (PSW bits 36-39), and the machine-check mask bit (PSW bit 13) indicate as a group the masked state of the CE. When a mask bit is one, the CE is interruptable for the corresponding interruptions. With the exception of certain CE element check conditions, when the mask bit is zero, these interruptions are always masked off. (See Chapter 9 for the masking of Abnormal Condition signals.) The system mask bits indicate the masked state of CE for the multiplexor and selector channels and the external signals. The program mask bits indicate the masked state for four of the nineteen types of program exceptions. The machine-check mask bit pertains to all machine checks. Program interruptions not maskable, as well as the supervisor-call interruption, are always taken. The masked states are not indicated on the operator sections of the CE control panel.

Most mask bits do not affect the execution of CE operations. The only exceptions are the significance mask bit, which determines the manner in which a floating-point operation is completed when a significance exception occurs, and the exponent-underflow mask bit, which determines the manner in

JUNE 1, 1971

| which a floating-point operation is completed when an exponent underflow
| occurs.

The interruptable state of the CE is switched by changing the mask bits in the PSW. The program mask may be changed separately by SET PROGRAM MASK, and the system mask may be changed separately by the privileged instruction SET SYSTEM MASK. The machine-check mask bit can be changed only by introducing an entire new PSW, as is the case with the Problem state and Wait state bits. Thus, a change in the entire Masked status may be achieved by the privileged instruction LOAD PSW, by an interruption such as for SUPERVISOR CALL, or by initial program loading. The new PSW may introduce a new Masked state, regardless of the preceding state. No explicit operator control is provided for changing the Masked state.

Timer updating is not affected by the choice between Masked or Interruptable states.

PROGRAMMING NOTE

To prevent an interruption-handling routine from being interrupted before necessary housekeeping steps are performed, the new PSW for that interruption should mask the CE for further interruptions of the kind that caused the interruption.

8.1.4 STOPPED STATE

When the CE is in the Stopped state, instructions and interruptions are not executed. In the Operating state, the CE executes instructions (if not waiting) and interruptions (if not masked off).

The stopped state is indicated on the operator control section of the CE control panel by the manual light. The stopped state is not identified by a bit in the PSW. A change in the stopped or operating state can be effected by manual intervention, external start or stop, or by machine malfunction. No interruptions can stop or start the CE. The CE is commanded to stop when the stop switch on the operator section of the CE control panel is pressed, when an external stop is received, when an address comparison indicates equality and the address-compare switches are set to STOP, or when the rate switch is set to INSTRUCTION STEP. In addition, the CE is placed in the stopped state after power is turned on. The CE is placed in the operating state when the start switch on the operator panel is pressed or an external start is received. The CE is also placed in the operating state when initial program loading is commenced.

The transition from Operating to Stopped state occurs at the end of instruction execution and prior to starting the next instruction execution. The transition takes place immediately when the CE is in the Wait state. All interruptions pending and not masked off are taken while the CE is still in the Operating state. They cause an old PSW to be stored and a new PSW to be fetched before entering the stopped state. If stop occurs due to Write Direct External Stop, no interrupts are taken, the CE is reset, and pending interrupts are eliminated. When the CE is in the Stopped state, interruptions are no longer taken, but remain pending.

The timer is not updated in the Stopped state.

PROGRAMMING NOTE

Except for timing considerations, execution of a program is not affected by stopping the CE.

| If, because of machine malfunction, the CE is unable to end an instruc-
| tion, then the STOP switch is not effective, and initial program loading
| reset or CE reset should be used.

JUNE 1, 1971

Input/output operations and IOCE-processor operations continue to completion while the CE is in the problem, wait, masked, or stopped state. However, no new I/O or IOCE-processor operations can be initiated while the CE is stopped, waiting or in the problem state. Also, the interruption caused by I/O completion or IOCE processor external interruption request remains pending when masked off, or when the CE is in the stopped state.

8.2 STORAGE PROTECTION

The storage-protection system consists of store protection and fetch protection for both main and display storage.

Storage protection is provided to protect the contents of certain areas of storage from destruction due to erroneous storing of information during the execution of a program. Fetch protection is provided to protect a program from making erroneous use of the contents of certain areas of storage as data or instructions. Protection is achieved by identifying blocks of storage with a storage key and comparing this key with a protection key supplied with the address of the location to be accessed. The detection of a mismatch is a protection exception and results in a program interruption or I/O interruption request.

8.2.1 AREA IDENTIFICATION

For protection purposes, both main storage and display storage are divided into blocks of 2,048 bytes, each block having an address which is a multiple of 2,048. A five-bit storage key is associated with each block. When data are stored in a storage block the four high-order bits of the storage key are compared with the protection key. When data are fetched, the fetch-protection bit (fifth, i.e., low-order bit) is inspected. When the fetch-protection bit is one, the four high-order bits of the storage key are compared with the protection key. The protection key of the current PSW is used as the comparand when a storage access is specified by an instruction. When a storage access is specified by a channel operation, the protection key supplied to the channel by the command address word is used as the comparand. The keys are said to match when they are equal or when either one is zero.

The storage key is not part of addressable storage. The key is changed by SET STORAGE KEY and is inspected by INSERT STORAGE KEY. The protection key in the PSW occupies bits 8-11 of that control word. The protection key of a channel is recorded in bits 0-3 of the channel status word, which is stored as a result of the channel operation.

8.2.2 PROTECTION ACTION

The store-protection system is always active. The fetch-protection system is active when the fetch-protection (low-order) bit of the storage key is set to one. The protection action is independent of the problem, supervisor, or masked state of the CE and of the type of instruction or I/O command being executed.

When an instruction causes a protection mismatch, execution of the instruction is suppressed or terminated, and program execution is altered by a program interruption. The content of the protected storage location always remains unchanged on a store-protection violation, and is never loaded into an addressable register or moved to another storage location on a fetch-protection violation.

In general, the violation of a protected location causes the instruction specifying this location to be suppressed, i.e., to be omitted entirely. In operations using multiple words or variable-length fields, part of the operation may already have been completed when the protected area is

referenced. In these operations the instruction cannot be suppressed and, hence, is terminated.

Protection mismatch due to an I/O operation causes data transmission to be terminated in such a manner that the content of the protected main storage location remains unchanged on a store-protection violation, and is not recorded on an output medium on a fetch-protection violation. The mismatch is indicated in the channel status word stored as a result of the operation.

Protection mismatch due to an IOCE-processor operation operates in the same manner as a protection mismatch in a CE access to main storage.

8.2.3 LOCATIONS PROTECTED

All main storage and display storage locations where information is stored in the course of an operation are subject to store protection. They are also subject to fetch protection provided the fetch protection bit is set to one in the storage key. A protection exception is only recognized when the access of a location is actually attempted, and not during an address manipulation operation prior to its use.

Locations whose addresses are generated by the CE for updating or interruption purposes, such as the timer, channel status word, or PSW addresses, are not protected. However, when the program specifies these locations, they are subject to protection. Protection mismatch due to an IOCE-processor operation operates in the same manner as a protection mismatch in a CE access to main storage.

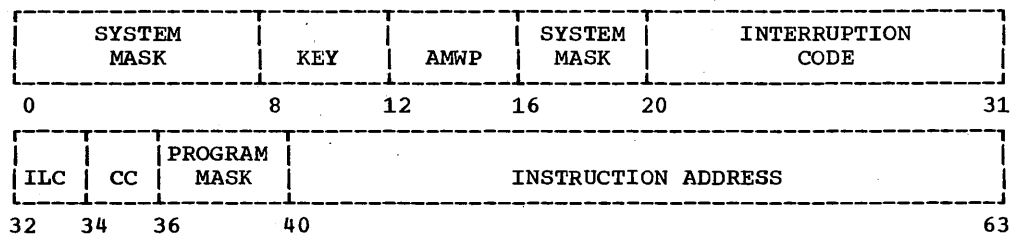
Storage protection (both store and fetch) is provided within display storage (DE) for computing element accesses only. The addressing of data fetched from display storage by display generators is determined by both the computing element and the display element and consequently storage protection is not provided on data transfers performed on the display generator interfaces.

8.3 PROGRAM STATUS WORD

The PSW contains all information not contained in storage or registers but which is required for proper program execution. By storing the PSW the program can preserve the detailed status of the CE for subsequent inspection. By loading a new PSW, or part of a PSW, the state of the CE may be changed.

In certain circumstances all of the PSW is stored or loaded; in others, only part of it. The entire PSW is stored and a new PSW is introduced when the CE is interrupted. The rightmost 32 bits are stored in BRANCH AND LINK. The LOAD PSW introduces a new PSW; SET PROGRAM MASK introduces a new condition code and program-mask field in the PSW; SET SYSTEM MASK introduces a new system-mask field.

The PSW has the following format:



JUNE 1, 1971

The following is a summary of the purposes of the PSW fields:

System Mask: Bits 0-7 and 16-19 of the PSW are associated with I/O channels and external signals as specified in the following table. When a mask bit is one, the source can interrupt the CE. When a mask bit is zero, the corresponding source cannot interrupt the CE and interruptions remain pending.

TABLE 8-I SYSTEM MASK ASSIGNMENT

SYSTEM MASK BIT	INTERRUPTION SOURCE
0	Multiplexor channel A
1	Selector channel 1A
2	Selector channel 2A
3	Selector channel 3A
4	Multiplexor channel B
5	Selector channel 1B
6	Selector channel 2B
7	Timer
7	Interrupt switch
7	External signal
7	PIR
7	DAR
16	Selector channel 3B
17	Multiplexor channel C
18	Selector channel 1C
19	Selector channel 2C

Protection Key: Bits 8-11 of the PSW form the CE protection key. The key is matched with the four high-order bits of storage key whenever data are to be stored, or whenever data are to be fetched and the fetch-protection bit in the storage key is one.

USASCII-8 (A): When bit 12 of the PSW is one, the codes preferred for the USASCII-8 code are generated for decimal results. When PSW bit 12 is zero, the codes preferred for the extended binary-coded-decimal interchange code are generated.

Machine-Check Mask (M): When PSW bit 13 is one, the machine-check interruption, element check (ELC) out-signal, and diagnostic log out occur upon malfunction detection. When bit 13 of the PSW is zero, the CE is masked for machine-check interruptions, and any associated diagnostic procedures do not take place. An ELC out-signal is, however, issued as before. The interruption remains pending.

Wait State (W): When bit 14 of the PSW is one, the CE is in the Wait state. When PSW bit 14 is zero, the CE is in the Running state.

Problem State (P): When bit 15 of the PSW is one, the CE is in the Problem state. When PSW bit 15 is zero, the CE is in the Supervisor state.

Interruption Code: Bits 20-31 of the PSW identify the cause of an I/O, program, supervisor call, machine check, or external interruption. Use of the code for all five interruption types is shown in a table appearing in the "Interruption Action" section (Chapter 9).

Instruction Length Code (ILC): The code in PSW bits 32 and 33 indicates the length, in halfwords, of the last-interpreted instruction, when a program or supervisor-call interruption occurs. The code is unpredictable for I/O,

external, or machine-check interruptions. Encoding of these bits is summarized in a table, "Instruction Length Recording", (Chapter 9).

Condition Code (CC): Bits 34 and 35 of the PSW are the two bits of the condition code. The condition codes for all instructions are summarized in a table, "Condition Code Setting" (Chapter 7).

Program Mask: Bits 36-39 of the PSW are the four program mask bits. Each bit is associated with a program exception, as specified in the following table. When the mask bit is one, the exception results in an interruption. When the mask bit is zero, no interruption occurs. The significance mask bit and the exponent-underflow mask bit also determine the manner in which floating-point operations are completed.

TABLE 8-II PROGRAM MASK ASSIGNMENT

PROGRAM MASK BIT	PROGRAM EXCEPTION
36	Fixed-point overflow
37	Decimal overflow
38	Exponent underflow
39	Significance

Instruction Address: Bits 40-63 of the PSW are the instruction address. This address specifies the leftmost eight-bit byte position of the next instruction.

Programming Note:

The new PSW is not checked for exceptions when the new PSW becomes current. These checks are made when the next instruction is executed.

8.4 MULTIPLE COMPUTING ELEMENT OPERATION

A number of provisions are required to avoid conflict and to allow coordination between the CEs in both the 9020D and 9020E Systems.

8.4.1 IDENTIFICATION OF A COMPUTING ELEMENT

Since a program in storage can be executed by any CE in the system, it is sometimes necessary for the program to identify the CE executing the program. This is particularly essential in connection with the task of reallocating the work load following a change in job priorities or a failure of some part of the system. The instruction LOAD IDENTITY provides a means for a program to load a value characteristic of the CE by which it is being executed.

8.4.2 SHARED MAIN STORAGE

Certain special storage addresses are automatically generated by a CE in connection with such operations as interruption handling and initial program loading. When several CEs share the same main storage, a means must be provided for the different CEs to use different storage locations for these purposes. This is done by grouping these addresses together into a preferential-storage area whose base address may be specified by the program. Each CE has a preferential-storage base address register that determines the location of this storage area. Instructions LOAD PREFERENTIAL-STORAGE BASE ADDRESS and STORE PREFERENTIAL-STORAGE BASE ADDRESS are provided for loading and storing this base address.

JUNE 1, 1971

PROGRAMMING NOTE:

Preferential storage areas are not allowed in display elements within a 9020E configuration.

When several CEs (or IOCE-processors) are cooperating in performing a single task, it is usually necessary to set up certain reference tables in storage to which each CE (or IOCE-processor) may refer to determine the status of the task. It is essential to provide some means of preventing other CEs (or IOCE-processors) from referring to such a table while one of the CEs (or IOCE-processors) is updating values in the table. This is done by associating a "lock" byte with each such table. The "lock" byte can be set by a CE (or IOCE-processor) to indicate that it is in the process of updating that table. Any other CEs (or IOCE-processors) wishing to refer to the table can then test this "lock" byte to determine whether or not it is free to access the table. The instruction TEST AND SET is provided to handle the necessary setting and testing of the "lock" byte.

Should a CE (or IOCE-processor) find a table busy, it is often desirable to program a short delay before trying again. The instruction DELAY provides this operation without making any main storage references.

8.4.3 CONFIGURATION CONTROL

The IBM 9020D and 9020E Systems, with their modularity and multiplicity of system elements, may be structured into many different configurations. Implementation of a system structuring vehicle is provided through programmed configuration control.

Within a 9020D System, configuration control is provided by the SET CONFIGURATION instruction which sets up control and data communication paths between major elements (i.e., CE, IOCE, SE, TCU, SCU, and PAM).

In a 9020E System, configuration control is provided in two forms. As in the 9020D, the SET CONFIGURATION instruction is provided to set up control and data communication paths between major elements (i.e., CE, IOCE, SE, DE, RCU and TCU). Additionally, the configuration console contains configuration control logic which functionally conditions the console for acceptance or transmission of DG, RKM and DAU configuration and status information.

The SET CONFIGURATION instruction establishes the system configuration by specifying to each configurable element the "state" it is to assume, the computing elements from which it may accept configuration changes, and the elements within the system with which it may exchange data and certain control signals.

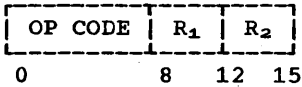
CE Direct Control provides two instructions (Section 8.5), READ DIRECT and WRITE DIRECT, which perform special functions via the external interruption lines into each computing element. The transfer of a single byte of data between two computing elements is provided by these instructions. WRITE DIRECT may also be used to cause the external start or external stop of another CE, to start, stop, or interrupt an IOCE-processor, or to cause the logout of another CE or of an IOCE.

8.4.4 INSTRUCTION FORMATS

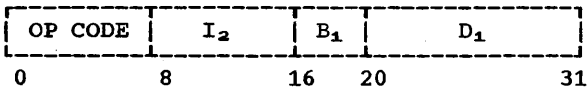
The instructions associated with multiple CE systems use the following instruction formats.

JUNE 1, 1971

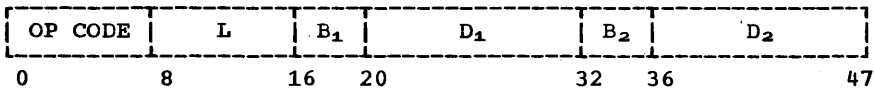
RR Format



SI Format



SS Format



In the RR format R₁ and R₂ specify general registers. In LOAD IDENTITY, R₂ is not used. In DELAY, R₁ and R₂ are treated as a single eight-bit count field.

In the SI format B₁ specifies a general register. The content of this register is added to D₁ to specify the location of the operand in storage. The I₂ field is not used.

A zero in the B₁ field indicates the absence of the corresponding address component.

8.4.5 MULTIPLE COMPUTING ELEMENT INSTRUCTIONS

The instructions associated with multiple CE systems, their mnemonics, formats, and operation codes follow. The table also indicates which exceptions may occur.

JUNE 1, 1971

TABLE 8-III MULTIPLE COMPUTING ELEMENT INSTRUCTIONS

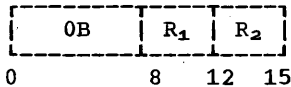
NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
DELAY	DLY	RR		0B	8.4.5.1
LOAD IDENTITY	LI	RR		0C	8.4.5.2
LOAD PS BASE ADDRESS	LPSB	SI	M P A S	A1	8.4.5.3
SET CONFIGURATION	SCON	RR	C M S	01	8.4.5.4
STORE PS BASE ADDRESS	SPSB	SI	M P A S	A0	8.4.5.5
TEST AND SET	TS	SI	C P A	93	8.4.5.6
SET ADDRESS TRANSLATOR	SATR	RR	C M S	0D	8.4.5.7
INSERT ADDRESS TRANSLATOR	IATR	RR		0E	8.4.5.8
MOVE WORD	MVW	SS	P A S	D8	8.4.5.9
START I/O PROCESSOR	SIOP	SI	C M S	9A	8.4.5.10

Legend

A Addressing interruption
 C Condition code is set
 M Privileged operation interruption
 P Protection interruption
 S Specification interruption

8.4.5.1 Delay

DLY R₁,R₂ [RR]



The operation of the CE is delayed for approximately 256N microseconds (usec) where N is the binary value of bits 8-15 of the instruction.

The R₁ and R₂ fields are treated as a single count field (N). A value of N equal to zero causes a negligible delay.

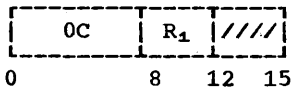
The instruction is terminated either by the count (N) reaching zero or by the occurrence of an unmasked I/O, external, or machine-check interruption condition. The timing of the delay is approximate, and may be increased by requests for service from the interval timer.

Condition Code: the code remains unchanged.

Program Interruptions: none.

8.4.5.2 Load Identity

LI R₁ [RR]



JUNE 1, 1971

The general register specified by R_1 is loaded with a value identifying the CE of a multiple CE system that executed this instruction.

The R_2 field is ignored.

The identifying value is a positive fixed-point integer, from 0 to 3.

Condition Code: the code remains unchanged.

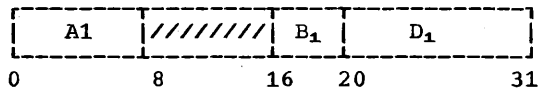
Program Interruptions: none

PROGRAMMING NOTE

Computing elements CE-1, CE-2, CE-3, and CE-4 take the identifying values of 0, 1, 2, and 3 respectively.

8.4.5.3 Load Preferential-Storage Base Address

LPSB $D_1(B_1)$ [SI]



The preferential-storage base address is loaded from the operand location.

The I_2 field is ignored. Bit positions 8-19 of the operand become the twelve high-order bits of the preferential-storage base address. The content of bit positions 0-7 and 20-31 of the operand is ignored.

The operand address must have its two low-order bits zero to specify a word. Bit positions 8-19 of the operand, interpreted as the twelve high-order bits of a storage address, must specify a location within a storage element (SE) provided in the storage address translator (ATR) which is configured to communicate with the CE executing this instruction. (A display element address is not acceptable). If any of these conditions are lacking, a specification exception causes a program interruption and the instruction is terminated.

Condition Code: the code remains unchanged.

Program Interruptions:
Privileged operation
Addressing
Specification
Protection

PROGRAMMING NOTE 1

Execution of LOAD PREFERENTIAL-STORAGE BASE ADDRESS causes bits 8-19 of the word fetched to be placed in the "logical" preferential-storage base address register (PSBAR) where they become the twelve high-order bits of the "logical" preferential-storage base address (PSBA). Bit positions 9-12 of the "logical" PSBAR are translated and select a slot of the storage address translator (ATR). This causes the value appearing in that position of the ATR to be placed in a four-bit "physical" PSBAR.

PROGRAMMING NOTE 2

JUNE 1, 1971

Whenever a storage address is detected whose twelve high-order bits are zeros, the access is directed to the storage element designated by "physical" PSBAR. The address of the location within this main storage element is developed by "ORing" the content of the bit positions 13-19 of the "logical" PSBAR into the corresponding positions in the address. The twelve low-order bits of the address are unaltered. Although the main storage element designated by "physical" PSBAR is obtained from the ATR, subsequent references are independent of the ATR. They are directed to the storage element designated by "physical" PSBAR and do not pass through the ATR (See Figure 8-1)

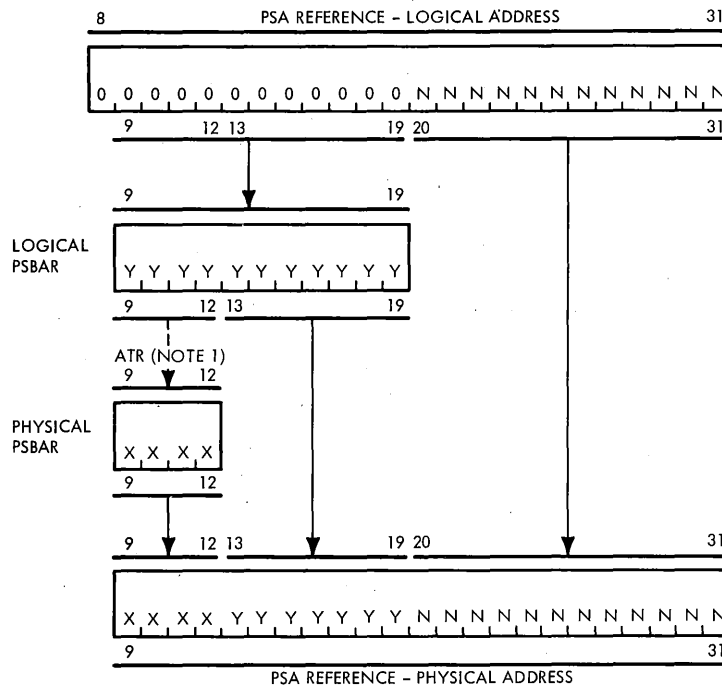


Figure 8-1 Load Preferential Storage Base Address Register

Note 1: Bits 9-12 of the logical PSBAR are initially translated to select an ATR slot from which the physical PSBAR will be loaded during the execution of the LPSB instruction.

PROGRAMMING NOTE 3

It should be noted that, when the "logical" PSBA is other than zero, references to the first 4096-byte block of "logical" main storage are always relocated. Consequently, this storage block is not available to the programmer for CE instructions or data. However, IOCE data transfers to and from main storage (other than those occurring during IPL, logout, or interruption handling) are never relocated and may reference any part of main storage.

Considerable care must be exercised in changing the preferential-storage base address, since this results in an immediate change of the PSWs that will be used in the case of an interruption. It also results in a new value of the interval timer.

Once set, the content of "physical" PSBAR is retained until changed by another LOAD PREFERENTIAL-STORAGE BASE ADDRESS, an IPL operation, a

JUNE 1, 1971

| PSW restart, an external start, or by a condition changing the PSBA to
| access the alternate PSA (Programming Note 4). Hardware is provided to
| automatically update "physical" PSBAR, following a change in "logical"
| PSBAR.

PROGRAMMING NOTE 4

Since the storage element (SE) containing the PSA may malfunction, or otherwise become unavailable, an alternate PSA in the corresponding segment of the "next higher" configured "logical" SE should be provided by the programmer.

Hardware is provided to automatically develop the alternate preferential-storage base address whenever the inhibit logout-stop bit is set off, and the CE recognizes that an access request to the primary PSA is not acknowledged or acknowledged but not granted while a PSA access request is pending. Hardware will also automatically develop the alternate PSA whenever either an address, data or fetch check is encountered or the SE is stopped during an access to the SE which contains the PSA. The alternate PSBA is developed by adding successive increments of 524,288 to the content of "logical" PSBAR (translating each successive logical address via the address translation register), until the first configured main storage element (SE) is encountered. | The original contents of "logical" PSBAR bits 9-12 is lost during this | process. When the maximum main storage is reached (last 524,288 byte | main storage address available to the system) the process wraps around | to zero and continues.

NOTE: In a 9020E System, the alternate PSBA process ignores display elements (i.e., the display elements are ignored with respect to alternate PSA search).

After the alternate PSBA is established, "physical" PSBAR is loaded from ATR with the SE identifier. Processing is then resumed using the alternate PSA.

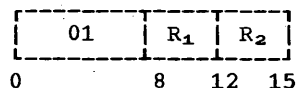
Once the alternate PSBA is established, hardware inhibits the CE from attempting to reach a second alternate PSA, should a check condition re-occur. Execution of the LOAD PREFERENTIAL-STORAGE BASE ADDRESS, the receipt of an external start signal (Chapter 8, Direct Control), the initiation of an IPL operation, or a PSW restart re-enables the | capability. Unless re-enabled, a check condition which would ordinarily | step PSBAR, detected in the alternate PSA, causes the CE to issue a | level element check (ELC) and to check stop.

| If the CE is configured to only one SE and the ILOS bit is on, the | primary and alternate PSBAs are the same. The first check condition | encountered is noted in hardware, and the CE attempts to continue | processing in the primary PSA. If a second check is encountered, a | level ELC is issued and the CE check stops. If the CE is configured to | no SE, or to an SE outside the available storage for the particular | installation, or outside the main storage assigned by the ATR, or is | configured to only one SE with the ILOS bit off and encounters a check | condition which would ordinarily step PSBAR, a level ELC is issued and | the CE check stops.

JUNE 1, 1971

8.4.5.4 Set Configuration

SCON R_1, R_2 [RR]



The content of the two general registers specified by the even register (R_1) of an even-odd pair, is interpreted as a configuration mask doubleword. The content of the general register specified by R_2 is interpreted as a Selection Mask Word (Figure 8-2).

NOTE It will be noted in figure 8-2 that the odd register of the even-odd pair specified by R_1 is ignored in a 9020D System.

The issuing CE must have its state bits set to zero or three and have its own scon bit on in its own configuration control register. The configuration mask must have a non-zero scon-field which references at least one CE available to the particular installation. Whenever an IOCE is selected, no more than one CE communications bit may be on in the configuration mask. Failure to satisfy any of the above conditions constitutes a specification exception causing a program interruption.

The elements selected by the selection mask will accept the applicable configuration mask information into their configuration control registers only if the scon bit for the issuing CE is already on in these registers.

Selected CEs, SEs, IOCEs, PAMs, SCUs and TCUs accepting the configuration mask information into their configuration control registers respond to the selector if correct parity is received with the mask. Those elements selected which detected improper mask parity will not respond to the SCON instruction.

In addition, within a 9020E System, the DES will respond to the selector only if (1) no more than four Display Generator communication bits are set, (2) no Display Generator has more than one bit set, (3) no more than one Display Generator is configured to any one of the data registers (i.e., A, B, C, D) and (4) correct parity is received in the CCR. Also within a 9020E System, the Reconfiguration Control Unit will respond to the selector only if no more than one IOCE communication bit is set and the CCR contains proper parity.

The results of all responses are placed in R_1 , using the selection mask format. Bit positions assigned to elements failing to respond are set to 1, all other bit positions are set to 0. The response result is tested for zero and the condition code set.

The content of the register specified by R_2 is unchanged.

The SCON instruction is valid in the supervisor state only.

Condition Code:

- 0 All selected elements have accepted the configuration mask information with correct parity
- 2 One or more of the selected elements failed to accept the configuration mask information, or accepted the configuration mask information with incorrect parity

JUNE 1, 1971

Program Interruptions:

Privileged operation
Specification

PROGRAMMING NOTE 1

The Configuration Control Register (CCR) in each Computing Element (CE), Input/Output Control Element (IOCE), Storage Element (SE), Display Element (DE), Reconfiguration Control Unit (RCU), Tape Control Unit (TCU), Storage Control Unit (SCU), and Peripheral Adapter Module (PAM) consists of a state field, scon field, and communications field. Additionally, the IOCEs and CEs have an inhibit logout-stop field, and the DEs, an inhibit display element stop field.

The content of an element's and/or unit's CCR may be altered under program control via the SET CONFIGURATION (SCON) instruction. Execution of SET CONFIGURATION allows a CE to select an element or unit (or several elements and/or units) and place configuration mask bits in its configuration control register.

Those elements and/or units selected by the CE executing SCON are designated in the Selection Mask (See Figure 8-2). The data to be inserted into the selected configuration control registers is presented in the Configuration Mask (See Figure 8-2).

The content of each CCR consists of similar state and scon fields. The length of the communications field depends upon a particular element's and/or unit's relationship within the system. The inhibit logout-stop bit is provided in the CCR of a CE and IOCE only, while the inhibit display element stop bit is provided in the CCR of a DE only.

An element is always in one of four states as determined by the state field in its CCR.

S0	S1	State
1	1	Three
1	0	Two
0	1	One
0	0	Zero

The scon field, in the configuration mask and in an element's CCR, contains a bit corresponding to each CE in the IBM 9020 D/E System.

Scon Bit	Computing Element
1	CE-1
2	CE-2
3	CE-3
4	CE-4 (spare)

A scon bit set in an element's CCR designates a CE from which it will accept configuration mask bits during the execution of SET CONFIGURATION. A CE must also have its own scon bit set in its CCR as a condition for execution of SET CONFIGURATION.

JUNE 1, 1971

The communications field of the configuration mask within a 9020D contains a bit corresponding to each element capable of inter-element exchange of data and control signals. Bits 8-17 of word 1 designate storage elements SE-1 through SE-10. Bits 20-23 designate computing elements CE-1 through CE-4, and bits 29-31 refer to input/output control elements IOCE-1 through IOCE-3, respectively. Word 2 of the configuration mask is not applicable in a 9020D configuration.

The communications field of the configuration mask within a 9020E System extends into two words as shown in Figure 8-2. Bits 8-12 of word 1 designate storage elements SE-1 through SE-5 while bits 13-17 designate display elements DE-1 through DE-5. Bits 20-23 designate computing elements CE-1 through CE-4, and bits 29-31 refer to input/output control elements IOCE-1 through IOCE-3, respectively. Bits 0-15 of word 2 designate display generator interface buffer combinations within the display elements.

A communications bit set in an element's CCR designates an element from which it will accept data and certain kinds of control information. With the exception of IOCE-TCU, IOCE-SCU and IOCE-PAM communication, two-way communication between an element pair in a 9020D configuration requires that the appropriate communications bits be set in each element's CCR. For example, to enable CE-2 to access SE-5, it is necessary that the communications bit for SE-5 be set in CE-2's CCR, and that the communication bit for CE-2 be set in SE-5's CCR. In a 9020E configuration, two-way communication between an element pair requires that the appropriate communications bits be set in each element's CCR with the exception of IOCE-TCU, IOCE-RCU, and DE-DG.

The inhibit logout-stop bit set on in the CCR of a CE, or an IOCE, prevents that element from issuing the logout-stop signal to any SE from which a storage check, or to any SE or DE from which an address or data check is received. (See Chapter 12 for IBM 9020 D/E System monitoring and checking features.) It also prevents hardware from incrementing the preferential-storage base address to reference the alternate PSA (Section 8.4.5.3, Programming Note 4).

The inhibit display element stop (IDES) bit set on in the CCR of a DE, prevents that element from entering a logout stop state upon detection of a DE malfunction encountered during a DG storage access. Should this bit be set off, the DE will automatically enter a logout stop state upon detecting this type of check condition.

JUNE 1, 1971

PROGRAMMING NOTE 2

This table summarizes functional state definitions for a computing element.

TABLE 8-IV SUMMARY OF FUNCTIONAL STATE DEFINITIONS FOR A COMPUTING ELEMENT

CE STATE	EXECUTE SCON(1)	ISSUE SCON(4)	ACCEPT SCON(6)	CE ELC(8) MASKABLE	CE PANEL CONTROLS			(10) SYSTEM CONSOLE CONTROLS
					(9) OPERATOR	MAINTENANCE	CCR, POWER	
THREE	YES	YES	YES	YES	NO	NO	NO	YES
TWO	NO(2)	N/A(5)	YES	NO	NO	NO	NO	YES
ONE	NO(2)	N/A(5)	YES	NO	YES	NO	NO	YES
ZERO SWITCH OFF	YES	YES	YES	NO	YES	YES	NO	YES
ZERO SWITCH ON	YES(3)	NO(3)	NO(7)	YES	YES	YES	YES(3)	NO

Notes:

- 1) "Execute SCON" means to execute the SET CONFIGURATION instruction as described in this manual.
- 2) Execution of SET CONFIGURATION is suppressed and a specification interruption is taken.
- 3) Manual control of CCR settings is provided. SET CONFIGURATION may be executed when the state bits are manually set to Three or Zero. The selection of other system elements and the propagation of the configuration mask are suppressed. All instruction exceptions are recognized and program interruptions are taken. If SET CONFIGURATION is attempted while the state bits are set to One or Two, note (2) applies.
- 4) "Issue SCON" refers to the ability to select external elements and present the configuration mask information when executing SET CONFIGURATION.
- 5) This column is not applicable (N/A) since note (2) applies.
- 6) "Accept SCON" means to accept the configuration mask bits into the CCR. The scon bit for the issuing CE must be already set in the receiving CCR. If either the scon bit is off or the mask is accepted with improper parity, condition code 2 is set in the issuing CE. (Two important exceptions to this rule should be noted: Whenever any receiving element has detected improper parity on the content of its CCR, or has its state bits set to Three, Two, or One and has all scon bits set off in its CCR, configuration mask bits are unconditionally accepted from any issuing CE.)

JUNE 1, 1971

- 7) All external configuration selection and mask signals are ignored. A self-issued configuration mask is accepted provided its own scon bit is already set in the CCR.
- 8) Incoming CE ELCs (element checks) are masked off when the scon bit for the issuing CE is off in the CCR of the receiving CE. Where "yes" is entered the CE ELC is also maskable by the DAR Mask and external interruption mask, PSW bit 7. "No" indicates that any additional masking provided is ignored. The receiving CE has its state bits set to Three, and an abnormal condition interruption is taken. (See Chapter 9.)
- 9) A limited number of operator's controls on the CE control panel are activated. (See Chapter 11.)
- 10) A number of the CE control panel facilities are enabled on the system console and configuration console. (See Chapter 11.)

PROGRAMMING NOTE 3

An IOCE with the state bits in its CCR set either to Three or to Two has all manual and power on/off controls disabled with the exception of its Element Master Power Off (MPO) switch. All other functions are enabled. When its state bits are set to One, certain manual controls are enabled to allow it to perform maintenance operations without a controlling CE. When its state bits are set to Zero, all manual controls are enabled; if the Test switch is on, power on/off controls are also enabled.

An SE or DE with the state bits in its CCR set either to Three, Two or One has all manual and power on-off controls disabled with the exception of its Element MPO switch. All other functions are enabled. When its state bits are set to Zero, its Test switch is enabled. With the Test switch set on, the SE or DE has all manual controls enabled, including its power on/off switch; no external signals are accepted, and none are issued.

State definitions for the TCU, RCU, SCU, and PAM are described in System Reference Library (SRL) documents for these units.

PROGRAMMING NOTE 4

An element configured to accept configuration mask information may have its CCR changed at any time. This happens independently of the operation the element is currently performing. The effect of the change depends upon the current operation, upon other elements involved in the current operation, and upon the fields changed in the CCR.

In general, changes in the state and scon fields do not affect the receiving element's current operation. However, changes in the communications field may affect the receiving element's current operation, as well as that of other elements involved in the operation. Since changing the communications field causes inter-element control and data paths to be broken and/or established, an operation currently using such paths is effected.

If a CE-IOCE, or a CE-CE interconnection is changed during the execution of the direct control instructions, results are unpredictable. Similarly, if a CE-IOCE inter-connection is changed during an I/O interruption, operation results are also unpredictable.

Changes requested in SE-CE, DE-CE or SE-IOCE inter-connections are always deferred until the completion of the current storage cycle.

JUNE 1, 1971

| Configuration changes involving the TCU, RCU, SCU or the PAM are described in SRL documents available for these units.

PROGRAMMING NOTE 5

Figure 8-3 is an example of a 9020D three-subsystem configuration. The lines connecting the system elements represent control and communications paths that are established by use of the SET CONFIGURATION instruction.

JUNE 1, 1971

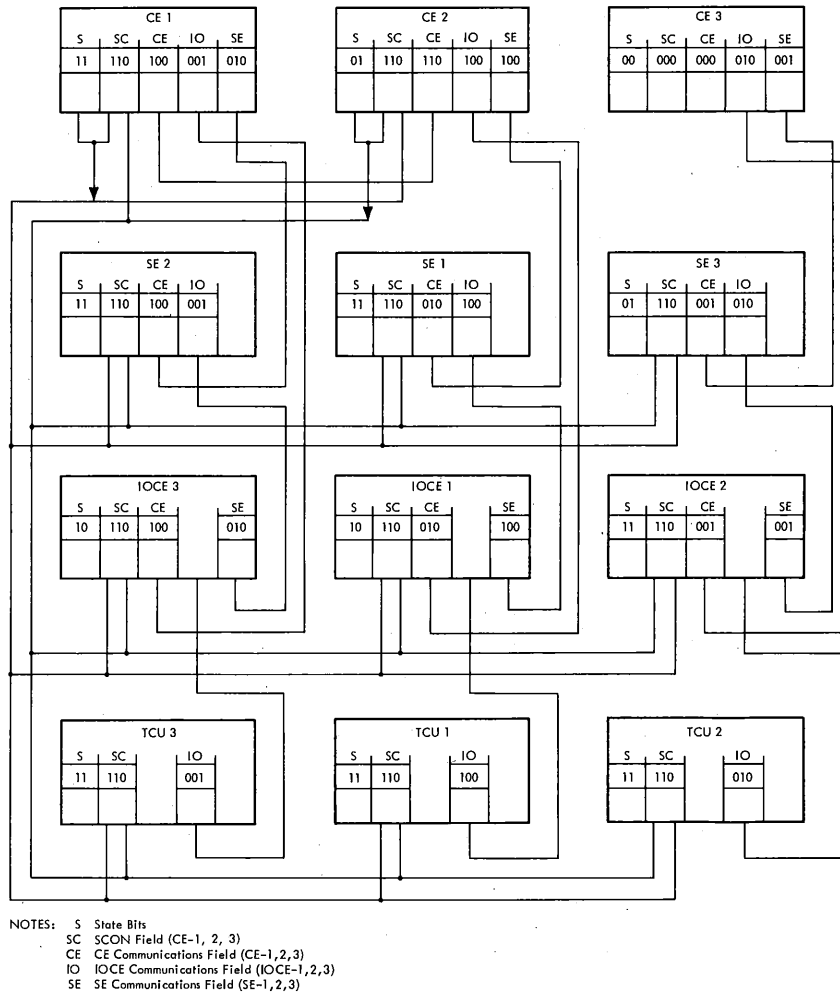


Figure 8-3 A 9020D Three-Subsystem Configuration

The first subsystem is composed of CE-1, IOCE-3, SE-2, and TCU-3. CE-1 is in state Three and therefore may recall units from the other subsystems by virtue of its ability to "execute" and "issue SCON." The second subsystem is composed of CE-2, IOCE-1, SE-1, and TCU-1. CE-2 is in state One and therefore has its operator controls activated, but may not execute SET CONFIGURATION. The second subsystem cannot add units except by CE-1 altering its configuration. Since maintenance functions are not required, IOCE-1 can be in either state Two or Three (state Two is shown). Similarly, TCU-1 and SE-1 may be in either state Three, Two, or One (state Three is shown). Note that all elements in the two subsystems will "accept SCON" from either CE-1 or CE-2, although CE-2 cannot "issue SCON."

This serves a very useful function should CE-1 fail. An element check (ELC) signal from CE-1 into CE-2 causes CE-2 to have the state bits in its CCR set to state Three, and to take an external interruption. The interruption routine can then cause CE-2 to take over CE-1's former tasks. All elements in the second subsystem are therefore available to assume any required task for the first subsystem, but may perform other tasks until required to do so.

JUNE 1, 1971

The third subsystem is composed of CE-3, IOCE-2, SE-3, and TCU-2. This subsystem contains three elements which are recallable by the first and second subsystems, and one element which is non-recallable. This is an arrangement which would normally be used to perform corrective maintenance on CE-3, for example. Since CE-3 is in state Zero with the scon field in its CCR set to zeros, it will not accept a new configuration mask. SE-3 is recallable; however, care should be exercised if this is done as some of its locations may contain incorrect parity due to use by faulty CE-3.

In general, for an element to be fully recallable it should have the scon bits in its CCR set for all CEs assigned to the primary subsystem, and for all CEs which are recallable to the primary subsystem. The scon bits for all other CEs should be set off.

PROGRAMMING NOTE 6

Assuming that an off-line CE is not in state Zero (with test switch on), several approaches for reconfiguring it into an active system suggest themselves. A "graceful" mode and a "forcing" mode are described here.

- 1) A "graceful" reconfiguration requires the off-line CE to be recallable (i.e., have the scon bit for the active CE set in its CCR), to have the communications bit for the active CE set in its CCR, and to have external interruptions unmasked. This mode also requires the active CE to have the communications bit for the off-line CE set in its CCR, and to have external interruptions unmasked.

The active CE executes WRITE DIRECT, specifying CE to CE data transfer to the off-line CE. The off-line CE takes an external interruption, determines its nature by analyzing the interruption code stored, and executes READ DIRECT, specifying the active CE, to retrieve the data byte on its direct-in line. The data byte is coded to alert the off-line CE that it is about to be recalled to the active system and allows it to terminate its current operation in an orderly manner. Execution of READ DIRECT causes the active CE to take an external interruption which may be notification that the off-line CE has received the message. Depending upon the complexity of the intercommunication, the active CE can allow the off-line CE a predetermined amount of time to perform these operations, or can wait on a subsequent WRITE DIRECT from the off-line CE to notify it that the housekeeping is complete.

Prior to changing the content of the off-line CE's CCR, the active CE must establish a PSA and an APSA for it, configure the on-line elements to communicate with it, and set up its configuration and selection masks.

The active CE now executes SET CONFIGURATION to the off-line CE, giving it the prepared CCR mask. This can be followed with WRITE DIRECT, specifying an external start (Section 8.5.1.7, Programming Note 2), causing it to start executing a program.

- 2) A "forcing" reconfiguration may be used when it is not possible to do a "graceful" recall because external interruptions are masked off in the off-line CE, or the communications bit for the active CE is not set in the CCR in the off-line CE, etc. The off-line CE must have the scon bit for the active CE set in its CCR as before, and the active CE must set up the PSA and APSA, and configure the other on-line elements to communicate with the CE being added to the system. The active CE can execute SET CONFIGURATION, followed with WRITE DIRECT, specify-

JANUARY 31, 1973

TNL: GN31-3005

ing an external start, causing the off-line CE to terminate its current operation and to start executing a desired program. No time is allowed for orderly termination of off-line operations.

PROGRAMMING NOTE 7

Reconfiguration of other off-line elements to an active system can be handled as follows (provided the proper scon bit is set in the off-line element's CCR):

- 1) An IOCE can be brought into the active system in two general ways. The first preserves current I/O operations; the second resets the IOCE.
 - a) The first method requires the off-line, controlling CE to be alerted that its IOCE is about to be recalled. It, therefore, stops issuing I/O instructions to the off-line IOCE and masks off I/O interruptions on all channels. The active CE, upon being notified that the IOCE is ready, masks off I/O interruptions on all channels and executes SET CONFIGURATION to the off-line IOCE to change its CE communications field from the off-line to the active CE. The IOCE's storage configuration must not be changed if an I/O operation is being performed. When reconfiguration is completed, the active CE may unmask I/O interruptions. It must be programmed to service outstanding I/O interruptions connected with unfinished off-line I/O operations.
 - b) The second method requires two SET CONFIGURATION instructions to be executed in sequence to the off-line IOCE. The first sets the IOCE State to 0 or 1 and the CE communications field in the IOCE's CCR to zeros. This places the IOCE in diagnostic mode. The second sets on the communications bit for the active CE in the IOCE's CCR. This sequence (i.e., going from an all-zero CE communications field when in State 0 or 1 to a field specifying a CE), causes the IOCE and all its channels to be reset. On-line I/O operation can now be initiated.
- 2) As indicated in PROGRAMMING NOTE 4, an SE or DE accepts a new configuration mask upon the completion of the current storage cycle.
- 3) Reconfiguration involving the TCU, RCU, SCU or the PAM is described in SRL documents available for these units.

PROGRAMMING NOTE 8

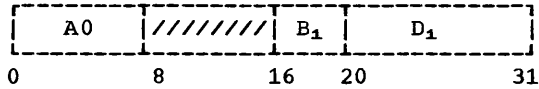
There is no priority determination of reconfiguration requests in elements or units and these elements or units will attempt to honor all requests simultaneously. The result of multiple reconfiguration requests is unpredictable and should be avoided through proper programming.

PROGRAMMING NOTE 9

Whenever an RCU is configured from State 0,1 or 2 into State 3, it is necessary to issue a Write Configuration to all outboard devices (RKM, DG, etc.) after configuration of the RCU. If this is not done, it is possible that the outboard devices may contain incorrect configuration information.

8.4.5.5 Store Preferential-Storage Base Address

SPSB D₁(B₁) [SI]



The contents of the "logical" and "physical" preferential-storage base address registers are stored at the operand location.

The content of the "logical" preferential base address register is stored in bit positions 8-19 at the operand location. The content of the "physical" preferential-storage base address register is stored in bit positions 28-31. Bit positions 0-7, and 20-27 are set to zeroes.

The operand address must have its two low-order bits zeros to specify a word; otherwise a specification exception causes a program interruption.

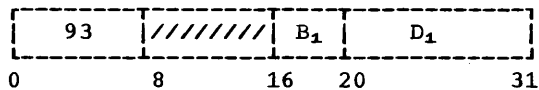
Condition Code: The code remains unchanged.

Program interruptions:

- Privileged operation
- Addressing
- Protection
- Specification

8.4.5.6 Test And Set

TS D₁(B₁) [SI]



The leftmost bit of the byte (bit 0) at the operand address sets the condition code and the entire byte is set to ones.

The byte in storage is fetched for the bit test and then set to all ones. No other access to this half (ODD or EVEN) of the SE is permitted between the moment of fetching and the moment of storing all ones.

Condition Code:

- 0 Leftmost bit of byte specified was zero.
- 1 Leftmost bit of byte specified was one.

Program Interruptions:

- Addressing
- Protection

JUNE 1, 1971

PROGRAMMING NOTE

This byte is interpreted as a "lock" as follows:

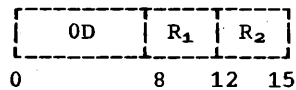
0xxx xxxx	Unlocked	No CE or IOCE has previously gained access to the "lock" byte. The accessing CE or IOCE "locks out" all other CEs or IOCEs. The condition code is set to zero.
1xxx xxxx	Locked	Another CE or IOCE has gained access to the "lock" byte. The accessing CE or IOCE is "locked out." The condition code is set to one.

x denotes a bit position which may be either zero or one.

"Unlocking" of the byte consists of changing it from the 1xxx xxxx state to the 0xxx xxxx state.

8.4.5.7 Set Address Translator

SATR R₁,R₂ [RR]



The content of the register specified by R₁ and of byte 0 of the register specified by R₂ are interpreted as a 40-bit storage element assignment mask. The content of byte 3 of the register specified by R₂ is interpreted as a selection mask. The initial content of bytes 1 and 2 is ignored.

If the selection mask contains all bits zero, the instruction is terminated and condition code three is set.

A CE or IOCE designated in the selection mask will accept the assignment mask into its address translator only if the scon bit for the issuing CE is already set on in the selected element's configuration control register; otherwise the instruction is terminated and condition code one is set.

Selected elements accepting the assignment mask into their address translators respond to the issuing CE. The results of all responses are placed in byte 2 of register R₂, using the selection mask format, if all selected elements did not respond. Bit positions assigned to elements failing to respond are set to 1. All other bit positions are set to 0. If all selected elements did respond, the results are not placed in byte 2. The content of register R₁ and of bytes 0, 1, and 3 of register R₂ are unchanged.

The CE executing the instruction must have its own scon bit set on in its own configuration control register, and have its state bits set to Zero or Three. The storage element assignment mask must reference a valid storage module available to a particular installation. In a 9020E System, it must not assign an SE module to a position reserved for a DE module in the particular installation, nor a DE module to other than positions 6, 7, 8, 9 or 10. Failure to satisfy any of these conditions causes a specification exception, and the instruction is terminated.

The instruction is valid in the supervisor state only.

Condition Code:

- 0 All selected elements have accepted the storage element assignment mask.
- 1 A selected element failed to accept the storage element assignment mask because the scon bit for the issuing CE was set off in its configuration control register, or either element did not have the other's communication bit set.
- 2 One or more of the selected elements failed to respond because of a parity error detected.
- 3 The selection mask is all zeros.

Program Interruptions

Privileged Operation Specification

PROGRAMMING NOTE 1

A ten-position address translation register (ATR) is contained in each CE and IOCE to provide dynamic storage address translation on a module basis. Associated with each position in the ATR is a block of addresses set on a 524,288 byte boundary. Each position has four bits, representing a hexadecimal digit which designates a module number (SE and/or DE 1 through A).

Whenever a "logical" address is developed by either a CE or IOCE, bits 9-12 of this address are decoded by hardware into one of the ten ATR positions (1 through 10). The hexadecimal digit contained in the corresponding ATR position is subsequently substituted into the "logical" address (bits 9-12) to provide a resultant "physical" address. This physical address determines which module contains the requested logical address (See Figure 8-4).

Since the storage module configurations on the 9020D and 9020E Systems differ in appearance and types, the requirements and interpretation of address translation on the two systems also differ.

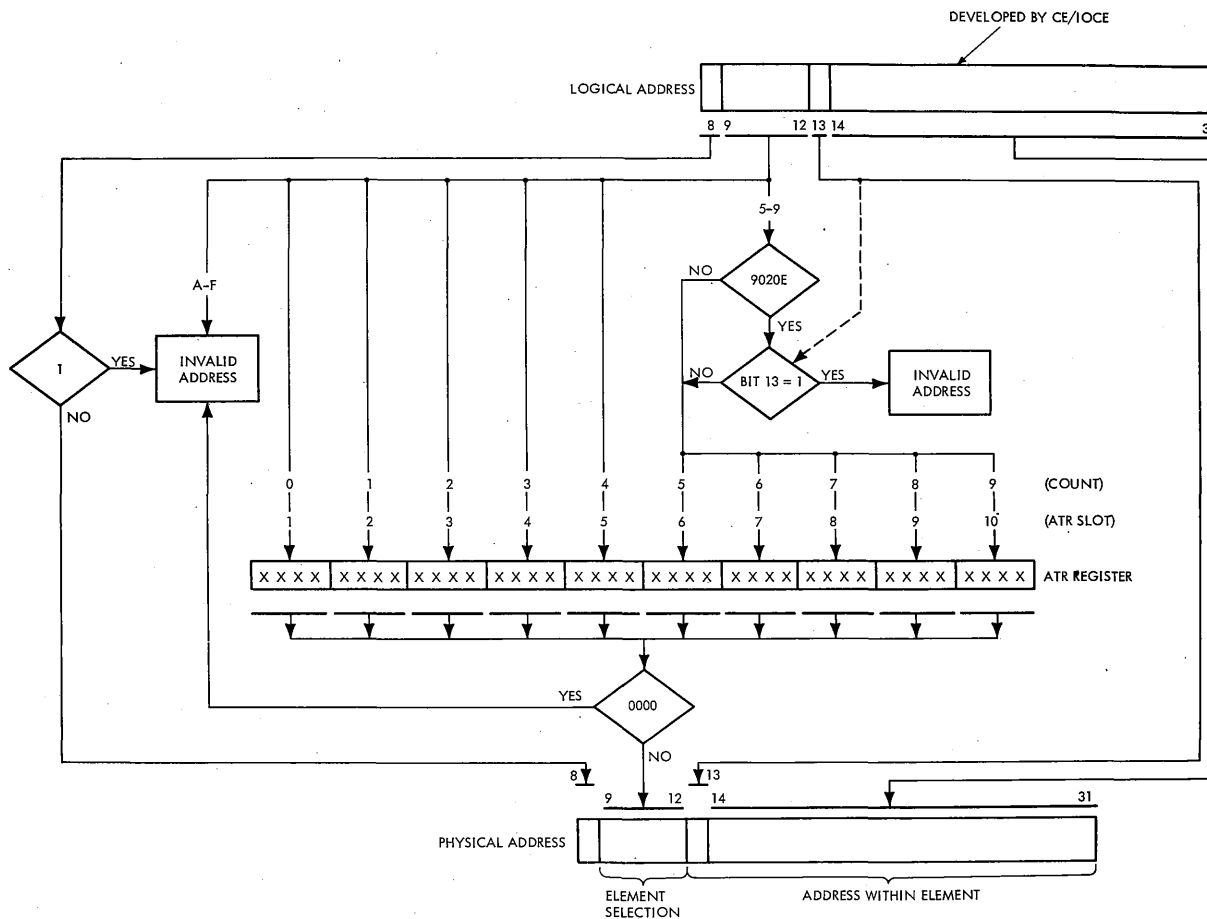
9020D

The IBM 9020D System uses storage elements (SE) in modules of 524,288 bytes. A particular system may have up to ten modules. The address range available to a particular system will contiguously extend from byte 0 through byte $\{(524,288) (N)\}-1$, where N equals the number of SEs existing in the system.

SE identifiers to be inserted into the address translation register must be valid identifiers for the particular system. That is, if a system has seven SEs, only identifiers 1-7 are valid. Identifiers B-F are always invalid. (See Table 8-VII)

Hardware does not prevent duplicate identifiers from being inserted. In addition, hardware does not prevent valid identifiers from being inserted into unused ATR positions (e.g., the tenth ATR position in a nine SE configuration). Zero identifiers are always accepted but will, if accessed, cause an invalid address condition.

JUNE 1, 1971



| Figure 8-4 9020D/E Address Translation

9020E

The IBM 9020E System uses storage elements (SE) in modules of 524,288 bytes and display elements (DE) in modules of 262,144 bytes. A particular system may have up to five SEs and up to five DEs. The SE address range available to a particular 9020E will contiguously extend from byte 0 through byte $\{(524,288) (N)\}-1$, where N equals the number of SEs existing in the system. The DE address range will start, regardless of the SE quantity, at byte 2,621,440. Each DE provides the low 262,144 byte address of subsequent 524,288 byte address ranges (i.e., set on 524,288 byte boundaries). This provides non-contiguous addressing as shown in Table 8-V.

Both SE and DE identifiers to be inserted into the address translation register must be valid identifiers for the particular system. For example, if the system has four SEs and three DEs, only identifiers 1-4 (SEs) and 6-8 (DEs) are valid. Identifiers B-F are always invalid.

In a 9020E System the first through fifth ATR positions are always associated with SEs, regardless of the SE quantity. Similarly, the sixth through tenth positions are always associated with DEs regardless of the DE quantity. Valid identifiers for SEs range from 1 through 5 (depending on the SE quantity) while valid identifiers for DEs range from 6 through A (depending on DE quantity).

Hardware does not prevent duplicate identifiers from being inserted. In addition, hardware does not prevent valid identifiers from being inserted into unused ATR positions. Zero identifiers are always accepted but will, if accessed, cause an invalid address condition. SE and DE identifiers are not allowed to be interchanged; that is, identifiers 1-5 (if valid) are accepted in ATR positions 1-5 only and identifiers 6-A (if valid) are accepted in ATR positions 6-10 only. An attempt to intermix SE and DE identifiers will cause a specification exception.

TABLE 8-V RANGE OF ADDRESSES

ADDRESS TRANSLATOR POSITION	9020D		9020E	
	FROM	TO	FROM	TO
1	000,000	524,287	000,000	524,287
2	524,288	1,048,575	524,288	1,048,575
3	1,048,576	1,572,863	1,048,576	1,572,863
4	1,572,864	2,097,151	1,572,864	2,097,151
5	2,097,152	2,621,439	2,097,152	2,621,439
6	2,621,440	3,145,727	2,621,440	2,883,583
7	3,145,728	3,670,015	3,145,728	3,407,871
8	3,670,016	4,194,303	3,670,016	3,932,159
9	4,194,304	4,718,591	4,194,304	4,456,447
10	4,718,592	5,242,879	4,718,592	4,980,735

JUNE 1, 1971

TABLE 8-VI ADDRESS TRANSLATION REGISTER SLOT SELECTION

LOGICAL ADDRESS BITS						ATR SLOT SELECTION	
8	9	10	11	12	13	9020D	9020E
0	0	0	0	0	X	1 (SE)	1 (SE)
0	0	0	0	1	X	2 (SE)	2 (SE)
0	0	0	1	0	X	3 (SE)	3 (SE)
0	0	0	1	1	X	4 (SE)	4 (SE)
0	0	1	0	0	X	5 (SE)	5 (SE)
0	0	1	0	1	0	6 (SE)	6 (DE)
0	0	1	0	1	1		INVALID ADDRESS
0	0	1	1	0	0	7 (SE)	7 (DE)
0	0	1	1	0	1		INVALID ADDRESS
0	0	1	1	1	0	8 (SE)	8 (DE)
0	0	1	1	1	1		INVALID ADDRESS
0	1	0	0	0	0	9 (SE)	9 (DE)
0	1	0	0	0	1		INVALID ADDRESS
0	1	0	0	1	0	10 (SE)	10 (DE)
0	1	0	0	1	1		INVALID ADDRESS
0	1	0	1	0	0	INVALID ADDRESS	INVALID ADDRESS
1	1	1	1	1	1		

X = Don't care

JUNE 1, 1971

TABLE 8-VII STORAGE/DISPLAY MODULE IDENTIFIERS

STORAGE/DISPLAY MODULE IDENTIFIER (Hexadecimal)	9020D	9020E
0	Unassigned Position	
1	SE-1	SE-1
2	SE-2	SE-2
3	SE-3	SE-3
4	SE-4	SE-4
5	SE-5	SE-5
6	SE-6	DE-1
7	SE-7	DE-2
8	SE-8	DE-3
9	SE-9	DE-4
A	SE-10	DE-5
B	Invalid identifiers	
C		
D		
E		
F		

PROGRAMMING NOTE 2

The formats of the contents of the registers specified by R_1 and R_2 when executing SET ADDRESS TRANSLATOR are:

Register specified by R_1 :

SE-n	SE-n	SE-n	SE-n	SE-n	SE-n or DE-n*	SE-n or DE-n*	SE-n or DE-n*
0	4	12	16	16	20	24	28 31

Register specified by R_2 :

SE-n or DE-n*	SE-n or DE-n*	IGNORED	RESPONSES	SELECTION MASK
0	4	8	16	24 31

* = 9020E System only.

Storage Element Assignment Mask -- Bits 0-31 of the register specified by R_1 , and bits 0-7 of the register specified by R_2 form the storage/display element assignment mask. Each 4-bit digit is interpreted as a storage/display module identifier when it lies in the range 1-A (hexadecimal). 0 denotes an unassigned position. B, C, D, E, and F are invalid identifiers.

RESPONSES -- Bits 16-23 of R_2 contain the response byte if any element(s) did not respond. The initial content of this byte is ignored so the programmer need not specify it when setting up R_2 . The format for responses is similar to that for the selection mask.

Selection Mask -- Bits 24-31 select the elements to which the storage assignment mask is presented. A bit position set to one causes the corresponding element to be selected. At least one bit must be set on

JUNE 1, 1971

to avoid terminating the instruction with condition code 3. Unassigned or spare bit positions, or those corresponding to elements unavailable to the installation must be set to zeros to avoid terminating the instruction with a condition code 1.

Element	Selection Mask Bit	Response Bit
CE-1	24	16
CE-2	25	17
CE-3	26	18
CE-4 (spare)	27	19
(Unassigned)	28	20
IOCE-1	29	21
IOCE-2	30	22
IOCE-3(spare)	31	23

PROGRAMMING NOTE 3

9020E Example: Consider a subsystem which is configured to use SE-1, SE-3, SE-4, DE-6, and DE-8.

When the content of an address translator (CE or IOCE) is set to

3410068000 (hexadecimal)

there are 2,097,152 bytes of SE/DE core available to the subsystem. SE-3, SE-4, and SE-1 are contiguously addressed by 0 - 1,572,863. An addressing gap extends from byte 1,572,864 through byte 2,621,439. Bytes 2,621,440 through 2,883,583 lie in DE-6 while bytes 3,145,728 through 3,407,871 lie in DE-8.

PROGRAMMING NOTE 4

When a CE is to receive a new storage assignment mask, the programmer may either place the CE in the wait state or present the mask asynchronously. A CE in the wait state accepts the mask into its ATR and then returns to the wait state. It may be restarted by an external start signal (Direct Control), or an external interrupt. A CE not in the wait state accepts the mask into its ATR at the completion of the current instruction and then returns to program execution. If the SE used is affected by the new ATR setting, results are unpredictable.

When an IOCE is to receive a new storage address assignment mask, the programmer should first place it in the wait state. An IOCE in the wait state accepts the storage address assignment mask into its ATR, and then returns to the wait state. An IOCE not in the wait state completes its current memory access, and then is forced into the wait state. Upon accepting the storage address assignment mask, it returns to the wait state. If an I/O operation or IOCE processor operation is in progress and the SE involved is affected by the new setting of ATR, results are unpredictable. An IOCE in the process of resetting does not accept a storage address assignment mask, and condition code 1 is returned.

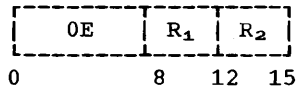
The occurrence of an unmasked machine check in the issuing CE terminates the SET ADDRESS TRANSLATOR instruction, causes a pulsed EIC to be issued, and diagnostic logout to occur (Chapter 9). When this happens, a CE selected to receive a new storage assignment mask returns to its previous operation; an IOCE returns to the wait state.

PROGRAMMING NOTE 5

When R_1 and R_2 specify the same register, it is likely, but not certain, that a specification exception will terminate the instruction.

8.4.5.8 Insert Address Translator

IATR R_1, R_2 [IRR]



The content of the storage address translator (ATR) is placed in a pair of registers specified by R_1 and R_2 .

The high-order 32 bits of the 40-bit address translator are placed in bit positions 0-31 of the register specified by R_1 . The low-order 8 bits are placed in bit positions 0-7 of the register specified by R_2 . Bit positions 8-31 of this register are set to zeros.

Conditions Code: The code remains unchanged.

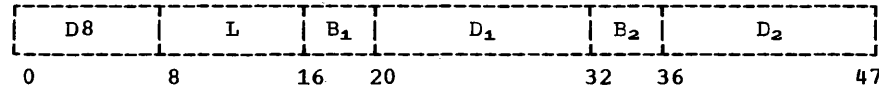
Program Interruptions: None.

PROGRAMMING NOTE

When R_1 and R_2 specify the same register, the 32 high-order bits of the ATR are placed in bit positions 0-31 of the register specified.

8.4.5.9 Move Word

MVW $D_1(L, B_1), D_2(B_2)$ [SS]



The second operand is placed in the first operand location. Both first and second operand locations are in main or display storage. The second operand address designates the left-most word of the operand field. The L field specifies the number of words to the right of this first word. The second operand field has the same length as the first operand field. Movement is left to right through each field.

The first and second operand addresses must specify full-word storage locations; otherwise a specification exception is detected and a program interruption is requested. The operation is suppressed.

Condition Code: the code remains unchanged.

Program Interruptions:

Protection (store or fetch violation)
Addressing
Specification

PROGRAMMING NOTE 1

JANUARY 31, 1973

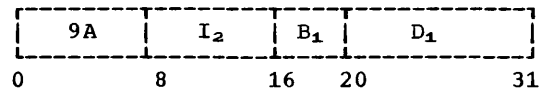
TNL: GN31-3005

The L field in the MOVE WORD instruction, unlike similar 9020 System instructions specifies the number of words rather than bytes involved in the move operation.

PROGRAMMING NOTE 2

During the execution of a MOVE WORD instruction, the original updated contents of the Instruction Counter are retained first in the Local Store Working Register, and later in the K register. To determine where the instruction address is valid at any particular time, examine the IC in LSWR indicator in conjunction with those indicators for either of the "Previous ROS Address Registers" as follows:

- 1) If the PROSAR contains "101" in bits 0-2 respectively, then IC is valid in the K register.
- 2) If the PROSAR contains some value other than "101" in bits 0-2, IC is valid in either IC or LSWR according to the condition of the "IC in LSWR" indicator as it is normally used for other instructions.

8.4.5.10 StartI/O ProcessorSIOP $D_1(B_1), I_2$ [SI]

The operand address and a protection key are made available to the designated IOCE-processor. The program status word (PSW), loaded by the IOCE-processor from the operand location, determines its subsequent action.

Bit positions 8-31 of the sum formed by the addition of the contents of register B₁ to the value of the D₁ field identify the storage location (in main or MACH storage) of a doubleword PSW. Bits 8-11 of the I₂ field specify the storage protection key for the PSW location on a main storage access. Bits 12-15 designate an IOCE-processor.

Bit	IOCE-Processor Selected
12	IOCE-1
13	IOCE-2
14	IOCE-3
15	0

Bit 15 must be a 0 or a specification exception will result. At least one, but not more than one of bits 12-14 must be set on at one time. The operand address must have three low-order bits zero to designate a doubleword. The protection key must be other than F (hexadecimal) since it is invalid in an IOCE-processor. If any of these conditions are lacking, a specification exception causes a program interruption and the instruction is suppressed.

Condition Code:

- 0 PSW loaded and IOCE-processor proceeding with its execution
- 1 Invalid PSW

2 --

3 Not operational

Program Interruptions:

Privileged operation
Specification

PROGRAMMING NOTE 1

Execution of START I/O PROCESSOR terminates in the CE, returning it to instruction fetching, when either the IOCE-processor returns condition code 0 or 1, or the CE stores condition code 3 on a read-only-storage timeout.

The meaning of the codes set follows:

Condition Code 0 is returned to signify the successful loading of a PSW in the IOCE-processor.

Condition Code 1 is returned when an invalid key, or PSW address causes the IOCE-processor to terminate its operation and go into wait state. Exceptions recognized are:

- a. Fetch-protection violation: The storage key for a main storage PSW location does not match the protection key supplied by the START I/O PROCESSOR instruction.
- b. Specification: The PSW address does not reference a double-word storage location. (Not normally possible since the CE checks this.)
- c. Invalid Key: The PSW loaded as a result of the START I/O PROCESSOR Instruction, contains a key of "F" (hexadecimal).
- d. Addressing: The PSW address designates a location outside the available main or MACH storage for the installed system, or outside the configured main storage, or main storage assigned by the storage address translator for a particular IOCE.

Condition Code 3 is set by the controlling CE when a non-operational IOCE-processor fails to return code 0 or 1, thus allowing a read-only-storage timeout to occur. Any of the following may cause the timeout:

- a. Configuration error: The communications bit for the issuing CE is not set in the configuration control register (CCR) of the selected IOCE-processor; or vice versa.
- b. The selected IOCE-processor is not available in the installation or is in power-down or check-stop state.
- c. The selected IOCE-processor is executing a move-type instruction, having its address translator (ATR) contents changed, or going through a reset operation associated with a SET CONFIGURATION instruction, an IPL, or PSW restart, and the duration of the operation exceeds the timeout interval.

PROGRAMMING NOTE 2

The selected IOCE-processor must have the communications bit for the issuing CE set in its configuration control register (CCR); otherwise the START I/O PROCESSOR signal is ignored. Condition code 3 will be stored. When the selected IOCE-processor is in the stopped or wait state, the operation begins at once. When it is in running state, the

JANUARY 31, 1973

TNL: GN31-3005

operation may take place at the end of the current instruction and before fetching the next instruction. However, when an instruction having long execution time is being processed, the current operation may not be completed in time to allow the START I/O PROCESSOR operation to be initiated.

Pending program, supervisor call and external (pushbutton, timer or CE Write Direct) interruptions are always eliminated in the IOCE-processor by a Start I/O Processor operation.

8.4.6 MULTIPLE-OPERATION INSTRUCTION EXCEPTIONS

Exceptional instructions or data cause a program interruption. When the interruption occurs the current PSW is stored as an old PSW and a new PSW is obtained. The interruption code in the old PSW identifies the cause. The following exceptions in the use of multiple CE instructions create program interruptions.

Privileged Operation: The LOAD PREFERENTIAL-STORAGE BASE ADDRESS, STORE PREFERENTIAL-STORAGE BASE ADDRESS, SET ADDRESS TRANSLATOR, START I/O PROCESSOR, or SET CONFIGURATION are encountered in the problem state.

Protection: The storage key of the location designated by a LOAD PREFERENTIAL-STORAGE BASE ADDRESS, STORE PREFERENTIAL-STORAGE BASE ADDRESS, MOVE WORD, or TEST AND SET does not match the protection key in the PSW.

Addressing: An address designates a location outside the available storage for the particular installation, or outside the configured storage, or storage assigned by the storage address translator for a particular CE.

Specification: The operand address of a LOAD PREFERENTIAL-STORAGE BASE ADDRESS, MOVE WORD, or STORE PREFERENTIAL-STORAGE BASE ADDRESS does not have the two low-order bits both zero.

Or bit positions 8-19 of the operand addressed by LOAD PREFERENTIAL-STORAGE BASE ADDRESS, interpreted as the 12 high-order bits of a storage address, do not specify a location within a storage element provided in the ATR, which is configured to communicate with the CE executing this instruction, or specifies a DE in a 9020E System.

Or execution of SET CONFIGURATION is attempted by a CE whose own scon bit is off in its configuration control register; or whose state bits are either set to One or Two. Or a configuration mask has all scon-field bits zeros, or does not reference at least one CE available to the system. Or a selection mask has an IOCE selection bit set when two or more CE communications bits are set in the configuration mask.

Or execution of SET ADDRESS TRANSLATOR is attempted by a CE whose scon bit is off in its configuration control register; or whose state bits are either set to One or Two. Or, a storage assignment mask references a particular storage module not available to a particular installation, or assigns an SE module to a portion reserved for a DE module in the particular installation, or assigns a DE module to other than positions 6-10 on a 9020E System.

Or START I/O PROCESSOR does not select an IOCE, or selects more than one IOCE, or the first operand address does not reference a doubleword storage location or a key of F (hexadecimal) was supplied to the IOCE-Processor, or bit 15 of the I₂ field was not zero.

JANUARY 31, 1973

In all of the above cases the instruction is suppressed, and data in storage and the registers remain unchanged. The instruction address is updated as for normal instruction sequencing, the instruction address and length code are stored as part of the old PSW, and a new PSW obtained.

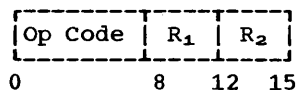
When an interruption is taken the instruction address stored as part of the old PSW has been updated by the number of halfwords indicated by the instruction-length code in the old PSW.

Operand addresses are tested only when used to address storage. The address restrictions do not apply to the components from which an address is generated, i.e., the content of the D_1 field, and the content of the register specified by B_1 .

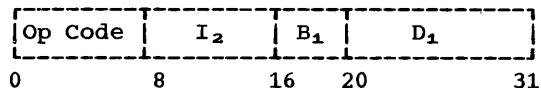
8.5 STATUS-SWITCHING INSTRUCTION FORMAT

Status-switching instructions use the following two formats:

RR Format



SI Format



In the RR format, the R_1 field specifies a general register, except for SUPERVISOR CALL. The R_2 field specifies a general register in SET STORAGE KEY and INSERT STORAGE KEY. The R_1 and R_2 fields in SUPERVISOR CALL contain an identification code. In SET PROGRAM MASK the R_2 field is ignored.

In the SI format, the eight-bit immediate field (I_2) of the instruction contains an identification code. The I_2 field is ignored in LOAD PSW, SET SYSTEM MASK. It is also ignored in DIAGNOSE when executed by a CE in state Three, Two, or One. The content of the general register specified by B_1 is added to the content of the D_1 field to form an address designating the location of an operand in storage. Only one operand location is required in status-switching operations.

A zero in the B_1 field indicates the absence of the corresponding address component.

8.5.1 STATUS-SWITCHING

The status-switching instructions and their mnemonics, formats, and operation codes follow. The table also indicates exceptions which cause a program interruption.

JUNE 1, 1971

TABLE 8-VIII STATUS-SWITCHING INSTRUCTIONS

NAME	MNEMONIC	TYPE	CC	EXCEPTIONS	CODE	SECTION
DIAGNOSE		SI		M,P,A,S	83	8.5.1.9
INSERT STORAGE KEY	ISK	RR		M A,S	09	8.5.1.6
LOAD PSW	LPSW	SI	L	M,P,A,S	82	8.5.1.1
READ DIRECT	RDD	SI		M,P,A,S	85	8.5.1.8
SET PROGRAM MASK	SPM	RR	L		04	8.5.1.2
SET STORAGE KEY	SSK	RR		M A,S	08	8.5.1.5
SET SYSTEM MASK	SSM	SI		M,P,A	80	8.5.1.3
SUPERVISOR CALL	SVC	RR			0A	8.5.1.4
WRITE DIRECT	WRD	SI	C	M,P,A,S	84	8.5.1.7

Legend

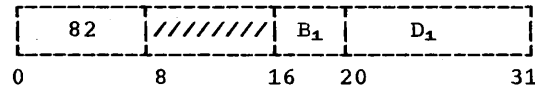
- A Addressing exception
- C Condition code set
- L New condition code loaded
- M Privileged-operation exception
- P Protection exception
- S Specification exception

PROGRAMMING NOTE

The program status is also switched by interruptions, initial program loading, and manual control.

8.5.1.1 Load PSW

LPSW $D_1(B_1)$ [SI]



The doubleword at the location designated by the operand address replaces the PSW.

The operand address must have its three low-order bits zero to designate a doubleword; otherwise, a specification exception results in a program interruption.

The doubleword which is loaded becomes the PSW for the next sequence of instructions. Bits 8-11 become the new protection key. Bits 40-63 of the doubleword become the new instruction address. The new instruction address is not checked for available storage or for an even byte address during a load PSW operation. These checks occur as part of the execution of the next instructions.

The interruption code in bit positions 20-31 of the new PSW is not retained as the PSW is loaded. When the PSW is subsequently stored

because of an interruption, these bit positions contain a new code. Similarly bits 32 and 33 of the PSW are not retained upon loading. They will contain the instruction-length code for the last-interpreted instruction when the PSW is stored during a branch-and-link operation or during a program or supervisor-call interruption.

Condition Code: the code is set according to bits 34 and 35 of the new PSW loaded.

Program Interruptions:

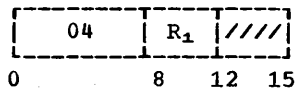
- Privileged operation
- Protection (fetch violation)
- Addressing
- Specification

PROGRAMMING NOTE

The CE enters the Problem state when LOAD PSW loads a doubleword with a one in bit position 15, and similarly enters the Wait state if bit position 14 is one. LOAD PSW and SUPERVISOR CALL are the only instructions available for entering the Problem state or the Wait state.

8.5.1.2 Set Program Mask

SPM R₁ [RR]



Bits 2-7 of the general register specified by the R₁ field replace the condition code and the program mask bits of the current PSW.

Bits 0, 1, and 8-31 of the register specified by the R₁ field are ignored. The contents of the register specified by the R₁ field remain unchanged.

The instruction permits setting of the condition code and the mask bits in either the Problem or Supervisor state.

Condition Code: the code is set according to bits 2 and 3 of the register specified by R₁.

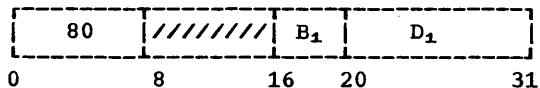
Program Interruptions: none

Programming Note:

Bits 2-7 of the general register may have been loaded from the PSW by BRANCH AND LINK.

8.5.1.3 Set System Mask

SSM D₁(B₁) [SI]



JUNE 1, 1971

The byte at the location designated by the operand address replaces the portion of the system mask in bits 0-7 of the current PSW. Bits 0-3 of the following byte replace the portion of the system mask in bits 16-19 of the current PSW. Bits 4-7 of this byte are ignored.

Condition Code: the code remains unchanged.

Program Interruptions:

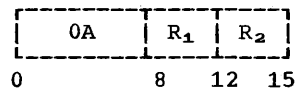
Privileged operation
Protection (fetch violation)
Addressing

PROGRAMMING NOTE

Consecutive Set System Mask instructions should not be issued. Allow at least one No-Op instruction between executions.

8.5.1.4 Supervisor Call

SVC R₁,R₂ [RR]



The instruction causes a supervisor-call interruption, with the R₁ and R₂ field of the instruction providing the interruption code.

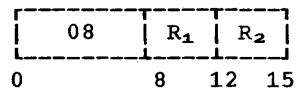
The contents of bit positions 8-15 of the instruction are placed in bit positions 24-31 of the old PSW which is stored in the course of the interruption. Bit positions 20-23 of the old PSW are made zero. The old PSW is stored at location 32, in the preferential-storage area, and a new PSW is obtained from location 96. The instruction is valid in both Problem and Supervisor state.

Condition Code: the code remains unchanged in the old PSW.

Program Interruptions: none

8.5.1.5 Set Storage Key

SSK R₁,R₂ [RR]



The key of the storage block addressed by the register designated by R₂ is set according to the key in the register designated by R₁.

The storage block of 2048 bytes, located on a multiple of the block length, is addressed by bits 8-20 of the register designated by the R₂ field. Bits 0-7 and 21-27 of this register are ignored. Bits 28-31 of the register must be zero; otherwise, a specification exception causes a program interruption.

The five-bit storage key is obtained from bits 24-28 of the register designated by the R₁ field. Bits 0-23 and 29-31 of this register are

JUNE 1, 1971

ignored. Bit 28 is the fetch-protection bit. When it is set to one, fetch protection is active.

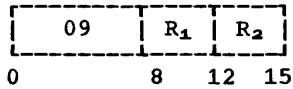
Condition Code: the code remains unchanged.

Program Interruptions:

Privileged operation
Addressing
Specification

8.5.1.6 Insert Storage Key

ISK R_1, R_2 [RR]



The key of the storage block addressed by the register designated by R_2 is inserted in the register designated by R_1 .

The storage block of 2048 bytes, located on a multiple of the block length, is addressed by bits 8-20 of the register designated by the R_2 field. Bits 0-7 and 21-27 of this register are ignored. Bits 28-31 of the register must be zero; otherwise a specification exception causes a program interruption. The five-bit storage key is inserted in bits 24-28 of the register specified by the R_1 field. Bits 0-23 of this register remain unchanged, and bits 29-31 are set to zero. Bit 28 is the fetch-protection bit. When it is set to one, fetch protection is active.

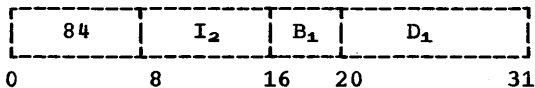
Condition Code: the code remains unchanged.

Program Interruptions:

Privileged operation
Addressing
Specification

8.5.1.7 Write Direct

WRD $D_1(B_1), I_2$ [SI]



The byte at the location designated by the operand address may be made available on a set of direct-out lines. An operation is specified and an element is selected by a unique direct-control signal-out line.

Bits 8-11 of the I_2 field specify an operation to be performed by an element designated by bits 12-15.

JUNE 1, 1971

TABLE 8-IX WRITE DIRECT - IMMEDIATE FIELD

Bit 9	Bit 10	Bit 11	Operation
0	0	0	CE-to-CE data transfer
0	0	1	CE external start
0	1	0	CE logout
0	1	1	IOCE logout
1	0	0	CE external stop
1	0	1	IOCE-processor start
1	1	0	IOCE-processor stop
1	1	1	IOCE-processor interrupt

Bit	Element Selected
12	CE-1, or IOCE-1
13	CE-2, or IOCE-2
14	CE-3, or IOCE-3
15	CE-4

NOTE: Bit 8 must always be zero.

Whenever bits 8-11 specify an operation to be performed by a computing element, bits 12-15 are interpreted to designate CE-1 through CE-4. Similarly, whenever an IOCE operation is specified, bits 12-14 are interpreted to designate IOCE-1 through IOCE-3.

If the instruction specifies a CE-to-CE data transfer operation, a unique direct-control signal-out is presented, and eight data bits and one parity bit of the byte at the location designated by the operand address are presented on a set of nine direct-out lines as static signals. If either an Addressing or a Protection Exception occur during operand fetching, the data on the direct-out lines will be zeros. The signals remain until the next WRITE DIRECT specifying a CE-to-CE data transfer is executed.

If the instruction specifies a CE external start, a CE logout, a CE external stop, an IOCE-processor start, an IOCE-processor stop, an IOCE-processor interrupt, or an IOCE logout, a unique direct-control signal-out is presented to the selected element. The location designated by the operand address must be valid or an addressing exception will result, but the signals on the direct-out lines remain unchanged.

When an IOCE operation is specified, bit 15 must be 0 or a specification exception will result. When a CE operation is specified, at least one, but not more than one of bits 12-15 must be set on at one time. Similarly, when an IOCE operation is specified, at least one, but not more than one of bits 12-14 must be set on at one time. Bit 8 must always be zero. If these conditions are lacking, a specification exception causes a program interruption and the instruction is suppressed.

Condition Code: The code remains unchanged when an operation specifies a CE. Condition Code 1 and 2 only apply to IOCE-processor Start-Stop.

When IOCE operation is specified:

- 0 IOCE operation completed.
- 1 IOCE operation completed, to/from Wait state.

JUNE 1, 1971

- 2 Redundant operation, no action taken.
- 3 IOCE operation not completed; timeout occurred.

Program Interruptions:

Privileged operation
Addressing
Specification
Protection

PROGRAMMING NOTE 1

A direct-control signal-out line of one CE connects into a signal-in line of another CE. Since a signal-in line is used to request an interruption from an external source (Chapter 9), interconnections of this kind allow one CE to interrupt another by executing WRITE DIRECT.

The direct-out lines of one CE are connected into direct-in lines of another CE. This interconnection provides a nine-bit data path for moving a byte (with parity) between computing elements.

Example -- If CE-2 executes WRITE DIRECT with bit 14 of the I_2 field set to one, CE-3 receives a CE-2 Write Direct interruption request. Provided CE-3 has the communications bit for CE-2 already on in its configuration control register, and has PSW mask bit 7 set to one, an interruption is taken with bit 23 set in the old PSW. CE-3 is presented a data byte on its direct-in lines. (The byte may be obtained by CE-3 executing READ DIRECT with CE-2 specified.)

Whenever the communications bit corresponding to the CE executing WRITE DIRECT is not already set on in the configuration control register of the receiving CE, the interruption request is ignored.

PROGRAMMING NOTE 2

WRITE DIRECT may be executed by one CE to reset and start another. A direct-control signal-out line from the first CE connects into the external-start line of the second. The "logical" preferential-storage base address in the selected CE is set to reference a preferential-storage area in the lowest block of 4096 bytes in its lowest configured "logical" storage element. This is the first configured storage element whose identifier appears in ATR, as ATR is inspected from left to right starting at position one. The current PSW is fetched from location 0 (relative to the preferential-storage base address), and processing is resumed.

External start does not request an interruption in the receiving CE. The receiving CE must, however, have the scon bit for the issuing CE set in its configuration control register; otherwise the external start signal is ignored. A CE cannot react to the receipt of any Write Direct operation immediately following the receipt of a Write Direct External Start. The amount of time required to react to the above is approximately equal to the time it takes for the CE to execute two No-Op instructions.

PROGRAMMING NOTE 3

WRITE DIRECT may be executed by a CE to cause another CE or IOCE to perform an automatic logout. A direct-control signal-out line from the first CE connects into the logout line of the selected element.

A CE logout does not request an external interruption in the receiving CE or IOCE. The receiving CE must have the scon bit for the requesting CE set in its configuration control register; otherwise the logout request is ignored. Similarly, the receiving IOCE must have the

JANUARY 31, 1973

TNL: GN31-3005

communications bit for the requesting CE set in its configuration control register.

PROGRAMMING NOTE 4

WRITE DIRECT may be executed by one CE to reset and stop another via a signal on a direct control signal-out line from the first CE connecting into the external stop line of the second. The transition from the operating state takes place at the end of instruction execution, prior to starting the next instruction sequence. The transition from check-stop state takes place immediately. All pending interruptions are eliminated. An element reset is performed and error status indicators (CR1, CR2), and DAR are reset to zero. The content of CCR and ATR are not changed, nor is the reset propagated to other system elements. When the CE is in the check-stop state its ELC signal is removed. The stopped state is indicated on the system console, configuration console and the CE control panel by the manual light.

The receiving CE must have the scon bit for the issuing CE set in its configuration control register; otherwise the external stop signal is ignored. A CE cannot react to the receipt of any Write Direct operation immediately following the receipt of a Write Direct External Stop. The amount of time required to react is approximately equal to the time it takes for the CE to execute two No-Op instructions.

If an operator wishes to manually restart a CE after it has been externally stopped (Write Direct Stop), he should use the PSW RESTART pushbutton. Use of the START pushbutton may cause unpredictable results.

PROGRAMMING NOTE 5

WRITE DIRECT may be executed by a CE to cause an IOCE-processor to start or stop via a signal on an appropriate direct control signal-out line connecting into the IOCE-processor start or stop line. The start signal causes a transition from the stopped state to the operating state to take place immediately. The IOCE-processor fetches an instruction from the location specified by the current value in its instruction address counter (IAR), if PSW bit 14 is off. If PSW bit 14 is on in the IOCE-processor, the processor starts to count its interval timer, and will take any interrupts if unmasked. When the receiving IOCE is stopped and the processor PSW bit 14 (Wait) is on, the receipt of a start signal causes condition code 1 to be set, indicating the IOCE-processor is in the wait state. Condition code 0 is set when the IOCE-processor is in the running state. When the receiving IOCE-processor is already in the operating state, the start signal is ignored, and condition code 2 is returned indicating a redundant operation.

The stop signal causes a transition from operating state to stopped state. If in the running state, the current instruction is finished, and all interruptions which are pending or become pending before the end of the instruction are preserved. Timer updating is suspended when the IOCE-processor is placed in stopped state. When the receiving IOCE-processor is in the wait state, the stop signal causes condition code 1 to be set, indicating the IOCE-Processor is in the wait state. Condition code 0 is set when the IOCE-Processor is in the running state. When the receiving IOCE-processor is already in stopped state, the stop signal causes no action. Condition code 2 is returned indicating a redundant operation.

The receiving IOCE-processor must have the communications bit for the requesting CE set in its configuration control register; otherwise the start (or stop) signal is ignored and condition code 3 is stored.

PROGRAMMING NOTE 6

WRITE DIRECT may be used by a CE to present an external interruption request to an IOCE-processor.

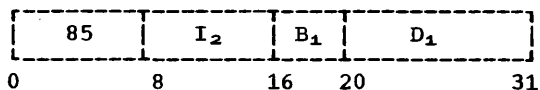
The receiving IOCE-processor must have the communications bit for the requesting CE set in its configuration control register; otherwise the interruption request is ignored and condition code 3 is stored.

When PSW bit 7 in the receiving IOCE-processor's PSW is one and the IOCE-processor is in the operating state, the interruption is taken; otherwise an interruption pending condition is created in the IOCE-processor. Either case causes condition code 0 to be set.

The interruption code stored is shown in Table H-1.

8.5.1.8 Read Direct

RDD $D_1(B_1), I_2$ [SI]



A data byte is accepted off a set of direct-in lines and placed in storage at the location designated by the operand address. A direct-control signal-out line is returned to the computing element which presented the data byte.

Eight data bits and one parity bit are accepted from the direct-in lines and stored.

Instruction bits 12-15 of the I_2 field are made available as four direct-control signal-out lines; one line for each computing element in the system.

Bit	Computing Element
12	CE-1
13	CE-2
14	CE-3
15	CE-4 (spare)

Bits 8-11 of the I_2 field are ignored. At least one, but not more than one of bits 12-15 must be set on at one time. If these conditions are lacking, a specification exception causes a program interruption, and the instruction is suppressed.

Condition Code: the code remains unchanged.

JUNE 1, 1971

Program Interruptions:

Privileged operation
Protection
Addressing
Specification

PROGRAMMING NOTE 1

A direct-control signal-out line of one CE connects into a signal-in line of another CE. Since a signal-in line is used to request an interruption from an external source (Chapter 9), interconnections of this kind allow one CE to interrupt another by executing READ DIRECT.

The direct-out lines of one CE are connected into the direct-in lines of another CE. This interconnection provides a nine-bit data path for moving a byte (with parity) between computing elements.

Example -- If CE-2 executes READ DIRECT with bit 14 of the I_2 field set to one, CE-3 receives a CE-2 Read Direct interruption request. Provided CE-3 has the communications bit for CE-2 already on in its configuration control register, and has PSW mask bit 7 set to one, an interruption is taken with bit 22 set in the old PSW. CE-2 accepts the data byte from its direct-in lines and places it in storage.

Whenever the communications bit corresponding to the CE executing READ DIRECT is not already set in the configuration control register of the receiving CE, the interruption request is ignored.

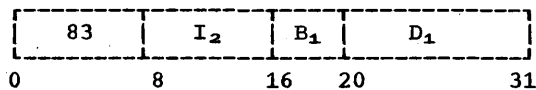
PROGRAMMING NOTE 2

Under certain circumstances the direct-in lines may appear busy during the execution of READ DIRECT. Normally, this condition is imposed on the lines during the interval when data are being changed by another CE executing WRITE DIRECT. The execution of READ DIRECT is delayed by a negligible amount, and the new data are then stored.

Excessive delay causes a CE machine-check interruption request to be presented. If permitted, the old PSW is stored with an interruption code of four, indicating a Read Direct timeout (Chapter 9).

8.5.1.9 Diagnose

[SI]



The Maintenance Control Word (MCW) designated by the first operand is loaded into the MCW register. The I_2 field is transferred to the SCAN Control Logic. A branch is then made to a location in Read Only Storage (ROS) as designated by the ROS Starting Address field (bits 8-19) of the MCW. Bit settings of the MCW and I_2 field cause certain predetermined

JUNE 1, 1971

functions to be performed.

Bit	Function
8	Disable Interleaving
9	Disable Interleaving and Reverse Storage Address
10	Diagnose FLT
11	Spare
12	Spare
13	Spare
14	Spare
15	Spare

The MCW must be located in main storage on a doubleword boundary. MCW bits 8-19 must specify an operational kernel ROS address for a computing element in state three, two, or one; otherwise a specification causes a program interruption and the instruction is suppressed.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	2	2	2	2	2	2	2	2	3	3	
R	L	R	R	R	L																	D											
T	A	S	M	S	O																	T											
P	C	A	P	A	C	0																											
	M	F	R																														
	P	S	P																														

3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	5	6	6	6	6		
2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	

- RTP - Reverse Tag Parity
- LACMP - Logout CE per Address Compare
- RSASFSP - Reverse Serial Address Full Sum Parity
- RMP - Reverse Mark Parity
- RSARP - Reverse Storage Address Parity
 - 00 - no parity reversal
 - 01 - reverse low-order parity bit (24-31)
 - 10 - reverse next-higher parity bit (16-23)
 - 11 - reverse high-order parity bit (8-15)
- LOC - Log On Count
- DT - Disable Timer
- RNOP - Reverse "Normal Op" to Storage
- FDGR - Force DG Request
- DSE1 - Diagnostic SE 1
- RERP - Reverse External Register Parity
- RC - Reset Checks

Figure 8-5 9020D/E Maintenance Control Word (MCW)

JUNE 1, 1971

An operational kernel may cause data to be automatically stored in one or more contiguous word or doubleword storage locations following the MCW location. An attempt to store data in a location that is either not within the available storage for a particular installation, or not within the configured storage, or storage allowed by the storage address translator for the particular computing element constitutes an addressing exception which causes a program interruption, and terminates the instruction. The storage key of a location stored must match the protection key in the PSW; otherwise a protection exception causes a program interruption and the instruction is terminated.

DIAGNOSE is completed by taking the next sequential instruction or by entering a Machine Check Interruption by obtaining a new PSW from location 112 of the preferential-storage area.

Condition Code: the code is unpredictable.

Program Interruptions:

Privileged operation
Protection
Specification
Addressing

PROGRAMMING NOTE 1

Whenever the DIAGNOSE instruction is executed by a CE in state three, two, or one, the bit settings of MCW 0-7, 20-31, 36-49, 51-63 and the I₂ field of the DIAGNOSE instruction are ignored and have no effect.

Programming details peculiar to the use of DIAGNOSE by a CE in state zero are described in maintenance publications.

PROGRAMMING NOTE 2

Whenever the DIAGNOSE instruction is executed by an IOCE in state three, two, or one the operation is suppressed. An operation exception is indicated and a Program Interruption is taken.

Programming details peculiar to the use of the DIAGNOSE instruction by an IOCE in state zero are described in maintenance publications.

JUNE 1, 1971

TABLE 8-X 9020D/E CE DIAGNOSE KERNELS (OPERATIONAL - ANY STATE)

KERNEL NAME	ROS STARTING ADDRESS	OTHER MCW BIT SETTINGS	STORAGE WORDS REQUIRED FOLLOWING MCW																																																																								
1. LOGOUT STORAGE	FD0	32-35 (SE or DE SELECT)	12																																																																								
		<table border="1"> <thead> <tr> <th colspan="4">MCW BITS</th> <th colspan="2">SELECTED</th> </tr> <tr> <th>32</th> <th>33</th> <th>34</th> <th>35</th> <th>9020D</th> <th>9020E</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>SE1</td><td>SE1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>SE2</td><td>SE2</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>SE3</td><td>SE3</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>SE4</td><td>SE4</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>SE5</td><td>SE5</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>SE6</td><td>DE1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>SE7</td><td>DE2</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>SE8</td><td>DE3</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>SE9</td><td>DE4</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>SE10</td><td>DE5</td></tr> </tbody> </table>	MCW BITS				SELECTED		32	33	34	35	9020D	9020E	0	0	0	1	SE1	SE1	0	0	1	0	SE2	SE2	0	0	1	1	SE3	SE3	0	1	0	0	SE4	SE4	0	1	0	1	SE5	SE5	0	1	1	0	SE6	DE1	0	1	1	1	SE7	DE2	1	0	0	0	SE8	DE3	1	0	0	1	SE9	DE4	1	0	1	0	SE10	DE5	
MCW BITS				SELECTED																																																																							
32	33	34	35	9020D	9020E																																																																						
0	0	0	1	SE1	SE1																																																																						
0	0	1	0	SE2	SE2																																																																						
0	0	1	1	SE3	SE3																																																																						
0	1	0	0	SE4	SE4																																																																						
0	1	0	1	SE5	SE5																																																																						
0	1	1	0	SE6	DE1																																																																						
0	1	1	1	SE7	DE2																																																																						
1	0	0	0	SE8	DE3																																																																						
1	0	0	1	SE9	DE4																																																																						
1	0	1	0	SE10	DE5																																																																						
2. RESET CHECKS	FD1	50 (RESET CHECKS)	0																																																																								
3. LOGOUT LOCAL STORE	FD2	NONE	48																																																																								
4. SET DAR MASK	FD3	NONE	1																																																																								
5. STORE DAR	FD4	NONE	1																																																																								
6. DEFINE STORAGE	FD5	NONE	1																																																																								
7. STORE PROCESSOR INTERRUPT REGISTER	FD6	NONE	1																																																																								
8. LOGOUT REGISTERS	FD7	NONE	16																																																																								

PROGRAMMING NOTE 3

This note defines the operation performed by each of the above diagnostic kernels.

Logout Storage Kernel

Execution of this kernel causes 6 doublewords of logout data from the selected SE or DE to be stored in contiguous storage locations starting at the doubleword following the MCW. (Refer to Chapter 12 for SE and DE logout formats). The logout data may not be directed to storage locations within the element being logged out.

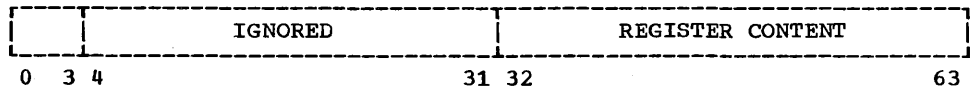
Reset Checks Kernel

JUNE 1, 1971

Execution of this kernel causes the check registers 1 and 2 in the CE that is executing the DIAGNOSE instruction to be reset. (See Chapter 12 for description of the check registers)

Logout Local Store Registers Kernel

Execution of this kernel causes the 24 words in the Local Store Registers and their associated parity bits to be stored in contiguous storage starting at the doubleword following the MCW. The content of each register will be placed in a doubleword with the contents of the register in bits 32-63 and the parity bits for the register in bits 0-3 of the doubleword.



PARITY BITS

Floating Point Register 7 will be the first register stored, then in descending order to Floating Point Register 0, then General Purpose Register 15 to General Purpose Register 0.

Set DAR Mask Kernel

Execution of this kernel causes the contents of the full word following the MCW to become the mask for the Diagnose Accessible Register (DAR).

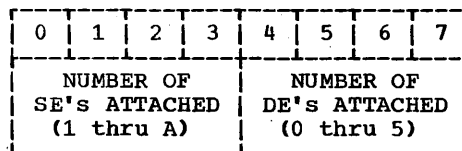
Store DAR Kernel

Execution of this kernel causes the contents of the Diagnose Accessible Register (DAR) to be stored in the full word following the MCW and the DAR to be reset.

Define Storage Kernel

Execution of this kernel will cause a byte of data, defining the number of SE's and DE's physically connected to the system to be stored. This byte will be stored in byte 3 of the full word following the MCW. Bytes 0, 1, and 2 will remain unchanged.

Storage Definition Byte



Store Processor Interrupt Kernel

Execution of this kernel will cause the contents of the Processor Interrupt Register to be stored in byte 3 (bits 28-30) of the word following the MCW, and the PIR to be reset.

Logout Register Kernel

Execution of this kernel will cause the content of the FAA Registers to be stored in 8 doublewords of contiguous storage starting at the doubleword following the MCW. Each register will take a doubleword with the contents of the register in bits 32-63 and the parity bits from the

JUNE 1, 1971

register in bits 0-3 of the doubleword. PIR will be stored in byte 3 of doubleword 1. The contents of the PIR and DAR are reset.

The order of logout is as follows:

<u>DOUBLE WORD</u>	<u>REGISTER</u>
1	PIR
2	CCR
3	DAR MASK
4	EXTERNAL
5	CHECK REG 2 & STOR CHK ADDR REG
6	ATR 1
7	ATR 2 & CE SELECT SATR
8	DAR

8.5.2 STATUS-SWITCHING EXCEPTIONS

Exceptional instructions or data cause a program interruption. When the interruption occurs the current PSW is stored as an old PSW, and a new PSW is obtained. The interruption code inserted in the old PSW identifies the cause of the interruption. The following exceptional conditions cause a program interruption in status-switching operations.

Privileged Operation: INSERT STORAGE KEY, WRITE DIRECT, READ DIRECT, LOAD PSW, SET STORAGE KEY, SET SYSTEM MASK, or DIAGNOSE is encountered while in the Problem state.

Protection: The storage key of the location designated by READ DIRECT, WRITE DIRECT, LOAD PSW, SET SYSTEM MASK, or of the location to be stored by DIAGNOSE does not match the protection key in the PSW.

Addressing: An address designates a location outside the available storage for the particular installation, or outside the configured storage, or storage allowed by the storage address translator for the particular CE.

Specification: The operand address of a LOAD PSW does not have all three low-order bits zero.

Or the block address specified by SET STORAGE KEY or INSERT STORAGE KEY does not have the four low-order bits all-zero.

Or WRITE DIRECT does not select an element; or selects more than one element; or bit 8 of the I_2 field is a 1; or bit 15 of the I_2 field is a 1 on an IOCE selection.

Or READ DIRECT does not select a computing element; or selects more than one computing element.

Or DIAGNOSE specified an operation whose execution is not permitted for a CE in state Three, Two or One, or specified an MCW not located on an integral boundary for a doubleword.

In most of the above interruption conditions the instruction is suppressed. Therefore, storage and external signals remain unchanged, and the PSW is not changed by information from storage. The only exception is READ DIRECT, which is terminated when a store protection or addressing violation is detected. Although storage remains unchanged, a direct-control signal-out signal may have been made available. In every case, the operation is terminated on a fetch-protection violation.

JUNE 1, 1971

When an interruption is taken the instruction address stored as part of the old PSW has been updated by the number of halfwords indicated by the instruction-length code in the old PSW.

Operand addresses are tested only when used to address storage. The address restrictions do not apply to the components from which an address is generated -- the content of the D_1 field, and the content of the register specified by B_1 .

Programming Notes:

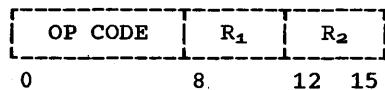
When a program interruption occurs, the current PSW is stored in the old PSW location. The instruction address stored as part of this old PSW is thus the updated instruction address, having been updated by the number of halfwords indicated in the instruction-length code of the same PSW. The interruption code in this old PSW identifies the cause of the interruption and aids in the programmed interpretation of the old PSW.

If the new PSW for a program interruption has an unacceptable instruction address, another program interruption occurs. Since this second program interruption introduces the same unacceptable instruction address, a string of program interruptions is established which may be broken only by an external or I/O interruption. If these interruptions also have an unacceptable new PSW, new supervisor information must be introduced by initial program loading or by manual intervention.

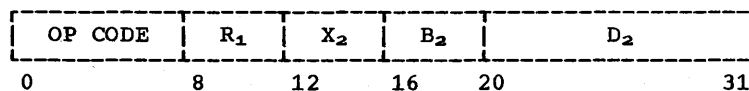
8.6 DISPLAY INSTRUCTION FORMATS

Display instructions use the following two formats:

RR Format



RX Format



| In the RR format, the R_1 field for CONVERT AND SORT SYMBOLS specifies whether primary radar and single symbol data or beacon data is to be processed by the instruction. The R_1 field is ignored in the CONVERT WEATHER LINES and REPACK SYMBOLS instructions.

| The R_2 field of the CONVERT AND SORT SYMBOLS and CONVERT WEATHER LINES instructions must contain the constant 9. The R_2 field of REPACK SYMBOLS is ignored.

| In the RX format, as used for the LOAD CHAIN instruction, the content of the register specified by R_1 is called the first operand.

| The contents of the general registers specified by the X_2 and B_2 fields are added to the content of the D_2 field to form the address of the second operand.

JUNE 1, 1971

A zero in the X_2 or B_2 fields indicates the absence of the corresponding address component.

An instruction can specify the same general register both for address modification and for operand location. Address modification is always completed prior to operation execution.

In the three RR instructions, the contents of some general registers and certain storage locations are likely to be changed in the operation execution.

8.6.1 DISPLAY INSTRUCTIONS

The display instructions, their mnemonics, formats, and operation codes are listed below. The table also indicates when the condition code is set and the exceptional conditions which cause a program interruption.

TABLE 8-XI DISPLAY INSTRUCTIONS

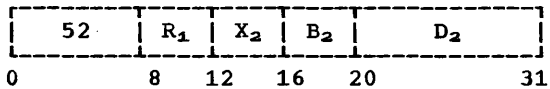
NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
LOAD CHAIN	LC	RX	C P A S	52	8.6.1.1
CONVERT AND SORT SYMBOLS	CSS	RR	C P A S	02	8.6.1.2
CONVERT WEATHER LINES	CVWL	RR	P A S	03	8.6.1.3
REPACK SYMBOLS	RPSB	RR	C P A S	0F	8.6.1.4

Legend

C Condition Code is set
P Protection exception
A Addressing exception
S Specification exception

8.6.1.1 Load Chain

LC $R_1, D_2 (X_2, B_2)$ [RX]



The first operand is placed in the third operand location. Then the second operand is placed in the first operand location. The condition code depends on bit 31 of the second operand.

The first operand is in the general register specified by R_1 , normally the even register of an even-odd pair. The third operand is an odd numbered register, either one larger than or equal to R_1 .

JANUARY 31, 1973

TNL: GN31-3005

Resulting Condition Code:

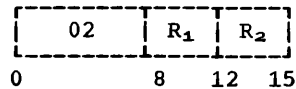
0 Bit 31 of second operand is 0
 1 -- --
 2 -- --
 3 Bit 31 of second operand is 1

Program Interruptions:

Protection
 Addressing
 Specification

PROGRAMMING NOTE:

The Load Chain instruction is useful in accessing chained or linked blocks of data (e.g., full data blocks).

8.6.1.2 Convert and Sort SymbolsCSS R_1, R_2 [RR]

The CONVERT AND SORT SYMBOLS instruction will process an input data stream for one PVD under control of information contained in or referenced by general registers 0-11 and floating point registers 0-6.

The R_1 field, when it contains the constant 2, specifies beacon input data, when zero specifies primary radar and/or single symbol input data. The R_2 field must contain the constant 9 (specifying the GPR which contains the address of the next doubleword in prime storage).

Figure 8-17 at the end of this chapter is a general flowchart of the operation for this instruction.

The input data stream consists of one or more contiguous data blocks which have one of two basic input formats, depending upon whether the data is primary radar and/or single symbol or beacon. The two formats are shown in Figure 8-6.

The primary radar or single symbol data consists of a header word (on a doubleword boundary), a word containing a symbol, followed by some number of data words. Each data word consists of an Y and X coordinate. If the number of data words in a batch is odd, a dummy data word must be included to assure doubleword boundary alignment of the next batch number. The dummy data word must contain 7FFF7FFF (bits 0 and 16 are zeros); these dummy words will never be displayed.

Y, X coordinates in the data words are 15-bit system coordinates which are always positive and are specified to the nearest 1/16 of a nautical mile over the range 0-2047.9375 (i.e., the binary point is

| always assumed to lie between bits 11 and 12 or bits 27 and 28 for the
| Y, X coordinates, respectively).

Beacon format requires a fixed six-word data block per X,Y data word. The sixth word is a header alignment word for doubleword alignment of the next header word, and is ignored by CSS.

As CSS processes the input data, it will generate output data into sixteen sort bins according to its X,Y position on the display. The data is sorted contiguously within each sort bin. Figure 8-7 shows the output data word format.

Both the input data and output data (in each of the sixteen sort bins) are paged. Each page consists of 512 bytes beginning on a page doubleword boundary. The starting address of a page is defined as $N512+8$ (i.e., the low-order nine bits of the address of the initial entry to the page is the binary value 000001000). The first 504 bytes of the page constitute the data field; the last eight bytes (page link field) of the page contain a chain address to the next page (in bytes 5-7).

All operands for this instruction are contained in assigned general purpose and floating point registers or pointed to by these registers. Specific input operand register assignment is shown in Figures 8-8 and 8-9 for primary radar/single symbol data and for beacon data, respectively (All bit positions shown as zero in figures 8-8 and 8-9 are required to be zero). The function of each of these operands is defined as follows:

NEXT ADDR BIN #N-(GPRs 0-7):

These are half-word operands, each of which is a displacement value for a particular sort bin. When the operand is added to the Sort Bin Base Address (GPR 8), the sum defines the next word address in the nth sort bin into which the output data word is to be stored. After each output data word is stored, the half-word operand displacement is incremented by four bytes. Each displacement value, upon initially entering CSS, may be assigned any word boundary value except for the following:

- (a) It must be greater than 27 (hex) since the first five doublewords specified by the Sort Bin Base Address are utilized as a dedicated save area by CSS.
- (b) It may not have the values of 1F8 (hex) or 1FC (hex) in the low order nine bits since CSS would store data into the page link field of the page.
- (c) It must be equal to or less than FE00 (hex) in order to contain a page within the 16 bit displacement field.
- (d) No set of displacement values should be assigned to more than one sort bin.

JANUARY 31, 1973

doubleword (containing the left hand word already processed) and to begin by processing the right hand data word. The programmer must not reset these bits prior to a CSS reentry following a sort bin page overflow termination. These bits are reset by a normal termination (word count zero) or by a program interrupt.

DS- (GPR 8):

The Data Stored (DS) bits 4 and 5 are reserved for use by CSS and should be set to zero by the programmer. Prior to any termination due to I/O or External interrupt or bin page overflow, the DS bits will be set if any output words had been stored in any sort bin in CSS execution up to that point. Their function is to indicate to CSS (upon re-entry from interrupt or bin page overflow handling) that output data was stored in a previous CSS execution and that the condition code must be set to one upon normal termination of CSS (word count zero), whether or not such termination occurs in the current CSS execution or subsequent CSS executions (i.e., re-entries due to a series of interrupts or bin page overflow conditions). These bits are reset by normal termination (word count zero), any program interrupt, or an invalid beacon header format termination.

SORT BIN BASE ADDR - (GPR 8):

This address must always lie on a N512+8 page doubleword boundary (i.e., the low-order nine bits must contain the binary value 000001000). It is used as the base address for referencing the sixteen sort bins as well as for accessing the CSS save area. It is not changed by instruction execution.

BIN NO. - (GPR 9):

Whenever CSS encounters a sort bin page overflow (126 output words have been stored in a sort bin page), it stores the affected sort bin number (0-F hex) in bits 4-7 of GPR 9 (BIN NO.). CSS is then terminated with CC 2. This indicates to the programmer which sort bin requires a new displacement address.

NEXT DW ADDR IN PRIME - (GPR 9):

Bits 8-31 maintain the address of the next input doubleword to be accessed from storage and are updated on a doubleword basis.

For any termination (with the exceptions noted below), this address will be left pointing to the next doubleword location in prime storage not yet processed. If, after processing the left-hand word of primary radar or single symbol data, (a) a sort bin page overflow occurs, or (b) a word count zero termination occurs illegally (from an initial odd word count), this address will point to the current doubleword just partially processed. In processing beacon data, if no header is encountered, termination occurs with CC 3 and this address is left pointing at the invalid doubleword location.

JANUARY 31, 1973

TNL: GN31-3005

NEXT ADDR BIN #0										NEXT ADDR BIN #1										GPR 0														
NEXT ADDR BIN #2										NEXT ADDR BIN #3										1														
NEXT ADDR BIN #4										NEXT ADDR BIN #5										2														
NEXT ADDR BIN #6										NEXT ADDR BIN #7										3														
NEXT ADDR BIN #8										NEXT ADDR BIN #9										4														
NEXT ADDR BIN #10										NEXT ADDR BIN #11										5														
NEXT ADDR BIN #12										NEXT ADDR BIN #13										6														
NEXT ADDR BIN #14										NEXT ADDR BIN #15										7														
RW	0	0	DS	0	0	SORT BIN BASE ADDRESS																				8								
0	1					4	5														8	31												
0	0	0	0	BIN NO.			NEXT DW ADDR IN PRIME																				9							
			0			3	4														7	31												
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																				WORD COUNT										10				
															15	16														31				
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CONVERSION CONSTANT										11			
0	1																					23	24	31										
SY	Y0G (ORIGIN)										(GEOGRAPHIC AREA)										SX	X0G (ORIGIN)										FPR 0		
0	Y1G										0	X1G										0												
(STERILE AREA #1)																														2				
0	Y0S1 (ORIGIN)										0	X0S1 (ORIGIN)										2												
0	Y1S1										0	X1S1										2												
(STERILE AREA #2)																														4				
0	Y0S2 (ORIGIN)										0	X0S2 (ORIGIN)										4												
0	Y1S2										0	X1S2										4												
(STERILE AREA #3)																														6				
0	Y0S3 (ORIGIN)										0	X0S3 (ORIGIN)										6												
0	Y1S3										0	X1S3										6												
0			1																15	16	17													31

Figure 8-8 Input Operand Register Assignment for Primary Radar/Single Symbol Data in CSS

NEXT ADDR BIN #0															NEXT ADDR BIN #1															GPR
NEXT ADDR BIN #2															NEXT ADDR BIN #3															0
NEXT ADDR BIN #4															NEXT ADDR BIN #5															1
NEXT ADDR BIN #6															NEXT ADDR BIN #7															2
NEXT ADDR BIN #8															NEXT ADDR BIN #9															3
NEXT ADDR BIN #10															NEXT ADDR BIN #11															4
NEXT ADDR BIN #12															NEXT ADDR BIN #13															5
NEXT ADDR BIN #14															NEXT ADDR BIN #15															6
0 0 0 0 DS 0 0															SORT BIN BASE ADDRESS															8
0 4 5 8															NEXT DW ADDR IN PRIME															9
0 3 4 7 8															DATA BLOCK COUNT															10
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															0 15 16 31															
R	PVD INDEX							ALTITUDE MASK							T ₁ T ₂ T ₃ 1	CONVERSION CONSTANT				11										
0	0 1 7 8							15 16 19							0 0 0 0	0 24 31														
(GEOGRAPHIC AREA)																														
SY	Y0G (ORIGIN)							SX							X0G (ORIGIN)				FPR											
0	Y1G							0							X1G				0											
(STERILE AREA #1)																														
0	Y0S1 (ORIGIN)							0							X0S1 (ORIGIN)				2											
0	Y1S1							0							X1S1															
(STERILE AREA #2)																														
0	Y0S2 (ORIGIN)							0							X0S2 (ORIGIN)				4											
0	Y1S2							0							X1S2															
(STERILE AREA #3)																														
0	Y0S3 (ORIGIN)							0							X0S3 (ORIGIN)				6											
0	Y1S3							0							X1S3															
0	0 1							15 16 17							31															

Figure 8-9 Input Operand Register Assignment for Beacon Data in CSS

INPUT DATA COUNT - (GPR 10):

GPR 10 initially must contain the input data count for the data to be processed by CSS. This count is subsequently maintained in GPR 10 by CSS during the instruction processing. The significance of the input data count is dependent on the type of data to be processed as shown below:

- (a) Primary radar/single symbol: The input data count must be even and must reflect the number of words to be processed. (i.e., the number of data words (Y,X coordinates) plus two (2) for each header doubleword). This count is decremented by one for each data word processed and by two for each header processed.
- (b) Beacon: The input data count must reflect the number of beacon data blocks to be processed. CSS will decrement this count by

JUNE 1, 1971

one upon completion of processing for a data block (i.e., each six words).

R-(GPR 11):

Bit 0 in GPR 11 is a control bit affecting the brightness bit (BR) in every output word generated by CSS. If R is zero, the BR bit in every output word is set according to bit 13 (BR bit) of the header last encountered in the input data stream; if R is one, the output BR bit is made zero regardless of the setting of bit 13 within the header.

PVD INDEX - (GPR 11 - Beacon Data Only):

The PVD INDEX field is contained in bits 1-7. It is used to address the index bits in words 2-4 of the beacon data block. The PVD INDEX value (bit displacement from bit 0, word 2 is as follows:

PVD Index Bits	Code	
1,2	00	Word 2
	01	Word 3
	10	Word 4
	11	Invalid
3,4	00	Byte 0 of selected word
	01	Byte 1 of selected word
	10	Byte 2 of selected word
	11	Byte 3 of selected word
5-7	000	Bit 0 of selected byte
	001	Bit 1 of selected byte
	010	Bit 2 of selected byte
	011	Bit 3 of selected byte
	100	Bit 4 of selected byte
	101	Bit 5 of selected byte
	110	Bit 6 of selected byte
	111	Bit 7 of selected byte

If PVD index bits 1 and 2 are 11 (i.e., are illegal), the PVD index bit is treated as zero and all beacon data blocks which pass through PVD index filtering are rejected.

When processing non-beacon data, the PVD INDEX field must be zeros.

ALTITUDE MASK - (GPR 11 - Beacon Data Only):

Bits 8-15 specify altitude filtering ranges. Each bit corresponds to a different altitude range.

TYPE FILTERING BITS - (GPR 11 - Beacon Data Only):

Bits 16, 17, 18 of GPR 11 are a type mask (T1, T2, T3). Bit 19 must contain a one.

CONVERSION CONSTANT - (GPR 11):

The conversion constant (bits 24-31) comprises a scaling control constant (bits 24-25) and a conversion multiplier (bits 26-31). The

JUNE 1, 1971

conversion constant is used to convert coordinates relative to the PVD origin from system scale to PVD scale. (The translation of system coordinates from system origin to PVD origin is performed during geographic filtering.)

Conversion constants may be derived from the following formulae:

$$(a) \quad M = \frac{4096}{S' \times 4^N}$$

$$(b) \quad S = \frac{4096}{M \times 4^N}$$

where S' is the nominal scale (PVD diameter in nautical miles)

S is the actual exact scale truncated to the nearest sixteenth of a mile.

M is the conversion multiplier and must be an integer in the range $1 \leq M \leq 42$.

N is the scaling control constant and must be an integer in the range $0 \leq N \leq 2$.

Given any nominal scale S' , formula (a) may first be used to solve for any acceptable combination of M and N with M rounded to the nearest integer. For example, in solving for $S'=28$, $N=0$ would indicate $M=146$ which is invalid; for $S'=28$, $N=1$ would indicate $M=37$ which is acceptable; for $S'=28$, $N=2$ would indicate $M=9$ which is also acceptable but a poorer approximation.

The actual exact scale S should then be determined from formula (b) using the M and N values. (Here, S must be truncated to the nearest sixteenth of a nautical mile, less than the actual quotient). Continuing our example, in solving for $M=9$, $N=2$ (or $M=36$, $N=1$) would indicate $S=28.44$ ($28 \frac{7}{16}$ 28.444); for $M=37$, $N=1$ would indicate $S=27.63$ ($27 \frac{10}{16}$ 27.67). (Note that in solving for $M=16$, $N=0$ would indicate $S=255.94$ ($255 \frac{15}{16}$ 256.00)).

Given two or three combinations of M and N , one should always pick the combination with the smaller N , since the conversion constant for this combination will always provide a scale conversion closest to the nominal scale S' .

Table 8-XII lists the standard set of conversion constants to be used in the 9020E System. The actual scales are truncated to the nearest sixteenth of a mile. The conversion constants are comprised of N (first two bits) and of M (last six bits), expressed in binary.

GPRs 12-15:

These registers may be used by the programmer. GPRs 12 and 13 are initially saved by CSS and subsequently restored before termination.

JANUARY 31, 1973

TNL: GN31-3005

TABLE 8-XII CONVERSION MULTIPLIERS

<u>SCALE</u>		<u>CONVERSION CONSTANT</u>	
<u>Nominal</u>	<u>Exact</u>	<u>Binary</u>	<u>Hex</u>
12	12.19	10010101	95
18	18.25	10001110	8E
28	28.44	01100100	64
43	42.63	01011000	58
64	63.94	01010000	50
102	102.38	00101000	28
128	127.94	00100000	20
158	157.50	00011010	1A
205	204.75	00010100	14
256	255.94	00010000	10
315	315.06	00001101	0D
410	409.56	00001010	0A
512	511.94	00001000	08
819	819.19	00000101	05

FPRs 0, 2, 4, 6:

The doubleword floating point registers define the PVD geographic area and the three sterile areas (areas in which radar and single symbol data cannot be displayed) in terms of pairs of 15-bit system coordinates given relative to the system origin. Each area is a rectangular area which is defined by two sets of Y, X coordinates, one representing the lower left corner, the other the upper right corner. Each coordinate is specified to the nearest 1/16 of a nautical mile over the range 0-2047.9375 (i.e., the binary decimal point is always assumed to be between bits 11 and 12 or bits 27 and 28 for the Y and X coordinates respectively).

YOG, XOG are the only system coordinates in CSS which are allowed to go negative. When either coordinate is negative, the coordinate must be in two's complement form with the corresponding sign bit (SY or SX) set to 1. In addition, if XOG is negative, YOG-0.0625 (where YOG is in true form if positive or in two's complement form if negative) must be entered into FPR 0 in lieu of the exact value of YOG in order that correct results are obtained. The maximum negative true value of the YOG or XOG coordinate should never exceed the actual scale.

Y1G may not exceed YOG plus the actual scale and X1G may not exceed XOG plus the actual scale plus 0.0625.

Resulting Condition Code:

- 0 Processing complete, no data stored in any sort bin.
- 1 Processing complete, data stored in one or more sort bins.
- 2 Page boundary encountered in the sort bin whose number (hex 0-F) is given by BIN NO. (bits 4-7) in GPR 9. Upon re-entering CSS, a new bin displacement value is expected in the appropriate GPR (0-7).
- 3 Invalid beacon format, first word of given data block was not a header.

Program Interruptions:

Protection

Addressing

Specification

PROGRAMMING NOTE 1:

The CSS instruction filters, converts, re-formats, and sorts primary radar, beacon, or single symbol data for one Plan View Display (PVD). It is used to prepare display data for use by the REPACK SYMBOLS instruction. It will operate in one of two modes (primary radar/single symbol or beacon), depending upon the content of the R₁ field.

Primary Radar/Single Symbol Mode:

The CSS instruction, upon finding the R₁ field zero (primary radar/single symbol mode), performs the following processing functions:

(a) Processing headers:

Bit zero of each input doubleword is examined to determine if the doubleword contains header and symbol information (bit 0 is one) or Y, X position data (bit 0 is zero).

If bit 0 is one, the doubleword for the header word and symbol word contains information (S, SYMBOL, BL, BR fields) which is stripped out, re-formatted, and saved in working storage until another header is accessed (see programming note 5). This information is inserted in the output word format for each input Y, X position passing the filtering process.

Upon initial entry (other than re-entry due to interrupt or sort bin page overflow handling), the first doubleword accessed should always be a header in order to provide a valid symbol description for the Y, X data to follow. (CSS makes no test to verify that the first doubleword is a header). Upon re-entry after interrupt or sort bin page overflow, the latest header information is retrieved from working storage.

(b) Geographic filtering:

The PVD constants (Y0G, X0G and Y1G, X1G) provided in FPR 0 define a rectangle (usually a square) approximating the coverage of the geographic area assigned to the PVD. Y, X position data not within this area is rejected for display. Data on the geographic boundaries of the PVD are treated as follows:

<u>AXIS</u>	<u>ACCEPTED</u>
X0G	yes
Y0G	yes
X1G (including end points)	no
Y1G	yes

For each Y, X point, the following is calculated (using 2's complement arithmetic and the PVD constants referenced as they were entered in FPR 0).

JANUARY 31, 1973

TNL: GN31-3005

$$\begin{aligned}x &= X - X0G \\y &= Y - Y0G \\x' &= X - X1G \\y' &= Y - Y1G\end{aligned}$$

Given that Y0G and X0G are both positive and/or zero, if end carries for both y and x are received (i.e., y and x are both positive and/or zero) and no end carries for both y' and x' are received (i.e., both y' and x' are negative), then the point Y, X passes the geographic filter test.

If Y0G-0.0625 is positive and/or zero and X0G is negative, an end carry for y (i.e., y is positive and/or zero) and no end carries for both y' and x' insure that the point Y, X passes the geographic filter.

If Y0G is negative and X0G is positive, an end carry for x (i.e., x is positive and/or zero) and no end carries for both y' and x' insure that the point Y, X passes the geographic filter.

If both Y0G-0.0625 and X0G are negative and no end carries for both y' and x' are received, then the point Y, X passes the geographic filter.

(c) Sterile area filtering

The constants in FPRs 2, 4, and 6 define three rectangular areas called sterile areas which are used for display of tabular data. Data falling within any of these sterile areas is rejected for display.

Data on the sterile area boundaries of a PVD are treated as follows:

<u>AXIS</u>	<u>ACCEPTED</u>
X0S	no
Y0S	no
X1S (including end points)	yes
Y1S	no

For each Y, X point the following is calculated (using 2's complement arithmetic):

$$\begin{aligned}x &= X - X0S \\y &= Y - Y0S \\x' &= X - X1S \\y' &= Y - Y1S\end{aligned}$$

If end carries for both y and x are received (i.e., both y and x are positive and/or zero) and no end carries for either y' and x' are received (i.e., both y' and x' are negative), then the point Y, X lies inside the sterile area and fails the sterile area test

(d) Coordinate conversion (system to PVD):

Scale conversion of system coordinates to PVD display coordinates is performed using the conversion constant provided in bits 24-31 of GPR 11.

(e) Corner filtering:

To approximate the roundness of the CRT display area, the data falling in the corners of the PVD window area (always square) are filtered out. Such corner filtering is performed by

determining if the converted PVD Y, X point falls within any of the four corners defined in the square area by diagonal lines between the following points:

256,0	and	0,256
767,0	and	1023,256
0,767	and	256,1023
767,1023	and	1023,767

If either the X 2-high order bits [X(H)] or the Y 2-high order bits [Y(H)] are either 01 or 10, then the Y, X point cannot be in a corner region and passes the corner filter.

If both X(H) and Y(H) are either 00 or 11, then the Y,X return is in one of the four corner squares. The Y,X point is then normalized so that it appears in the corner square [X(H)=00, Y(H)=00]. Normalization occurs as follows for the X low order 8-bits [X(L)] and Y low order 8-bits [Y(L)] as follows:

If		Normalization is required for	New Values	
X(H)	Y(H)		X'(L)	Y'(L)
00	00	--	X(L)	Y(L)
11	00	X(L)	-X(L)	Y(L)
00	11	Y(L)	X(L)	-Y(L)
11	11	X(L), Y(L)	-X(L)	-Y(L)

Thus the low order 8 bits of a coordinate [X(L) or Y(L)] are complemented (one's complement) if the hi-order bits of the coordinate are 11.

After normalization, if $X'(L) + Y'(L) - 256$, or if $X'(L), Y'(L) = 256,0$ or $0,256$, the return passes the corner filter. Thus with the exception of the end points of the diagonals, points on the diagonals are rejected.

(f) Sort bin storage of data:

For each point passing all filters, the essential display data, S, SYMBOL, BL, BR, and the Y, X display coordinates, are formatted into a single output word (Figure 8-7). (If the R bit in GPR 11 is one, the BR bit is set to zero in the output word at this time). It is subsequently stored in the appropriate sort bin for the given PVD.

The high order two bits of both the Y and X display coordinates are used to sort the data into one of the sixteen sort bins (Figure 8-10). Upon determining the appropriate sort bin, the bin address is generated by the addition of the sort bin base address (bits 8-31, GPR 8) and the appropriate displacement from one of the GPRs 0 through 7. The data is subsequently stored at this bin location and the related sort bin displacement value is updated on a word basis.

Input data is continually processed from prime storage (repeating steps a-f) until the word count (GPR 10) is decremented to zero.

JANUARY 31, 1973

TNL: GN31-3005

Hi-Order Two-Bits of Y	11	1100 6L	1101 6R	1110 7L	1111 7R
	10	1000 4L	1001 4R	1010 5L	1011 5R
	01	0100 2L	0101 2R	0110 3L	0111 3R
	00	0000 0L	0001 0R	0010 1L	0011 1R
		00	01	10	11
		Hi-Order Two-bits of X			

Top Row - Binary bin number (Bin No.) formed from the two highest order bits of Y and X, respectively.

Bottom Row - GPR halfword location for bin displacement address
 e.g. 4L - GPR 4, left halfword
 4R - GPR 4, right halfword

Figure 8-10 Bin Addressing

Beacon Mode:

The CSS instruction, upon finding the R1 field equal to 2 (beacon mode), performs the following functions:

(a) Identifying headers and data blocks:

Each beacon data block in the input stream of beacon data is checked for a header. The header is distinguished from the remaining five words in the data block by the 1 bit in bit 0. Should this test for a header in any data block fail, the instruction is terminated with a condition code of 3.

Each header word is saved throughout the filtering process following. Matching is performed on the filtering field (altitude mask, type field, and selected beacon bit and index bits (words 2-4), in the order listed). Should any match succeed, the Y, X data word (fifth word) is passed on to the geographic, sterile area, and corner filters. If all three filters (altitude, type, and index) fail, the instruction proceeds to the next beacon data block.

Header alignment words in beacon data blocks are bypassed completely by CSS.

(b) Altitude filtering:

Altitude filtering is performed by comparing the ALTITUDE KEY in bits 16-23 of the header word with the ALTITUDE MASK provided in GPR 11, bits 8-15. (Each bit corresponds to a different altitude range). A match on any bit position will cause CSS to directly enter the geographic filter--bypassing both type and index filtering. A no match condition will cause CSS to enter the type filter.

(c) Type filtering:

Type filtering is performed by comparing type bits 8-10 (T1, T2, T3) of the header word with the mask provided in GPR 11, bits 16-18. A match on any bit position will cause CSS to directly enter the geographic filter--bypassing index filtering. A no match condition will cause CSS to enter the index filter.

(d) Index filtering:

Index filtering is performed by checking the selected beacon bit (bit 11) of the header word. If zero, CSS will step ahead to the next header on a doubleword boundary. If one, CSS will access the PVD INDEX in GPR 11, decode, and check the corresponding index bit in word 2, 3, or 4. If either the PVD INDEX is zero or if the referenced index bit is zero, CSS will proceed to the next header. If one, CSS will proceed to the geographic filter.

(e) Geographic filtering:

Same as other mode.

(f) Sterile area filtering:

Same as other mode.

(g) Coordinate conversion (system to PVD):

Same as other mode.

(h) Corner filtering:

Same as other mode.

(i) Sort bin storage of data:

Same as other mode.

Input data is continually processed from prime storage (repeating steps a-i) until the data block count (GPR 10) is decremented to zero.

PROGRAMMING NOTE 2:

Input data in prime storage is stored in 512-byte pages. Upon a 512 carry in incrementing the NEXT DW ADDR in PRIME, GPR 9, CSS will internally link to the next page using the data chain address contained in bytes 5-7 of the 64th doubleword in a page. The data chain address must reference a doubleword boundary; otherwise, a specification exception will occur. No flag indication of a page link will be made by CSS.

Paging is also performed when the output data words are stored in the 16 PVD sort bins. However, when a page has been filled up for a given sort bin, CSS sets the RW indicator (GPR 8) if the left word of the

JANUARY 31, 1973

TNL: GN31-3005

input data doubleword has just been processed, stores the affected sort bin number in bits 4-7 of GPR 9, and then terminates with CC 2. CC 2 specifies to the program that this termination is for bin paging. The operational program must assign a new page to the sort bin (specified by bits 4-7 in GPR 9) by inserting the starting displacement value of the new page in the appropriate GPR (GPR 0 to 7). After the program reissues CSS, CSS processing continues from the point left off as controlled by the RW indicator.

The new initial displacement value for bin paging is subject to the same restrictions indicated for NEXT ADDR BIN #N. Additional restrictions necessary to insure proper operation of the Repack Symbols instruction are:

- (a) Not more than two pages for each sort bin may be used to store CSS output symbols during any CSS execution by batch. The last word location in the second page, into which a symbol is stored, is addressed by a displacement with 1F0 in the low order nine bits.
- (b) The data chain address at the end of the first page must point to the first doubleword in the next page.
- (c) The second page assigned (during bin paging) to any sort bin must have an initial page boundary displacement lower than that of the first page.

PROGRAMMING NOTE 3:

When operating CSS in the beacon mode, paging cannot be performed within a beacon data block. Therefore, the programmer must position any set of beacon data blocks contiguously with the page link field (last doubleword of page), except for the remaining data on the last page processed by CSS. (A set of 21 beacon data blocks will exactly fill the 126-word data field on a page). Otherwise, if a beacon data block does cross a page boundary, the result will be unpredictable.

PROGRAMMING NOTE 4:

CSS tests for an initial zero data count, and for pending interrupts, requests for time clock stepping, or for zero data count after each header or data doubleword if processing primary radar or single symbol data, and after each beacon data block if processing beacon data. If either a Program interrupt or a zero data count condition exists, CSS will set a condition code of 0 (no data stored on this or related instruction executions) or 1 (data stored on this or previous related instruction executions - i.e., DS=11), reset the DS and RW bits in GPR 8, and terminate. (The instruction address in the PSW upon termination will point to the next instruction after CSS and the NEXT DW ADDR IN PRIME (GPR 9) will be updated (except for initial zero data count) to the next doubleword location in prime storage not yet processed).

If neither a Program interruption nor a word count of zero exists, CSS will set a condition code of 1 if output data was stored during this or previous related instruction executions (DS bits on). If no data was stored during this or previous related instruction executions (DS bits off), a condition code of 0 is set.

If the pending condition is not a Program interrupt and the word count is not zero, CSS next determines whether the interrupt pending is a time clock step request or a non-program interrupt (Input/Output or External). If a time clock step request is pending, CSS will decrement the time clock. (If the time clock stepped from + to -, an External interrupt will be generated). Then CSS will test for another non-

program interrupt (I/O or External). If no interrupt is now pending, CSS will automatically re-enter the processing loop of the instruction.

If the interrupt was either an I/O interrupt or an External interrupt (which may be the result of a time clock step), CSS will adjust the instruction address backward so that it points to the given instruction to accommodate instruction re-entry, and terminate.

Interruption then takes place upon termination of the instruction. The current PSW is stored in the old PSW location and the new PSW accesses the appropriate interrupt routine (Program, I/O, or External).

If the interrupt was a non-program interrupt (I/O or External), a Load PSW instruction at the end of the interrupt routine causes the old PSW to be fetched and control is passed to this PSW. The given CSS instruction is re-initiated and its operation is continued from the point left off.

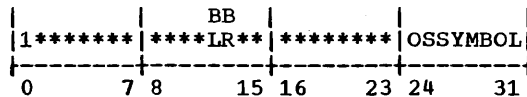
If an I/O, External, or Program interrupt takes place during CSS, the instruction can, if desired, be restarted in some CE other than the given CE, provided that the contents of the general and floating point registers are transferred (via software) from the given CE to the second CE.

PROGRAMMING NOTE 5:

The SORT BIN BASE ADDRESS (GPR 8) itself refers to the first doubleword of a five doubleword area which is reserved as a dedicated working or save area for use by the CSS instruction only.

The first doubleword in the save area is used to retain the original contents of GPRs 12 and 13 during CSS execution. These registers are restored before instruction termination.

The second doubleword in the save area is used to retain old header information for primary radar or single symbol data in the event of any non-program interrupts, time clock steps, or bin page overflow conditions. The format of bits 0-31 of the second doubleword is as follows:



The second word of the second doubleword is used for diagnostic purposes.

PROGRAMMING NOTE 6:

For either bin page overflow or interrupt handling, the following must be maintained by the operational program until the CSS instruction is reissued:

- (a) GPRs 0-11.
- (b) FPRs 0, 2, 4, 6.
- (c) The dedicated CSS working area addressed by SORT BIN BASE ADDR (GPR 9).

In addition, the R₁ and R₂ fields of the reissued instruction must be the same as before.

JANUARY 31, 1973

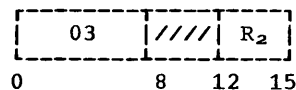
TNL: GN31-3005

PROGRAMMING NOTE 7:

The R bit (GPR 11) is normally used in a range/scaling or off-centering operation, during which the BR bit for all output words generated for history data from prime storage must be made zero.

PROGRAMMING NOTE 8:

CSS will detect and terminate on an odd input word count when in primary/single symbol mode. However, before termination, the prime storage address is left at the current doubleword boundary and paging is not detected. It is extremely important therefore, to use only even input word counts when processing primary-single symbol data.

8.6.1.3 Convert Weather LinesCVWL R₂ [RR]

The Convert Weather Lines instruction will process on a PVD basis, an input data stream of weather line data under control of information contained in or referenced by general registers 0-11 and floating point registers 0-6.

The R₂ field must contain the constant 9, specifying the GPR which contains the address of the next doubleword in prime storage.

Figure 8-18 at the end of this chapter is a flowchart of the operation for this instruction.

The input weather-line data stream consists of zero or more contiguous data blocks which have the basic format shown in Figure 8-11.

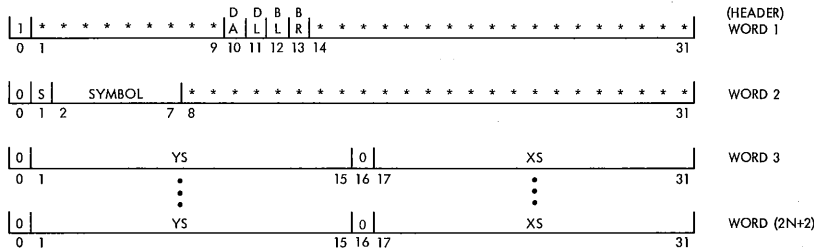
A weather line input data block consists of a header word (on a doubleword boundary), a word containing a symbol, and 2N data words, where N is the number of weather lines. Each line is defined by two points (two Y,X data words). CVWL will accept and process one or more contiguous data blocks in one instruction execution.

Y, X coordinates in the data words are 15-bit system coordinates which are always positive and are specified to the nearest 1/16 of a nautical mile over the range 0-2047.9375 (i.e., the binary point is always assumed to lie between bits 11 and 12 or bits 27 and 28 for the Y, X coordinates, respectively).

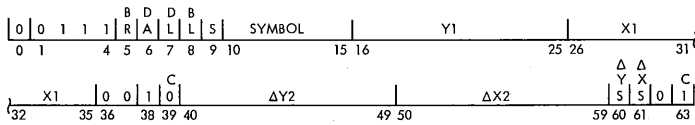
CVWL processes the input data stream and generates output data into refresh storage consisting of contiguous doublewords. Each contains display information for one weather line (two Y,X points). The format of the doubleword is shown in Figure 8-11.

All input data is paged. Each page consists of 512 bytes beginning on a page doubleword boundary. The starting address of a page is defined as N512+8 (i.e., the low-order nine bits of the address of the initial entry to the page is the binary value 000001000). The first 504 bytes of the page constitute the data field; the last eight bytes (page link field) of the page contain a chain address to the next page (in bytes 5-7).

INPUT FORMAT



OUTPUT FORMAT (POSITION-AND-LINE)



Notes: Multiple data blocks are accepted for input.
 (X1, Y1) is origin of weather line.
 (ΔX2, ΔY2) are x and y increments from origin for weather line.
 X1, Y1, ΔX2, ΔY2 are 10-bit values.

Legend:
 DA Dash
 DL Dash length
 BL Blink
 BR Brightness
 S Symbol Size
 ΔY2 Sign of ΔY2
 ΔX2 Sign of ΔX2
 N Number of weather lines
 C0 Character at major position
 C1 Character at final position
 * Ignored by CVWL

Figure 8-11 Input and Output Formats for Weather Line Data

All operands for this instruction are contained in assigned general purpose and floating point registers or pointed to by these registers. Specific input operand register assignment is shown in Figure 8-12. (All bit positions shown as zero in Figure 8-12 are required to be zero).

The function of each of these operands is defined as follows:

GPRs 0-2, 4:

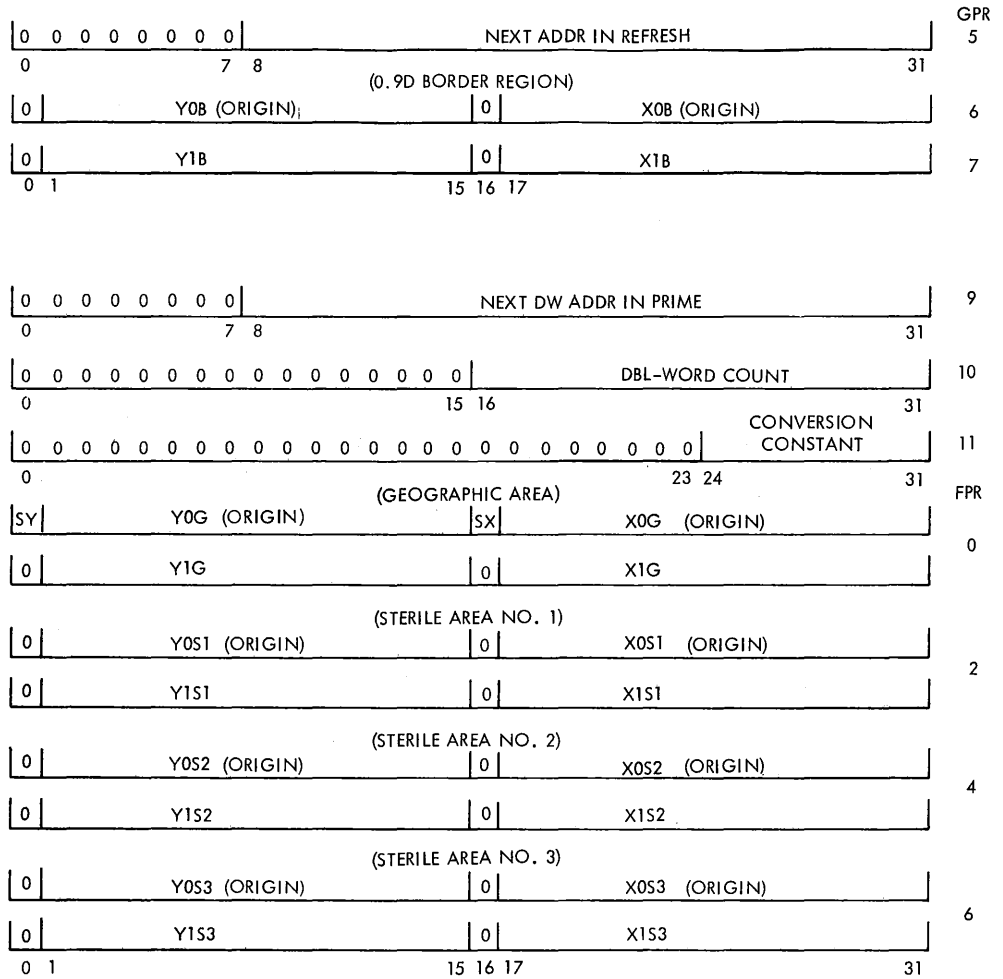
These general registers are reserved for use as working registers for CVWL, and the original contents will be destroyed during CVWL execution.

LAST PRIME STORAGE ADDR - (GPR 3):

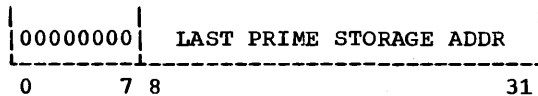
GPR 3 is used by CVWL to retain the last prime storage data address processed (per GPR 9) if and only if, there is a program interrupt. This address is required by the program to restart CVWL after certain program interrupt conditions (See Programming Note 4). The format for this information is as follows:

JANUARY 31, 1973

TNL: GN31-3005



| Figure 8-12 Input Operand Register Assignment for CVWL



NEXT ADDR IN REFRESH - (GPR 5):

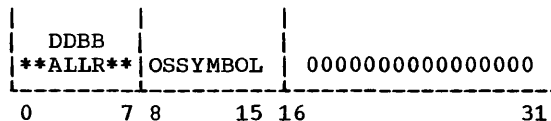
Bits 8-31 define the next refresh storage doubleword location into which the doubleword weather line result will be stored. This address will be maintained in GPR 5 by CVWL during instruction execution. On normal termination, this address will point to the next doubleword location in refresh storage.

0.9 DIAMETER BORDER COORDINATES - (GPRs 6, 7):

Y0B, X0B and Y1B, X1B define the origin (lower left corner) and the upper right corner, respectively, of a rectangular area (usually a square nominally 0.9 times as large as and centered within the PVD area, if the PVD area is also a square, defined by positive coordinates in FPR 0) within the PVD geographic area. CVWL does not verify the validity of these coordinates. Each coordinate is specified to the nearest 1/16 of a nautical mile over the range 0-2047.9375 (i.e., the binary decimal point is assumed to be between bits 11 and 12 or bits 27 and 28 for the Y and X coordinates, respectively).

GPR 8:

This general register is used by CVWL to store header information in the event of an interrupt. The format of the header information in this register is shown below:



The contents of this register must be maintained during an interrupt.

NEXT DW ADDR IN PRIME - (GPR 9):

Bits 8-31 maintain the address of the next input doubleword to be accessed from storage and are updated by CVWL on a doubleword basis. On normal termination this address will point to the location of the next doubleword in prime storage.

DBL-WORD COUNT - (GPR 10):

GPR 10 (bits 16-31) must initially contain the input doubleword count for the data to be processed by CVWL. This count includes both headers and data. CVWL decrements this count by one for each doubleword processed (either header or data).

CONVERSION CONSTANT - (GPR 11):

The conversion constant bits 24-31 comprise a scaling control constant (bits 24-25) and a conversion multiplier (bits 26-31). The conversion constant is used to convert coordinates relative to the PVD origin from system scale to PVD scale. (The translation of system coordinates from system origin to PVD origin is performed during geographic filtering).

Conversion constants may be derived from the following formulae:

JANUARY 31, 1973

TNL: GN31-3005

$$(a) \quad M = \frac{4096}{S' \times 4^N}$$

$$(b) \quad S = \frac{4096}{M \times 4^N}$$

where S' is the nominal scale (PVD diameter in nautical miles)
 S is the actual exact scale truncated to the nearest sixteenth of a mile.
 M is the conversion multiplier and must be an integer in the range $1 \leq M \leq 42$.
 N is the scaling control constant and must be an integer in the range $0 \leq N \leq 2$.

Given any nominal scale S' , formula (a) may first be used to solve for any acceptable combination of M and N with M rounded to the nearest integer. For example, in solving for $S'=28$, $N=0$ would indicate $M=146$ which is invalid; for $S'=28$, $N=1$ would indicate $M=37$ which is acceptable; for $S'=28$, $N=2$ would indicate $M=9$ which is also acceptable but a poorer approximation.

The actual exact scale S should then be determined from formula (b) by using the M and N values. (Here, S must be truncated to the nearest sixteenth of a nautical mile, less than the actual quotient). Continuing our example, in solving for $M=9$, $N=2$ (or $M=36$, $N=1$) would indicate $S=28.44$ ($28 \frac{7}{16}$ 28.444); for $M=37$, $N=1$ would indicate $S=27.63$ ($27 \frac{10}{16}$ 27.67). (Note that in solving for $M=16$, $N=0$ would indicate $S=255.94$ ($255 \frac{15}{16}$ 256.000).

Given two or three combinations of M and N , one should always pick the combination with the smaller N , since the conversion constant for this combination will always provide a scale conversion closest to the nominal scale S' .

Table 8-XII lists the standard set of conversion constants to be used in the 9020E System. The actual scales are truncated to the nearest sixteenth of a mile. The conversion constants are comprised of N (first two bits) and of M (last six bits), expressed in binary.

GPRs 12-15:

These registers may be used by the programmer and are not accessed during CVWL execution.

FPRs 0, 2, 4, 6:

The floating point registers define the PVD geographic area and the three sterile areas in terms of 15-bit system coordinates given relative to the system origin. Each coordinate is specified to the nearest 1/16 of a nautical mile over the range 0-2047.9375 (i.e., the binary decimal point is always assumed to be between bits 11 and 12 or bits 27 and 28 for the Y and X coordinates respectively).

| YOG, XOG are the only system coordinates in CVWL which are allowed to
 | go negative. When either coordinate is negative, the coordinate must be
 | in two's complement form with the corresponding sign bit (SY or SX) set
 | to 1. In addition, if XOG is negative, YOG-0.0625 (where YOG is in true
 | form if positive or in two's complement form if negative) must be
 | entered into FPR 0 in lieu of the exact value of YOG in order that
 | correct results are obtained. The maximum negative true value of the
 | YOG or XOG coordinate should never exceed the active scale.

Y1G may not exceed Y0G plus the actual scale. X1G may not exceed X0G plus the actual scale plus 0.0625.

Resulting Condition Code:

The condition code remains unchanged.

Program Interruptions:

Protection

Addressing

Specification

PROGRAMMING NOTE 1:

The CVWL instruction filters, converts, and re-formats weather line data. It performs the following processing functions:

(a) Processing headers:

Bit zero of each input doubleword is examined to determine if the doubleword contains header and symbol information (bit 0 is one) or Y, X position data (bit 0 is zero).

If bit 0 is one, the doubleword for the header word and symbol word contains information (S, SYMBOL, BR, BL, DA, DL fields) which is extracted, saved in GPR 8, and used to format subsequent output data doublewords until another header is accessed. This information is inserted in the output position-and-line doubleword format for each line passing the filtering process.

Upon initial entry (other than re-entry due to interrupt handling), the first doubleword accessed should always be a header in order to provide a valid symbol description for the line data to follow. CVWL makes no test to verify that the first doubleword is a header, and not finding one, will utilize the contents of GPR 8. Upon re-entry after an interrupt, the latest header information is retrieved from GPR 8.

(b) Geographic filtering:

Geographic filtering is performed utilizing the PVD constants provided (Y0G, X0G and Y1G, X1G) in FPR 0. These constants define an area approximating the coverage of the geographic area assigned to the PVD. Decisions relevant to geographic filtering are always performed on the basis of both of the two points which define the weather line.

If at least one end point of the line lies within the PVD area, the line is always accepted for display (subject to sterile area filtering). If the other end point lies anywhere within the PVD area, the line is displayed in full (subject to sterile area filtering); if outside the PVD area, truncation is performed (see below).

For each Y,X point, the following is calculated (using 2's complement arithmetic and the PVD constants referenced as they were entered in FPR 0).

JANUARY 31, 1973

TNL: GN31-3005

$$\begin{aligned}x &= X - X0G \\y &= Y - Y0G \\x' &= X - X1G \\y' &= Y - Y1G\end{aligned}$$

Given that Y0G and X0G are both positive and/or zero, if end carries for both y and x are received (i.e., y and x are both positive and/or zero) and no end carries for both y' and x' are received (i.e., both y' and x' are negative), then the point Y, X passes the geographic filter test. End points on the geographic boundaries of a PVD are treated as follows:

<u>AXIS</u>	<u>ACCEPTED</u>
X0G	yes
Y0G	yes
X1G (including end points)	no
Y1G	yes

If Y0G-0.0625 is positive and/or zero and X0G is negative, an end carry for y (i.e., y is positive and/or zero) and no end carries for both y' and x' insure that point Y, X passes the geographic filter.

If Y0G is negative and X0G is positive, an end carry for x (i.e., x is positive and/or zero) and no end carries for both y' and x' insure that the point Y, X passes the geographic filter.

If both Y0G-0.0625 and X0G are negative and no end carries for both y' and x' are received, then the point Y, X passes the geographic filter.

(c) Truncation:

When a weather line extends from within the PVD area to some point outside the PVD area, the line is truncated inwards until some point lying within the border region is obtained. (The border region consists of the area between two rectangular areas; if those areas are squares, then they are a PVD square per FPR 0 and an inner square per GPRs 6, 7 which has a side nominally equal to 0.9 times the width of the PVD square). Truncation is performed by successively halving line segments in the appropriate direction. To prevent excessive truncation time, CVWL terminates the operation and rejects the line, if not successful in nine attempts.

No symbol is displayed at any truncated position.

Truncation proceeds as follows, given two end points on a line, Y(I),X(I) (initially the major position) and Y(R),X(R) (initially the rejected position):

1. Compute

$$X(T) = \frac{X(I) + X(R)}{2}$$

$$Y(T) = \frac{Y(I) + Y(R)}{2}$$

2. Pass Y(T),X(T) through the geographic filter area per (b). If it fails the geographic test, substitute Y(T),X(T) for Y(R),X(R) respectively, and return to

step (1) for another inward truncation. If it passes, proceed to step (3).

3. Pass Y(T),X(T) through the border filter (identical in operation to the sterile area filter per (d)). If it fails the border test (i.e., is within the border rectangle), substitute Y(T),X(T) for Y(I),X(I) and return to step (1) for an outward truncation. If it passes, the point lies within the border region and truncation has been completed.

(d) Sterile area filtering:

The sterile areas within the PVD area may or may not overlap into the border region. The three sterile areas are defined by the system coordinates provided in FPRs 2, 4, and 6. If either end point of a weather line is found within a sterile area, the line is not displayed. A weather line which straddles a sterile area (neither end point lies in the sterile region), however, will be displayed. If one end point lies within the PVD area, not in a sterile area, and the other end point lies outside the PVD area, so that truncation was required, then the line will always be displayed even if the truncated point should lie in a sterile area. No symbol will be displayed at the truncated position.

For each Y, X point not a truncated position, the following is calculated (using 2's complement arithmetic):

$$\begin{aligned}x &= X - X0S \\y &= Y - Y0S \\x' &= X - X1S \\y' &= Y - Y1S\end{aligned}$$

If end carries for both y and x are received (i.e., both y and x are positive and/or zero) and no end carries for both y' and x' are received (i.e., both y' and x' are negative), then the point Y, X lies within the sterile area and fails the sterile area test. End points on the sterile area boundaries of a PVD are tested as follows:

<u>AX1S</u>	<u>ACCEPTED</u>
X0S	no
Y0S	no
X1S (including end points)	yes
Y1S	no

(e) Coordinate conversion (system to PVD):

Scale conversion of system coordinates to PVD display coordinates is performed using the conversion constant provided in bits 24-31 of GPR 11.

(f) Position-and-line formatting:

All position-and-line weather lines which have been accepted for display are formatted as shown in Figure 8-11. As is indicated, the first point of the line is given by Y1, X1, and the second point is presented as delta Y2, delta X2 (all four values are 10-bit display coordinates). The major position Y1, X1 in the output format is always the first of the two input Y, X points if that point lies within the PVD area; otherwise, it is the second input point.

JANUARY 31, 1973

TNL: GN31-3005

Other fields in this format not explained in the figure are as follows:

Bits 1-4: These bits (0111) identify the output doubleword format as position-and-line. CVWL does not generate either the short-line or long-line formats for weather line data.

Bit 39: (Denoted by C0). With one exception, CVWL sets this bit to 1 to signify that the character is placed in the major position (at the head of the line). CVWL sets this bit to zero if the header symbol code is a blank (hex 00), in which case, C1 (bit 63) is also zero.

Bit 40-59: If either delta Y2 or delta X2 (or both) is negative, it is given in two's complement form with the negative sign in the respective sign bit, delta YS or delta XS.

Bit 60,61: (Denoted by delta YS, delta XS, respectively). These bits are the signs of $\Delta Y2 = Y2 - Y1$ and $\Delta X2 = X2 - X1$, respectively, where Y1, X1 is the major position.

Bit 63: (Denoted by C1). With two exceptions, CVWL sets this bit to 1 to signify that the character is placed at the end of the line. CVWL sets this bit to zero if this end position is a truncated point and/or the header symbol code is a blank (hex 00).

- (g) CVWL will store the doublewords generated into a refresh storage per GPR 5. This address is updated on a doubleword basis by CVWL as used. Paging is not performed on output data stored in refresh storage.

Input data is continually processed from prime storage (repeating steps a-g) until the doubleword count (GPR 10) is decremented to zero.

PROGRAMMING NOTE 2:

Input data in prime storage is stored in 512-pages. Upon a 512 carry in incrementing the NEXT DW ADDR IN PRIME GPR 9, CVWL will internally link to the next page using the data chain address contained in bytes 5-7 of the 64th doubleword in the page. The data chain address must reference a doubleword boundary; otherwise, a specification exception will occur. No flag indication of a page link will be made by CVWL.

PROGRAMMING NOTE 3:

CVWL tests for an initial zero data count and for pending interrupts, requests for time clock stepping, or zero data count after each header or data doubleword. (Reference Figure 8-17). If either a program interrupt or a zero data count condition exists, CVWL terminates. (The instruction address in the PSW upon termination will point to the next instruction after CVWL and the NEXT DW ADDR IN PRIME (GPR 9) will be updated to the next doubleword location in prime storage not yet processed).

If the interrupt condition is a Program interrupt, the interrupt routine can provide for recovery in the event of an invalid DE address (see programming note 4).

If the pending condition is not a Program interrupt and the word count is not zero, CVWL next determines whether the interrupt pending is

JANUARY 31, 1973

a time clock step request or a non-program interrupt (Input/Output or External). If a time clock step request is pending, CVWL will decrement the time clock. (If the time clock stepped from + to -, an External interrupt will be generated). Then CVWL will test for another non-program interrupt (I/O or External). If no interrupt condition is now pending, CVWL will automatically re-enter the processing loop of the instruction.

If the interrupt was either an I/O interrupt or an External interrupt (which may be the result of a time clock step), CVWL will adjust the instruction address backward so that it points to the given instruction to accommodate instruction re-entry, and terminate.

Interruption then takes place upon termination of the instruction. The current PSW is stored in the old PSW location and the new PSW accesses the appropriate interrupt routine (Program, I/O, or External).

If the interrupt was a non-program interrupt (I/O or External), a Load PSW instruction at the end of the interrupt routine causes the old PSW to be fetched and control is passed to this PSW. The given CVWL instruction is re-initiated and its operation is continued from the point left off.

If an I/O, External, or Program interrupt takes place during CVWL the instruction can, if desired, be restarted in some CE other than the

JUNE 1, 1971

given CE, provided that the contents of the general and floating point registers are transferred (via software) from the given CE to the second CE.

PROGRAMMING NOTE 4:

CVWL can, in the formatting and storing of output data, attempt to store data into an invalid display storage address since display storage addressing is not contiguous from one display element to the next. That is, CVWL could step the NEXT ADDR IN REFRESH (GPR 5) out of a display element.

This condition will result in a Program interrupt (invalid address) and when identified by the interrupt routine, the instruction may be restarted after completing a few steps. To determine if a particular Program interrupt is due to a DE overrun, GPR 5 should be interrogated. If GPR 5 contains an address equal to the address of the last doubleword in a DE plus sixteen (i.e., GPR 5 equals XX0008 hex, where XX equals 2C, 34, 3C, 44, 4C hex), the Program interrupt (invalid address) is due to this condition. The program must then move the data contained in the last doubleword of the display element to a new refresh storage area and replace it with an appropriate TRANSFER command. The next doubleword address after the display data just stored in the new refresh location should be inserted into GPR 5. The program must then increment the input DW count (GPR 10) by one and transfer the Last Prime Data Address (GPR 3) into GPR 9. The program must then re-issue CVWL via Load PSW.

PROGRAMMING NOTE 5:

In the unlikely event that the last output doubleword (word count now zero) is stored in the last doubleword location of the display element, CVWL is terminated normally (without any Program interrupt for invalid address). The NEXT ADDR IN REFRESH (GPR 5) will point at the first location just beyond the end of the display element. Therefore, the program should provide means for detecting and recovering from this unusual condition (GPR 5 can be used, when desired, to determine when the new refresh storage lies outside of the display element). The program should move the data contained in the last doubleword of the display element to a new refresh storage area and replace it with an appropriate TRANSFER command. The next doubleword address in the new refresh location should be inserted into GPR 5 for the next store operation into the new refresh area.

PROGRAMMING NOTE 6:

Whenever an interrupt occurs, the operational program must maintain the contents of the following registers until the CVWL instruction is re-issued.

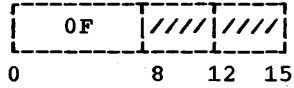
- | (a) GPRs 0 - 11
- | (b) FPRs 0, 2, 4, 6

PROGRAMMING NOTE 7:

All weather lines processed by CVWL must be non-contiguous; that is, no line can be joined to any other. Otherwise, bits 39 and 63 in the output formats of two lines would cause an over-printing of one symbol upon another.

8.6.1.4 Repack Symbols

RPSB [RR]



The REPACK SYMBOLS instruction will perform refresh storage management of radar and single symbol data under control of information contained in or referenced by general registers 0-11 and floating point registers 0-6, the Work Control Table, and the Old Descriptor Table.

Figure 8-19 at the end of this chapter is a flowchart of the operation for this instruction.

RPSB updates and manages one bin of radar and/or single symbol data for one PVD during each instruction execution.

A sort bin represents one-sixteenth of the geographic area covered by a PVD. All radar and single symbol data (previously processed by CSS) to be displayed in this area are contained in that sort bin.

Because of differing radar scan rates and non-unique batch numbers, it is necessary to distinguish data from different radars in the management of the refresh image. This may be done by executing RPSB separately for each radar, i.e., by utilizing separate Work Control Tables and descriptor tables for each radar associated with a PVD or, alternatively, by defining pseudo-class/types, where a pseudo-class/type corresponds to the data for a class/type for a particular radar. Using pseudo-class/types, RPSB would only have to be executed once for each PVD sort bin but would cycle internally once for each pseudo-class/type associated with a PVD. To simplify the discussion in this section, it is assumed that data from only one radar is associated with a PVD.

Each PVD bin can consist of multiple class/types of data, and RPSB requires one pass per class/type. Multiple passes can be performed per instruction execution; that is, a RPSB execution for a bin consists of as many internal passes as there are class/types associated with that PVD.

The function of RPSB is to manage the display data (Old Descriptor Table (ODT) and the old refresh image) in a particular bin for a PVD according to the set of orders in the Work Control Table (WCT). This management consists of generating a New Descriptor Table (NDT) from the ODT and a new refresh data image from the old refresh data image and from the new data derived by prior CSS execution.

The generation of a NDT may require the deletion of some of the descriptors in the ODT, the movement of those not affected, the movement of some of the ODT descriptors from current to history (modification of radar data), and the insertion of new descriptors. Upon completion of the NDT, the old refresh data is similarly processed to form the new refresh data image. Subsequently, on the next update cycle for this PVD, the NDT and new refresh area become the ODT and old refresh area.

Each descriptor table describes the content of refresh storage for one sort bin in terms of descriptors. Thus there are sixteen descriptor tables, one for each sort bin, only one of which is used for any given RPSB execution. A descriptor is a halfword operand, the first byte of which is a batch number (data identification), the second byte is the

JUNE 1, 1971

count of symbols in refresh memory for that particular batch. (see figure 8-13).

There are two basic class/types with respect to the descriptor table format: radar and single/symbol. Radar data contains both history batches and current batches, these being separated by an all-zero descriptor. Single symbol data is specified by one group of descriptors, there being no history or current designation. Each group of descriptors for a class/type (within a sort bin of the PVD) is separated from the next class/type by an all zero descriptor. A descriptor with batch number of zero and non-zero symbol count cannot be used because it will be interpreted as an all-zero descriptor by RPSB; in addition, the non-zero symbol count will invalidate symbol count calculations in RPSB execution.

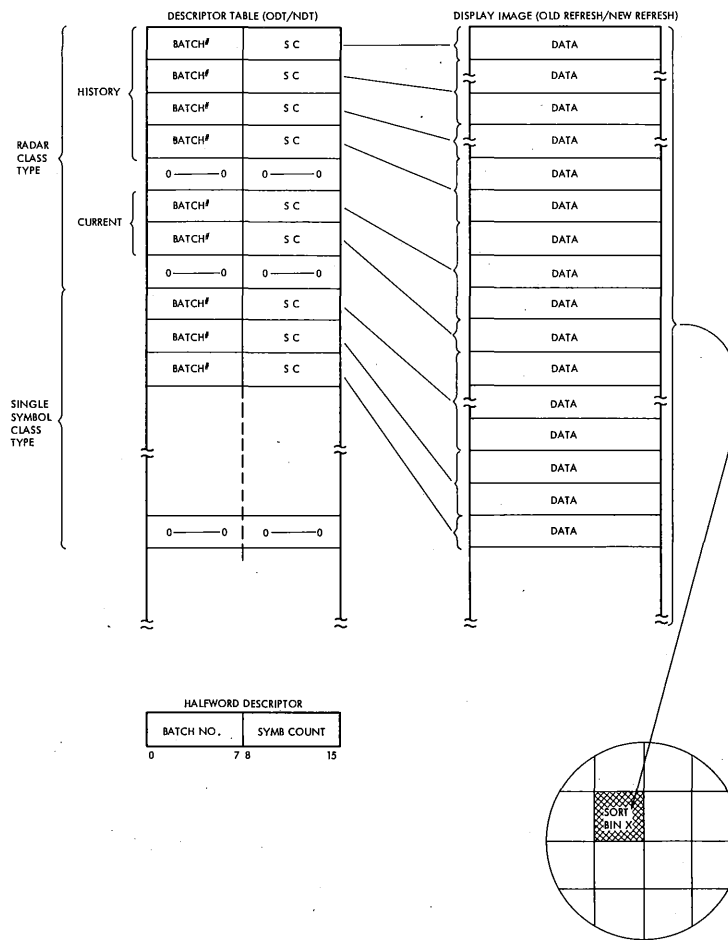


Figure 8-13 Example of a descriptor table for PVD Y sort bin X

All operands for this instruction are referred to by or contained in general registers 0-11, floating point registers 0-6, and the bin displacement values in the WCT. Specific register contents are shown in Figures 8-14 and 8-15 for the general registers and the WCT, respectively. (All bit positions shown as zero in these figures are required to be zero).

JUNE 1, 1971

The function of each of the general registers 0-11 and floating point registers 0-6 is as follows:

GPRs 0,1 - HISTORY COUNT (HCT), CURRENT COUNT (CCT)

Bits 16-31 of these registers are reserved for use by RPSB in accumulating a symbol count (in bytes) of history and current data, respectively. RPSB will not restore the original contents of these registers at the end of its execution.

GPR 2 - NEXT ODT ADDR (ODTA)

Bits 8-31 maintain the address of the next halfword descriptor in the Old Descriptor Table. It is updated on a two-byte (descriptor) basis.

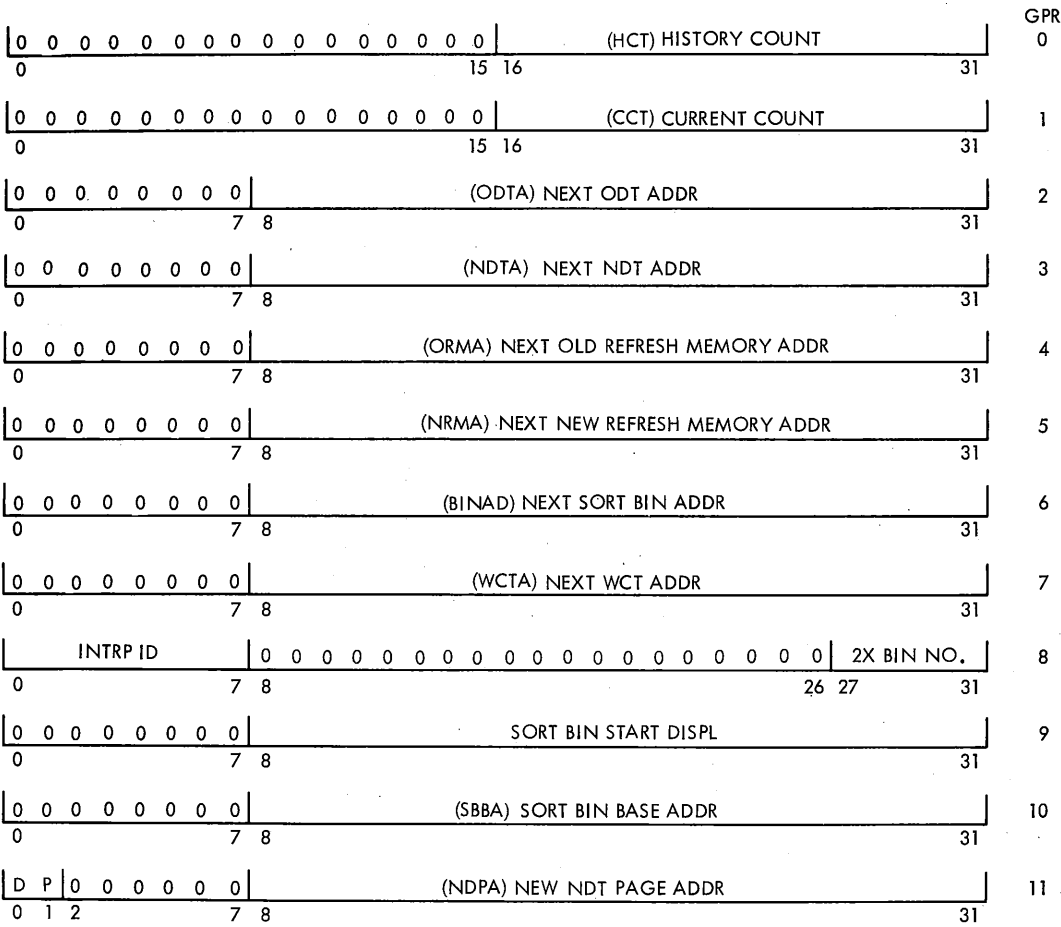


Figure 8-14 Input Operand and Address Assignment for RPSB

Whenever the last descriptor has been processed from the previous doubleword, a new doubleword is fetched from the location indicated by this address (on a doubleword boundary).

JANUARY 31, 1973

TNL: GN31-3005

GPR 3 - NEXT NDT ADDR (NDTA)

Bits 8-31 maintain the address of the next halfword descriptor in the New Descriptor Table. It is updated on a two-byte (descriptor) basis.

GPR 4 - NEXT OLD MEMORY REFRESH ADDRESS (ORMA)

Bits 8-31 maintain the current doubleword address of the next pair of symbols in old refresh and are updated on a doubleword basis (2 symbols per doubleword). This address must initially be on a doubleword boundary, otherwise a specification exception will occur. RPSB accesses Old Refresh Memory even when there is no actual data required for a given bin. Efficient microprogramming causes automatic prefetching of data in anticipation of its later use. It is necessary, therefore, to maintain a valid address in ORMA to avoid addressing interrupts.

GPR 5 - NEXT NEW MEMORY REFRESH ADDR (NRMA)

Bits 8-31 maintain the current doubleword address of the next pair of symbols in new refresh and are updated on a doubleword basis (2 symbols per doubleword). This address must initially be on a doubleword boundary, otherwise a specification exception will occur.

GPR 6 - NEXT SORT BIN ADDR (BINAD)

Bits 8-31 are reserved for use by RPSB for holding the word address in the sort bin from which the next symbol for the current class/type may be fetched. RPSB initially calculates the address from the sum of the SORT BIN BASE ADDR (SBBA) (GPR 10) and the appropriate BIN N DISP value from the bin displacement value table at the beginning of the Work Control Table. This absolute address may be on either a word or doubleword boundary and is not used or modified until RPSB begins to fetch symbols from the sort bin, re-format them, and insert them into new refresh. At this time it is updated on a doubleword basis (2 symbols per doubleword). The referenced sort bin data is derived from previous CSS operation. At the end of the insertion of all new symbols for any given class/type, BINAD will be pointing at the proper location for the next class/type. The original contents of GPR 6 will not be preserved during RPSB execution.

GPR 7 - NEXT WCT ADDR (WCTA)

Bits 8-31 contain the next address in the Work Control Table. The address is always on a halfword boundary. On instruction initialization, this address must be on a word boundary.

GPR 8 - INTRP ID

The INTRP ID code (bits 0 through 7) are reserved for use by RPSB and are normally set to zero to indicate that instruction execution is to start from the beginning. Whenever RPSB detects a pending interrupt condition at one of the several interruptable points in the instruction, it sets the associated INTRP ID code and initializes the instruction for re-entry to the point from which the interrupt was acknowledged.

The INTRP ID code is also set whenever a time clock step request is detected. However, in this case, the time clock stepping is accomplished within RPSB itself. Upon stepping the time clock, RPSB tests for an interrupt condition. If an interrupt is pending, the action described above is performed and the instruc-

tion is terminated; if no interrupt request was made, RPSB re-initializes and re-enters the instruction at the point from which the time clock step was acknowledged (without termination).

Bits 0-3 specify whether the interrupt pertains to descriptor table updating or to display image updating. These codes are as follows:

Bits 0-3 (Hex)

- 0 Descriptor table updating - no delete adjustment is required on the first valid descriptor to be moved after the delete operations.
- 1 Descriptor table updating - the symbol count of the first valid descriptor to be moved (for the current class/type) after descriptor deletions will be increased by one.
- 9 Display image updating - all codes with hex 9 for bits 0-3 pertain to moving history/current symbols from the old refresh area to the new refresh area (DE to DE) or to the insertion of new symbols into the new refresh area (SE to DE).

Bits 4-7 identify specific interrupt subcategories of descriptor table or display image updating and are defined below:

Descriptor Table Updating (Bits 0-3 = hex 0 or 1):

Bits 4-7 (Hex)

- 1 Exit at end of delete routines. Re-entry at beginning of move history descriptors routine.
- 2 Exit at end of move history descriptors routine. Re-entry at beginning of move and modify current radar descriptors routine.
- 4 Exit at end of move and modify current radar descriptors routine. Re-entry at beginning of move current descriptors routine.
- 6 Exit at end of move current descriptors routine. Re-entry at beginning of insert new descriptors routine.

Display Image Updating (bits 0-3 = Hex 9)

All of the following codes will cause re-entry into the instruction at the same point, i.e., at the beginning of the move history current symbols routine. Program interrupts will be caused by either an invalid DE address or a DE storage protection exceptional condition.

Bits 4-7 (Hex)

- 1 Non-program interrupt exit from end of insert descriptors routine or from within the move history symbols loop.
- 2 Program interrupt exit from within the move history symbols loop.
- 3 Non-program interrupt exit from within the move current symbols loop.
- 4 Program interrupt exit from within the move current symbols loop.

JANUARY 31, 1973

TNL: GN31-3005

- 5 Non-program interrupt exit from end of move history and current symbols routine.
- 6 Program interrupt exit from end of move history and current symbols routine.
- 7 Non-program interrupt exit from within the insert new symbols loop when NRMA (GPR 5) and BINAD (GPR 6) are on alternate word boundaries.
- 8 Program interrupt exit from within the insert new symbols loop when NRMA (GPR 5) and BINAD (GPR 6) are on alternate word boundaries.
- 9 Non-program interrupt exit from within the insert new symbols loop when NRMA (GPR 5) and BINAD (GPR 6) are on the same word boundaries.
- A Program interrupt exit from within the insert new symbols loop when NRMA (GPR 5) and BINAD (GPR 6) are on the same word boundaries.
- B Non-program interrupt exit after last new refresh memory store.
- C Program interrupt exit after last new refresh memory store.
- D Non-program interrupt exit before last new refresh memory store.
- E Program interrupt exit before last new refresh memory store.

In the event of a non-program interrupt, all addresses and symbol counts are adjusted to reflect the symbols actually moved to the new refresh memory area. For a program interrupt, however, residual symbol counts and data addresses vary depending upon the particular interrupt condition.

GPR 8 - 2X BIN NO.

Bits 27-31 contain a number which is twice the sort bin number (0-15) for the sort bin currently being processed. It is used for two main functions: (a) to indicate to RPSB the proper bin displacement value to be accessed from any bin displacement value table in the WCT, and (b) to specify the number of the sort bin processed by RPSB to the operational program in the event of termination of the instruction (The validity of GPR8 contents is not guaranteed on specification error termination).

GPR 9 - SORT BIN START DISPL

This register is reserved for use by RPSB. For any particular INSERT order in process during the insert operation, bits 8-31 of GPR 9 hold a start displacement value for sort bin data. It is used for the purpose of calculating a symbol count for the given INSERT. For the first INSERT order in the first class/type processed, this displacement value is taken from the initial bin displacement value table in the WCT. For each succeeding INSERT order (for every INSERT order in all class/types processed in any RPSB execution), this displacement is the stop displacement value for the previous INSERT order (the stop displacement value for any INSERT order is taken from the associated bin displacement value table). The original contents of this register will not be preserved during instruction execution.

GPR 10 - SORT BIN BASE ADDR (SBBA)

Bits 8-31 contain the sort bin base address used in calculating the BINAD (GPR 6) from the sort bin displacement values in the WCT. It must always be on a N512+8 page boundary (i.e., the low order 9 bits must contain the binary value 000001000).

GPR 11 - DP

The DP descriptor paging code (bits 0 and 1) is reserved for use by RPSB and is normally set to zero to indicate that instruction execution is to start from the beginning. Whenever RPSB encounters a NDT page overflow (63 doublewords of new descriptors have been stored in a NDT page), it sets the DP bits to one, inserts the NEW NDT PAGE ADDR (GPR 11) in the page link field of the old page, and replaces the NDTA in GPR 3 with the NEW NDT PAGE ADDR from GPR 11. Later, when RPSB reaches the end of that class/type (all symbols moved), the DP bits, being on, will cause RPSB to terminate the instruction with CC 2.

GPR 11 - NEXT NDT PAGE ADDR (NDPA)

Bits 8-31 contain an address of a new page in the New Descriptor Table, to be used for replacement of NDTA in GPR 3 in the event of a NDT page overflow.

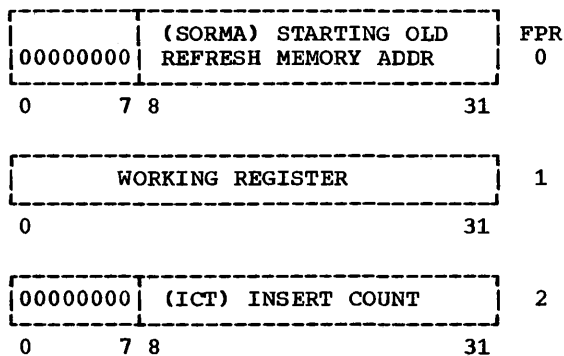
GPRs 12-15:

These registers may be used by the programmer.

FPRs 0-6

These registers are reserved for use by RPSB as working registers. The original contents of these registers will not be preserved during instruction execution.

In the event of an interruption, these registers contain information which must be preserved throughout the interruption. In addition, FPRs 0,2,3,5,6, will be involved in recovery for certain types of program interrupts - invalid DE address and storage protection exceptional conditions (see Programming Note 4). The formats of data contained in FPRs 0-6 are as follows:



JUNE 1, 1971

address) up to and including the first null descriptor (all zero) encountered. The right hand byte of the DELETE N2 order (normally containing a batch number) is ignored by RPSB. This order will, in addition to deleting descriptors from the ODT, delete all old refresh display data associated with these batch descriptors. Only one DELETE N2 order may be given for a class/type.

JUNE 1, 1971

BIN 0 DISPL		BIN 1 DISPL	
BIN 2 DISPL		BIN 3 DISPL	
BIN 4 DISPL		BIN 5 DISPL	
BIN 6 DISPL		BIN 7 DISPL	
BIN 8 DISPL		BIN 9 DISPL	
BIN 10 DISPL		BIN 11 DISPL	
BIN 12 DISPL		BIN 13 DISPL	
BIN 14 DISPL		BIN 15 DISPL	
DELETE	BATCH#	DELETE	BATCH#
DELETE	BATCH#	MODIFY	BATCH#
MODIFY	BATCH#	MODIFY	BATCH#
MODIFY	BATCH#	INSERT	BATCH#
BIN 0 DISPL		BIN 1 DISPL	
BIN 2 DISPL		BIN 3 DISPL	
BIN 4 DISPL		BIN 5 DISPL	
BIN 6 DISPL		BIN 7 DISPL	
BIN 8 DISPL		BIN 9 DISPL	
BIN 10 DISPL		BIN 11 DISPL	
BIN 12 DISPL		BIN 13 DISPL	
BIN 14 DISPL		BIN 15 DISPL	
		EOCT	
DELETE	BATCH#	DELETE	BATCH#
DELETE	BATCH#	INSERT	BATCH#
BIN 0 DISPL		BIN 1 DISPL	
BIN 2 DISPL		BIN 3 DISPL	
BIN 4 DISPL		BIN 5 DISPL	
BIN 6 DISPL		BIN 7 DISPL	
BIN 8 DISPL		BIN 9 DISPL	
BIN 10 DISPL		BIN 11 DISPL	
BIN 12 DISPL		BIN 13 DISPL	
BIN 14 DISPL		BIN 15 DISPL	
		INSERT	BATCH#
BIN 0 DISPL		BIN 1 DISPL	
BIN 2 DISPL		BIN 3 DISPL	
BIN 4 DISPL		BIN 5 DISPL	
BIN 6 DISPL		BIN 7 DISPL	
BIN 8 DISPL		BIN 9 DISPL	
BIN 10 DISPL		BIN 11 DISPL	
BIN 12 DISPL		BIN 13 DISPL	
BIN 14 DISPL		BIN 15 DISPL	
		EOCT	
EOB			

INITIAL BIN
DISPLACEMENT
TABLE

RADAR
CLASS TYPE A

SINGLE SYMBOL
CLASS TYPE B

ORDERS:	HEX CODE:
DELETE N1	10
DELETE N2	20
DELETE	30
MODIFY	40
INSERT	50
END OF C/T (EOCT)	60
END OF BIN (EOB)	70

Figure 8-15 Work Control Table

JUNE 1, 1971

DELETE N1 (10) A DELETE N1 order instructs RPSB to delete descriptors from the ODT (beginning with the current ODT address) up to but not including the first null descriptor (all zero) encountered. The right hand byte of the DELETE N1 order (normally containing a batch number) is ignored by RPSB. This order will, in addition to deleting descriptors from the ODT, delete all old refresh data associated with these batch descriptors. Only one DELETE N1 order may be given for a class/type.

DELETE (30) A DELETE order instructs RPSB to delete the descriptor at the current ODT address and its associated data in old refresh if there is a match between the batch numbers contained in the DELETE order and the descriptor. If a match, RPSB will delete the descriptor and the old refresh data associated with that batch according to descriptor symbol count. ODT descriptors must be deleted sequentially, that is, each descriptor in the ODT must be deleted before the next descriptor. Should there be no match between the batch numbers of the DELETE order and the present descriptor, RPSB will step to the next order in the WCT. RPSB does not step to the next descriptor (e.g., if the next order is also a DELETE, RPSB will compare the batch number contained in this order against the same descriptor used for the previous order). Successive DELETE orders may be given for a class/type.

Dummy DELETE orders are occasionally required to provide INSERT order alignment (discussed under INSERT order). A dummy DELETE order is defined as a DELETE order with a batch number of 255. (The batch number comparison will be unsuccessful since no valid batch of data will receive this batch number). RPSB does not detect this batch number as a special case and will do a normal descriptor batch number comparison.

MODIFY (40) A MODIFY order instructs RPSB to move a radar descriptor from the current classification in the ODT to the history classification in the NDT if there is a match between the batch numbers contained in the MODIFY order and the present ODT current descriptor. In addition, if there is a match, RPSB will modify [reset the brightness bit (BR)] the old refresh data associated with that batch and move the data to new refresh. The number of data words to be moved and modified is specified by the descriptor symbol count (right-hand byte of the descriptor). No modification of the descriptor occurs. ODT descriptors must be either deleted by delete orders or moved to the NDT as history descriptors or as modified current descriptors in sequence (before a new ODT descriptor can be compared for a batch match, all preceding ODT descriptors must have been deleted or moved). Should there be no match between the batch numbers of the MODIFY order and the present descriptor, RPSB will step to the next order in the WCT. RPSB does not step to the next descriptor. Successive MODIFY orders may be given for a class/type.

JUNE 1, 1971

RPSB determines that the class/type of data being processed in a class/type pass is radar data only if a MODIFY order is encountered before any inserts or EOCT. The WCT for a radar data class/type must therefore contain at least one MODIFY order before any INSERT or EOCT orders. (Otherwise the radar data class/type will be treated as a single symbol class/type). If no valid MODIFY order is required, a dummy MODIFY should be inserted. (A dummy MODIFY is defined as a MODIFY order with a batch number of 255. The batch number comparison will be unsuccessful since no valid batch of data will receive this batch number. RPSB does not detect this batch number as a special case and will do a normal descriptor batch number comparison).

A dummy MODIFY order can also be used in two other instances. A dummy MODIFY may be used for INSERT order alignment (discussed under the INSERT order). Another use of a dummy MODIFY occurs when it is placed after an INSERT order. Any MODIFY encountered in this position instructs RPSB to insert a null descriptor (all zero) in the next available location in the NDT. The right hand byte of such a MODIFY order is ignored. RPSB will always ignore the next halfword order position following a MODIFY after an INSERT (i.e., the WCT address is stepped 4 bytes to the next order).

INSERT (50)

An INSERT order instructs RPSB to extract data (previously processed by CSS) from the sort bin, reformat it, insert it into new refresh, and construct a descriptor for this data. RPSB receives the batch number for this data in the right hand byte of the INSERT order. The symbol count is calculated by RPSB by subtracting the sort bin stop displacement (provided in one of the sixteen displacement values following the INSERT order) from the sort bin start displacement (GPR 9). Successive INSERT orders may be given in the WCT for a class/type.

All INSERT orders must lie on odd halfword boundaries (i.e., not on word boundaries). Additionally, the order following any INSERT must be positioned 18 halfwords from the INSERT. The 17th halfword order position immediately following the 16 bin displacement addresses is ignored by RPSB. Alignment of the first INSERT order within a class/type may be facilitated with appropriate dummy DELETE or dummy MODIFY orders.

Every INSERT order is immediately followed by its own table of bin displacement values (always aligned on a word boundary).

EOCT (60)

The EOCT (End of Class/Type) order instructs RPSB to internally re-initialize and to proceed to the next class/type, if any. The right hand byte of this order is ignored by RPSB.

EOB (70)

The EOB (End of Bin) order instructs RPSB to terminate the instruction. This normally signifies the end of all processing for a particular bin. This order must always be preceded by an EOCT

JUNE 1, 1971

order. The right hand byte of the order is ignored by RPSB.

Control of refresh storage management via the WCT and the ODT is illustrated below for both single symbol class/types of data and radar class/types of data.

I. SINGLE SYMBOLS

For each class/type of single symbol data, the ODT contains some number of descriptors, each describing some number of symbols stored in old refresh. A null descriptor (all zeros) in the ODT separates this class/type from the descriptors of the next class/type.

The WCT may contain orders for a single symbol class/type as shown in the following examples:

- | | |
|-------------------------|--------------------------|
| (a) Normal Update Cycle | (b) Delete all data from |
| 0-n DELETE orders | a Class/Type |
| 0-n INSERT orders | 1 DELETE N1 |
| 1 EOCT order | 1 EOCT order |
- (c) Insert data for a Class/Type
- 0 or 1 DELETE #255 order
 - 0-n INSERT orders
 - 1 EOCT order

The number of DELETE orders (0-n) must always be such that the first INSERT order if any, lies on an odd halfword boundary. One dummy DELETE (batch number 255) order may be required to insure proper alignment.

II. RADAR

For each class/type of radar data, the ODT contains some number of history descriptors, a null descriptor (all zero), some number of current descriptors, and a final null descriptor (all zero). The first null descriptor separates history from current while the second null descriptor separates this class/type from the next. Each of the non-zero descriptors define some number of symbols in old refresh.

The WCT may contain orders for a radar class/type as shown in the following examples.

- | | |
|-------------------------|------------------------|
| (a) Normal Update Cycle | (b) Delete Data from a |
| 0-n DELETE orders | Class/Type |
| 1-n MODIFY orders | 1 DELETE N2 |
| 0-n INSERT orders | 1 DELETE N1 |
| 1 EOCT order | 1 EOCT order |
- (c) Insert data into a Class/Type
- 0-n INSERT orders
 - 1 MODIFY (255) order
 - 0-n INSERT orders
 - 1 EOCT order

A dummy MODIFY order may be required to align the INSERT order on an odd halfword boundary. (Another dummy MODIFY may precede this order to establish the class/type as a radar class/type).

RPSB processing is terminated upon recognition of an EOB order following an EOCT order in the last class/type sequence.

JUNE 1, 1971

Radar data is distinguished from single symbol data by the presence of at least one MODIFY order. If no data is to be modified, a dummy MODIFY (batch number 255) order should be used.

Radar data can also be treated as single symbol data if no history exists. In this case, the null descriptor separating history (there is no history) from current should be eliminated.

Resulting Condition Code:

- 0 Processing Complete
- 1 -----
- 2 Page Boundary encountered in the NDT. Upon re-issuing RPSB, a new NDT page address is required in GPR 11.
- 3 Invalid Work Control Table order sequence.

Program Interruptions:

Protection
Addressing
Specification

PROGRAMMING NOTE 1

The RPSB instruction provides for updating of old display data and for re-formatting and insertion of new radar and single symbol data in refresh storage. RPSB processes data for only one sort bin at a time for one PVD, however, it will internally execute once per class/type and process a series of class/types. Since this instruction explicitly requires all data and management tables to be chronological, sequential, and contiguous, extreme care must be exercised in providing the proper instruction inputs.

The instruction will perform the following processing functions:

(a) Re-entry from interrupt or NDT page overflow handling:

RPSB may be terminated prematurely before completion because of an I/O interrupt, External interrupt, Program interrupt, or because a new page must be assigned to the New Descriptor Table. Therefore, upon entering RPSB, the instruction must first determine from the INTRP ID bits (GPR 8) and the DP bits (GPR 11) whether this entry is an initial one or a re-entry due to interrupt or NDT page overflow. If an initial entry, it proceeds to process the first order of the WCT (step b). If a re-entry due to NDT paging, it resets the DP bits and proceeds to process the next class/type of data at the point in the WCT where it previously left off (step n). If a re-entry due to interrupt handling, it re-initializes and branches to the point of previous interruption within RPSB routine as specified by the code. Specific re-entry points corresponding to the interrupt codes are:

<u>INTRP ID (Hex)</u>	<u>Re-Entry Point</u>	<u>Step</u>
01 or 11	Move History Descriptors	(f)
02 or 12	Modify-Current Radar Descriptors	(g)
04 or 14	Move Current Radar Descriptors	(h)
06 or 16	Insert Descriptors	(i)
91 to 9E	Move History Symbols	(k)

(b) Delete N2:

RPSB examines the first order in the WCT (as specified by GPR 7). If it is not a DELETE N2 order, RPSB proceeds to step (c). If the order is a DELETE N2 order, RPSB will step through the ODT descriptors accumulating the symbol counts within each descriptor

JUNE 1, 1971

(for each step, the ODT address is increased by 2) until a null descriptor is encountered. Upon detection of a null descriptor, the ODT address is incremented (by 2) to the first descriptor following the null descriptor. The symbol count accumulated is saved by RPSB and the WCT address is stepped to the next order.

(c) Delete N1:

RPSB examines the next order in the WCT (as specified by GPR 7). If it is not a DELETE N1 order, RPSB proceeds to step (d), still retaining any delete symbol count accumulated by a previous DELETE N2 order, if any. If the order is a DELETE N1 order, RPSB will step through the ODT descriptors, accumulating the symbol count from each descriptor (on top of any count accumulated from step (b)) until a null descriptor is detected. The ODT address (GPR 2) is successfully incremented (by 2 per descriptor) as each descriptor symbol count is accumulated. The ODT address is left pointing at the null descriptor. The WCT address is stepped to the next order.

(d) Delete:

RPSB examines the next order in the WCT. If not a DELETE order, RPSB proceeds to step (e). If the order is a DELETE (dummy DELETES included) the instruction will compare the batch number accompanying the order with the first ODT entry. Should the batch numbers not compare, RPSB will step to the next WCT entry and interrogate the order code again. (The ODT address (GPR 2) is not changed).

If a batch number compare between a DELETE order and the given descriptor table entry is made, RPSB will accumulate the delete symbol count from the given descriptor (on top of any count accumulated by previous DELETE N2 or DELETE orders). In addition, it will increment the ODT address (GPR 2) to the next halfword descriptor. The latter action effectively deletes the descriptor by not transferring it to the NDT. The instruction will continue interrogation of successive orders until all DELETES have been exhausted.

(e) Delete Symbols:

At this point, it will increment the NEXT OLD REFRESH MEMORY ADDR (GPR 4) by the accumulated delete symbol counts (including all DELETE N2, DELETE N1, and DELETE results). None of the deleted symbols are moved to new refresh from the old area.

If the number of symbols deleted is odd and the next old descriptor is not a null descriptor, then one is added to the symbol count of this descriptor. If a radar class/type with both history and current data is being processed, the delete symbol count is odd, and the next descriptor is the history/current null descriptor, then one is added to the symbol count in the first current descriptor. This correction is required because, when the first symbol (either history or current) not deleted is moved from old refresh to new refresh in step (k) for history data or step (l) for current data, RPSB reformats this symbol from word 2 to word 1 position and places a blank symbol in word 2.

After all delete orders, if any, have been processed and the symbol count correction has been made (for an odd delete count), the instruction tests for a time clock step request or a pending interrupt condition. If neither condition exists, RPSB proceeds to step (f). If either condition occurred, RPSB exits here with an INTRP ID code of hex 01 or 11.

JUNE 1, 1971

Each descriptor in the ODT must be deleted or moved (in subsequent RPSB operation) to allow interrogation of further descriptors. The ODT address (GPR 2) is never incremented until the given descriptor has been deleted by a DELETE entry or moved to the NDT (in a subsequent operation).

(f) Move history descriptors (radar class/type) or move single symbol descriptors:

Upon completion of the delete portion of the instruction (i.e., the next WCT entry is not a DELETE), RPSB moves the old descriptors related to the history data if radar data, or to single symbol data, constructing a new descriptor table (NDT). RPSB first examines each descriptor in the ODT at the location specified by NEXT ODT ADR (GPR 2). If non-zero, the descriptor is moved to the NDT at the location specified by NEXT NDT ADDR (GPR 3). At the same time, the symbol count associated with each descriptor is accumulated. Successive descriptors are moved until an all-zero descriptor is encountered. RPSB then tests for an interrupt or a time clock step request; if one is pending, an interrupt exit is taken with INTRP ID code of 02 or 12. If no condition is pending, RPSB then proceeds to step (g).

For radar data, the all-zero descriptor delineates the boundary between the end of history descriptors and the beginning of current descriptors. For single symbol data, the all-zero descriptor marks the end of single symbol descriptors for the given class/type in old refresh.

Single symbol data is considered by RPSB to be current data even though single symbol descriptors are moved in this step. Thus the symbol count accumulated is a current count (rather than a history count as for radar data). Thus single symbol data is moved in step (l) and not in step (k) where the brightness bit is reset.

(g) Modify-current radar descriptors:

Following the move routine, RPSB will interrogate the WCT for a MODIFY order. If there is no MODIFY order, the instruction assumes that single symbol data is being processed, stores zero into GPR 0 (there is no history count for single symbol data; the count of symbols moved in step (f) is treated as a current count), and proceeds directly to step (i).

If a MODIFY order is present, then it is known that radar data is being inputted. The ODT address is stepped to the first current descriptor following the null separator. As in the delete operation, RPSB will step through the orders in the WCT until either a batch number match is found or the MODIFY orders have been exhausted. Should a match be found, RPSB will move the descriptor from the ODT (per GPR 2) to the NDT (GPR 3). In addition, the symbol count associated with the descriptor is accumulated (on top of the move history symbol count). When all MODIFYs (including dummy MODIFY orders) have been interrogated, RPSB will insert an all-zero halfword descriptor into the NDT to provide a separation of history descriptors from current descriptors to be moved or inserted in a later step. After RPSB stores the accumulated symbol count (history plus modified current) into GPR 0, it tests for a pending interrupt or a time clock step request. If one exists, an interrupt exit is taken with an INTRP ID code of 04 or 14. If no condition is pending, RPSB proceeds to step (h).

(h) Move current radar descriptors:

JUNE 1, 1971

When all MODIFY orders have been looked at, RPSB then looks at the next descriptor in the ODT per GPR 2. If the descriptor is all-zero, it signals the end of current radar data. If the descriptor is not zero, the instruction will move successive descriptors from the ODT to the NDT until the end of the current radar descriptors is indicated by an all-zero descriptor. Each time a descriptor is moved, its symbol count is accumulated as a current count (starting with zero).

When the all zero descriptor has been reached, a test is made for a pending interrupt or time clock step; if the test is successful, an interrupt exit is made with an INTRP ID code of 06 or 16. If no interrupt condition is pending, RPSB proceeds to step (i).

(i) Insert descriptors:

First, the symbol count acquired up to this point is stored in GPR 1 (except for re-entry into this step from step (j)). It represents the count of symbols of current data to be moved (without modification of the brightness bit) for either radar or single symbol data.

Next, RPSB checks the WCT for an INSERT order. If there is no INSERT order (i.e., the next order is an EOCT), it stores an all-zero descriptor into the NDT (per GPR 3) to designate the end of current descriptors, and then proceeds to step (k). If an INSERT order is found, the instruction will proceed to generate a descriptor for the INSERT. The batch number is obtained from the order itself and the symbol count is calculated from the stop displacement and the start displacement for the given INSERT order. A descriptor is added to the NDT per GPR 3. The insert symbol count is accumulated from the descriptor (starting from zero).

When calculating the symbol count for any given INSERT order, the first step is to determine the stop and start displacements for the given order. The stop displacement (denoted STOP) is taken from the associated bin displacement value table immediately following the INSERT order. The start displacement (denoted START) for the given INSERT order is given by SORT BIN START DISPL in GPR 9 which is the stop displacement for the previous INSERT order (or the initial start displacement, if the given order was the first INSERT). The symbol count is given by one of the following two expressions:

$$S/C = \frac{(STOP - START)}{4} \quad (STOP > START)$$

$$S/C = \frac{(504 + STOP' - START')}{4} \quad (STOP < START)$$

where STOP' = low order 9 bits of STOP
START' = low order 9 bits of START

This symbol count calculation is correct only if the three restrictions indicated in PROGRAMMING NOTE 2 are met. If the stop displacement equals the start displacement, no descriptor is generated.

After any given INSERT order has been processed, the STOP displacement is placed into GPR 9 as the start displacement for the next INSERT order, if any. The WCT address is then stepped by 36 (18 halfwords) to the next order and the instruction proceeds to step (j).

JUNE 1, 1971

(j) Modify (after Insert)

Upon completion of each INSERT order, RPSB examines the next order for an EOCT, another INSERT, or a MODIFY. If the order is an EOCT, RPSB inserts an all zero descriptor into the NDT to terminate the list of current descriptors, then proceeds to step (k). If the order is an INSERT, RPSB proceeds directly back to step (i) to process the order. If the order is a MODIFY order, RPSB inserts a null descriptor in the NDT (per GPR 3), updates the WCT address by 4 to the next order, and proceeds back to step (i) to process more INSERT orders, if any. If the next order was none of these, then RPSB terminates with CC 3 - invalid WCT order sequence.

(k) Move History Symbols

An interrupt exit with INTRP ID code 91 is taken here if a time clock step request or interrupt request is pending; otherwise RPSB proceeds with this step.

Up to this point, only descriptors have been moved; it is necessary now to move the symbol data from old refresh to new refresh. RPSB will fetch the history symbol count from GPR 0 and proceed to transfer the designated number of symbols (2 symbols per doubleword) from old refresh (per GPR 4) to new refresh (per GPR 5). As each pair of symbols is moved, the brightness bits for the symbols are reset. After completion of this move, the instruction continues to step (l).

If the delete operation has left the first history symbol to be moved on an odd word boundary, the instruction will transfer the first symbol and associated control bits from word 2 to word 1 in the display format (see Figure 8-16). The brightness bit is reset if not already off. The P2 bit will be reset to prevent display of word 2. In addition, a "blank" symbol is inserted in word 2. All remaining history symbols are transferred as given, except that the brightness bits of all symbols are reset.

In this move of radar history, history consists not only of old history symbols but also of those current symbols which are to be modified as they are moved.

As each pair of history symbols is moved from old refresh to new refresh, a test is made within the move loop for a pending interrupt or time clock step request; if an interrupt condition exists, an exit is taken with an interrupt code of 91 or 92.

If the history count is zero, (e.g., for single symbol data), RPSB proceeds directly to step (l).

(l) Move current symbols:

RPSB will transfer current symbols (2 symbols per doubleword) from old refresh (per GPR 4) to new refresh (per GPR 5). The number of symbols moved is given by the current symbol count contained in GPR 1. (The brightness bits are left unchanged). Upon completion of the data move, the instruction continues with step (m).

In moving history symbols in the previous step, the last history symbol to be moved may be in the same doubleword as a second symbol which remains current (first symbol to be moved in this step). In this case, as the doubleword is moved, only the brightness bit for the history symbol will be reset.

JUNE 1, 1971

If all history was deleted and there was no current data which was modified and moved, (i.e., the history count in step (k) is zero), and the first current symbol to be moved (without resetting the brightness bit) is on an odd word boundary, the instruction will transfer this symbol and associated control bits from word 2 to word 1 in the display format. The P2 bit will be reset to prevent display of word 2. In addition, a "blank" symbol is inserted in word 2. All remaining symbols are transferred as given (brightness bit not reset).

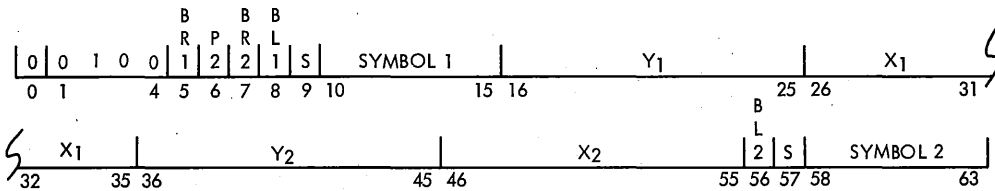
As each pair of current symbols is moved from old refresh to new refresh, a test is made within the move loop for a pending interrupt or time clock step request; if any exists, an interrupt exit with code 93 or 94 is taken.

If the current symbol count is zero, RPSB continues to step (m).

(m) Format and insert new symbols:

If the history and current moves end up on a doubleword boundary, an interrupt exit (codes 95, 96) is taken if an interrupt or time clock step request is pending.

RPSB will extract all symbols (one symbol per word) from the sort bin for the given class/type, re-format them, and add them to the current symbol data in new refresh.



Legend:
 BR Brightness
 BL Blink
 S Symbol Size
 P2 Second Major Position Is Valid

Figure 8-16 Major Position and Single Symbol Output Format

The beginning address for the first symbol in the sort bin is given by the address contained in BINAD (GPR 6).

Consecutive pairs of symbols (two symbols per doubleword) are fetched from the sort bin from the address given by GPR 6. Each pair of symbols will be formatted into the doubleword format shown in Figure 8-16. The two-symbol doubleword is then stored into new refresh per GPR 5. The addresses (NRMA and BINAD) in GPRs 5 and 6 are updated as the symbols are transferred.

Repacking of symbols will be required whenever the left symbol (from sort bin doubleword) is to be inserted into the right half position in the output doubleword and vice versa.

JUNE 1, 1971

If the last symbol from the sort bin is inserted into the left half position (word 1) of an output doubleword, the P2 bit will be reset to prevent display of the right half position (word 2) and a blank symbol is inserted into word 2.

The number of symbols extracted from the sort bin is given by the insert symbol count, obtained in step (i).

As each pair of symbols are fetched from the sort bin, formatted, repacked if necessary, and stored in new refresh, a test is made for a pending interrupt or a time clock step request. If such a condition exists, an interrupt exit is taken with code 97 or 98 if BINAD and NRMA are on alternate word boundaries, or with code 99 or 9A if BINAD and NRMA are on the same word boundaries.

Another interrupt test is made, (INTRP ID = 9D or 9E) just before the last doubleword is stored in new refresh at the end of the insert symbols loop (when BINAD and NRMA are on the same word boundaries).

When the given number of symbols have been inserted into new refresh, RPSB takes step (n).

If the insert symbol count is zero, RPSB proceeds directly to step (n).

(n) Recycling or Termination:

At this point, RPSB has completed refresh storage management for one class/type. The current order in the WCT should always be an EOCT order.

Before re-cycling to the next class/type, a final test for a pending interrupt or time clock step request is made. If one exists, an interrupt exit is made here with an interrupt code of 9B or 9C.

If a page overflow for the NDT had occurred while processing the previous class/type, the DP bits (GPR 11) will have been set. If these are on at this time, RPSB will terminate with CC 2. The operational program will be expected to insert a new NDT page address in GPR 11 and then to reissue RPSB. RPSB will return to this point after resetting the DP bits.

If there was no NDT page overflow or if RPSB had just recovered control after an NDT page overflow, the instruction will proceed with the next order in the WCT. If an EOB order, RPSB is terminated with CC 0 and GPR 7 is updated to the address of the next order. If any other order, RPSB will repeat steps (b) through (n).

PROGRAMMING NOTE 2:

The sort bin data is stored in 512-byte pages. Upon a 512 carry in incrementing BINAD (GPR 6), RPSB will internally link to the next page using the data chain address contained in bytes 5-7 of the 64th doubleword in the page. The data chain address must reference a doubleword boundary; otherwise, a specification exception will occur. No flag indication of a page link will be made by RPSB.

In order that the symbol counts be properly calculated by RPSB during the insert descriptors routine (see PROGRAMMING NOTE 1, step (i)), the following three restrictions must be met:

(a) Not more than two sort bin pages may be processed by any INSERT order (i.e., the stop displacement must occur in the same

JUNE 1, 1971

page as the start displacement or in the second page allocated to the given INSERT order).

(b) The data chain address at the end of the first page must point to the first doubleword in the next page.

(c) The second page assigned to any sort bin must have an initial page boundary displacement lower than that of the first page.

Paging is also performed on the old and new descriptor tables. It is performed internally for the ODT but externally by the operational program for the NDT. When a page has been filled up for the NDT (512 carry), RPSB sets the DP bits in GPR 11 to one and obtains a new page address from GPR 11 and continues the processing. When the processing of a class/type is complete (i.e., all symbols moved), the DP bits are sensed. If off, RPSB recycles and processes the next class/type; if on, RPSB terminates with CC 2. CC 2 specifies to the program that this termination is for NDT paging. The operational program must assign a new NDT page address to GPR 11 and then return control to the instruction. After the instruction is reissued, the DP bits cause RPSB to continue with the next order in the WCT (i.e., first order of next class/type).

NDT paging is thus subject to one restriction, namely that not more than 252 halfword descriptors can be stored in any one class/type pass. (The limit of 252 includes the class/type all-zero entry for all class/types and the history/current all-zero separator for radar data). Otherwise, RPSB may overrun the two pages allocated to NDT per class/type pass (one page is the initial page, the second page is given by GPR 11).

PROGRAMMING NOTE 3:

RPSB tests for an interrupt or time clock step condition at various break points in the instruction as indicated in PROGRAMMING NOTE 1 (see also instruction flowchart Figure 8-19). If an interrupt is pending, RPSB sets the INTRP ID code (GPR 8) to identify the interrupt point. If the interrupt is a non-program type (I/O or External), RPSB will step back the instruction address in the current PSW so that it references the instruction itself (no such stepping back takes place for any program interrupt). Then the instruction is terminated (except for time clock step requests).

An interruption takes place during which the current PSW is stored in the old PSW location and the new PSW accesses the interrupt routine. The particular interrupt is then serviced.

Program interrupts (invalid DE address, DE storage protection exception) are identified by even INTRP ID codes of 92 to 9E. Odd codes 91 to 9D indicate non-program interrupts (I/O or External) or a time clock step request. (see PROGRAMMING NOTE 4).

If the interrupt is a program interrupt, it will be evaluated by the operational program to determine the action to be taken. RPSB may be re-entered in the event of an invalid DE address or a DE storage protect exception.

If a non-program interrupt (I/O or External), the old PSW is refetched by a LPSW instruction and control is again passed to the given RPSB instruction. The INTRP ID bits being on, cause the instruction to branch back to the exit point. Any register contents or status internally stored prior to the interrupt must be restored prior to resumption of RPSB processing.

JANUARY 31, 1973

TNL: GN31-3005

If a time clock step request occurred, an internal routine within RPSB decrements the clock. (If the clock should go from + to - during the clock updating, an External interrupt condition is generated). Following this, another interrupt test is made. Should a pending interrupt condition exist, the instruction is terminated. If no interrupt is pending, RPSB returns control to the point indicated by INTRP ID (see step (a), PROGRAMMING NOTE 1).

If an I/O, External, or Program interrupt takes place during RPSB, the instruction can, if desired, be restarted in some CE other than the given CE, provided that contents of local store are transferred (via software) from the given CE to the second CE.

PROGRAMMING NOTE 4:

In a program interruption (protection, addressing, or specification), the instruction address which is stored in the old PSW location is the address of the next instruction after the current RPSB instruction (i.e., it is not stepped back as in PROGRAMMING NOTE 3).

RPSB does not differentiate among program interrupt types; all program interrupts (addressing, storage protect, etc), once detected, are handled in exactly the same way.

GPRs and FPRs significant for program interrupt recovery are:

	GPR 0	HCT--Residual history count (in bytes)
	GPR 1	CCT--Residual current count (in bytes)
	GPR 4	Updated ORMA*
	GPR 5	Updated NRMA
	GPR 8 (bits 0-7)	INTRP ID code
	FPR 0	SORMA*--ORMA at the beginning of the most recent move symbols sequence**
	FPR 2	ICT--Residual insert count (in bytes)
	FPR 3	SNRMA--NRMA at beginning of most recent move symbols sequence**
	FPR 5	Starting history count (in bytes) at the beginning of the most recent move symbols sequence**
	FPR 6 (bit 0)	M--Multiple interrupt indicator. If this bit is on, the most recent move symbols sequence** was initiated from an interrupt re-entry point
	FPR 6 (bits 8-31)	Starting current count (in bytes) at the beginning of the most recent move symbols sequence**

Floating point registers 5 and 6 (bits 8-31) are especially pertinent for software recovery from a storage protect interrupt.

Any deviation in the recovery procedure described in this programming note should be very carefully investigated because of the great complexity involved.

* If an interrupt occurs prior to the move symbols sequence (i.e., no symbols, history or current, have been moved) and if an odd delete had occurred, this address (ORMA or SORMA) will be on an odd word boundary. However, if the move symbols sequence had been entered and an odd delete had occurred, the address will have been reset backwards to the nearest doubleword boundary.

** Whenever the move symbols sequence is entered, one or more recoverable interrupts can occur. Upon recovery from each interrupt, re-entry into the move symbols sequence initiates a new 'most recent move symbols sequence'.

Program Interrupt (Code 005, Addressing)

An invalid ORMA (in one of the RPSB move symbols routine) or an invalid NRMA (in one of either the move symbols or insert symbols routines) will generate a program interrupt (addressing exception).

When RPSB is terminated by an addressing interrupt in a move symbols routine, an attempt is made to back up the data count and address parameters before instruction termination so that the last DE doubleword (assumed to contain a DE transfer command - TRF) is not included. As this is not always possible, a number of special situations can arise (indicated below) that require interrogation and manipulation by the interrupt program. The program should first test for a NRMA interrupt before any ORMA interrupt in order to determine the interrupt situation.

NRMA Interrupt

| For any NRMA addressing interrupt, NRMA will be backed up to either the last valid DE doubleword address (LADDR), LADDR+8, or LADDR-8, depending upon where the interrupt occurred.

| If NRMA (GPR 5) contains the last valid DE doubleword address (LADDR)*, the program interrupt was caused by NRMA having been stepped out of bounds. Although NRMA will have been stepped back to a valid address (LADDR) by RPSB, formatted data will have stored in the last NRM doubleword, destroying any TRF command there. If this data were moved from ORM, i.e., the residual history count (HCT) plus the residual current count (CCT) was greater than or equal to 8, or residual insert count (ICT) equals 0, it will be necessary only to insert a transfer command (TRF) in LADDR, to assign a new NRMA with the transfer address (from the TRF command), and to re-issue RPSB. However, if this location LADDR contains one or two insert words (HCT + CCT = 4 or 0, respectively) and the residual insert count (ICT) is not zero, it is necessary to either:

1. If $HCT + CCT = 4$ (one insert word): adjust the HCT and CCT to zero, reduce ICT by 4, increment ORMA by 8, or
2. If $HCT + CCT = 0$ (two insert words): reduce the ICT by 8 (but not less than zero).

The program then should move the contents of LADDR (old NRM) to new NRM (per the transfer address), insert a TRF command into LADDR and the transfer address into NRMA, and increase the new NRMA by 8.

If NRMA contains the first invalid DE doubleword address (LADDR + 8), the last DE doubleword address (LADDR) contains symbol data which should be moved to new NRM (per the transfer address). In addition, a TRF command should be loaded into LADDR, and the transfer address plus 8 should be inserted in NRMA.

| If NRMA contains the last valid DE doubleword address (LADDR-8) and the history count plus the current count is greater than or equal to 8,

JANUARY 31, 1973

TNL: GN31-3005

the program interrupt was caused by NRMA having been stepped out of bounds. Both the ORMA and NRMA were stepped back a quadword before termination. In this case, the interrupt routine should store a transfer command at LADDR, add 8 to ORMA, and the transfer address should be inserted in NRMA. The interrupt routine should then reduce the history count in GPR0 by 8, or subtract 8 from the current count in GPR1 if HCT equals 0, or subtract 4 from the HCT and CCT if the HCT is not equal to zero but less than 8.

ORMA Interrupt

If ORMA (GPR 4) is equal to or greater than the address of the third valid doubleword from the end of the DE (NRMA is valid and not equal to or greater than LADDR), the interrupt was caused by ORMA stepping out of bounds. If ORMA is less than this address and NRMA is a valid address not equal to LADDR, then the program interrupt is for an invalid address other than ORMA or NRMA and is not recoverable under the scheme described in this programming note.

$M = 1$ (where M is bit 0 of FPR 6) indicates that the present interrupt is not the first encountered during the move symbols routines for the present class/type.

If ($M = 0$) and ($SORMA < LADDR$) or if ($M = 1$) and ($ORMA = LADDR - 8$ or $LADDR - 16$), then ORMA was stepped out of bounds in the process of moving ORM data and was backed up to a valid DE address before termination. In this case, when ORMA (GPR4) is a valid doubleword address, no further adjustment to ORMA is necessary, however, if ORMA is on a word boundary, then ORMA must be backed up to the previous doubleword address.

The interrupt routine should then reduce the history count in GPR0 (HCT) by the difference $D = (LADDR - ORMA)$ if $HCT \geq D$; or subtract D from the current count in GPR1 (CCT) if $HCT = 0$; or if HCT is not equal to zero but is less than D , subtract HCT from D , set HCT to zero, then subtract D from CCT. Finally, the interrupt routine should increment NRMA by D and insert the transfer address from LADDR into ORMA (GPR4).

* LADDR may be 2BFFF8, 33FFF8, 3BFFF8, 43FFF8, or 4BFFF8 depending on which DE RPSB is working.

($M = 0$) and ($SORMA \geq LADDR$) indicates that ORMA has incremented beyond the last valid symbol location by the delete routine and that no ORM data has been moved. (For an odd delete, ORMA and SORMA will still be on an odd word boundary). The interrupt routine should reassign a new address to ORMA given by the transfer address (per LADDR) plus the delete overflow ($SORMA - LADDR$). For interrupt re-entry, ORMA can be on an odd word boundary (illegal for normal or NDT re-entry).

($M = 1$) and ($ORMA = LADDR + 8$), where ($LADDR + 8$) is the first invalid DE doubleword address, indicates a rare coincidence between either an internal time clock step or a non-program interrupt and an ORMA program interrupt in such a manner as to defeat direct recovery. The RPSB instruction should be re-initialized and restarted from the beginning.

Program Interrupt (Code 004, Storage Protection)

Due to the differences in interrupt detection timing, RPSB does not provide direct recovery for storage protect check interrupts. However, residual address and symbol count parameters should be sufficient for developing software recovery. Significant GPR and FPR contents will be identical to those for program interrupt code 005 (Addressing). RPSB processes a storage protect, once it is detected, in exactly the same way as any other program interrupt (e.g., addressing) is handled.

If NRMA is stepped into a storage protected area, symbol data will have been stored up to and including the last unprotected doubleword location.

PROGRAMMING NOTE 5

For either interrupt or NDT page overflow handling, the following registers must be maintained by the operational program until the RPSB instruction is reissued:

(a) GPRs 0-11

(6) FPRs 0-6

PROGRAMMING NOTE 6

Upon initial entry, both ORMA (GPR 4) and NRMA (GPR 5) must be valid addresses on doubleword boundaries (i.e., for the first class/type to be processed). Thereafter, RPSB restarts each class/type in new refresh on a doubleword boundary regardless of the boundary on which the last symbol was stored. Hence, each class/type always starts with ORMA and NRMA on doubleword boundaries.

PROGRAMMING NOTE 7

Data in other than major position and single symbol format (first byte P0100xxxx) cannot be moved with RPSB. Bits 1-4 of each doubleword moved are always forced to this bit configuration (0100).

PROGRAMMING NOTE 8

The next sort bin page address at the end of a bin page should always be a valid address. If it is not and RPSB fetches data to the end of the bin, prefetching can cause an invalid addressing or specification interrupt.

PROGRAMMING NOTE 9

The old refresh memory address specified should always be valid even when no old data is to be moved, since RPSB prefetches old refresh data to save execution time in its iterative data operations even when the data will not be used. The same is true when old refresh memory data runs to a doubleword from the end of a DE. (i.e., RPSB may attempt to prefetch beyond the valid address area with a resultant extraneous addressing interrupt). Transfer addresses should be valid.

8.6.2 DISPLAY INSTRUCTION EXCEPTIONS

Exceptional instructions or data cause a program interruption. When a program interruption occurs, the current PSW is stored as an old PSW and a new PSW is obtained. The following exceptional conditions cause a program interruption in display instruction operations.

Protection: When the storage key of an accessed location does not match the protection key in the PSW and neither the storage key nor the protection key is zero, a protection exception is recognized.

Addressing: An address specifying any part of data, an instruction, or a control word is outside the available storage for the particular installation, or outside the configured storage, or storage assigned by the storage address translator for a particular CE.

Specification: The second operand address of the LOAD CHAIN instruction is not on a doubleword boundary.

JANUARY 31, 1973

TNL: GN31-3005

In CSS, CVWL, and RPSB instructions, the data chain address contained in bytes 5-7 of the doubleword following a 512 byte page is not on a doubleword boundary.

In the RPSB instruction, the NEXT OLD REFRESH MEMORY ADDR (ORMA) in GPR 4 or the NEXT NEW REFRESH MEMORY ADDR (NRMA) in GPR 5 is not initially on a doubleword boundary.

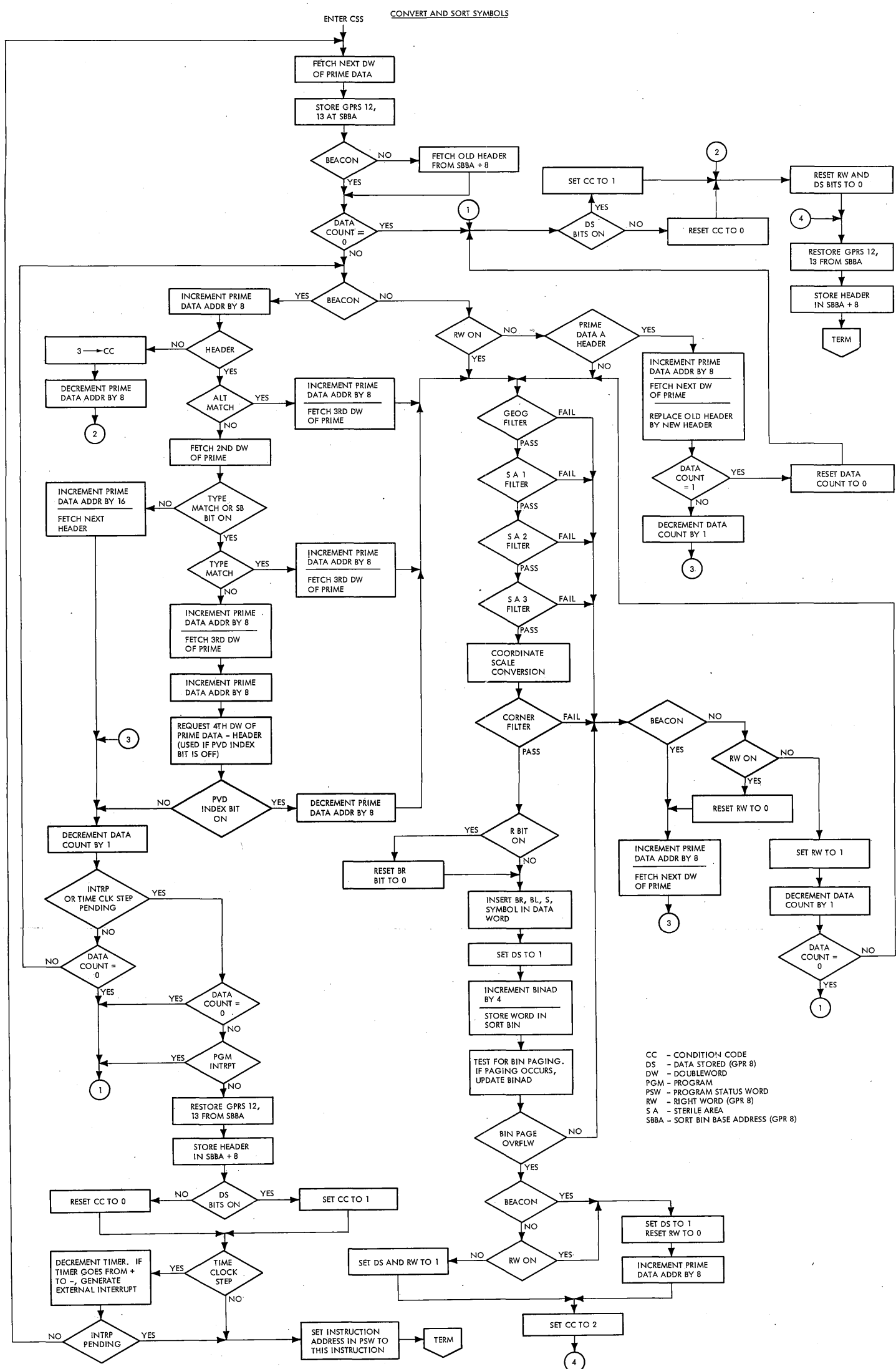


Figure 8-17 CSS Operation Flowchart

JUNE 1, 1971

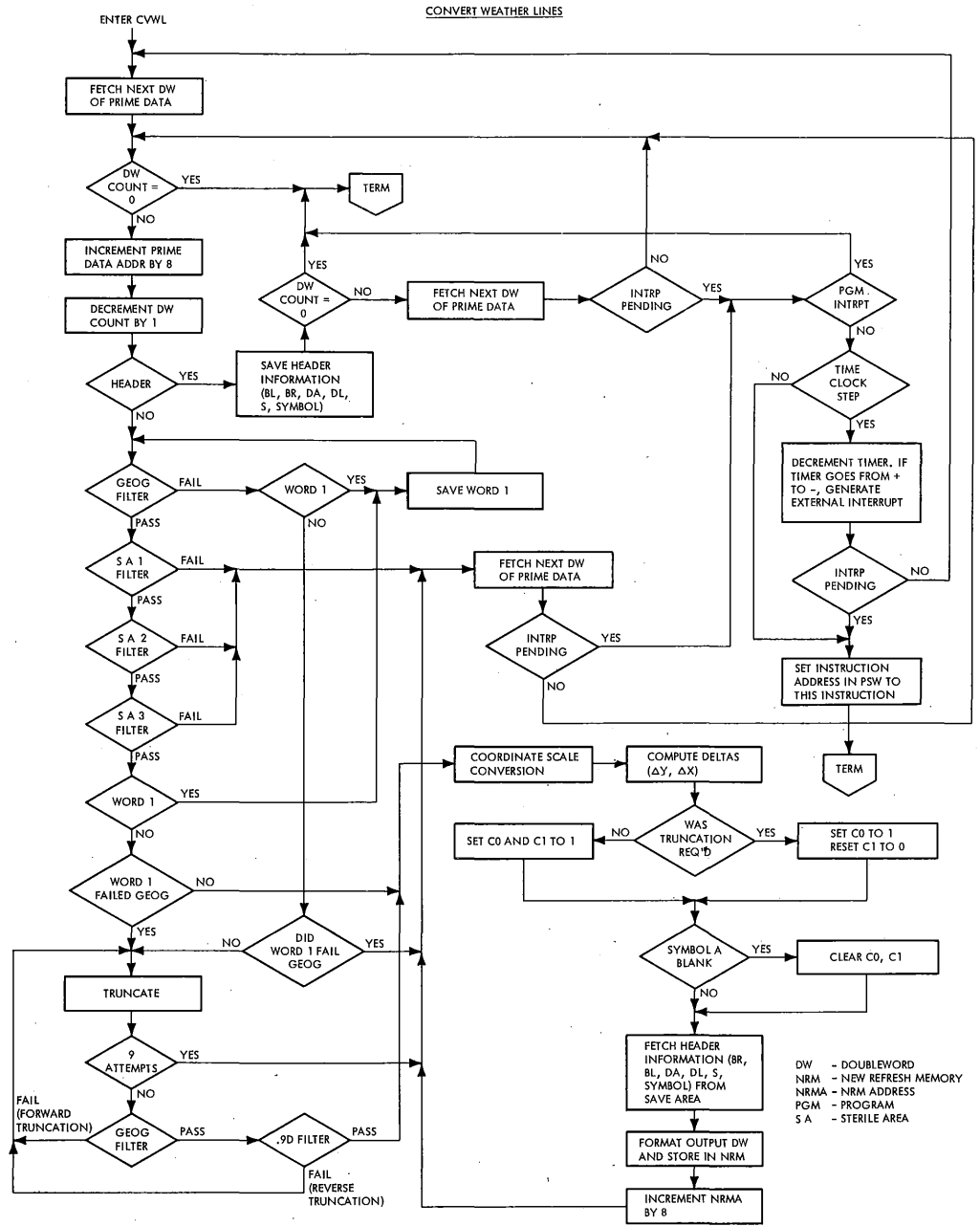


Figure 8-18 CVWL Operation Flowchart

JUNE 1, 1971

CHAPTER 9: INTERRUPTIONS

The interruption system permits the CE to change its state as a result of conditions arising outside the system, in I/O units, or in the CE itself. The five classes of these conditions are input-output, program, supervisor-call, external, and machine-check interruptions.

9.1 INTERRUPTION ACTION

An interruption consists of storing the current PSW as an old PSW and fetching a new PSW.

Processing resumes in the state indicated by the new PSW. The new PSW is not checked for programming errors when it becomes the current PSW. These checks are made when the next instruction is executed. The old PSW contains the address of the instruction that would have been executed next if an interruption had not occurred and the instruction-length code of the last-interpreted instruction.

Interruptions are taken only when the CE is interruptable for the interruption source. Input/output and external interruptions may be masked by the system mask, four of the program interruptions may be masked by the program mask, and the machine-check interruptions may be masked by the machine-check mask.

With the exception of DELAY, CONVERT AND SORT SYMBOLS, CONVERT WEATHER LINES, and REPACK SYMBOLS, an interruption always takes place after one instruction interpretation is finished and before a new instruction interpretation is started. However, the occurrence of an interruption may affect the execution of the current instruction. To permit proper programmed action following an interruption, the cause of the interruption is identified and provision is made to locate the last-interpreted instruction.

When the CE is commanded to stop, the current instruction is finished and all interruptions which are pending or become pending before the end of the instruction, and which are not masked, are taken.

The details of instruction execution, source identification, and location determination are explained in the following sections, and are summarized in Table 9-III, Interruption Action.

PROGRAMMING NOTE

A pending interruption will be taken even if the CE becomes interruptable during only one instruction.

9.1.1 INSTRUCTION EXECUTION

With the exception of DELAY, CONVERT AND SORT SYMBOLS, CONVERT WEATHER LINES, and REPACK SYMBOLS, an interruption occurs when the preceding instruction is finished and the next instruction is not yet started. Reference Chapter 8 for discussion of interrupt handling during these instructions. The manner in which the preceding instruction is finished may be influenced by the cause of the interruption. The instruction is said to have been completed, terminated, or suppressed.

JUNE 1, 1971

In the case of instruction completion, results are stored and the condition code is set as for normal instruction operation, although the result may be influenced by the exception which has occurred.

In the case of instruction termination, all, part, or none of the result may be stored. Therefore, the result data are unpredictable. The setting of the condition code, if called for, may also be unpredictable. In general, the results should not be used for further computation.

In the case of instruction suppression, the execution proceeds as if no operation were specified. Results are not stored, and the condition code is not changed.

9.1.2 SOURCE IDENTIFICATION

The five classes of interruptions are distinguished by the storage locations in which the old PSW is stored and from which the new PSW is fetched. The detailed causes are further distinguished by the interruption code of the old PSW. The bits of the interruption code are numbered 20-31, corresponding to their position in the PSW.

For I/O interruptions, additional information is provided by the contents of the channel status word, stored as part of the I/O interruption.

For machine-check interruptions (except for Read Direct Timeout), additional information is provided by the diagnostic procedure which is part of the interruption.

Interruption PSWs are assigned to an area of storage called the preferential-storage area. The location of the preferential-storage area is determined by a base address in each CE, thus, allowing different areas to be assigned for each CE. All addresses generated automatically by the CE are constructed using the current value of the preferential-storage base address.

The following table lists the main storage locations in the preferential-storage area. The addresses shown are relative to the preferential-storage base address.

JUNE 1, 1971

TABLE 9-I PREFERENTIAL/STORAGE ASSIGNMENTS

ADDRESS	LENGTH	PURPOSE	
0	0000 0000	double word	Initial program loading PSW
8	0000 1000	double word	Initial program loading CCW1
16	0001 0000	double word	Initial program loading CCW2
24	0001 1000	double word	External old PSW
32	0010 0000	double word	Supervisor call old PSW
40	0010 1000	double word	Program old PSW
48	0011 0000	double word	Machine old PSW
56	0011 1000	double word	Input/output old PSW
64	0100 0000	double word	Channel status word
72	0100 1000	word	Channel address word
76	0100 1100	word	Unused
80	0101 0000	word	Timer
84	0101 0100	word	Unused
88	0101 1000	double word	External new PSW
96	0110 0000	double word	Supervisor call new PSW
104	0110 1000	double word	Program new PSW
112	0111 0000	double word	Machine-check new PSW
120	0111 1000	double word	Input/output new PSW
128	1000 0000		Diagnostic logout area*

* The diagnostic logout area extends through byte location 511.

PROGRAMMING NOTE

Whenever a CE encounters a fetch, data or address check in, or fails to gain access to, the storage element (SE) containing its preferential-storage area (PSA), and provided the inhibit logout-stop bit is set off in its CCR, hardware automatically develops a new preferential-storage base address which references its alternate PSA (Chapter 8).

If the access is attempted because of an instruction fetch (including SUPERVISOR CALL), an operand fetch, or an interval timer update, an addressing exception is recognized and a program interruption is taken using the alternate PSA; or, a storage timeout, fetch, data or address check is recognized and a machine check interruption is taken using the alternate PSA. If the access is attempted in order to take a program interruption, input/output interruption, external interruption, or a supervisor-call interruption, the original interruption request is preserved and is taken using the alternate PSA. (See Chapter 12 for machine-check handling involving a split-logout.)

Whenever a CE encounters a fetch, data or address check in, or fails to gain access to, the SE containing its alternate PSA, an element check (CE-ELC) is issued and the CE performs a check stop (Chapter 12).

No indication of the change in the preferential-storage base address from the primary PSA to the alternate PSA is made by the hardware. The programmer may take cognizance of such a change in the interruption routines entered via the alternate PSA.

9.1.3 LOCATION DETERMINATION

For some interruptions, it is desirable to locate the instruction being interpreted when the interruption occurred. Since the instruction address in the old PSW designates the instruction to be executed next, it is necessary to know the length of the preceding instruction. This

JUNE 1, 1971

length is recorded in bit positions 32 and 33 of the PSW as the instruction-length code.

The instruction-length code is predictable only for program and supervisor-call interruptions. For I/O and external interruptions, the interruption is not caused by the last-interpreted instruction, and the code is not predictable for these instructions. For machine-check interruptions, the setting of the code may be affected by the malfunction and therefore, is unpredictable.

For the supervisor-call interruption the instruction-length code is 1, indicating the halfword length of SUPERVISOR CALL. For program interruptions, the codes 1, 2, and 3 indicate the instruction length in halfwords. The code 0 is reserved for those program interruptions where the length of the instruction is not available because of certain overlapping conditions in instruction fetching. In code-0 cases, the instruction address in the old PSW does not represent the next instruction address. Instruction-length code 0 can occur for a program interruption only when the interruption is caused by a protected or an unavailable data address. The following table shows the states of the instruction-length code.

TABLE 9-II INSTRUCTION LENGTH RECORDING

ILC	PSW 32-33	INSTRUCTION BITS 0-1	INSTRUCTION LENGTH	FORMAT
0	00		Not available	
1	01	00	One halfword	RR
2	10	01	Two halfwords	RX
2	10	10	Two halfwords	RS or SI
3	11	11	Three halfwords	SS

Programming Notes:

When a program interruption is due to an incorrect branch address, the location determined from the instruction address and instruction-length code is the branch address and not the location of the branch instruction.

When an interruption occurs while the CE is in the Wait state, the instruction-length code is always unpredictable.

When an interruption occurs during an execute operation, the instruction-length code does not reflect the length of the instruction executed, but is 2, the length of EXECUTE.

9.2 INPUT/OUTPUT INTERRUPTION

The I/O interruption provides a means by which the CE responds to signals from I/O devices.

A request for an I/O interruption may occur at any time, and more than one request may occur at the same time. The requests are preserved in the IOCE until accepted by the CE. Priority is established among requests so that only one interruption request is processed at a time.

An I/O interruption can occur only after execution of the current instruction is completed (with the exception of CSS, CVWL, RPSB, and DELAY) and while the CE is interruptable for the channel presenting the request. Channels are masked by system mask bits 0-6 and 16-19. Interruptions masked off remain pending.

JUNE 1, 1971

The I/O interruption causes the old PSW to be stored at location 56 in the preferential-storage area and causes the channel status word associated with the interruption to be stored at location 64. Subsequently a new PSW is loaded from location 120.

The interruption code in the old PSW identifies the channel and device causing the interruption in bits 20-23 and 24-31 respectively. The instruction-length code is unpredictable.

PROGRAMMING NOTE

Whenever the status bits of the channel status word (CSW) associated with the interruption indicate that the storing of the CSW was due to a channel control check, or an interface control check, additional information is provided by a diagnostic logout into the preferential-storage area. To prevent the contents of the logout area from being destroyed by an immediately following logout, all channels of the selected IOCE must be masked until the logout information has been acted upon or stored elsewhere for later use.

9.3 PROGRAM INTERRUPTION

Exceptions resulting from improper specification or use of instructions and data, or from conditions preventing an IOCE from accessing the preferential-storage area, cause a program interruption.

The current instruction is completed, terminated, or suppressed. Only one program interruption occurs for a given instruction and is identified in the old PSW. The occurrence of a program interruption does not preclude the simultaneous occurrence of other program-interruption causes. Which of several causes is identified may vary from one occasion to the next.

A program interruption can occur only when the corresponding mask bit, if any, is one. When the mask bit is zero, the interruption is ignored. Program interruptions do not remain pending. Program mask bits 36-39 permit masking of four of the interruption causes.

The program interruption causes the old PSW to be stored at location 40 in the preferential-storage area and a new PSW to be fetched from location 104.

The cause of the interruption is identified by interruption-code bits 24-31. The remainder of the interruption-code, bits 20-23 of the PSW, is made zero. The instruction-length code indicates the length of the preceding instruction in halfwords. For a few cases, the instruction length is not available. These cases are indicated by code 0.

If the new PSW for a program interruption has an unacceptable instruction address, another program interruption occurs. Since this second program interruption introduces the same unacceptable instruction address, a string of program interruptions is established which may be broken only by an external or I/O interruption. If these interruptions also have an unacceptable new PSW, new supervisor information must be introduced by initial program loading or by manual intervention.

A description of the individual program exceptions follows. The application of these rules to each class of instructions is further described in the applicable sections. Some of the exceptions listed may also occur in operations executed by I/O channels. In that event, the exception is indicated in the channel status word stored with the I/O interruption (as explained under "Input/Output Operations") Chapter 10.

JUNE 1, 1971

9.3.1 OPERATION EXCEPTION

When an operation code is not assigned an operation exception is recognized. The operation is suppressed.

The instruction-length code is 1, 2, or 3.

9.3.2 PRIVILEGED-OPERATION EXCEPTION

When a privileged instruction is encountered in the problem state, a privileged-operation exception is recognized. The operation is suppressed.

The instruction-length code is 1 or 2.

9.3.3 EXECUTE EXCEPTION

When the subject instruction of EXECUTE is another EXECUTE, an execute exception is recognized. The operation is suppressed.

The instruction-length code is 2.

9.3.4 PROTECTION EXCEPTION

When the storage key of an accessed location does not match the protection key in the PSW and either the storage key or the protection key is not zero, a protection exception is recognized.

The operation is suppressed on a store-protection violation, except in the case of STORE MULTIPLE, READ DIRECT, and variable-length operation which are terminated. Zeros may be stored in the destination location for variable-length operations. The operation is terminated on a fetch-violation.

The instruction-length code is 0, 2, or 3.

9.3.5 ADDRESSING EXCEPTION

When an address specifies any part of data, an instruction, or a control word outside the available storage for the particular installation, or outside the configured storage, or storage assigned by the storage address translator for a particular CE or IOCE, an addressing exception is recognized.

The operation is terminated for an invalid data address. Data in storage remain unchanged, except when designated by valid addresses. The operation is suppressed for an invalid instruction address.

The instruction-length code normally is 1, 2 or 3; but may be 0 in the case of a data address.

9.3.6 SPECIFICATION EXCEPTION

A specification exception is recognized when:

- 1) A data, instruction, or control-word address does not specify an integral boundary for the unit of information.
- 2) The R_1 field of an instruction specifies an odd register address for a pair of general registers which contain a 64 bit operand.

JUNE 1, 1971

- 3) A floating-point register address other than 0, 2, 4, or 6 is specified.
- 4) The multiplier or divisor in decimal arithmetic exceeds 15 digits and sign.
- 5) The first operand field is shorter than or equal to the second operand field in decimal multiplication or division.
- 6) The block address specified in SET STORAGE KEY or INSERT STORAGE KEY has the four low-order bits not all zero.
- 7) SET CONFIGURATION is attempted by a CE whose own scon bit is off in its configuration control register; or whose state bits are either set to One or Two.

Or a configuration mask has its scon-field bits all zeros, or does not reference at least one CE available to the system.

Or a selection mask has an IOCE selection bit set when two or more CE communications bits are set in the configuration mask.

- 8) The operand address of LOAD PREFERENTIAL-STORAGE BASE ADDRESS, or STORE PREFERENTIAL-STORAGE BASE ADDRESS does not have the two low-order bits both zero.

Or bit positions 8-19 of the operand addressed by LOAD PREFERENTIAL-STORAGE BASE ADDRESS, interpreted as the 12 high-order bits of a storage address, do not specify a location within a storage element provided in the ATR, which is configured to communicate with the CE executing this instruction, or specifies a DE within a 9020E system.

- 9) WRITE DIRECT does not select an element; or selects more than one element, or an invalid operation is specified by bits 8-11 of the I_2 field, or bit 15 of the I_2 field is a 1 on an IOCE selection.

Or READ DIRECT does not select a computing element; or selects more than one computing element.

- 10) The Maintenance Control Word is not on an integral boundary for a doubleword; it specifies an operation whose execution is not permitted for a CE in state Three, Two, or One.

- 11) An execution of SET ADDRESS TRANSLATOR is attempted by a CE whose own scon bit is off in its configuration control register; or whose state bits are either set to One or Two. Or a storage assignment mask references a particular storage module not available to a particular installation or assigns an SE module to other than positions 1-5 on a 9020E system, or assigns a DE module to other than positions 6-10 on a 9020E system.

- 12) A START I/O PROCESSOR (SIOP) does not select an IOCE, or selects more than one IOCE, or the first operand address does not reference a double-word storage location or a key of F (hexadecimal) was supplied to the IOCE-processor, or bit 15 of the I_2 field was not zero.

- 13) In CSS, CVWL, and RPSB instructions, the data chain address contained in bytes 5-7 of the doubleword following a 512 byte page is not on a doubleword boundary; or in CSS the R_1 field is not equal to 0 or 2; or in CSS or CVWL the R_2 field is not equal to 9.

JUNE 1, 1971

- 14) In the RPSB instruction, the NEXT OLD REFRESH MEMORY ADDR (ORMA) in GPR 4 or the NEXT NEW REFRESH MEMORY ADDR (NRMA) in GPR 5 is not initially on a doubleword boundary.

In all of these cases the operation is suppressed with the exception of CONVERT AND SORT SYMBOLS, SET ADDRESS TRANSLATOR, SET CONFIGURATION, REPACK SYMBOLS and CONVERT WEATHER LINES which may be either suppressed or terminated.

The instruction-length code is 1,2, or, 3.

9.3.7 DATA EXCEPTION

A data exception is recognized when:

- 1) The sign or digit codes of operands in decimal arithmetic or editing operations, or in CONVERT TO BINARY are incorrect.
- 2) Fields in decimal arithmetic overlap incorrectly.
- 3) The decimal multiplicand has too many high-order significant digits.

The operation is terminated. The instruction-length code is 2 or 3.

9.3.8 FIXED-POINT-OVERFLOW EXCEPTION

When a high-order carry occurs or high-order significant bits are lost in fixed-point add, subtract, shift, or sign-control operations, a fixed-point-overflow exception is recognized.

The operation is completed by ignoring the information placed outside the register. The interruption may be masked by PSW bit 36.

The instruction-length code is 1 or 2.

9.3.9 FIXED-POINT-DIVIDE EXCEPTION

A fixed-point-divide exception is recognized when a quotient exceeds the register size in fixed-point division, including division by zero, or the result of CONVERT TO BINARY exceeds 31 bits.

Division is suppressed. Conversion is completed by ignoring the information placed outside the register.

The instruction-length code is 1 or 2.

9.3.10 DECIMAL-OVERFLOW EXCEPTION

When the destination field is too small to contain the result field in a decimal operation, a decimal-overflow exception is recognized.

The operation is completed by ignoring the overflow information. The interruption may be masked by PSW bit 37.

The instruction-length code is 3.

JUNE 1, 1971

9.3.11 DECIMAL-DIVIDE EXCEPTION

When a quotient exceeds the specified data field size, a decimal-divide exception is recognized. The operation is suppressed.

The instruction-length code is 3.

9.3.12 EXPONENT-OVERFLOW EXCEPTION

When the result characteristic in floating-point addition, subtraction, multiplication, or division exceeds 127 and the result is not zero, an exponent-overflow exception is recognized. The operation is completed. The fraction is normalized, and the sign and fraction of the result remain correct. The result characteristic is made 128 smaller than the correct characteristic.

The instruction-length code is 1 or 2.

9.3.13 EXPONENT-UNDERFLOW EXCEPTION

When the result characteristic in floating-point addition, subtraction, multiplication, halving, or division is less than zero and the result fraction is not zero, an exponent-underflow exception is recognized. The operation is completed.

The setting of the exponent-underflow mask (PSW bit 38) affects the results of the operation. When the mask bit is zero, the sign, characteristic, and fraction are set to zero, making the result a true zero. When the mask bit is one, the fraction is normalized, the characteristic is made 128 larger than the correct characteristic, and the sign and fraction remain correct.

The instruction-length code is 1 or 2.

9.3.14 SIGNIFICANCE EXCEPTION

When the result of a floating-point addition or subtraction has an all-zero fraction, a significance exception is recognized.

The operation is completed. The interruption may be masked by PSW bit 39. The manner in which the operation is completed is determined by the mask bit.

The instruction-length code is 1 or 2.

9.3.15 FLOATING-POINT-DIVIDE EXCEPTION

When division by a floating-point number with zero fraction is attempted, a floating-point divide exception is recognized. The operation is suppressed.

The instruction-length code is 1 or 2.

9.3.16 PREFERENTIAL-STORAGE AREA LOCKOUT EXCEPTION

When an IOCE fails to gain access to the preferential storage area on a CAW fetch, on an I/O interruption, on a CSW store pertaining to an I/O instruction, or on a store during a logout operation, a preferential storage area lockout exception is recognized.

JUNE 1, 1971

If the CE is executing an I/O instruction, the current operation is terminated.

The instruction-length code is unpredictable.

For other than the above, the instruction operation is completed.

The instruction-length code is 1, 2 or 3.

9.3.17 SE/DE STOPPED EXCEPTION

When a CE attempts to gain access to a storage element which is in logout-stop status, a SE/DE stopped exception is recognized.

The operation is suppressed on an instruction fetch, and terminated on a data access.

The instruction-length code is 1, 2 or 3.

9.4 SUPERVISOR-CALL INTERRUPTION

The supervisor-call interruption occurs as a result of the execution of SUPERVISOR CALL.

The supervisor-call interruption causes the old PSW to be stored at location 32 in the preferential-storage area and a new PSW to be fetched from location 96.

The contents of bit positions 8-15 of the SUPERVISOR CALL become bits 24-31 in the interruption code of the old PSW. PSW bit positions 20-23 in the old PSW are made zero. The instruction-length code is 1, indicating the halfword length of SUPERVISOR CALL.

PROGRAMMING NOTE

The name "supervisor call" indicates that one of the major purposes of the interruption is the switching from Problem to Supervisor state. This major purpose does not preclude the use of this interruption for other types of status-switching.

The interruption code may be used to convey a message from the calling program to the supervisor.

When SUPERVISOR CALL is performed as the subject instruction of EXECUTE, the instruction-length code is 2.

9.5 EXTERNAL INTERRUPTION

The external interruption provides a means by which the CE responds to signals from the timer, from the INTERRUPT switch, from external units and in certain circumstances from itself.

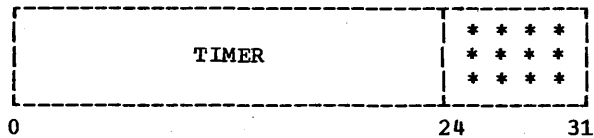
A request for an external interruption may occur at any time, and requests from different sources may occur at the same time. Requests are preserved until honored by the CE. All pending requests are presented simultaneously when an external interruption occurs. With the exception of abnormal condition requests preserved in the Diagnose Accessible Register (DAR), and IOCE-processor interrupts preserved in the Processor Interrupt Register (PIR), which are preserved until the register has been read and reset by a diagnostic operation, each request is presented only once. When several requests from one source are made before the interruption is taken, only one interruption occurs.

JUNE 1, 1971

With the exception of a request for a CE element check (ELC) presented to a CE which has the SCON bits set for the issuing CE and is in state Zero (with test switch off), One, or Two, an external interruption can occur only when system mask bit 7 is one and after the execution of the current instruction is completed (with the exception of DELAY, CONVERT AND SORT SYMBOLS, CONVERT WEATHER LINES, and REPACK SYMBOLS, which are terminated). The interruption causes the old PSW to be stored at location 24 in the preferential-storage area and a new PSW to be fetched from location 88. The source of the interruption is identified by interruption-code bits 20-31. The source of interruption-code bit 30 is further identified by the contents of the PIR. The source of interruption-code bit 31 is further identified by the contents of the Diagnose Accessible Register. The instruction-length code is unpredictable for external interruptions.

9.5.1 TIMER

A timer value changing from positive to negative causes an external interruption with PSW bit 24 set to one.



The timer occupies a 32-bit word at storage location 80 in the preferential-storage area. The contents of the timer are reduced by a one in bit position 21 and in bit position 23 every 1/60th of a second, as determined by the line frequency. The gross result is equivalent to reducing the timer by one in bit position 23 every 1/300th of a second. The full cycle time is approximately 15.5 hours.

The count is treated as a signed integer by following the rules for fixed-point arithmetic. The negative overflow, occurring as the timer is counted from a large negative number to a large positive number, is ignored. The interruption is initiated as the count proceeds from a positive number, including zero, to a negative number.

The timer is updated whenever access to storage permits. An updated timer value is normally available at the end of each instruction execution. Timer updating maybe omitted when I/O data transmission approaches the limit of storage capability.

The timer may have been updated several times after an interruption is initiated, before the CE is actually interrupted, depending upon external masking and instruction execution time. The timer remains unchanged when the CE is in the stopped state, or when the rate switch on the operator intervention panel is set to INSTRUCTION STEP. The timer value may be changed at any time by storing a new value in storage location 80 in the preferential-storage area.

In the DELAY, CSS, CVWL, and RPSB instructions, the timer is updated when required at various points within these instructions. After the usual timer update, control is immediately returned to the instruction, continuing its operation from where it left off. If, however, a timer update results in a change in timer value from positive to negative, an external interruption is generated causing the instruction to be terminated at its next internal interrupt exit point.

PROGRAMMING NOTE

JUNE 1, 1971

The timer in association with a program can serve both as a real-time clock and as an interval timer.

The timer is not updated whenever the CE is in state Zero, with the Disable Interval Timer switch on.

9.5.2 INTERRUPT SWITCH

Pressing the interrupt switch on the operator control section of the CE control panel causes an external interruption with bit 25 of the interruption code turned on, provided the CE is in state Zero, or state One; or provided the CE is in state Two or Three with the CE control panel interlock switch on.

PROGRAMMING NOTE

Pressing the interrupt switch on the System Console also causes an external interruption with bit 25 of the interruption code turned on, provided that the CE is selected by the CE Select switch and the CE is not in state Zero with the test switch on, and the System Console Interlock switch is on.

9.5.3 EXTERNAL SIGNAL

An external signal causes an external interruption, with the corresponding bit in the interruption code turned on.

The pattern presented in interruption code bits 20-31 depends upon the pattern received before the interruption is taken. Three categories of external signals are recognized: those due to execution of the CE DIRECT CONTROL instruction, those due to execution of the WRITE DIRECT instruction by the IOCE-processor, and those due to the detection of Abnormal conditions.

9.5.3.1 CE Direct Control Signals

Pairs of direct-control signal-out lines, originating in each CE, are connected into corresponding pairs of signal-in lines in each of up to three other CEs. The signal-in lines are designated CE(x) Write Direct and CE (x) Read Direct, where "x" identifies the source of the signal (Table 9-III). An interconnection of this kind allows one CE to interrupt another by executing the direct control instructions READ DIRECT and WRITE DIRECT. A similar interconnection of the direct-out lines of one CE to the direct-in lines of other CEs, and vice versa, provides an eight-bit data path (plus parity) for moving information between CEs.

A direct control interruption can occur only while the CE is interruptable for the CE presenting the request. Interruptions will be taken, provided PSW mask bit 7 is one, and the communications bit for the requesting CE is set on in the configuration control register of the receiving CE.

PROGRAMMING NOTE

The function of each of the direct-control signal-out lines is defined for IBM 9020D/E System use in Chapter 8.

9.5.3.2 Abnormal Condition Signals

The abnormal condition interruption provides a means by which the CE responds to certain hardware generated signals from a CE, IOCE, SE, DE, TCU, RCU, SCU, or PAM. In certain circumstances, a CE also responds to

JUNE 1, 1971

signals from itself. Three categories of abnormal condition signals are recognized: element check (ELC), out-of-tolerance check (OTC), and on-battery-signal (OBS).

A request for an abnormal condition interruption may occur at any time, and up to 29 requests may occur at the same time. The requests are preserved in the diagnose accessible register (DAR) until accepted by the CE.

An abnormal condition interruption can occur only while the CE is interruptable for the element presenting the request. The manner in which an interruption is handled depends upon the masking conditions prevailing in the CE, the type of element requesting the interruption, and upon the content of the configuration control registers in both the CE and the requesting element.

An abnormal condition interruption requested by an IOCE, DE, RCU, SE, SCU, TCU, PAM, or by a CE (own) OBS signal sets an identifying bit in DAR. These bits are maskable by the DAR mask bits 0, 2, 4, 6-26 or 31. (See Table 9-IV.) An unmasked bit set in DAR causes an interruption to be taken with bit 31 set in the old PSW, provided PSW mask bit 7 is one.

An abnormal condition interruption request from a CE (i.e., a CE-ELC or a CE (own) OTC) also sets an identifying bit in DAR which is maskable by the DAR mask (bits 27-30).

Any CE-ELC (except CE (own) OTC) bit set in DAR causes an interruption to be taken with bit 31 set in the old PSW, provided the scon bit for the requesting CE is already set in the CCR of the receiving CE, and either (a) or (b) is also satisfied.

- a) The receiving CE is in state Three, or in state Zero with the test switch on, and both the corresponding DAR mask bit and PSW mask bit 7 are ones.
- b) The receiving CE is in state One, Two, or Zero with the test switch off. Both the corresponding DAR mask bit and PSW bit 7 are ignored. The state field in the configuration control register of the receiving CE is set to Three (Chapter 8).

When the scon bit for the requesting CE is not set in the CCR of the receiving CE, any CE-ELC bit set in DAR causes an interruption to be taken regardless of the state of the receiving CE, provided the corresponding DAR mask bit and the PSW mask bit 7 are ones.

PROGRAMMING NOTE 1

Abnormal condition signals are out-of-tolerance check (OTC), on-battery signal (OBS), and element check (ELC).

The OTC signal indicates that a rise in temperature above the normal operating range has been detected.

The OBS check indicates that the element is switched from its normal power supply to its battery supply.

The ELC signal indicates that an element failure is detected. Although precise interpretation of an ELC may require diagnostic analysis, generally a CCR parity check, machine check, or a power supply check is indicated.

Definitions for the abnormal condition signals peculiar to a TCU, RCU, SCU or PAM are described in System Reference Library (SRL) publications provided for each unit. Those peculiar to a CE, IOCE, DE or SE are described in this manual (Chapter 12).

PROGRAMMING NOTE 2

A bit in the diagnose accessible register (DAR) is unconditionally set on the receipt of an abnormal condition interruption request. Once set, a bit remains in DAR until reset by the execution of the store DAR kernel initiated by the DIAGNOSE instruction. The entire content of DAR is then obtained without regard to the DAR masking.

If the condition which requested the interruption prevails after DAR reset, the bit is again set on.

The DAR mask is set by the execution of the set DAR mask kernel initiated by the DIAGNOSE instruction. Once set, a DAR mask bit remains on until altered by another set operation.

9.5.3.3 IOCE Direct Control Signals

The IOCE Direct Control signals provide the means by which the IOCE-processor can request an external interruption. A request for an external interruption from an IOCE-processor may occur at any time, and requests from different IOCE-processors may occur at the same time. A processor interruption line is provided to each CE from each IOCE-processor in the system. Requests are preserved in the processor interruption register (PIR) until removed by a diagnostic operation (STORE PIR kernel). The interruption can occur only when the CE's PSW bit 7 is one and after the execution of the current instruction is completed (with the exception of DELAY, CONVERT AND SORT SYMBOLS, CONVERT WEATHER LINES and REPACK SYMBOLS which are terminated). The interruption causes the old PSW to be stored at PSA location 24 and a new PSW to be fetched from PSA location 88. The type of the interruption is identified by interruption-code bit 30. The source of the interruption is identified by the contents of the processor interruption register (PIR) as shown below.

PROCESSOR INTERRUPTION REGISTER

<u>Interruption Source Identification</u>	<u>Processor Interruption Register Code Bit</u>
IOCE-Processor - 1	28
IOCE-Processor - 2	29
IOCE-Processor - 3	30

9.6 MACHINE-CHECK INTERRUPTION

The machine-check interruption provides a means for recovery from, and fault location of CE and IOCE machine malfunctions.

9.6.1 CE MACHINE CHECK

When the machine-check mask bit is one, occurrence of a machine check due to a CE malfunction terminates the current instruction, issues a signal on the element check (ELC) out-line, delays approximately 11 ms., initiates a CE diagnostic procedure, and subsequently causes the machine-check interruption. Occurrence of a machine check due to a read direct timeout terminates the current instruction and subsequently causes the machine-check interruption. No ELC is issued, and no logout takes place.

The state of the CE is logged out into the preferential-storage area, starting with location 128 and extending through as many words as the CE requires. The old PSW is stored at location 48 in the preferential-storage area with an interruption code of Zero. (An interruption code

JUNE 1, 1971

of four is stored to identify a Read Direct timeout.) The new PSW is fetched from location 112. Proper execution of these steps depends upon the nature of the machine check.

When the machine-check mask bit is zero, an attempt is made to complete the current instruction upon the occurrence of a machine check, and to proceed with the next sequential instruction. The element check (ELC) out-line is signaled upon occurrence of the CE malfunction machine-check condition. The diagnostic procedure and interruption occur upon the machine-check mask bit being changed to one.

A change in the machine-check mask bit due to the loading of a new PSW results in a change in the treatment of machine checks. Depending upon the nature of a machine check, the old treatment may still be in force for several cycles.

Following emergency power turn-off and turn-on, or system reset, incorrect parity may exist in storage or registers. Unless new information is loaded a machine check may occur erroneously. Once storage and registers are cleared a machine check can be caused only by machine malfunction and never by data or instructions.

PROGRAMMING NOTE

The indication of the check condition which caused the machine-check interruption is preserved in the CE check registers until reset by either a Logout or a Diagnose operation (Chapter 12).

9.6.2 IOCE MACHINE CHECK

When the machine-check mask bit is one, occurrence of a machine check due to an IOCE malfunction issues a signal on the IOCE element check (ELC) out-line, initiates an IOCE diagnostic procedure, and subsequently causes the machine-check interruption. The current instruction is terminated in the controlling CE if it is an I/O instruction, otherwise the diagnostic procedure and the interruption take place after the current instruction interpretation is finished.

The state of the IOCE is logged out into the controlling CE's preferential-storage area, starting with location 324 and extending through as many words as the IOCE requires. The old PSW is stored at location 48 in the preferential-storage area with an interruption code of 1, 2 or 3 to identify the particular IOCE. The new PSW is fetched from location 112. Proper execution of these steps depends upon the nature of the machine check.

When the machine-check mask bit is zero, the IOCE waits in check-stop status. The IOCE element check (ELC) out-line is signaled upon occurrence of the machine-check condition. The IOCE diagnostic procedure and interruption occur upon the machine-check mask bit being changed to one.

PROGRAMMING NOTE

The indication of the check condition which caused the machine-check interruption is preserved in the IOCE check registers until reset during any IOCE logout (Chapter 12).

9.7 PRIORITY OF INTERRUPTIONS

During execution of an instruction, several interruption-causing events may occur simultaneously. The instruction may give rise to a program interruption, an external interruption may occur, a machine

JUNE 1, 1971

check may occur, and an I/O interruption request may be made. Instead of the program interruption, a supervisor-call interruption might occur, however, both can not occur, since these two interruptions are mutually exclusive. Simultaneous interruption requests are honored in a predetermined order.

The machine-check interruption has highest priority. When it occurs the current operation is terminated. Program and supervisor-call interruptions that would have occurred as a result of the current instruction are eliminated. Every reasonable attempt is made to limit the side-effects of a machine check. Normally, I/O and external interruptions, as well as the progress of the I/O data transfer and the updating of the timer, remain unaffected.

When no machine check occurs, the program interruption or supervisor-call interruption is taken first, the external interruption is taken next, and the I/O interruption is taken last. The action consists of storing the old PSW and fetching the new PSW belonging to the interruption first taken. This new PSW is subsequently stored without any instruction execution and the next interruption PSW is fetched. This storing and fetching continues until no more interruptions are to be serviced. The external and I/O interruptions are taken only if the immediately preceding PSW indicates that the CE is interruptible for these causes.

Instruction execution is resumed using the last-fetched PSW. The order of executing interruption subroutines is therefore the reverse of the order in which the PSWs are fetched.

The interruption code of a new PSW is not loaded since a new interruption code is always stored. The instruction-length code in a new PSW is similarly ignored, since it is unpredictable for all interruptions other than program or supervisor call. The protection key of a new PSW is stored unchanged.

PROGRAMMING NOTE

When interruption sources are not masked off, the order of priority in handling the interruption subroutines is machine check, I/O, external, and program or supervisor call. This order can be changed to some extent by masking. The priority rule applies to interruption requests made simultaneously. An interruption request made after some interruptions have already been taken is honored according to the priority prevailing at the moment of request.

9.8 INTERRUPTION EXCEPTIONS

The instruction address in a new PSW is not tested for availability or resolution as the PSW is fetched during an interruption. However, an unavailable or odd instruction address is detected as soon as the instruction address is used to fetch an instruction. These exceptions are described in the section on normal sequential operation.

If the new PSW for the program interruption has an unacceptable instruction address, another program interruption occurs. Since this second program interruption introduces the same unacceptable instruction address, a string of program interruptions is established. This string may be broken by an external or I/O interruption. If these interruptions also have an unacceptable new PSW, new supervisor information must be introduced by initial program loading or by external or manual intervention.

JUNE 1, 1971

TABLE 9-III INTERRUPTION ACTION

INTERRUPTION SOURCE IDENTIFICATION	INTERRUPTION CODE PSW BITS 20-31	MASK BITS	ILC SET	INSTRUCTION EXECUTION
<u>Input/Output</u> (old PSW 56, new PSW 120, priority 4)				
Multiplexor channel A	0000 aaaaaaaaa	0	x	completed
Selector channel 1A	0001 aaaaaaaaa	1	x	completed
Selector channel 2A	0010 aaaaaaaaa	2	x	completed
Selector channel 3A	0011 aaaaaaaaa	3	x	completed
Multiplexor channel B	0100 aaaaaaaaa	4	x	completed
Selector channel 1B	0101 aaaaaaaaa	5	x	completed
Selector channel 2B	0110 aaaaaaaaa	6	x	completed
Selector channel 3B	0111 aaaaaaaaa	16	x	completed
Multiplexor channel C	1000 aaaaaaaaa	17	x	completed
Selector channel 1C	1001 aaaaaaaaa	18	x	completed
Selector channel 2C	1010 aaaaaaaaa	19	x	completed
<u>Program</u> (old PSW 40, new PSW 104, priority 2)				
Operation	0000 00000001		1,2,3	suppressed
Privileged operation	0000 00000010		1,2	suppressed
Execute	0000 00000011		2	suppressed
Protection	0000 00000100		0,2,3	suppressed/ terminated
Addressing	0000 00000101		0,1,2,3	suppressed/ terminated
Specification	0000 00000110		1,2,3	suppressed/ terminated
Data	0000 00000111		2,3	terminated
Fixed-point overflow	0000 00001000	36	1,2	completed
Fixed-point divide	0000 00001001		1,2	suppressed/ completed
Decimal overflow	0000 00001010	37	3	completed
Decimal divide	0000 00001011		3	suppressed
Exponent overflow	0000 00001100		1,2	completed
Exponent underflow	0000 00001101	38	1,2	completed
Significance	0000 00001110	39	1,2	completed
Floating-point divide	0000 00001111		1,2	suppressed
IOCE-3 PSA lockout	0000 00010000		1,2,3	terminated/ completed
IOCE-2 PSA lockout	0000 00100000		1,2,3	terminated/ completed
IOCE-1 PSA lockout	0000 01000000		1,2,3	terminated/ completed
SE/DE stopped	0000 10000000		1,2,3	suppressed/ terminated

JUNE 1, 1971

Interruption Action (continued)					
INTERRUPTION SOURCE IDENTIFICATION	INTERRUPTION CODE PSW BITS 20-31	MASK BITS	ILC SET	INSTRUCTION EXECUTION	
<u>Supervisor Call</u> (old PSW 32, new PSW 96, priority 2)					
Instruction bits	0000 rrrrrrrr		1	completed	
<u>External</u> (old PSW 24, new PSW 88, priority 3)					
DAR	xxxx xxxxxxxx1	7	x	completed	
PIR	xxxx xxxxxxx1x	7	x	completed	
CE4 Write Direct	xxxx xxxxx1xx	7	x	completed	
CE4 Read Direct	xxxx xxxx1xxx	7	x	completed	
CE3 Write Direct	xxxx xxx1xxxx	7	x	completed	
CE3 Read Direct	xxxx xx1xxxxx	7	x	completed	
Interrupt switch	xxxx x1xxxxxx	7	x	completed	
Timer	xxxx 1xxxxxxx	7	x	completed	
CE2 Write Direct	xxx1 xxxxxxxx	7	x	completed	
CE2 Read Direct	xx1x xxxxxxxx	7	x	completed	
CE1 Write Direct	x1xx xxxxxxxx	7	x	completed	
CE1 Read Direct	1xxx xxxxxxxx	7	x	completed	
<u>Machine Check</u> (old PSW 48, new PSW 112, priority 1)					
CE malfunction	0000 00000000	13	x	terminated	
IOCE-1 malfunction	0000 00000001	13	x	completed	
IOCE-2 malfunction	0000 00000010	13	x	/terminated completed	
IOCE-3 malfunction	0000 00000011	13	x	/terminated completed	
Read Direct timeout	0000 00000100	13	x	/terminated terminated	
<u>Legend</u>					
a	Device address bits				
r	Bits of R ₁ and R ₂ fields of SUPERVISOR CALL				
X	Unpredictable				
DAR	Diagnose Accessible Register				
PSA	Preferential Storage Area				
PIR	Processor Interruption Register				

JUNE 1, 1971

TABLE 9-IV ABNORMAL CONDITION INTERRUPTION ACTION

INTERRUPTION SOURCE IDENTIFICATION		DIAGNOSE ACCESSIBLE REGISTER CODE BIT	DIAGNOSE ACCESSIBLE REGISTER MASK BIT
9020D	9020E		
IOCE-1a*	IOCE-1a*	0	0
IOCE-1b	IOCE-1b	1	0
IOCE-2a*	IOCE-2a*	2	2
IOCE-2b	IOCE-2b	3	2
IOCE-3a*	IOCE-3a*	4	4
IOCE-3b	IOCE-3b	5	4
SE-1 ELC	SE-1 ELC	6	6
SE-2 ELC	SE-2 ELC	7	7
SE-3 ELC	SE-3 ELC	8	8
SE-4 ELC	SE-4 ELC	9	9
SE-5 ELC	SE-5 ELC	10	10
SE-6 ELC	DE-1 ELC	11	11
SE-7 ELC	DE-2 ELC	12	12
SE-8 ELC	DE-3 ELC	13	13
SE-9 ELC	DE-4 ELC	14	14
SE-10 ELC	DE-5 ELC	15	15
SPARE	SPARE	16	16
SPARE	SPARE	17	17
PAM-1 ELC	RCU-1 ELC	18	18
PAM-2 ELC	RCU-2 ELC	19	19
PAM-3 ELC	SPARE	20	20
TCU-1 ELC	TCU-1 ELC	21	21
TCU-2 ELC	TCU-2 ELC	22	22
TCU-3 ELC	TCU-3 ELC	23	23
SCU-1 ELC	SCU-1 ELC	24	24
SCU-2 ELC	SCU-2 ELC	25	25
SCU-3 ELC	SCU-3 ELC	26	26
CE-1 ELC**	CE-1 ELC**	27	27
CE-2 ELC**	CE-2 ELC**	28	28
CE-3 ELC**	CE-3 ELC**	29	29
CE-4 ELC**	CE-4 ELC**	30	30
CE (own)OBS	CE (own)OBS	31	31
Legend			
ELC - Element Check			
OTC - Out of tolerance Check			
OBS - On Battery Supply			
RCU - Reconfiguration Control Unit			

JUNE 1, 1971

NOTE

* External signals from IOCEs are encoded as a two-bit field, designated a and b.

Interruption Source <u>identification</u>	Bit Setting	
	a	b
Normal Operation	0	0
OBS	0	1
OTC	1	0
ELC	1	1

** The CE ELC bits 27-30 are other CE ELC or CE (own) OTC.

JUNE 1, 1971

CHAPTER 10: INPUT/OUTPUT OPERATIONS

Transfer of information to and from storage, other than to or from the computing element (CE), input/output control element (IOCE-processor), or via the direct control path, is referred to as an input and output operation. An input/output (I/O) operation involves the use of an input/output device. Input/output devices perform I/O operations under control of control units, which are attached to the CE by means of input/output control element (IOCE) channels.

This portion of the manual describes the control of I/O devices by the channels and the CE from the programming point of view. The programmed control procedures apply to all I/O operations, and are independent of the type of I/O device, its speed, or its mode of operation.

Programming details peculiar to individual devices are described in System Reference Library (SRL) publications provided for each device.

10.1 ATTACHMENT OF INPUT/OUTPUT DEVICES

10.1.1 INPUT/OUTPUT DEVICES

Input/output devices provide external storage and a means of communication between data-processing systems or between a system and its environment. Input/output devices include such equipment as card read punches, magnetic tape units, printer-keyboard devices, page printers, peripheral adapter modules, data adapter units, storage control units and channel-to-channel adapters.

Most types of I/O devices, such as printers, card equipment, disks or tape devices, deal directly with external documents, and these devices are physically distinguishable and identifiable. Other types such as data adapter units and channel-to-channel adapters consist only of electronic equipment and do not directly handle physical recording media. The Peripheral Adapter Module is available to handle transmission of information between the data processing system and a remote station, and its input and output are signals on a transmission line. Furthermore, the equipment in this case may be time-shared for a number of concurrent operations, and it is denoted as a particular I/O device only during the time period associated with the operation on the corresponding remote station.

Input/output devices may be accessible from one or more channels. Devices that are accessible from one channel normally are attached only to one control unit. A device can be made accessible to two or more channels by switching it between two or more control units, each attached to a different channel, or by switching the control unit between two or more channels.

10.1.2 CONTROL UNITS

The control unit provides the logical capabilities necessary to operate and control an I/O device, and adapts the characteristics of each device to the standard form of control provided by the channel.

All communication between the control unit and the channel takes place over the I/O interface. The control unit accepts control signals

JUNE 1, 1971

from the channel, controls the timing of data transfer over the I/O interface, and provides indications concerning the status of the device.

The I/O interface provides an information format and a signal sequence that is common to all I/O devices. The interface consists of a set of lines that can connect a number of control units to the channel. Except for the signal used to establish priority among control units, all communications to and from the channel occur over a common bus, and any signal provided by the channel is available to all control units. At any one instant, however, only one control unit is logically connected to the channel.

The selection of a control unit for communication with the channel is controlled by a signal from the channel that passes serially through all control units and permits, sequentially, each control unit to respond to the signals provided by the channel. A control unit remains logically connected on the interface until it has transferred the information it needs or has, or until the channel signals it to disconnect, whichever occurs earlier.

The I/O device attached to the control unit may be designed to perform only certain limited operations, or it may perform many different operations. A typical operation is moving the recording medium and recording data. To accomplish these functions, the device needs detailed signal sequences peculiar to the type of device. The control unit decodes the commands received from the channel, interprets them for the particular type of device, and provides the signal sequence required for execution of the operation.

A control unit may be housed separately, housed within an IOCE or CE, or it may be physically and logically integral with the I/O device. In the case of most electromechanical devices, a well-defined interface exists between the device and the control unit because of the difference in the type of equipment the control unit and the device contain. These electromechanical devices often are of a type where only one device of a group is required to operate at a time (magnetic tape drives), and the control unit is shared among a number of I/O devices.

From the user's point of view, most functions performed by the control unit can be merged with those performed by the I/O device. Therefore, this manual normally does not make specific mention of the control unit function; the execution of I/O operations is described as if the I/O devices communicated directly with the channel. Reference is made to the control unit only when emphasizing a function performed by it or when sharing of the control unit among a number of devices affects the execution of I/O operations.

10.1.3 CHANNELS

The channel directs the flow of information between I/O devices and storage. It relieves the CE of the task of communicating directly with the devices and permits data processing to proceed concurrently with I/O operations.

The channel provides a standard interface for connecting different types of I/O devices to the CE and to storage. It accepts control information from the CE in the format supplied by the program and changes it into a sequence of signals acceptable to a control unit. After the operation with the device has been initiated, the CE is released for other work and the channel assembles or disassembles data and synchronizes the transfer of data bytes over the interface with storage cycles. To accomplish this, the channel maintains and updates an address and a count that describe the destination or source of data in storage. Similarly, when an I/O device provides signals that should

JUNE 1, 1971

be brought to the attention of the program, the channel converts the signals to a format compatible to that used in the CE.

The channel contains all the common facilities for the control of I/O operations. I/O operations are completely overlapped with the activity in the CE. The only storage cycles required during I/O operations are those needed to transfer data and control information to or from the final locations in storage. These cycles do not interfere with the CE program, except when both the CE and the channel concurrently attempt to refer to the same main storage element.

10.1.3.1 Modes of Operation

Data transfer between storage and an I/O device occurs in one of two modes: burst or multiplex.

In burst mode, the I/O device monopolizes the I/O interface and stays logically connected to the channel for the transfer of a burst of information. No other device can communicate over the interface during the time a burst is transferred. The burst can consist of a few bytes, a whole block of data, or a sequence of blocks with associated control and status information.

In multiplex mode, the facilities in the channel may be shared by a number of concurrently operating I/O devices. The multiplex mode causes all I/O operations to be split into short intervals of time during which only a segment of information is transferred over the interface. During an interval, only one device is logically connected to the channel. The intervals associated with the concurrent operation of multiple I/O devices are sequenced in response to demands from the devices. The channel controls are occupied with any one operation only for the duration of time required to transfer a segment of information. The segment can consist of a single byte of data, a few bytes of data, a status report from the device, or a control sequence used for initiation of a new operation.

A short burst of data can be handled in either multiplex or burst mode. The distinction between a short burst occurring in the multiplex mode and an operation in the burst mode is in the length of the bursts. Whenever the burst causes the device to be connected to the channel for more than approximately 100 microseconds, the channel is considered to be operating in the burst mode.

Operation in burst and multiplex modes is differentiated because of the way the channels respond to I/O instructions. A channel operating in the burst mode appears busy to new I/O instructions, whereas a channel operating in the multiplex mode is available for the initiation of new operations. A channel that can operate in both modes determines its mode of operation by time-out. If such a channel happens to be communicating with an I/O device at the instant a new I/O instruction is issued, action upon the instruction is delayed until the current mode of operation is established. New I/O operations are initiated only after the channel has serviced all outstanding requests for data transfer for previously initiated operations.

10.1.3.2 Types of Channels

A system is equipped with two types of channels: selector and multiplexor. Channels are classified according to the modes of operation they can sustain.

The channel facilities required for sustaining a single I/O operation are termed a subchannel. The subchannel consists of the channel storage used for recording the addresses, count, and any status and control

JUNE 1, 1971

information associated with the I/O operation. The mode in which a channel can operate depends upon whether it has one or more subchannels.

The selector channel has one subchannel and always forces the I/O device to transfer data in the burst mode. The burst extends over the whole block of data, or, when command chaining is specified, over the whole sequence of blocks. The selector channel cannot perform any multiplexing and therefore can be involved in only one data-transfer operation at a time. In the meantime, other I/O devices attached to the channel can be executing previously initiated operations that do not involve communication with the channel, such as rewinding tape to load point. When the selector channel is not executing an operation or a chain of operations and is not processing an interruption, it monitors the attached devices for status information.

The multiplexor channel contains multiple subchannels and can operate in either multiplex or burst mode. The mode of data transfer is determined by the I/O device and can change at any time. An operation on any one subchannel can occur partially in the multiplex and partially in the burst mode.

When the multiplexor channel operates in the multiplex mode it can sustain concurrently one I/O operation per subchannel, provided that the total load on the channel does not exceed its throughput capacity. Except for those aspects of communication that pertain to the physical channel, each subchannel appears to the program as an independent selector channel. When the multiplexor channel is not servicing an I/O device, it monitors its devices for data and for interruption conditions.

When the multiplexor channel operates in the burst mode, the subchannel associated with the burst operation monopolizes all channel facilities and appears to the program as a single selector channel. The remaining subchannels on the multiplexor must remain dormant and cannot respond to devices until the burst is completed.

10.1.4 SYSTEM OPERATION

Input/output operations are initiated and controlled by information with three types of formats: instructions, commands, and orders. Instructions are decoded and executed by the CE, and are part of the CE program. Commands are decoded and executed by the IOCE channels, and initiate I/O operations such as reading and writing. Instructions are fetched from main storage and commands may be fetched from either main storage or the IOCE's Mach storage.

Functions peculiar to a device, such as rewinding tape or spacing a line on the printer are specified by means of orders. Orders are decoded and executed by I/O devices. The control information specifying an order may appear in the modifier bits of a Control command code, may be transferred to the device as data during a control or write operation, or may be made available to the device by other means.

The CE program initiates I/O operations with the instruction START I/O. This instruction identifies the channel and device and causes the IOCE channel to fetch the channel address word (CAW) from a fixed location in main storage. The CAW contains the protection key and designates the location in storage from which the channel subsequently fetches the first channel command word (CCW). The CCW specifies the command to be executed and the storage area, if any, to be used.

NOTE: Either main storage or the IOCE MACH storage may be used for I/O data transfers or CCW fetching. Selection of the storage to be used is under program control. (See Appendix H).

JUNE 1, 1971

If the channel is not operating in burst mode and if the subchannel associated with the addressed I/O device is available, the channel attempts to select the device by sending the address of the device to all control units attached to the channel. The control unit that recognizes the address connects itself logically to the channel and responds to the selection by returning the address. The channel subsequently sends the command code part of the CCW over the interface, and the device responds with a status byte indicating whether it can execute the command.

At this time, the execution of START I/O is completed. The results of the attempt to initiate the execution of the command are indicated by setting the condition code in the program status word (PSW), and, under certain conditions, by storing pertinent information in the channel status word (CSW).

If the operation is initiated at the device and its execution involves transfer of data, the subchannel is set up to respond to service requests from the device and assumes further control of the operation. In the case of operations that do not require any data to be transferred to or from the device, the device may signal the end of the operation immediately upon receipt of the command code.

An I/O operation may involve transfer of data to one storage area, designated by a single CCW, or to a number of noncontiguous storage areas. In the latter case, a list of CCWs is used for execution of the I/O operation, each CCW designating a contiguous storage area, and the CCWs are said to be coupled by data chaining. Data chaining is specified by a flag in the CCW and causes the channel to fetch another CCW upon the exhaustion or filling of the storage area designated by the current CCW. The storage area designated by a CCW fetched on data chaining pertains to the I/O operation already in progress at the I/O device, and the I/O device is not notified when a new CCW is fetched.

Termination of the I/O operation normally is indicated by two types of conditions: Channel End and Device End. The Channel End condition indicates that the I/O device has received or provided all information associated with the operation and no longer needs channel facilities. The Device End signal indicates that the I/O device has terminated execution of the operation. The Device End condition can occur concurrently with the Channel End condition or later.

Operations that keep the control unit busy after releasing channel facilities may, under certain conditions, cause a third type of signal. This signal, called Control Unit End, may occur only after Channel End and indicates that the control unit has become available for initiation of another operation.

The conditions signaling the termination of an I/O operation can be brought to the attention of the program by I/O interruptions or, when the channel is masked, by programmed interrogation of the I/O device. In either case, these conditions cause storing the CSW, which contains additional information concerning the execution of the operation. At the time the Channel End condition is generated, the channel identifies to the program the last CCW used and provides its residual byte count, thus indicating the extent of storage used. Both the channel and the device can provide indications of unusual conditions with Channel End. The Control Unit End and Device End conditions can be accompanied by error indications from the device.

Facilities are provided for the program to initiate execution of a chain of commands with a single START I/O. When the chaining flags in the current CCW specify command chaining and no unusual conditions have been detected in the operation, the receipt of the Device End signal causes the channel to fetch a new CCW and to initiate a new command at

JUNE 1, 1971

the device. A chained command is initiated by means of the same sequence of signals over the I/O interface as the first command specified by START I/O. The ending signals occurring at the termination of an operation caused by a CCW specifying command chaining are not made available to the program when another operation is initiated by the command chaining; the channel continues execution of the channel program. If, however, an unusual condition has been detected, the ending signals cause suppression of command chaining and termination of the channel program.

Conditions that initiate I/O interruptions are asynchronous to activity in the CE, and more than one condition can occur at the same time. The channel and the CE establish priority among the conditions so that only one interruption request is processed at a time. The conditions are preserved in the I/O devices and subchannels until accepted by the CE.

Execution of an I/O operation or chain of operations thus involves up to five levels of participation:

1. The CE is busy for the duration of execution of START I/O, which lasts at most until the addressed I/O device responds to the first command.
2. The channel is busy with the execution during transfers of data to or from the device.
3. The subchannel is busy with the execution from the initiation of the operation at the I/O device until the Channel End condition for the last operation of the command chain is accepted by the CE.
4. The control unit may remain busy after the subchannel has been released and may generate the Control Unit End condition when it becomes free.
5. The I/O device is busy from the initiation of the first command until the Device End condition associated with the last operation is cleared by the CE.

A pending Device End condition causes the associated device to appear busy, but does not affect the state of any other part of the system. A pending Control Unit End blocks communications through the control unit to any device attached to it, and a pending Channel End normally blocks all communications through the subchannel.

10.1.5 COMPATIBILITY OF OPERATION

The organization of the I/O system provides for a uniform method of controlling I/O operations. Channels are provided with different data-transfer capabilities, and an I/O device designed to transfer data only at a specific rate (a magnetic tape drive) can operate only on a channel that can accommodate at least this data rate.

The data rate a channel can accommodate depends also on the way the I/O operation is programmed. The channel can sustain its highest data rate when no data chaining is specified. Data chaining reduces the maximum allowable rate, and the extent of the reduction depends on the frequency at which new CCWs are fetched and on the address resolution of the first byte, i.e., the location of the first byte relative to a word boundary in the new area. Furthermore, since in most instances the channel may share main storage with the CE and other channels, activity in the rest of the system affects the accessibility of main storage and, hence, the instantaneous load the channel can sustain.

JUNE 1, 1971

In view of the dependence of channel capacity on programming and on activity in the rest of the system, an evaluation of the ability of a specific I/O configuration to function concurrently must be based on a consideration of both the data rate and the way the I/O operations are programmed.

10.2 CONTROL OF INPUT/OUTPUT DEVICES

The CE controls I/O operations by means of five I/O instructions:

START I/O,
TEST I/O,
HALT I/O,
TEST CHANNEL, and
SET PCI.

The instruction TEST CHANNEL addresses a channel; it does not address an I/O device.

10.2.1 INPUT/OUTPUT DEVICE ADDRESSING

An I/O device and the associated access path are designated by an I/O address. The I/O address is a 16-bit binary number and consists of two parts: a channel address in the eight high-order bit positions and a device address in the eight low-order bit positions.

The channel address specifies the IOCE and channel to which the instruction applies while the device address identifies the particular I/O device on that channel. Any number in the range 0-255 can be used as a device address. Facilities are thus provided for addressing up to 256 devices per channel. The assignment of I/O addresses is listed in the following table.

TABLE 10-I INPUT/OUTPUT ADDRESS ASSIGNMENTS

ADDRESS	ASSIGNMENT
0000 0000 xxxx xxxx	Devices on multiplexor channel A
0000 0001 xxxx xxxx	Devices on selector channel 1A
0000 0010 xxxx xxxx	Devices on selector channel 2A
0000 0011 xxxx xxxx	Devices on selector channel 3A
0000 0100 xxxx xxxx	Devices on multiplexor channel B
0000 0101 xxxx xxxx	Devices on selector channel 1B
0000 0110 xxxx xxxx	Devices on selector channel 2B
0000 0111 xxxx xxxx	Devices on selector channel 3B
0000 1000 xxxx xxxx	Devices on multiplexor channel C
0000 1001 xxxx xxxx	Devices on selector channel 1C
0000 1010 xxxx xxxx	Devices on selector channel 2C
0000 1011 0000 0000 to	Invalid
1111 1111 1111 1111	

On the multiplexor channel the device address identifies the subchannel as well as the I/O device.

For devices sharing a control unit (e.g., magnetic tape drives) the high-order bit positions of the device address identify the control unit. The number of bit positions in the common field depends upon the

JUNE 1, 1971

number of devices attached. When 16 or less devices are attached, at least four high-order bits of the addresses are common. Control units with more than 16 devices may be assigned noncontiguous sets of 16 addresses. The low-order bit positions of the address identify the device on the control unit.

When the control unit is designed to accommodate less devices than can be addressed with the common field, the control unit does not recognize the address not assigned to it. As an example, if a control unit is designed to control devices having only bits 0000-1001 in the low-order positions of the address, it does not recognize addresses containing 1010-1111 in these bit positions. However, when a control unit has less than 16 devices installed, but is designed to accommodate 16 or more, it may respond to all 16 addresses and may indicate Unit Check for the invalid addresses.

Input/output devices accessible through more than one channel have a distinct address for each path of communications. This address identifies the channel, the subchannel, and the control unit. For sets of devices connected to two or more control units, the portion of the address identifying the device on the control unit is fixed, and does not depend on the path of communications.

When two or more devices which are assigned identical addresses share a channel, they share a common subchannel. Any of the devices may initiate action in the subchannel, but the subchannel can only initiate action at the device with highest priority on the interface cable. This priority may be fixed (RCU, DAU) or may be alterable under program control (Pass Address command used with PAM).

Except for the rules described, the assignment of device addresses is arbitrary. The assignment is made at the time of installation, and the addresses normally remain fixed thereafter.

10.2.2 INSTRUCTION EXCEPTION HANDLING

Before the channel is signaled to execute an I/O instruction, the instruction is tested for validity by the CE. Exceptional conditions detected at this time cause a program interruption. When the interruption occurs, the current PSW is stored as the old PSW and is replaced by a new PSW. The interruption code in the old PSW identifies the cause of the interruption.

The following exception may cause a program interruption.

Privileged Operation: An I/O instruction is encountered when the CE is in the problem state. The instruction is suppressed before the channel has been signaled to execute it. The CSW, the condition code in the PSW, and the state of the addressed subchannel and I/O device remain unchanged.

10.2.3 STATES OF THE INPUT/OUTPUT SYSTEM

The state of the I/O system identified by an I/O address depends on the collective state of the channel, subchannel, and I/O device. Each of these components of the I/O system can have up to four states, as far as the response to an I/O instruction is concerned. These states are listed in the following table. The name of the state is followed by its abbreviation and a brief definition of the state.

JUNE 1, 1971

TABLE 10-II INPUT/OUTPUT STATES

NAME	ABBREVIATION AND DEFINITION	
<u>Channel</u>		
Available	A	None of the following states
Interruption pending	I	Interruption immediately available from channel
Working	W	Channel operating in burst mode
Not operational	N	Channel not operational
<u>Subchannel</u>		
Available	A	None of the following states
Interruption pending	I	Information for CSW available in subchannel
Working	W	Subchannel executing an operation
Not operational	N	Subchannel not operational
<u>I/O Device</u>		
Available	A	None of the following states
Interruption pending	I	Interruption condition pending in device
Working	W	Device executing an operation
Not operational	N	Device not operational

A channel, subchannel, or I/O device that is available, that contains a pending interruption condition, or that is working, is said to be operational. The states of containing an interruption condition, working, or being not operational are collectively referred to as "not available".

In the case of the multiplexor channel, the channel and subchannel are easily distinguishable and, provided the channel is operational, any combination of channel and subchannel states are possible. Since the selector channel can have only one subchannel, the channel and subchannel are functionally coupled, and certain states of the channel are related to those of the subchannel. In particular, the Working state can occur only concurrently in both the channel and subchannel and, whenever an interruption condition is pending in the subchannel, the channel also is in the same state. The channel and subchannel, however, are not synonymous, and an interruption condition not associated with data transfer, such as Attention, does not affect the state of the subchannel. Thus, the subchannel may be available when the channel has an interruption condition pending. Consistent distinction between the subchannel and channel permits the two types of channels, selector and multiplexor, to be covered uniformly by a single description.

The device referred to in the preceding table includes both the device proper and its control unit. For some types of devices, such as magnetic tape drives, the Working and the Interruption Pending states can be caused by activity in the addressed device or control unit. A shared control unit imposes its state on all devices attached to the control unit. The states of the devices are not related to those of the channel and subchannel.

When the response to an I/O instruction is determined on the basis of the states of the channel and subchannel, the components further removed are not interrogated. Thus, ten composite states are identified as conditions for the execution of the I/O instruction. Each composite

JUNE 1, 1971

state is identified in the following discussion by three alphabetic characters, in which the first character position identifies the state of the channel, the second identifies the state of the subchannel, and the third refers to the state of the device. Each character position can contain A, I, W, or N, denoting the state of the component. The symbol X in place of a letter indicates that the state of the corresponding component is not significant for the execution of the instruction.

Available (AAA): The addressed channel, subchannel, control unit, and I/O device are operational, are not engaged in the execution of any previously initiated operations, and do not contain any pending interruption conditions.

Interruption Pending in Device (AAI) or Device Working (AAW): The addressed channel and subchannel are available. The addressed control unit or I/O device is executing a previously initiated operation or contains a pending interruption condition. The following situations are possible:

1. The device is executing an operation after signaling the Channel End condition, such as rewinding tape or seeking on a disk file.
2. The control unit associated with the device is executing an operation after signaling the Channel End condition, such as backspacing file on a magnetic tape drive.
3. The device or control unit is executing an operation on another subchannel or channel.
4. The device or control unit contains the Device End, Control Unit End, or Attention condition or, on the selector channel, the Channel End condition associated with an operation terminated by HALT I/O.

Device Not Operational (AAN): The addressed channel and subchannel are available. The addressed I/O device is not operational. A device appears not operational when no control unit recognizes the address. This occurs when the control unit is not provided in the system, when power is off in the control unit, when the control unit has been logically switched off the I/O interface by the I/O Interface Disable switch, or when the control unit is not configured to communicate with the IOCE housing the addressed channel. The Not Operational state is indicated when the control unit is provided and is designed to attach the device, but the device has not been installed and the address has not been assigned to the control unit.

If the addressed device is not installed or has been logically removed from the control unit, but the associated control unit is operational and the address has been assigned to the control unit, the device is said to be not ready. When an instruction is addressed to a device in the Not Ready state, the control unit responds to the selection and indicates Unit Check whenever the Not Ready state precludes a successful execution of the operation. See "Unit Check".

Interruption Pending In Subchannel (AIX): The addressed channel is available. An interruption condition is pending in the addressed subchannel due to the termination of the portion of the operation involving the use of channel facilities. The subchannel is in a position to provide information for a complete CSW. The interruption condition can indicate termination of an operation at the addressed I/O device or at another device on the subchannel. In

JUNE 1, 1971

the case of the multiplex channel, the channel is available. The state of the addressed device is not significant, except when TEST I/O is addressed to the device associated with the terminated operation, in which case the CSW contains status information provided by the device.

The state AIX does not occur on the selector channel. On the selector channel, the existence of an interruption condition in the subchannel immediately causes the channel to assign to this condition the highest priority for I/O interruptions and, hence, leads to the state IIX.

Subchannel Working (AWX): The addressed channel is available. The addressed subchannel is executing a previously initiated operation or chain of operations in the multiplex mode and has not yet reached the Channel End for the last operation. The state of the addressed device is not significant, except when HALT I/O is issued, in which case the CSW contains status provided by the device.

The Subchannel Working state does not occur on the selector channel since all operations on the selector channel are executed in the burst mode and cause the channel to be in the Working State (WWX).

Subchannel Not Operational (ANX): The addressed channel is available. The addressed subchannel on the multiplexor channel is not operational. A subchannel is not operational when it is not provided in the system. This state cannot occur on the selector channel.

Interruption Pending in Channel (IIX): The addressed channel is not working and has established which device or, in the case of the PCI condition on the multiplexor channel, which sub-channel will cause the next I/O interruption from this channel. The state where the channel contains a pending interruption condition is distinguished only by the instruction TEST CHANNEL. This instruction does not cause the subchannel and I/O device to be interrogated. The other I/O instructions consider the channel available when it contains a pending interruption condition. When the channel assigns priority for interruption among devices, the interruption condition is also preserved in the I/O device or subchannel.

Channel Working (WXX): The addressed channel is operating in the burst mode. In the case of the multiplexor channel, a burst of data is currently being handled. In the case of the selector channel, an operation or a chain of operations is currently being executed, and the Channel End for the last operation has not yet been reached. The states of the addressed device and, in the case of the multiplexor channel, of the subchannel are not significant.

Channel Not Operational (NXX): The addressed channel is not operational, or the channel address in the instruction is invalid, or has the wrong number of high-order zeros. A channel is not operational when it is not provided in the system, when power is off in the IOCE housing the channel, or when the IOCE housing the channel is not configured to communicate with the CE. The states of the addressed I/O device and subchannel are not significant.

10.2.4 RESETTING OF THE INPUT/OUTPUT SYSTEM

Two types of resetting can occur in the I/O system. The reset states overlap the hierarchy of states distinguished for the purpose of responding to the CE during the execution of I/O instructions. Reset-

JUNE 1, 1971

ting terminates the current operation, disconnects the device from the channel, and may place the device in certain modes of operation. The meaning of the two reset states for each type of I/O device is specified in the Systems Reference Library (SRL) publication for the device.

10.2.4.1 Input/Output Control Element Reset

The Input/Output Control Element reset (IOCE reset) function is performed when initial program loading is performed, when an all-zero CE communications field is changed to a non-zero field while in state Zero or One, or when a system power-on sequence is completed. (For IOCE operation on a set address translator instruction occurring during the reset interval, see Section 8.4.5.7.)

An IOCE reset causes each channel housed in that IOCE to terminate operations on all subchannels. Status information and interruption conditions in the subchannels are reset, and all subchannels are placed in the available state. The channel sends the reset signal to all I/O devices attached to it.

If the device is currently communicating over the I/O interface, the device immediately disconnects from the channel. Data transfer and any operation using the facilities of the control unit are immediately terminated, and the I/O device is not necessarily positioned at the beginning of a block. Mechanical motion not involving the use of the control unit, such as rewinding magnetic tape, proceeds to the normal stopping point, if possible. The device appears in the working state until the termination of mechanical motion or the inherent cycle of operation, if any, whereupon it becomes available. Status information in the device and control unit is reset, and no interruption condition is generated upon completing the operation.

A control unit accessible by more than one channel is reset if it is currently associated with the channel on the IOCE generating the reset.

10.2.4.2 Malfunction Reset

The Malfunction Reset function is performed when the channel detects equipment malfunctioning.

Execution of Malfunction Reset in the channel depends on the type of malfunction. It may cause all operations in the channel to be terminated and all subchannels to be reset to the Available state. The channel may send the Malfunction Reset signal to the device connected to the channel at the time the malfunction is detected.

When the channel signals Malfunction Reset over the interface, the device immediately disconnects from the channel. Data transfer and any operation using the facilities of the control unit are immediately terminated, and the I/O device is not necessarily positioned at the beginning of a block. Mechanical motion not involving the control unit, such as rewinding magnetic tape proceeds to the normal stopping point, if possible. The device appears in the Working state until the termination of mechanical motion or the inherent cycle of operation, if any. Status information associated with the addressed device is reset, but an interruption condition may be generated upon completing any mechanical operation.

When a Malfunction Reset occurs, the program is alerted by an I/O interruption or, when the malfunction is detected during the execution of an I/O instruction, by the setting of the condition code. In either case the CSW identifies the condition. The device addressed by the I/O instruction or the device identified by the I/O interruption, however, is not necessarily the one placed in the Malfunction Reset state.

JUNE 1, 1971

10.2.5 CONDITION CODE

The results of certain tests by the CE and by the channel and device, and the original state of the addressed part of the I/O system are used during the execution of an I/O instruction to set one of four condition codes in bit positions 34 and 35 of the PSW. The condition code is set at the time the execution of the instruction is completed, that is, the time the CE is released to proceed with the next instruction. The condition code indicates whether or not the channel address contains the correct number of high-order zeros (10.2.6), or the channel has performed the function specified by the instruction and, if not, the reason for the rejection. The code can be used for decision-making by subsequent branch on condition operations.

The following table lists the conditions that are identified and the corresponding condition codes for each instruction. The states of the system and their abbreviations were previously defined in "States of the Input/Output System" (Section 10.2.3). The digits in the table represent the numeric value of the code. The instruction START I/O can set code 0 or 1 for the AAA state, depending on the type of operation that is initiated.

JUNE 1, 1971

TABLE 10-III CONDITION CODE SETTING FOR INPUT/OUTPUT INSTRUCTIONS

CONDITIONS		CONDITION CODE FOR:				
		START I/O	TEST I/O	HALT I/O	SET PCI	TEST CHANNEL
Available	AAA	0,1*	0	1*	0	0
Interruption pending in device	AAI	1*	1*	1*	0	0
Device working	AAW	1*	1*	1*	0	0
Device not operational	AAN	3	3	3	0	0
Interruption pending in subchannel	AIX#					
For the addressed device		2	1*	0	2	0
For another device		2	2	0	2	0
Subchannel working	AWX#	2	2	1*	2	0
Subchannel not operational	ANX#	3	3	3	3	0
Interruption pending in channel	IXX	see note below				1
Channel working	WXX#	2	2	2	2	2
Channel not operational	NXX#	3	3	3	3	3
Error						
Channel equipment error		1*	1*	1*	1*	-
Channel programming error		1*	-	-	-	-
Device error		1*	1*	-	-	-

Legend

* The CSW or its status portion is stored at location 64 during execution of the instruction.

The symbol X stands for A, I, W, and N, and indicates that the state of the corresponding component is not significant. As an example, AIX denotes the state AIA, AII, AIW, and AIN, while IXX represents a total of 16 states, some of which do not occur.

- The condition cannot be identified during execution of the instruction.

Note: For the purpose of executing START I/O, TEST I/O, SET PCI, and HALT I/O, a channel containing a pending interruption condition appears the same as an available channel, and the condition-code setting depends upon the states of the subchannel and device. The condition codes for the IXX state are the same as for the AXX states, where the Xs represent the states of the subchannel and the device. As an example, the condition code for the IAA state is the same as for the AAA state, and the condition code for the IAW state is the same as for the AAW state.

The available condition is indicated only when no errors are detected during the execution of the I/O instruction. When a programming error occurs in the information placed in the CAW or CCW and the addressed channel or subchannel is working, condition code 2 is set. Either code 1 or 3 may be set when a programming error occurs and a part of the addressed I/O system is not operational.

When a subchannel on the multiplexor channel contains a pending interruption condition (state AIX), the I/O device associated with the terminated operation normally is in the Interruption Pending state. When the channel detects during execution of TEST I/O that the device is

JUNE 1, 1971

| not operational, condition code 3 is set. Similarly, condition code 3 is set when HALT I/O is addressed to a subchannel in the Working state and operating in the multiplex mode (state AWX), but the device turns out to be not operational. The Not Operational state in both situations | can be caused by operator intervention, reconfiguration or by equipment malfunctioning.

The error conditions listed in the preceding table include all equipment or programming errors detected by the channel or the I/O device during execution of the I/O instruction. The status portion of the CSW identifies the error. Three types of errors can occur:

Channel equipment error: The channel can detect the following equipment errors during execution of START I/O, TEST I/O, SPCI and HALT I/O:

The device address that the channel received on the interface during initial selection either has a parity error or is not the same as the one the channel sent out. Some device other than the one addressed may be malfunctioning.

The unit-status byte that the channel received on the interface during initial selection has a parity error.

A signal from the I/O device occurred during initial selection at an invalid time or had invalid duration.

The channel detected an error in its control equipment.

The channel may perform the Malfunction Reset function, depending on the type of error. If a CSW is stored, Channel Control Check or Interface Control Check is indicated, depending on the type of error.

Channel programming error: The channel can detect the following programming errors during execution of START I/O:

- Invalid CCW address in CAW
- Invalid CCW address specification in CAW
- Invalid CAW format
- Location of first CCW protected for fetching
- First CCW specifies transfer in channel
- Invalid command code in first CCW
- Invalid count in first CCW
- Invalid format of first CCW
- | First CCW resides in an SE which is in Logout-Stop State

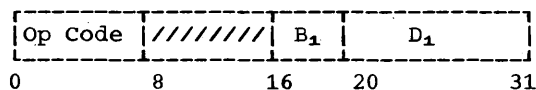
| The CSW indicates Program, Protection or Chaining Check.

Device error: Programming or equipment errors detected by the device during the execution of START I/O are indicated by Unit Check or Unit Exception in the CSW.

The conditions responsible for Unit Check and Unit Exception for each type of I/O device are detailed in the SRL publication for the device.

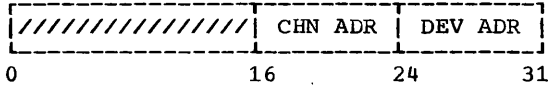
10.2.6 INSTRUCTION FORMAT

All I/O instructions use the following SI format:



JUNE 1, 1971

Bit positions 8-15 of the instruction are ignored. The content of the B₁ field designates a register. The sum obtained by the addition of the content of register B₁ and content of the D₁ field identifies the channel and the I/O device. A zero in the B₁ field indicates the absence of the B₁ address component. This sum has the format:



Bit positions 0-7 are not part of the address. Bit positions 8-15, which constitute the high-order portion of the address, are ignored. Bit positions 16-23 of the sum contain the channel address, while bit positions 24-31 identify the device on the channel and, in the case of the multiplexor channel, designate the subchannel.

Bit positions 16-19 of the sum must contain zeros. A condition code 3 is set for any I/O instruction that fails to develop zeros in these bit positions and execution is completed without an attempt to select a channel.

10.2.7 INSTRUCTIONS

The mnemonics, format, and operation codes of the I/O instructions follow. The table also indicates that all I/O instructions cause program interruption when they are encountered in the Problem state, and that all I/O instructions set the condition code.

TABLE 10-IV INPUT/OUTPUT INSTRUCTIONS

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
START I/O	SIO	SI	C M	9C	10.2.7.1
TEST I/O	TIO	SI	C M	9D	10.2.7.2
HALT I/O	HIO	SI	C M	9E	10.2.7.3
TEST CHANNEL	TCH	SI	C M	9F	10.2.7.4
SET PCI	SPCI	SI	C M	9B	10.2.7.5

Legend

C Condition code is set
M Privileged-operation exception

PROGRAMMING NOTE

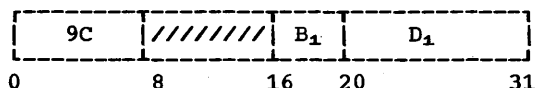
The instructions START I/O, TEST I/O, SET PCI, and HALT I/O may cause a CSW to be stored. To prevent the contents of the CSW stored by the instruction from being destroyed by an immediately following I/O interruption, all channels must be masked before issuing START I/O, TEST I/O, SET PCI, or HALT I/O and must remain masked until the information in the CSW provided by the instruction has been acted upon or stored elsewhere for later use.

JUNE 1, 1971

Whenever the status bits of the CSW associated with the interruption indicate that the storing of the CSW was due to a channel control check, or an interface control check, additional information is provided by a diagnostic logout into the preferential-storage area. To prevent the contents of the logout area from being destroyed by an immediately following logout, all channels of the selected IOCE must be masked until the logout information has been acted upon or stored elsewhere for later use.

10.2.7.1 Start I/O

SIO $D_1(B_1)$ [SI]



A write, read, read backward, control or sense operation is initiated at the addressed I/O device and subchannel. The instruction START I/O is executed only when the CE is in the Supervisor state.

Bit positions 16-31 of the sum formed by the addition of the content of register B_1 and the content of the D_1 field identify the channel, subchannel, and I/O device to which the instruction applies. The CAW at location 72 of the CE preferential-storage area contains the protection key for the subchannel and the address of the first CCW. The CCW so designated specifies the operation to be performed, the storage area to be used, and the action to be taken when the operation is completed.

The I/O operation specified by START I/O is initiated if the addressed I/O device and subchannel are available, the channel is Available or is in the Interruption Pending state, and errors or exceptional conditions have not been detected. The I/O operation is not initiated when the addressed part of the I/O system is in any other state or when the channel or device detects any error or exceptional condition during execution of the instruction.

When any of the following conditions occurs, with the channel either Available or in the Interruption Pending state and with the subchannel available before the execution of the instruction, START I/O causes the status portion, bit positions 32-47, of the CSW at location 64 of the CE preferential-storage area to be replaced by a new set of status bits. The status bits pertain to the device addressed by the instruction. The contents of the other fields of the CSW are not changed.

An immediate operation was executed, and either no command chaining is specified, or chaining is suppressed because of unusual conditions detected during the operation. An operation is called immediate when the I/O device signals the Channel End condition immediately on receipt of the command code. The CSW contains the Channel End bit and any other indications provided by the channel or the device. The busy bit is off. The I/O operation has been initiated, but no information has been transferred to or from the storage area as designated by the CCW. No interruption conditions are generated at the device or subchannel, and the subchannel is available for a new I/O operation.

The I/O device contains a pending interruption condition due to Device End or Attention, or the control unit contains a pending Control Unit End for the addressed device, or, on the selector channel, the control unit contains for the addressed device a pending Channel End following the execution of HALT I/O. The CSW

JUNE 1, 1971

unit-status field contains the busy bit, identifies the interruption condition, and may contain other bits provided by the device or control unit. The interruption condition is cleared. The channel-status field indicates any error conditions detected by the channel and contains the PCI bit if specified in the first CCW.

The I/O device or the control unit is executing a previously initiated operation, or the control unit has pending an interruption condition associated with a device other than the one addressed. The CSW unit-status field contains the busy bit or, if the control unit is busy, the busy and status-modifier bits. The channel-status field indicates any error conditions detected by the channel and contains the PCI bit if specified in the first CCW.

The I/O device or channel detected an equipment or programming error during execution of the instruction. The CSW identifies the error condition. The Channel End and busy bits are off, unless the error was detected after the device was selected, and the device was found to be busy, in which case the busy bit, as well as any bits indicating pending interruption conditions, are on. The interruption conditions indicated in the CSW have been cleared at the device. The I/O operation has not been initiated. No interruption conditions are generated at the I/O device or subchannel.

On the multiplexor channel, START I/O causes the addressed device to be selected and the operation to be initiated only after the channel has serviced all outstanding requests for data transfer.

Condition Code:

- 0 I/O operation initiated and channel proceeding with its execution
- 1 CSW stored
- 2 Channel or subchannel busy
- 3 Not operational or invalid I/O address format

Program Interruptions:

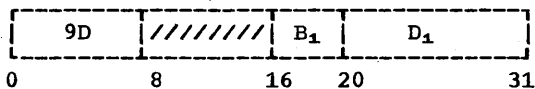
Privileged operation.

PROGRAMMING NOTE

When the channel detects a programming error during execution of START I/O and the addressed device contains an interruption condition, with the channel in either the available or interruption pending state and the subchannel in the Available state, START I/O may or may not clear the interruption condition. If the instruction has caused the device to be interrogated, as indicated by the presence of the busy bit in the CSW, the interruption condition has been cleared, and the CSW contains Program, Chaining or Protection Check, as well as the status from the device.

10.2.7.2 Test I/O

TIO $D_1(B_1)$ [SI]



The state of the addressed channel, subchannel, and device is indicated by setting the condition code in the PSW and, under certain conditions, by storing the CSW. Pending interruption conditions may be

JUNE 1, 1971

cleared. The instruction TEST I/O is executed only when the CE is in the Supervisor state.

Bit positions 16-31 of the sum formed by the addition of the content of register B₁ and the content of the D₁ field identify the channel, subchannel, and I/O device to which the instruction applies.

When any of the following conditions occurs with the channel either available or in the Interruption Pending state, TEST I/O causes the CSW at location 64 of the CE preferential-storage area to be stored. The content of the CSW pertains to the I/O device addressed by the instruction.

The subchannel contains a pending interruption condition due to a terminated operation at the addressed device. The CSW identifies the interruption condition, and the interruption condition is cleared. The protection key, command address, and count fields contain the final values for the I/O operation, and the status may include other bits provided by the channel and the device. The interruption condition in the subchannel is not cleared, and the CSW is not stored if the interruption condition is associated with an operation on a device other than the one addressed.

The subchannel is available and the I/O device contains a pending interruption condition due to Device End or Attention, the control unit contains a pending Control Unit End for the addressed device, or, on the selector channel, the control unit contains for the addressed device a pending Channel End following the execution of HALT I/O. The CSW unit-status field identifies the interruption condition and may contain other bits provided by the device or control unit. The interruption condition is cleared. The busy bit in the CSW is off. The other fields of the CSW contain zeros unless an equipment error is detected.

The subchannel is available and the I/O device or the control unit is executing a previously initiated operation, or the control unit has a pending interruption condition, other than Device End or Attention, associated with a device other than the one addressed. The CSW unit-status field contains the busy bit or, if the control unit is busy, the busy and status-modifier bits. Other fields of the CSW contain zeros unless an equipment error is detected.

The subchannel is available and the I/O device or channel detected an equipment error during execution of the instruction or the addressed device is in the Not Ready state and does not have any pending interruption condition. The CSW identifies the error conditions. If the device is not ready, Unit Check is indicated. No interruption conditions are generated at the I/O device or the subchannel.

When TEST I/O is used to clear an interruption condition from the subchannel and the channel has not yet accepted the condition from the device, the instruction causes the device to be selected and the interruption condition in the device to be cleared. During certain I/O operations, some types of devices cannot provide their current status in response to TEST I/O. The tape control unit, for example, is in such a state when it has provided the Channel End condition and is executing the backspace-file operation. When TEST I/O is issued to a control unit in such a state, the unit-status field of the CSW contains the busy and status-modifier bits, with zeros in the other CSW fields. The interruption condition in the device and in the subchannel is not cleared. On some types of devices the device never provides its current status in response to TEST I/O, and an interruption condition can be cleared only by permitting an I/O interruption. When TEST I/O is issued to such a device, the unit-status field contains the status-modifier bit, with

zeros in the other CSW fields. The interruption condition in the device and in the subchannel, if any, is not cleared.

However, at the time the channel assigns the highest priority for interruptions to a condition associated with an operation at the subchannel, the channel accepts the status from the device and clears the corresponding condition at the device. When TEST I/O is addressed to a device for which the channel has already accepted the interruption condition, the device is not selected, and the condition in the subchannel is cleared regardless of the type of device and its present state. The CSW contains unit status and other information associated with the interruption condition.

On the multiplexor channel, TEST I/O causes the addressed device to be selected only after the channel has serviced all outstanding requests for data transfer.

Condition Code:

- 0 Available
- 1 CSW stored
- 2 Channel or subchannel busy
- 3 Not operational or invalid I/O address format

Program Interruptions:

Privileged operation

PROGRAMMING NOTES

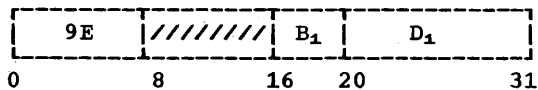
Masking of channels provides the program a means of controlling the priority of I/O interruptions selectively by channels. The priority of devices attached on a channel is fixed and cannot be controlled by the program. The instruction TEST I/O permits the program to clear interruption conditions selectively by I/O device.

When a CSW is stored by TEST I/O, the Interface Control Check and Channel Control Check indications may be due to a condition already existing in the channel or due to a condition created by TEST I/O. Similarly, presence of the Unit Check bit in the absence of Channel End, Control Unit End or Device End bits may be due to a condition created by the preceding operation, the Not Ready state, or an equipment error detected during the execution of TEST I/O.

TEST I/O cannot be used to clear a pending interruption condition due to the PCI flag while the subchannel is in the Working state.

10.2.7.3 Halt I/O

HIO D₁(B₁) [SI]



Execution of the current I/O operation at the addressed I/O device, subchannel, or channel is terminated. The subsequent state of the subchannel depends on the type of channel. The instruction HALT I/O is executed only when the CE is in the Supervisor state.

Bit positions 16-31 of the sum formed by the addition of the content of register B₁ and the content of the D₁ field identify the channel,

JUNE 1, 1971

and, when the channel is not working, identify the subchannel and the I/O device to which the instruction applies.

When the channel is either Available or in the Interruption Pending state, with the subchannel either available or working, HALT I/O causes the addressed device to be selected and to be signaled to terminate the current operation, if any. If the subchannel is available, its state is not affected. If, on the multiplexor channel, the subchannel is working, data transfer is immediately terminated, but the subchannel remains in the Working state until the device provides the next status byte, whereupon the subchannel is placed in the Interruption Pending state.

When HALT I/O is issued to a channel operating in the burst mode, data transfer for the burst operation is terminated, and the device performing the burst operation is immediately disconnected from the channel. The subchannel and I/O device address in the instruction, in this case, is ignored.

The termination of a burst operation by HALT I/O on the selector channel causes the channel and subchannel to be placed in the Interruption Pending state. Generation of the interruption condition is not contingent on the receipt of a status byte from the device. When HALT I/O causes a burst operation on the multiplexor channel to be terminated, the subchannel associated with the burst operation remains in the Working state until the device provides Channel End, where-upon the subchannel enters the Interruption Pending state.

On the multiplexor channel operating in the multiplex mode, the device is selected and the instruction is executed only after the channel has serviced all outstanding requests for data transfer for previously initiated operations. If the control unit does not accept the HALT I/O signal because it is in the Not Operational or Control Unit Busy state, the subchannel, if working, is set up to signal termination of device operation the next time the device requests or offers a byte of data. If command chaining is indicated in the subchannel and the device presents status next, chaining is suppressed.

When the addressed subchannel has a pending interruption condition, with the channel in the Available or Interruption Pending state, HALT I/O does not cause any action.

When any of the following conditions occur, HALT I/O causes the status portion, bit positions 32-47, of the CSW at location 64 of the CE preferential-storage area to be replaced by a new set of status bits. The contents of the other fields of the CSW are not changed. The CSW stored by HALT I/O pertains only to the execution of HALT I/O and does not describe under what conditions the I/O operation at the addressed subchannel is terminated. The extent of data transfer, and the conditions of terminations of the operation at the subchannel, are provided in the CSW associated with the interruption condition due to the termination.

The addressed device has been selected and signaled to terminate the current operation. The CSW contains zeros in the status field unless an equipment error is detected.

The channel attempted to select the addressed device, but the control unit could not accept the HALT I/O signal because it is executing a previously initiated operation or has an interruption condition, other than Device End or Attention, associated with a device other than the one addressed. The signal to terminate the operation has not been transmitted to the device, and the subchannel, if in the working state, has been set up to signal termination the next time the device identifies itself. The CSW

JUNE 1, 1971

unit-status field contains the busy and status-modifier bits. The channel-status field contains zeros unless an equipment error is detected.

The channel detected an equipment malfunction during the execution of HALT I/O. The status bits in the CSW identify the error condition. The state of the channel and the progress of the I/O operation are unpredictable.

When HALT I/O causes data transfer to be terminated, the control unit associated with the operation remains unavailable until the data-handling portion of the operation in the control unit is terminated. Termination of data-transfer portion of the operation is signaled by generation of Channel End, which may occur at the normal time for the operation, earlier, or later, depending on the operation and type of device. If the control unit is shared, all devices attached to the control unit appear in the Working state until the Channel End condition is accepted by the channel. The I/O device executing the terminated operation remains in the Working state until termination of the inherent cycle of the operation, at which time Device End is generated. If blocks of data at the device are defined, such as reading on magnetic tape, the recording medium is advanced to the beginning of the next block.

When HALT I/O is issued at a time when the subchannel is available and no burst operation is in progress, the effect of the HALT-I/O signal depends on the type of device and its state. The HALT-I/O signal has no effect on devices that are not in the Working state or are executing an operation of a fixed duration, such as rewinding tape or positioning a disk-access mechanism. If the device is executing a type of operation that is variable in duration, the device interprets the signal as one to terminate the operation. Pending Attention or Device End conditions at the device are not reset.

Condition Code:

- 0 Interruption pending in subchannel
- 1 CSW stored
- 2 Burst operation terminated
- 3 Not operational or invalid I/O address format

Program Interruptions:

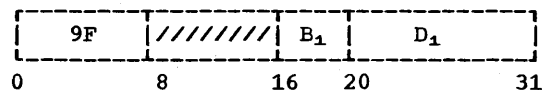
Privileged operation

PROGRAMMING NOTE

The instruction HALT I/O provides the program a means of terminating an I/O operation before all data specified in the operation have been transferred. It permits the program to immediately free the selector channel for an operation of higher priority. On the multiplexor channel, HALT I/O provides a means of controlling real-time operations and permits the program to terminate data transmission on a communication line.

10.2.7.4 Test Channel

TCH D₁(B₁) [SI]



JUNE 1, 1971

The condition code in the PSW is set to indicate the state of the addressed channel. The state of the channel is not affected, and no action is caused. The instruction TEST CHANNEL is executed only when the CE is in the Supervisor state.

Bit positions 16-23 of the sum formed by the addition of the content of register B₁ and the content of the D₁ field identify the channel to which the instruction applies. Bit positions 24-31 of the address are ignored.

The instruction TEST CHANNEL inspects only the state of the addressed channel. It tests whether the channel is operating in the burst mode, is aware of any outstanding interruption conditions from its devices, or is not operational. When the channel is operating in the burst mode and contains a pending interruption condition, the condition code is set as for operation in the burst mode. When none of these conditions exists, the Available state is indicated. No device is selected, and, on the multiplexor channel, the subchannels are not interrogated.

Condition Code:

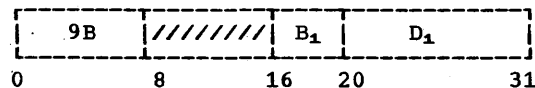
- 0 Channel available
- 1 Interruption pending in channel
- 2 Channel operating in burst mode
- 3 Channel not operational or invalid I/O address format

Program Interruptions:

Privileged operation

10.2.7.5 Set PCI

SPCI D₁(B₁) [SI]



An I/O interruption is requested from the subchannel controlling the addressed device. The instruction SET PCI is executed only when the CE is in the supervisor state.

Bit positions 16-31 of the sum formed by the addition of the content of register B₁ and the content of the D₁ field identify the I/O device to whose subchannel or channel the instruction applies.

If the channel is operating in the multiplex mode, and the addressed subchannel is in the interruption pending state (Channel End) or working, the PCI bit is set in the subchannel. If the channel is operating in the burst mode, the PCI bit for the operating subchannel is set. When the addressed part of the I/O system is in any other state, or when the channel or subchannel detects any error or exceptional condition during execution of the instruction, the PCI bit is not set.

Setting of the PCI bit will cause the associated subchannel to request an I/O interruption. An I/O interruption caused solely by SET PCI will cause CSW data to be stored without waiting for the current CCW to be completed and without changing device operation. Also, all unit-status bits in the CSW will be off. A CSW stored as a result of SET PCI plus other causes will have all appropriate status bits set.

JUNE 1, 1971

If the channel detected an equipment malfunction during the execution of SET PCI, the status portion (bits 32-47) of the CSW at location 64 of the CE preferential-storage area will be set to indicate the error condition. In this case the state of the channel is unpredictable.

Condition Code:

- 0 Channel available or interruption pending and subchannel available or interruption pending (not Channel End); no action has been caused.
- 1 CSW stored
- 2 PCI flag set
- 3 Channel not operational or Invalid I/O address format

Program Interruptions:

Privileged operation

PROGRAMMING NOTE

If a number of SET PCI instructions are executed addressing subchannels of a channel which is masked off, only the first PCI issued will be effective when the CE becomes interruptible for that channel. All other PCI's remain pending until after a data transfer has occurred for the associated subchannels.

10.3 EXECUTION OF INPUT/OUTPUT OPERATIONS

The channel can execute the following seven commands:

- Write,
- Read,
- Read Backward,
- Control,
- Sense, and
- Transfer In Channel.
- Search

Each of the seven commands, except Transfer In Channel initiates a corresponding I/O operation. The term "I/O operation" refers to the activity initiated by a command in the I/O device and associated subchannel. The subchannel is involved with the execution of the operation from the initiation of the command until the Channel End signal is received or, in the case of command chaining, until the Device End signal is received. The operation in the device lasts until Device End occurs.

10.3.1 BLOCKING OF DATA

Data recorded by an I/O device may be divided into blocks. A block of data is defined for each type of I/O device as the amount of information recorded in the interval between adjacent starting and stopping points of the device. The length of a block depends on the document; for example, a block can be a card, a line of printing, the information recorded between two consecutive gaps on magnetic tape, or the information in one record on disk.

The maximum amount of information that can be transferred in one I/O operation is one block. An I/O operation is terminated when the associated storage area is exhausted or the end of the block is reached, whichever occurs first. For some operations, such as writing on a magnetic tape unit, blocks are not defined, and the amount of information transferred is controlled only by the program.

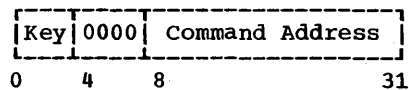
JUNE 1, 1971

10.3.2 CHANNEL ADDRESS WORD

The channel address word (CAW) specifies the storage protection key and the address of the first CCW associated with START I/O. It is assigned location 72 of the CE preferential-storage area. The channel refers to the CAW only during the execution of START I/O. The pertinent information thereafter is stored in the subchannel, and the program is free to change the content of the CAW. Fetching of the CAW by the channel does not affect the contents of location 72.

When the IOCE fails to gain access to the preferential-storage area on a CAW fetch, a preferential-storage area lockout exception is recognized by the controlling CE and a program interruption is taken.

The CAW has the following format:



The fields in the CAW are allocated for the following purposes:

Protection Key: Bits 0-3 form the protection key for all commands associated with START I/O. This key is matched with a storage key whenever data are placed in main storage, or whenever data are fetched from main storage and the fetch-protection bit is one in the storage key. (Protection is described in Chapter 2.)

Command Address: Bits 8-31 designate the location of the first CCW in storage.

Bit positions 4-7 of the CAW must contain zeros. The three low-order bits of the command address must be zero to specify the CCW on integral boundaries for double words. If any of these restrictions are violated or if the command address specifies a location protected for fetching or outside the storage of the particular installation, or outside the configured storage, or storage assigned by the address translator for a particular IOCE, START I/O causes the status portion of the CSW to be stored with the Protection Check or Program Check bit on. In this event, the I/O operation is not initiated.

PROGRAMMING NOTE 1

Bit positions 4-7 of the CAW, which presently must contain zeros, may in the future be assigned for the control of new functions. It is therefore recommended that these bit positions not be set to one for the purpose of obtaining an intentional Program Check indication.

PROGRAMMING NOTE 2

The protection key is matched with a key in storage whenever reference is made to main storage for a CCW or data. The key is ignored whenever reference is made to MACH storage.

10.3.3 CHANNEL COMMAND WORD

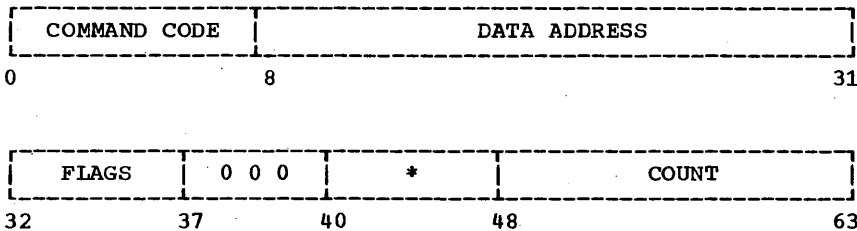
The channel command word (CCW) specifies the command to be executed and, for commands initiating I/O operations, it designates the storage area associated with the operation and the action to be taken whenever transfer to or from the area is completed. The CCWs can be located anywhere in storage, and more than one can be associated with a START

JUNE 1, 1971

I/O. The channel refers to a CCW in storage only once, whereupon the pertinent information is stored in the channel.

The first CCW is fetched during the execution of START I/O. Each additional CCW in the sequence is obtained when the operation has progressed to the point where the additional CCW is needed. Fetching of the CCWs by the channel does not affect the contents of the location in storage.

The CCW has the following format:



The fields in the CCW are allocated for the following purposes:

Command Code: Bits 0-7 specify the operation to be performed.

Data Address: Bits 8-31 specify the location of an eight-bit byte in storage. It is the first location referred to in the area designated by the CCW.

Chain-Data (CD) Flag: Bit 32, when one, specifies chaining of data. It causes the storage area designated by the next CCW to be used with the current operation.

Chain-Command (CC) Flag: Bit 33, when one, and when the CD flag is zero, specifies chaining of commands. It causes the operation specified by the command code in the next CCW to be initiated on normal completion of the current operation.

Suppress-Length-Indication (SLI) Flag: Bit 34 controls whether or not an incorrect-length condition is to be indicated to the program. When this bit is one and the CD flag is zero in the last CCW used, the incorrect-length indication is suppressed. When both the CC and SLI flags are one and the CD flag is zero, command chaining takes place regardless of the presence of an incorrect-length condition.

Skip Flag: Bit 35, when one, specifies suppression of transfer of information to storage during a read, read backward, or sense operation.

Program-Controlled-Interruption (PCI) Flag: Bit 36, when one, causes the channel to generate an interruption request condition upon fetching the CCW. When bit 36 is zero, normal operation takes place.

Count: Bits 48-63 specify the number of eight-bit byte locations in the storage area designated by the CCW.

Bit positions 37-39 of every CCW other than one specifying Transfer In Channel must contain zeros. Violation of this restriction generates the Program Check condition. When the first CCW designated by the CAW does not contain the required zeros, the I/O operation is not initiated, and the status portion of the CSW with the Program Check indication is stored during execution of START I/O. Detection of this condition during data chaining causes the I/O device to be signaled to terminate the operation. When the absence of these zeros is detected during

JUNE 1, 1971

command chaining, the new operation is not initiated, and an interruption condition is generated.

The content of bit positions 40-47 of the CCW is ignored.

PROGRAMMING NOTE 1

Bit positions 37-39, of the CCW, which presently must contain zeros, may in the future be assigned for the control of new functions. It is therefore recommended that these bit positions not be set to one for the purpose of obtaining an intentional Program Check indication.

PROGRAMMING NOTE 2

The command address in the CAW designates the location of the first CCW in main or MACH storage. Since chaining takes place between CCW's located in successive double-word storage locations, if the first CCW is fetched from main storage the chained CCW's are also fetched from main storage. Conversely, when the first CCW is fetched from MACH storage, subsequent chained CCW's are fetched from MACH storage. Two chains of CCW's located in noncontiguous storage areas, including the case of areas in both main and MACH storage, can be coupled by a transfer in channel (TIC) command. If the first CCW resides in an SE which is in Logout Stop state, the operation is not initiated, and the status portion of the CSW is stored during execution of Start I/O, with a Chaining Check indicated. When a subsequent CCW is found to be in a Logout-Stopped SE during Command Chaining, the new operation is not initiated, and an interruption condition is generated in the subchannel. If this condition is detected during Data Chaining, the I/O device will be signalled to terminate its operation.

10.3.4 COMMAND CODE

The command code, bit positions 0-7 of the CCW, specifies to the channel and the I/O device the operation to be performed.

The two low-order bits or, when these bits are 00, the four low-order bits of the command code identify the operation to the channel. The channel distinguishes among the following four operations:

- Output forward (write, control, search)
- Input forward (read, sense)
- Input backward (read backward)
- Branching (transfer in channel)

The channel ignores the high-order bits of the command code.

Commands that initiate I/O operations (write, read, read backward, control, and sense) cause all eight bits of the command code to be transferred to the I/O device. In these command codes, the high-order bit positions contain modifier bits. The modifier bits specify to the device how the command is to be executed. They may cause, for example, the device to compare data received during a write operation with data previously recorded, and they may specify such conditions as recording density and parity. For the Control command, the modifier bits may contain the order code specifying the control function to be performed. The meaning of the modifier bits depends on the type of I/O device and is specified in the SRL publication for the device.

The command-code assignment is listed in the following table. The symbol x indicates that the bit position is ignored; m identifies a modifier bit.

TABLE 10-V COMMAND-CODE ASSIGNMENT

CODE	COMMAND
xxxx 0000	Invalid
mmmm 0100	Sense
xxxx 1000	Transfer In Channel
mmmm 1100	Read Backward
mmmm mm01	Write
mmmm mm10	Read
mmmm mm11	Control
mmmm mm01	Search

Whenever the channel detects an invalid command code during the initiation of a command, the Program Check condition is generated. When the first CCW designated by the CAW contains an invalid command code, the status portion of the CSW with the Program Check indication is stored during execution of START I/O. When the invalid code is detected during command chaining, the new operation is not initiated, and an interruption condition is generated. The command code is ignored during data chaining, unless it specifies Transfer In Channel.

10.3.5 DEFINITION OF STORAGE AREA

The storage area associated with an I/O operation is defined by CCWs. A CCW defines an area by specifying the address of the first eight-bit byte to be transferred and the number of consecutive eight-bit bytes contained in the area. The address of the first byte appears in the data-address field of the CCW. The number of bytes contained in the storage area is specified in the count field.

PROGRAMMING NOTE 1

I/O data transfers may be made between an I/O device and either main or MACH storage. The selection of storage is under program control, with bits 8 and 9 of the data address in the channel command word (CCW) designating the storage to be used. When bits 8 and 9 are not both ones, the data address refers to a main storage location; when bits 8 and 9 are both ones, the data address refers to a MACH storage location.

In write, read, control, and sense operations storage locations are used in ascending order of addresses. As information is transferred to or from storage, the content of the address field is incremented, and the content of the count field is decremented. The read-backward operation causes data to be placed in storage in a descending order of addresses, and both the count and the address are decremented. When the count in any operation reaches zero, the storage area defined by the CCW is exhausted.

Any storage location provided in the system can be used to transfer data to or from an I/O device, provided that during an operation the location is not protected. Similarly, the CCWs can be specified in any part of available storage, provided the location is not protected for a fetch-type reference. When the channel attempts to refer to a protected location, the Protection Check condition is generated. The device is signaled to terminate the operation, if Data Chaining is in progress. An interruption condition is set up in the subchannel, and the new operation is not initiated, if Command Chaining is in progress.

PROGRAMMING NOTE 2

JUNE 1, 1971

Storage protection applies only the main storage references. Storage protection is not provided for MACH storage.

In the event the channel refers to a location not provided in the system, or outside the configured storage, or storage assigned by the storage address translator for a particular IOCE, the program-check condition is generated. Invalid data addresses detected after initiation of the operation or detection of an invalid CCW address during chaining is indicated to the program with the interruption condition at the termination of the operation or chain of operations. Storage addresses do not wrap around to location 0.

During an output operation, the channel may fetch data from the storage prior to the time the I/O device requests the data. As many as 12 bytes may be prefetched and buffered. Similarly, on data chaining during an output operation, the channel may fetch the new CCW when as many as 12 bytes remain to be transferred under the control of the current CCW. When the I/O operation uses data and CCWs from locations near the end of the available storage, such prefetching may cause the channel to refer to locations that do not exist. Invalid addresses detected during prefetching of data or CCWs do not affect the execution of the operation and do not cause error indications until the I/O operation actually attempts to use the information. If the operation is terminated by the I/O device or by HALT I/O before the invalid information is needed, the condition is not brought to the attention of the program.

The count field in the CCW can specify any number of bytes up to 65,535. Except for a CCW specifying Transfer In Channel, the count field may not contain the value zero. Whenever the count field in the CCW initially contains a zero, the Program Check condition is generated. When this occurs in the first CCW designated by the CAW, the operation is not initiated, and the status portion of the CSW with the Program Check indication is stored during execution of START I/O. When a count of zero is detected during data chaining, the I/O device is signaled to terminate the operation. Detection of a count of zero during command chaining suppresses initiation of the new operation and generates an interruption condition.

10.3.6 CHAINING

When the channel has performed the transfer of information specified by a CCW, it can continue the activity initiated by START I/O by fetching a new CCW. Such fetching of a new CCW is referred to as chaining, and the CCWs belonging to such a sequence are said to be chained.

Chaining takes place only between CCWs located in successive double-word locations in storage. It proceeds in an ascending order of addresses; i.e., the address of the new CCW is obtained by adding eight to the address of the current CCW. Two chains of CCWs located in noncontiguous storage area can be coupled for the purpose of chaining by means of Transfer In Channel. All CCWs in a chain apply to the I/O device specified in the original START I/O.

Two types of chaining are provided: chaining of data and chaining of commands. Chaining is controlled by the Chain Data (CD) and Chain Command (CC) flags in conjunction with the Suppress Length Indication (SLI) flag in the CCW. These flags specify the action to be taken by the channel upon the exhaustion of the current CCW and upon receipt of ending status from the device, as shown in Table 10-VI.

The specification of chaining is effectively propagated through a Transfer In Channel command. When in the process of chaining a Transfer

JUNE 1, 1971

In Channel command is fetched, the CCW designated by the Transfer In Channel is used for the type of chaining specified in the CCW preceding the Transfer In Channel.

The CD, CC and SLI flags are ignored in the Transfer In Channel command.

PROGRAMMING NOTE 3

There is a difference in definition of "PSA Access" for the CE and for the IOCE. A CE makes a "PSA Access" whenever bits 8-19 of any logical address are all zeros. An IOCE makes a "PSA Access" only to fetch a CAW, store a CSW, store an I/O Old PSW, or perform a Logout. All CCW fetches and all data transfers, even though bits 8-19 may be zero (or the address specifies a storage block containing a PSA) are defined as "Non-PSA Access" for the IOCE.

TABLE 10-VI ACTION IN CHANNEL UPON EXHAUSTION OF COUNT OR RECEIPT OF CHANNEL END

FLAGS IN CURRENT CCW	IMMEDIATE OPERATION	REGULAR OPERATION		
CD CC SLI	Count not exhausted and channel end from device	Count exhausted, end of block at device not reached	Count exhausted and channel end from device	Count not exhausted and channel end from device
0 0 0	End, -	Stop, IL	End, -	End, IL
0 0 1	End, -	Stop, -	End, -	End, -
0 1 0	Chain command	Stop, IL	Chain command	End, IL
0 1 1	Chain command	Chain command	Chain command	Chain command
1 0 0	End, -	Chain data	*	End, IL
1 0 1	End, -	Chain data	*	End, IL**
1 1 0	End, -	Chain data	*	End, IL
1 1 1	End, -	Chain data	*	End, IL**

Explanation:

End - Operation is terminated and an interruption condition is generated in the subchannel.

Stop - Device is signaled to terminate data transfer, but subchannel remains in the working state until channel end is received.

IL - Incorrect length is indicated with the interruption condition.

Chain command - The channel performs command chaining upon receipt of device end.

Chain data - The channel immediately fetches a new CCW for the same operation.

* The situation where the count is zero but data chaining is indicated at the time the device provides channel end cannot validly occur. When data chaining is indicated, the channel fetches the new CCW after transferring the last byte of data designated by the current CCW but before the device provides the next request for data or status transfer. As a result, the channel recognizes the channel end from the device only after it has fetched the new CCW, which cannot contain a count of zero unless a programming error has been made.

** Note that the SLI flag does not suppress the incorrect length indication when the CD flag is on.

JUNE 1, 1971

10.3.6.1 Data Chaining

During data chaining, the new CCW fetched by the channel defines a new storage area for the original I/O operation. Execution of the operation at the I/O device is not affected. Data chaining occurs only when all data designated by the current CCW have been transferred to or from the device and causes the operation to continue, using the storage area designated by the new CCW. The content of the command-code field of the new CCW is ignored, unless it specifies Transfer In Channel.

Data chaining is considered to occur immediately after the last byte of data designated by the current CCW has been transferred to or from the device. When the last byte has been placed in storage or accepted by the device, the new CCW takes over the control of the operation and replaces the pertinent information in the subchannel. If the device sends Channel End after exhausting the count of the current CCW but before transferring any data to or from the storage area designated by the new CCW, the CSW associated with the termination pertains to the new CCW.

If programming errors are detected in the new CCW or during its fetching, the error indication is generated, and the device is signaled to terminate the operation when it attempts to transfer data designated by the new CCW. If the device signals the Channel End condition before transferring any data designated by the new CCW, Program Check or Protection Check is indicated in the CSW associated with the termination. Unless the address of the new CCW is invalid, the location is protected for fetching, or programming errors are detected in an intervening Transfer In Channel command, the content of the CSW pertains to the new CCW. A data address referring to a nonexistent or protected area causes an error indication only after the I/O device has attempted to transfer data to or from the invalid location.

Data chaining during an input operation causes the new CCW to be fetched when all data designated by the current CCW have been placed in storage. On an output operation, the channel may fetch the new CCW from storage ahead of the time data chaining occurs. The earliest such prefetching may occur is when 12 bytes still remain to be transferred under the control of the current CCW. Any programming errors in the prefetched CCW, however, do not affect the execution of the operation until all data designated by the current CCW have been transferred to the I/O device. If the device terminates the operation before all data designated by the current CCW have been transferred, the conditions associated with the prefetched CCW are not indicated to the program.

Only one CCW describing a data area may be prefetched and buffered in the channel. If the prefetched CCW specifies transfer in channel, only one more CCW is fetched before the exhaustion of the current CCW.

PROGRAMMING NOTES:

Data chaining permits information to be rearranged as it is transferred between storage and an I/O device. Data chaining also permits a block of information to be transferred to or from noncontiguous areas of storage, and, when used in conjunction with the skipping function, it permits the program to place in storage selected portions of a block of data.

When, during an input operation, the program specifies data chaining to a location into which data have been placed under the control of the current CCW, the channel fetches the new contents of the location, even if the location contains the last byte transferred under the control of the current CCW. The program, therefore, can use self-describing records; i.e., it can chain to a CCW that has been read under the control of the current CCW. However, since the program is not notified

JUNE 1, 1971

This page
intentionally
left blank.

JUNE 1, 1971

of any data errors until the end of the operation, there is no assurance that the CCW is correct. The CCW in storage may be invalid even though its parity is correct.

10.3.6.2 Command Chaining

During command chaining, the new CCW fetched by the channel specifies a new I/O operation. The channel fetches the new CCW and initiates the new operation upon the receipt of the Device End signal for the current operation. When command chaining takes place, the completion of the current operation does not cause an I/O interruption, and the count indicating the amount of data transferred during the current operation is not made available to the program. For operations involving data transfer, the new command always applies to the next block of data at the device.

Command chaining takes place and the new operation is initiated only if no unusual conditions have been detected in the current operation. If a condition such as Attention, Unit Check, Unit Exception, Incorrect Length, Program Check, Chaining Check or Protection Check has occurred, the sequence of operations is terminated, and the status associated with the current operation causes an interruption condition to be generated. The new CCW in this case is not fetched. The incorrect-length condition does not suppress command chaining if the current CCW has the SLI flag on.

An exception to sequential chaining of CCWs occurs when the I/O device presents the Status Modifier condition with the Device End signal. When command chaining is specified and no unusual conditions have been detected, the combination of Status Modifier and Device End bits causes the channel to fetch and chain to the CCW whose storage address is 16 higher than that of the current CCW.

When both command and data chaining are used, the first CCW associated with the operation specifies the operation to be executed, and the last CCW indicates whether another operation follows.

PROGRAMMING NOTE

Command chaining makes it possible for the program to initiate transfer of multiple blocks of data by means of a single START I/O. It also permits a subchannel to be set up for execution of auxiliary functions, and for data-transfer operations without interference by the program at the end of each operation. Command chaining, in conjunction with the Status Modifier condition, permits the channel to modify the normal sequence of operations in response to signals provided by the I/O device.

10.3.7 SKIPPING

Skipping is the suppression of storage references during an I/O operation. It is defined only for read, read backward, and sense operations and is controlled by the skip flag, which can be specified individually for each CCW. When the skip flag is one, skipping occurs; when zero, normal operation takes place. The setting of the skip flag is ignored in all other operations.

Skipping affects only the handling of information by the channel. The operation at the I/O device proceeds normally, and information is transferred to the channel. The channel keeps updating the count but does not place the information in storage. If the Chain Command or Chain Data flag is one, a new CCW is obtained when the count reaches zero. In the case of data chaining, normal operation is resumed if the skip flag in the new CCW is zero.

JUNE 1, 1971

No checking for invalid or protected data addresses takes place during skipping.

PROGRAMMING NOTE

Skipping, when combined with data chaining, permits the program to place in storage selected portions of a block of information from an I/O device.

10.3.8 PROGRAM-CONTROLLED INTERRUPTION

The program-controlled interruption (PCI) function permits the program to cause an I/O interruption during execution of an I/O operation. The function may be requested by SET PCI instruction or by setting the PCI flag in the CCW. When requested via the CCW this flag can be on either in the first CCW specified by START I/O or in a CCW fetched during chaining. Neither the PCI flag nor the associated interruption affects the execution of the current operation.

Whenever the PCI flag in the CCW is on, the channel attempts to interrupt the program. In the multiplexor channel at least one data byte is transferred to or from storage on a read or write operation before the interruption occurs. In the selector channel, the interruption occurs sometime after the CCW is fetched and is dependent upon channel conditions. Actual interruption will be delayed if this channel is masked.

If chaining occurs before the interruption due to the PCI flag has taken place, the PCI condition is carried over to the new CCW. This carryover occurs both on data and command chaining, and in either case, the condition is propagated through the Transfer In Channel command. The PCI conditions are not stacked; i.e., if another CCW is fetched with a PCI flag before the interruption due to the PCI flag of the previous CCW has occurred, only one interruption takes place.

A CSW containing the PCI bit may be stored by an interruption while the operation is still proceeding or by an interruption or TEST I/O upon the termination of the operation. It cannot be stored by TEST I/O while the subchannel is in the Working state.

When the CSW is stored by an interruption before the operation or chain of operations has been terminated, the command address is eight higher than the address of the current CCW, and the count is correct on multiplexor channels, 0 to 3 bytes high on selector channels. All unit-status bits in the CSW are off. If the channel has detected any unusual conditions, such as Channel Data Check, Program Check, or Protection Check by the time the interruption occurs, the corresponding channel-status bit is one, although the condition in the subchannel is not reset and is indicated again upon the termination of the operation.

Presence of any unit-status bit in the CSW indicates that the operation or chain of operations has been terminated. The CSW in this case has its regular format with the PCI bit added.

However, when the interruption condition due to the PCI flag has been delayed until the operation at the subchannel has been terminated, two interruptions from the subchannel still may take place, with the first interruption indicating and clearing only the PCI condition, and the second providing the CSW associated with the ending status. Whether one or two interruptions occur depends on whether or not the PCI condition has been assigned the highest priority for interruption at time of termination. The TEST I/O addressed to the device associated with an interruption condition in the subchannel clears the PCI condition as well as the one associated with the termination.

JUNE 1, 1971

The setting of the PCI flag is inspected in every CCW except those specifying Transfer In Channel. In a CCW specifying Transfer In Channel, the setting of the flag is ignored. The PCI flag is ignored also during initial program loading.

PROGRAMMING NOTE

Since no unit-status bits are placed in the CSW associated with the termination of an operation of the selector channel by HALT I/O, the presence of a unit-status bit with the PCI bit is not a necessary condition for the operation to be terminated. When the selector channel contains the PCI bit at the time the operation is terminated by HALT I/O, the CSW associated with the termination is indistinguishable from the CSW provided by an interruption during execution of the operation.

Program-controlled interruption provides a means of alerting the program of the progress of chaining during an I/O operation. It permits programmed dynamic main-storage allocation.

10.3.9 COMMANDS

The following table lists the command codes for the six commands and indicates which flags are defined for each command. The flags are ignored for all commands for which they are not defined.

TABLE 10-VII INPUT/OUTPUT COMMANDS

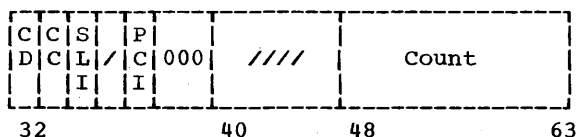
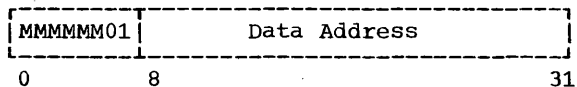
NAMES	FLAGS	CODES	SECTION
Invalid		xxxx 0000	10.3.4
Write	CD CC SLI PCI	mmmm mm01	10.3.9.1
Read	CD CC SLI SKIP PCI	mmmm mm10	10.3.9.2
Read Backward	CD CC SLI SKIP PCI	mmmm 1100	10.3.9.3
Control	CD CC SLI PCI	mmmm mm11	10.3.9.4
Sense	CD CC SLI SKIP PCI	mmmm 0100	10.3.9.5
Transfer In Channel		xxxx 1000	10.3.9.6
Search	CD CC SLI PCI	mmmm mm01	10.3.9.7

Legend

CD = Chain data
CC = Chain command
SLI = Suppress length indication
SKIP = Skip
PCI = Program-controlled interruption
m = Modifier bits
x = Ignored

All flags have individual significance, except that the CC and SLI flags are ignored when the CD flag is on. The SLI flag is ignored on immediate operations, in which case the incorrect-length indication is suppressed regardless of the setting of the flag. The PCI flag is ignored during initial program loading.

10.3.9.1 Write



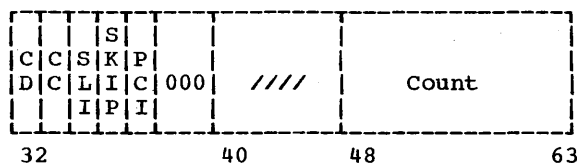
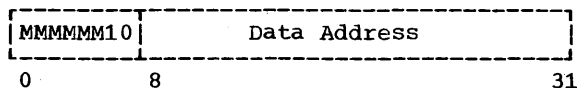
A write operation is initiated at the I/O device, and the subchannel is set up to transfer data from storage to the I/O device. Data in storage are fetched in an ascending order of addresses, starting with the address specified in the CCW.

A CCW used in a write operation is inspected for the CD, CC, SLI, and the PCI flags. The setting of the SKIP flag is ignored. Bit positions 0-5 of the CCW contain modifier bits.

PROGRAMMING NOTE

When writing on devices for which block length is not defined, such as a magnetic tape unit or an inquiry station, the amount of data written is controlled only by the count in the CCW. Every operation terminated under count control causes the incorrect-length indication, unless the indication is suppressed by the SLI flag.

10.3.9.2 Read

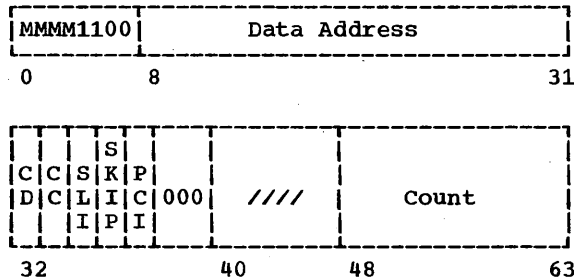


A read operation is initiated at the I/O device, and the subchannel is set up to transfer data from the device to storage. For devices such as magnetic tape drives and card equipment, the bytes of data within a block are provided in the same sequence as written by means of a write command. Data in storage are placed in an ascending order of addresses, starting with the address specified in the CCW.

A CCW used in a read operation is inspected for every one of the five flags -- CD, CC, SLI, SKIP, and PCI. Bit positions 0-5 of the CCW contain modifier bits.

JUNE 1, 1971

10.3.9.3 Read Backwards



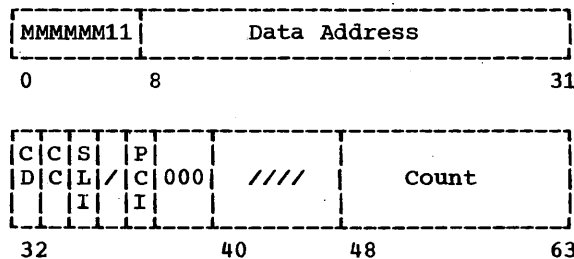
A read-backward operation is initiated at the I/O device, and the subchannel is set up to transfer data from the device to storage. On magnetic tape drives, Read Backward causes reading to be performed with the tape moving backwards. The bytes of data within a block are sent to the channel in a sequence opposite to that on writing. The channel places the bytes in storage in a descending order of address, starting with the address specified in the CCW. The bits within an eight-bit byte are in the same order as sent to the device on writing.

A CCW used in a read-backward operation is inspected for every one of the five flags -- CD, CC, SLI, SKIP, and PCI. Bit positions 0-3 of the CCW contain modifier bits.

PROGRAMMING NOTE

When data chaining is used during a read-backward operation the channel places data in storage in a descending sequence, but fetches CCWs in an ascending sequence. Consequently, if a magnetic tape is to be written so that it can be read in either the forward or backward direction, the CCW must be written at both the beginning and the end of the physical record. If more than one CCW is to be used, the order of the CCWs must be reversed at the end of the record, since the storage areas associated with the CCWs are used in reverse sequence. Furthermore, a CCW used for reading backward must describe the associated storage area by specifying the highest address of the area, whereas it normally contains the lowest address.

10.3.9.4 Control



A control operation is initiated at the I/O device, and the subchannel is set up to transfer data from storage to the device. The device interprets the data as control information. The control information, if any, is fetched from storage in an ascending order of addresses, starting with the address specified in the CCW. A Control

JUNE 1, 1971

command is used to initiate at the I/O device an operation not involving transfer of data, such as backspacing or rewinding magnetic tape.

For most control functions, the entire operation is specified by the modifier bits in the command code, and the function is performed over the I/O interface as an immediate operations, see Immediate Operations (Section 10.4.1.2). If the command code does not specify the entire control function, the data-address field of the CCW designates the location containing the required additional information. This control information may include an order code further specifying the operation to be performed or an address, and is transferred in response to requests by the device.

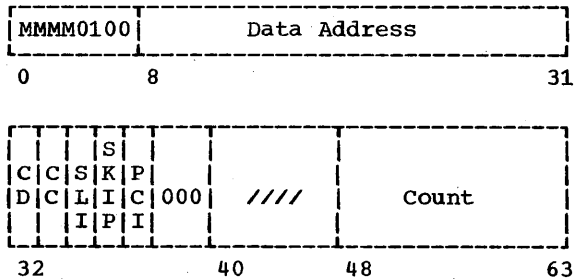
A Control command code containing zeros for the six modifier bits is defined as No Operation. The No Operation order causes the addressed device to respond with Channel End and Device End without causing any action at the device. The order can be executed as an immediate operation, or the device can delay the status until after the initiation sequence is completed. Other operations that can be initiated by means of the Control command depend upon the type of I/O device. These operations and their codes are specified in the SRL publication for the device.

A CCW used in a control operation is inspected for the CD, CC, SLI, and the PCI flags. The setting of the skip flag is ignored. Bit positions 0-5 of the CCW contain modifier bits.

PROGRAMMING NOTE

Since a CCW with a count of zero is invalid, the program cannot use the CCW count field to specify that no data be transferred to the I/O device. Any operation terminated before data have been transferred causes the incorrect-length indication, provided the operation is not immediate and has not been rejected during the initiation sequence. The incorrect-length indication is suppressed when the SLI flag is on.

10.3.9.5 Sense



A sense operation is initiated at the I/O device, and the subchannel is set up to transfer data from the device to storage. The data are placed in storage in an ascending order of addresses, starting with the address specified in the CCW.

Data transferred during a sense operation provide information concerning unusual conditions detected in the last operation and concerning the status of the device. The status information provided by the Sense command is more detailed than that supplied by the unit-status byte, and may describe reasons for the Unit Check indication. It may also indicate, for example, if the device is in the Not Ready state, if the

JUNE 1, 1971

tape drive is in the file-protected state, or if magnetic tape is positioned beyond the end-of-tape mark.

For most devices the first six bits of the sense data describe conditions detected during the last operation. These bits are common to all devices having this type of information and are designated as follows:

TABLE 10-VIII SENSE BIT DESIGNATION

BIT	DESIGNATION
0	Command Reject
1	Intervention Required
2	Bus Out Check
3	Equipment Check
4	Data Check
5	Overrun

The following is the meaning of the first six bits:

Command Reject: The device has detected a programming error. A command has been received which the device is not designed to execute, such as Read issued to a printer, or which the device cannot execute because of its present state, such as Backspace issued to a tape drive with the tape at load point. Command Reject is also indicated when the program issues an invalid sequence of commands, such as Write to a direct-access storage device without previously designating the data block.

Intervention Required: The last operation could not be executed because of a condition requiring some type of intervention at the device. This bit indicates conditions such as an empty hopper in a card punch or the printer being out of paper. It is also turned on when the addressed device is in the Not Ready state, is in test mode, or is not provided on the control unit.

Bus Out Check: The device or the control unit has received a data byte or a command code with an invalid parity over the I/O interface.

During writing, bus-out check indicates that incorrect data have been recorded at the device, but the condition does not cause the operation to be terminated prematurely. Errors on command codes and control information cause the operation to be immediately terminated and suppresses checking for Command Reject and Intervention Required conditions.

Equipment Check: During the last operation, the device or the control unit has detected equipment malfunctioning, such as an invalid card hole count or printer buffer parity error.

Data Check: The device or the control unit has detected a data error other than those included in Bus Out Check. Data Check identifies errors associated with the recording medium and includes conditions such as reading an invalid card code or detecting invalid parity on data recorded on magnetic tape.

On an input operation, Data Check indicates that incorrect data may have been placed in storage. The control unit forces correct parity on data sent to the channel. On writing, this condition indicates that incorrect data may have been recorded at the device. Unless the operation is of a type where the error precludes meaningful continua-

JUNE 1, 1971

tion, data errors on reading and writing do not cause the operation to be terminated prematurely.

Overrun: The channel has failed to respond on time to a request for service from the device. Overrun can occur when data are transferred to or from a nonbuffered control unit operating with a synchronous medium, and the total activity initiated by the program exceeds the capability of the channel. When the channel fails to accept a byte on an input operation, the following data in main storage are shifted to fill the gap. On an output operation, Overrun indicates that data recorded at the device may be invalid. The Overrun bit is also turned on when the device receives the new command too late during command chaining.

All information significant to the use of the device normally is provided in the first two bytes. Any bit positions following those used for programming information contain diagnostic information, which may extend to as many bytes as needed. The amount and the meaning of the status information are peculiar to the type of I/O device and are specified in the SRL publication for the device. The basic Sense command has modifier bits set to zero. This command initiates a sense operation on all devices and cannot cause the Command Reject, Intervention Required, Data Check, or Overrun bits to be turned on. If the control unit detects an equipment error, or invalid parity of the Sense command code, the Equipment Check or Bus Out Check bits are turned on, and Unit Check is sent with Channel End.

Devices that can provide special diagnostic sense information or can be instructed to perform other special functions by means of the Sense command, may define modifier bits for the control of these functions. The special sense operations may be initiated by a unique combination of modifier bits, or a group of codes may specify the same function. Any remaining Sense command codes may be considered invalid, thus causing the Unit Check indication, or may cause the same action as the Basic Sense command, depending upon the type of device.

The sense information pertaining to the last I/O operation is reset the next time the program causes the associated control unit to be selected, unless the selection is due to the execution of TEST I/O, or HALT I/O, or unless the basic sense operation, or a No Operation order is initiated at the control unit.

A CCW used in a sense operation is inspected for every one of the five flags -- CD, CC, SLL, SKIP, and PCI. Bit positions 0-3 of the CCW contain modifier bits.

10.3.9.6 Transfer In Channel

The next CCW is fetched from the location designated by the data-address field of the CCW specifying Transfer In Channel. The Transfer In Channel command does not initiate any I/O operation at the channel, and the I/O device is not signaled of the execution of the command. The purpose of the Transfer In Channel command is to provide chaining between CCWs not located in adjacent double-word locations in an ascending order of addresses. The command can occur in both data and command chaining.

The first CCW designated by the CAW may not specify Transfer In Channel. When this restriction is violated, no I/O operation is initiated, and the Program Check condition is generated. The error causes the status portion of the CSW with the Program Check indication to be stored during the execution of START I/O.

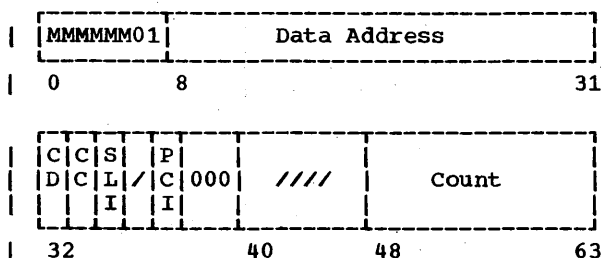
To address a CCW on integral boundaries for double words, a CCW specifying Transfer In Channel must contain zeros in bit positions 29-31. Furthermore, a CCW specifying a Transfer In Channel may not be

JUNE 1, 1971

fetched from a location designated by an immediately preceding Transfer In Channel. When either of these errors is detected or when an invalid address is specified in Transfer In Channel, the Program Check condition is generated. When the Transfer In Channel command designates a CCW in a location protected for fetching, the Protection Check condition is generated. When the Transfer In Channel command designates a CCW in an SE which is Logout-Stopped, the Chaining Check condition is generated. Detection of these errors during data chaining causes the operation at the I/O device to be terminated, whereas during command chaining they cause an interruption condition to be generated.

The contents of the second half of the CCW, bit positions 32-63, are ignored. Similarly, the contents of bit positions 0-3 of the CCW are ignored.

10.3.9.7 Search



On a SEARCH command, a read operation is initiated at the I/O device, and the subchannel is set up to transfer data from storage to the control unit as in a write operation. The control unit compares the information coming from storage with the information read from a record on the I/O device. The Search command operates on only one record. If the search condition is not satisfied, the channel fetches the next CCW in the command chain (8 bytes higher). If the search condition is satisfied, a status modifier indication occurs and the channel skips the next CCW in the command chain and fetches the CCW following (16 bytes higher).

A CCW used in a search operation is inspected for the CD, CC, SLI and PCI flags. The setting of the SKIP flag is ignored. Bit positions 0-5 of the CCW following (16 bytes higher).

PROGRAMMING NOTE

A Search command may be used for consecutive records if the following CCW contains a Transfer In Channel command which specifies the address of the Search CCW. This TIC CCW will be skipped when the search operation is satisfied. The TIC CCW is normally followed by a Read or Write CCW to handle the data transfer.

10.4 TERMINATION OF INPUT/OUTPUT OPERATIONS

When the operation or sequence of operations initiated by START I/O is terminated, the channel and the device generate status conditions. These conditions can be brought to the attention of the program by means of an I/O interruption, by TEST I/O, or, in certain cases, by START I/O. The status conditions, as well as an address and a count indicating the extent of the operation sequence, are presented to the program in the form of a channel status word (CSW).

JUNE 1, 1971

10.4.1 TYPES OF TERMINATION

Normally an I/O operation at the subchannel lasts until the device signals Channel End. The Channel End condition can be signaled during the sequence initiating the operation, or later. When the channel detects equipment malfunctioning or a system reset is performed, the channel disconnects the device without receiving Channel End. The program can force a device on the selector channel to be disconnected prematurely by issuing HALT I/O.

10.4.1.1 Termination at Operation Initiation

After the addressed channel and subchannel have been verified to be in a state where START I/O can be executed, certain tests are performed on the validity of the information specified by the program, and on the availability of the addressed control unit and I/O device. This testing occurs both during the execution of START I/O and during command chaining.

A data-transfer operation is initiated at the subchannel and device only when no programming or equipment errors are detected by the channel, and when the device responds with zero status during the initiation sequence. When the channel detects or the device signals any unusual condition during the initiation of an operation, but Channel End is off, the command is said to be rejected.

Rejection of the command during the execution of START I/O is indicated by the setting of the condition code in the PSW. Unless the device is not operational, the conditions that precluded the initiation are detailed by the portion of the CSW stored by START I/O. The device is not started, no interruption conditions are generated, and the subchannel is not tied up beyond the initiation sequence. The device is immediately available for the initiation of another operation, provided the command was not rejected because of the busy or non-operational condition.

When an unusual condition causes a command to be rejected during initiation of an I/O operation by command chaining, an interruption condition is generated, and the subchannel is not available until the condition is cleared. The conditions are indicated to the program by means of the corresponding status bits in the CSW. The Not Operational condition, which during the execution of START I/O causes condition code 3 to be set, is indicated by means of the Interface Control Check bit. The new operation at the I/O devices is not started.

10.4.1.2 Immediate Operations

Instead of accepting or rejecting a command, the I/O device can signal the Channel End condition immediately upon receipt of the command code. An I/O operation causing the Channel End condition to be signaled during the initiation sequence is called an immediate operation.

When the first CCW designated by the CAW initiates an immediate operation, no interruption condition is generated. If no command chaining occurs, the Channel End condition is brought to the attention of the program by causing START I/O to store the CSW status portion, and the subchannel is immediately made available to the program. The I/O operation however, is initiated, and if Channel End is not accompanied by Device End, the device remains busy. Device End, when subsequently provided by the device, causes an interruption condition to be generated.

When command chaining is specified after an immediate operation and no unusual conditions have been detected during the execution, START I/O does not cause storing of CSW status. The subsequent commands in the

JUNE 1, 1971

chain are handled normally, and the Channel End condition for the last operation generates an interruption condition even if the device provides the signal immediately upon receipt of the command code.

Whenever immediate completion of an I/O operation is signaled, no data have been transferred to or from the device. The data address in the CCW is not checked for validity.

Since a count of zero is not valid, any CCW specifying an immediate operation must contain a nonzero count. When an immediate operation is executed however, incorrect length is not indicated to the program, and command chaining is performed when so specified.

PROGRAMMING NOTE

Control operations for which the entire operation is specified in the command code may be executed as immediate operations. Whether or not the control function is executed as an immediate operation depends on the operation and type of device and is specified in the SRL publication for the device.

10.4.1.3 Termination of Data Transfer

When the device accepts a command, the subchannel is set up for data transfer. The subchannel is said to be working during this period. Unless the channel detects equipment malfunctioning or, on the selector channel, the operation is terminated by HALT I/O, the Working state lasts until the channel receives the Channel End signal from the device. When no command chaining is specified or when chaining is suppressed due to unusual conditions, the Channel End condition causes the operation at the subchannel to be terminated and an interruption condition to be generated. The status bits in the associated CSW indicate Channel End and the unusual conditions, if any. The device can signal Channel End at any time after initiation of the operation, and the signal may occur before any data have been transferred.

For operations not involving data transfer, the device normally controls the timing of the Channel End condition. The duration of data transfer operations may be variable and may be controlled by the device or the channel.

Excluding equipment errors and HALT I/O, the channel signals the device to terminate data transfer whenever any of the following conditions occurs:

- The storage areas specified for the operation are exhausted or filled.
- Program Check condition is detected.
- Protection Check condition is detected.
- Chaining Check condition is detected.

The first of these conditions occurs when the channel has stepped the count in the last CCW associated with the operation to zero. A count of zero indicates that the channel has transferred all information specified by the program. The other three conditions are due to errors and cause premature termination of data transfer. In either case, the termination is signaled in response to a service request from the device and causes data transfer to cease. If the device has no blocks defined for the operation (such as writing on magnetic tape), it terminates the operation and generates the Channel End condition.

The device can control the duration of an operation and the timing of Channel End by blocking of data. On certain operations for which blocks are defined (such as reading on magnetic tape), the device does not provide the Channel End signal until the end of the block is reached,

JUNE 1, 1971

regardless of whether or not the device has been previously signaled to terminate data transfer.

Complete check for the validity of the data address is performed only as data are transferred to or from storage. When the initial data address in the CCW is invalid, no data are transferred during the operation, and the device is signaled to terminate the operation in response to the first service request. On writing, devices such as magnetic tape drives request the first byte of data before any mechanical motion is started, and, if the initial data address is invalid, the operation is terminated before the recording medium has been advanced. However, since the operation has been initiated, the device provides Channel End, and an interruption condition is generated. Whether or not a block at the device is advanced when no data are transferred depends on the type of device and is specified in the SRL publication for the device.

When command chaining takes place, the subchannel appears in the Working state from the time the first operation is initiated until the device signals the Channel End condition of the last operation of the chain. On the selector channel, the device executing the operation stays connected to the channel and the whole channel appears to be in the Working state for the duration of the execution of the chain of operations. On the multiplexor channel an operation in the burst mode causes the channel to appear to be in the Working state only for the duration of the transfer of the burst of data. If Channel End and Device End do not occur concurrently, the device disconnects from the channel after providing Channel End, and the channel can in the meantime communicate with other devices on the interface.

Any unusual conditions cause command chaining to be suppressed and an interruption condition to be generated. The unusual conditions can be detected by either the channel or the device, and the device can provide the indications with Channel End, Control Unit End, or Device End. When the channel is aware of the unusual condition by the time the Channel End signal for the operation is received, the chain is terminated as if the operation during which the condition occurred were the last operation of the chain. The Device End signal subsequently is processed as an interruption condition. When the device signals Unit Check or Unit Exception with Control Unit End or Device End, the subchannel terminates the Working state upon receipt of the signal from the device. The Channel End indication in this case is not made available to the program.

10.4.1.4 Termination by HALT I/O

The instruction HALT I/O causes the current operation at the addressed channel or subchannel to be terminated immediately. The method of termination differs from that used upon exhaustion of count or upon detection of programming errors to the extent that termination by HALT I/O is not contingent on the receipt of a service request from the device.

When HALT I/O is issued to a channel operating in the burst mode, the channel issues the HALT I/O signal to the device regardless of the current activity in the channel and on the interface. If the channel is involved in the data-transfer portion of an operation, data transfer is immediately terminated, and the device is disconnected from the channel. If HALT I/O is addressed to a selector channel executing a chain of operations and the device has already provided Channel End for the current operation, the instruction causes the device to be disconnected and the Chain Command flag to be removed.

When HALT I/O is issued to the multiplexor channel and the channel is not operating in the burst mode, HALT I/O causes the device to be

JUNE 1, 1971

selected, and the HALT I/O signal is issued as the device responds. When command chaining is indicated in the subchannel, HALT I/O causes the Chain Command flag to be turned off.

Termination of an operation by HALT I/O on the selector channel results in up to four distinct interruption conditions. The first one is generated by the channel upon execution of the instruction and is not contingent on the receipt of status from the device. The command address and count in the associated CSW indicate how much data have been transferred, and the channel-status bits reflect the unusual conditions, if any, detected during the operation. If HALT I/O is issued before all data specified for the operation have been transferred, incorrect length is indicated, subject to the control of the SLI flag in the current CCW. The execution of HALT I/O itself is not reflected in CSW status, and all status bits in a CSW due to this interruption condition can be zero. The channel is available for the initiation of a new I/O operation as soon as the interruption condition is cleared.

The second interruption condition on the selector channel occurs when the control unit generates the Channel End condition. The selector channel handles this condition as any other interruption condition from the device after the device has been disconnected from the channel, and provides zeros in the protection key, command address, count, and channel status fields of the associated CSW. The Channel End condition is not made available to the program when HALT I/O is issued to a channel executing a chain of operations and the device has already provided Channel End for the current operation.

Finally, the third and fourth interruption conditions occur when Control Unit End, if any, and Device End are generated. These conditions are handled as for any other I/O operation.

Termination of an operation by HALT I/O on the multiplexor channel causes the normal interruption conditions to be generated. If the instruction is issued when the subchannel is in the data-transfer portion of an operation, the subchannel remains in the Working state until Channel End is signaled by the device, at which time the subchannel is placed in the Interruption Pending state. If HALT I/O is issued after the device has signaled Channel End and the subchannel is executing a chain of operations, the Channel End condition is not made available to the program, and the subchannel remains in the working state until the next status byte from the device is received. Receipt of a status byte subsequently places the subchannel in the Interruption Pending state.

The CSW associated with the interruption condition in the subchannel contains the status bytes provided by the device and the channel, and indicates at what point data transfer was terminated. If HALT I/O is issued before all data areas associated with the current operation have been exhausted or filled, incorrect length is indicated, subject to the control of the SLI flag in the current CCW. The interruption condition is processed as for any other type of termination.

PROGRAMMING NOTE

The CSW associated with a write operation terminated by HALT I/O indicates how many bytes the channel has sent to the device. Since the execution of HALT I/O may cause the loss of the byte of data in transit over the I/O interface and may cause the device to suppress recording of data contained in its buffer, if any, all bytes that have left the channel may not necessarily be recorded at the I/O device.

JUNE 1, 1971

10.4.1.5 Termination Due to Equipment Malfunction

When channel equipment malfunctioning is detected or invalid signals are received over the I/O interface, the recovery procedure and the subsequent states of the subchannels and devices on the channel depend on the type of error. Normally the program is alerted of the termination by an I/O interruption or setting of the condition code and the associated CSW indicates the Channel Control Check or Interface Control Check condition. Equipment malfunctioning may cause the channel to perform the malfunction-reset function.

10.4.2 INPUT/OUTPUT INTERRUPTIONS

Input/output interruptions provide a means for the CE to change its state in response to conditions that occur in I/O devices or channels. These conditions can be caused by the program or by an external event at the device.

10.4.2.1 Interruption Conditions

The conditions causing requests for I/O interruptions to be initiated are called I/O interruption conditions. An I/O interruption condition can be brought to the attention of the program only once and is cleared when it causes an interruption. Alternately, an interruption condition can be cleared by TEST I/O, and conditions generated by the I/O device following the termination of the operation at the subchannel can be cleared by START I/O. The latter include the Attention, Device End, Control Unit End, and the Channel End condition when provided by a device on the selector channel after termination of the operation by HALT I/O.

The device attempts to initiate a request to the channel for an interruption whenever it detects any of the following conditions:

- Channel End
- Control Unit End
- Device End
- Attention
- Unit Check or Unit Exception

When command chaining is specified and is not suppressed due to error conditions, Channel End and Device End do not cause interruption conditions and are not made available to the program. Unit-check and unit-exception conditions cause interruption to be requested only when the conditions are detected during the initiation of a chained command. Once the command has been accepted by the device, Unit Check and Unit Exception do not occur in the absence of Channel End, Control Unit End, Device End, or Attention.

When the channel detects any of the following conditions, it initiates a request for an I/O interruption without having received the status byte from the device:

- PCI flag in a CCW, or
- Execution of HALT I/O on selector channel.

The interruption conditions from the channel can be accompanied by other channel status indications, but none of the device status bits are on when the channel initiates the interruption.

A request for an I/O interruption due to a programming error condition detected during command chaining (e.g., invalid or protected CCW address, invalid command code, count of zero, or two sequential transfer-in-channel commands) may be initiated either by the I/O device

JUNE 1, 1971

or by the channel, depending upon the type of channel. To stack the interruption condition in the device, as occurs on the multiplexor channel, the channel signals the device to respond with a unit-status byte consisting of all zeros on a subsequent scan for interruption conditions. The error indication is preserved in the subchannel.

More than one interruption condition can be cleared concurrently. As an example, when the PCI condition exists in the subchannel at the termination of an operation, the PCI condition is indicated with Channel End, and only one I/O interruption occurs, or only one TEST I/O is needed. Similarly, if the channel-end condition is not cleared until Device End is generated, both conditions may be indicated in the CSW and cleared at the device concurrently.

However, at the time the channel assigns highest priority for interruptions to a subchannel, the channel accepts the status from the device and clears the condition at the device. The interruption condition and the associated status indication are subsequently preserved in the channel. Any subsequent status generated by the device is not included with the status in the channel, even if the status is generated before the CE accepts the condition.

10.4.2.2 Priority of Interruptions

All requests for I/O interruption are asynchronous to the activity in the CE, and interruption conditions associated with more than one I/O device can exist at the same time. The priority among requests is controlled by two types of mechanisms -- one establishes the priority among interruption conditions associated with devices attached to the same channel, and another establishes priority among requests from different channels. A channel requests an I/O interruption only after it has established priority among requests from its devices. The conditions responsible for the requests are preserved in the devices, subchannels or channels until accepted by the CE.

Assignment of priority to requests for interruption associated with devices on any one channel is a function of the type of interruption condition and the position of the device on the I/O interface cable. A device's position on the cable is not related to its address.

The selector channel assigns the highest priority to conditions associated with the portion of the operation in which the channel is involved. These conditions include Channel End, Program Controlled Interruption, execution of HALT I/O in the channel, and errors prematurely terminating a chain of operations. The selector channel cannot handle any interruption conditions other than those due to the PCI flag while operation is in progress.

As soon as the selector channel has cleared the interruption conditions associated with data transfer it starts monitoring devices for Attention, Control Unit End, Device End and Channel End conditions associated with operations terminated by HALT I/O. The highest priority is assigned to the I/O device that first identifies itself on the interface.

On the multiplexor channel the priority among requests for interruption is based only on response from devices. The highest priority is assigned to the device that first identifies itself with an interruption condition or which requests service for data transfer and contains the PCI condition in the subchannel. The PCI, as well as any other condition in the subchannel, cannot cause an I/O interruption unless the device initiates a reference to the subchannel.

Except for conditions associated with termination of data transfer, the current assignment of priority for interruption among devices on a

JUNE 1, 1971

| selector channel may be canceled when START I/O, TEST I/O, or HALT I/O is issued to the channel. Whenever the assignment is canceled, the selector channel resumes monitoring for interruption conditions and reassigns the priority on completion of the activity associated with the I/O instruction.

| The assignment of priority among requests for interruption from channels is based on the type of channel and its address assignment. The priorities of channels are in the order of their addresses, with channel 1 having the highest priority. The interruption priority of multiplexor channels is not fixed, and depends upon the current activity in the channels. Their priorities may be above, below, or between those of the selector channels.

10.4.2.3 Interruption Action

An I/O interruption can occur only when the channel accomodating the device is not masked and after the execution of the current instruction in the CE has been terminated. If a channel has established the priority among requests for interruption from devices while it is masked, the interruption occurs immediately after the termination of the instruction removing the mask and before the next instruction is executed. This interruption is associated with the highest priority condition on the channel. If more than one channel is unmasked concurrently, the interruption occurs from the channel having the highest priority among those requesting interruption.

If the priority among interruption conditions has not yet been established in the channel by the time the mask is removed, the interruption does not necessarily occur immediately after the termination of the instruction removing the mask. This delay can occur regardless of how long the interruption condition has existed in the device or the subchannel.

The interruption causes the current program status word (PSW) to be stored as the old PSW at location 56 of the CE's preferential-storage area and causes the CSW associated with the interruption to be stored at location 64 of the same area. Subsequently, a new PSW is loaded from location 120 of this area, and processing resumes in the state indicated by this PSW. The I/O device or, in the case of Control Unit End, the control unit causing the interruption is identified by the channel address in bit positions 20-23 and by the device address in bit positions 24-31 of the old PSW. The CSW associated with the interruption identifies the condition responsible for the interruption and provides further details about the progress of the operation and the status of the device.

| When an IOCE fails to gain access to the preferential-storage area on an I/O interruption, a preferential-storage area lockout exception is recognized by the controlling CE and a program interruption is taken.

PROGRAMMING NOTE

| When a number of I/O devices on a shared control unit are concurrently executing operations, such as rewinding tape, the initial Device End signals generated on completion of the operations are provided in the order of generation, unless command chaining is specified for the operation last initiated. In the latter case, the control unit provides the Device End signal for the last initiated operation first, and the other signals are delayed until the subchannel is freed. Whenever interruptions due to the Device End signals are delayed either because the channel is masked or the subchannel is busy, the original order of the signals is destroyed.

JUNE 1, 1971

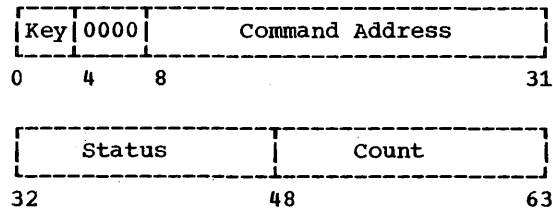
10.4.3 CHANNEL STATUS WORD

The channel status word (CSW) provides to the program the status of an I/O device or the conditions under which an I/O operation has been terminated. The CSW is formed, or parts of it are replaced, in the process of I/O interruptions and during execution of START I/O, TEST I/O, SPCI and HALT I/O. The CSW is placed in the interrupted CE's preferential main storage at location 64 and is available to the program at this location until the time the next I/O interruption occurs or until another I/O instruction causes its content to be replaced, whichever occurs first.

When the CSW is stored as a result of an I/O interruption, the I/O device is identified by the I/O address in the old PSW. The information placed in the CSW by START I/O, TEST I/O, SPCI or HALT I/O pertains to the device addressed by the instruction.

When an IOCE fails to gain access to the preferential-storage on a CSW store pertaining to an I/O instruction, a preferential-storage area lockout exception is recognized by the controlling CE and a program interruption is taken.

The CSW has the following format:



The fields in the CSW are allocated for the following purposes:

Protection Key: Bits 0-3 form the protection key used in the chain of operations at the subchannel.

Command Address: Bits 8-31 form an address that is eight higher than the address of the last CCW used.

Status: Bits 32-47 identify the conditions in the device and the channel that caused the storing of the CSW. Bits 32-39 are obtained over the I/O interface and indicate conditions detected by the device or the control unit. Bits 40-47 are provided by the channel and indicate conditions associated with the subchannel. Each of the 16 bits represents one type of condition, as follows:

JUNE 1, 1971

TABLE 10-IX STATUS BIT DESIGNATION

BIT	DESIGNATION	BIT	DESIGNATION
32	Attention	40	Program Controlled Interruption
33	Status Modifier	41	Incorrect Length
34	Control Unit End	42	Program Check
35	Busy	43	Protection Check
36	Channel End	44	Channel Data Check
37	Device End	45	Channel Control Check
38	Unit Check	46	Interface Control Check
39	Unit Exception	47	Chaining Check

Count: Bits 48-63 form the residual count for the last CCW used.

10.4.4 UNIT STATUS CONDITIONS

The following conditions are detected by the I/O device or control unit and are indicated to the channel over the I/O interface. The timing and causes of these conditions for each type of device are specified in the SRL publication for the device.

When the I/O device is accessible from more than one channel, status due to channel-initiated operations is signaled to the subchannel that initiated the associated I/O operation. The handling of conditions not associated with I/O operations, such as Attention or Device End due to transition from the Not Ready to the Ready state, depends on the type of device and condition and is specified in the SRL publication for the device.

The channel does not modify the status bits received from the I/O device. These bits appear in the CSW as received over the interface.

10.4.4.1 Attention

Attention is generated when the device detects an asynchronous condition that is significant to the program. The condition is interpreted by the program and is not associated with the initiation, execution, or termination of an I/O operation.

The device can signal the attention condition to the channel only when no operation is in progress at the I/O device, control unit, or subchannel. Attention can be indicated with Device End upon completion of an operation, and it can be presented to the channel during the initiation of a new I/O operation. Otherwise, the handling and presentation of the condition to the channel depends on the type of device.

When the device signals Attention during the initiation of an operation, the operation is not initiated. Attention accompanying Device End causes command chaining to be suppressed.

10.4.4.2 Status Modifier

Status Modifier is generated by the device when the device cannot provide its current status in response to TEST I/O, to indicate that the control unit is busy, or when the normal sequence of commands has to be modified.

JUNE 1, 1971

When the Status Modifier condition is signaled in response to TEST I/O and the bit appears in the CSW in the absence of any other status bit, presence of the bit indicates that the device cannot execute the instruction and has not provided its current status. The interruption condition, which may be pending at the device or subchannel, has not been cleared, and the CSW stored by TEST I/O contains zeros in the key, command address, and count fields.

When the Status Modifier bit appears in the CSW together with the Busy bit, it indicates that the busy condition pertains to the control unit associated with the addressed I/O device. The control unit appears busy when it is executing a type of operation that precludes the acceptance and execution of any command or the instructions TEST I/O and HALT I/O or contains an interruption condition for a device other than one addressed. The interruption condition may be due to Control Unit End or, on the selector channel, due to Channel End following the execution of HALT I/O. The Busy state occurs for operations such as backspace tape file, in which case the control unit remains busy after providing Channel End, and for operations terminated on the selector channel by HALT I/O. A control unit accessible from two or more channels appears busy when it is communicating with another channel.

Once the execution of a command has been initiated, the status-modifier indication can be provided only together with Device End. The handling of this set of bits by the channel depends on the operation. If command chaining is specified in the current CCW and no unusual conditions have been detected, presence of Status Modifier and Device End causes the channel to fetch and chain to the CCW whose storage address is 16 higher than that of the current CCW. If the I/O device signals the Status Modifier condition at a time when no command chaining is specified, or when any unusual conditions have been detected, no action is taken in the channel, and the Status Modifier bit is placed in the CSW.

PROGRAMMING NOTE

When the multiplexor channel detects a programming error during command chaining, the interruption condition is queued at the I/O device. On some devices, queuing of the condition may generate the Status Modifier indication, which subsequently appears in the CSW associated with the termination of the operation.

10.4.4.3 Control Unit End

Control Unit End indicates that the control unit has become available for use for another operation.

The Control Unit End condition is provided only by control units that are shared by I/O devices or that are accessible by two or more channels and only when one or both of the following conditions has occurred:

1. The program had previously caused the control unit to be interrogated while the control unit was in the Busy state. The control unit is considered to have been interrogated in the busy state when a command or the instructions TEST I/O or HALT I/O has been issued to a device on the control unit, and the control unit has responded with Busy and Status Modifier in the unit status byte. See "Status Modifier".
2. The control unit detected an unusual condition during the portion of the operation after Channel End had been signaled to the channel. The indication of the unusual condition accompanies Control Unit End.

JUNE 1, 1971

If the control unit remains busy with the execution of an operation after signaling Channel End, but has not been interrogated by the program, Control Unit End is not generated. Similarly, Control Unit End is not provided when the control unit has been interrogated and could perform the indicated function. The latter case is indicated by the absence of Busy and Status Modifier in the response to the instruction causing the interrogation.

When the Busy state of the control unit is temporary, Control Unit End is included with Busy and Status Modifier in response to the interrogation even though the control unit has not yet been freed. The busy condition is considered to be temporary if its duration is short with respect to the program time required to handle an I/O interruption.

The Control Unit End condition can be signaled with Channel End, Device End, or between the two. When Control Unit End is signaled by means of an I/O interruption in the absence of any other status conditions, the interruption may be identified by any address assigned to the control unit. A pending Control Unit End causes the control unit to appear busy for initiation of new operations.

10.4.4.4 Busy

Busy indicates that the I/O device or control unit cannot execute the command or instruction because it is executing a previously initiated operation or because it contains a pending interruption condition. The interruption condition for the addressed device, if any, accompanies the busy indication. If the busy condition applies to the control unit, Busy is accompanied by Status Modifier. On command chaining the busy indication is caused only when Attention is generated at the device.

The following table lists the conditions when the Busy bit (B) appears in the CSW and when it is accompanied by the Status Modifier bit (SM). A double hyphen (--) indicates that the Busy bit is off; an asterisk (*) indicates that the CSW status is not stored or an I/O interruption cannot occur; and the (cl) indicates that the interruption condition is cleared and the status appears in the CSW. The abbreviation DE stands for Device End, while CU stands for control unit.

TABLE 10-X INDICATION OF THE BUSY CONDITION IN CHANNEL STATUS WORD

CONDITION	CSW STATUS STORED BY:			
	START I/O	TEST I/O	HALT I/O	I/O INT.
Subchannel available				
DE or Attention in device	B,cl	--,cl	*	--,cl
Device working, CU available	B	B	*	*
CU End or Channel End in CU:				
for the addressed device	B,cl	--,cl	*	--,cl
for another device	B,SM	B,SM	*	--,cl
CU working	B,SM	B,SM	*	*
Interruption pend. in subchannel for the addressed device because of:				
chaining terminated by attention condition (NOTE)	*	B,cl	*	B,cl
other type of termination	*	--,cl	*	--,cl
Subchannel working				
CU available	*	*	--	*
CU working	*	*	B,SM	*

JUNE 1, 1971

NOTE:

The busy bit is included in the status associated with a pending interruption condition from the subchannel only when a chain of commands has been prematurely terminated due to Attention and no interruption was pending in the channel at the time of chaining.

10.4.4.5 Channel End

Channel End is caused by the completion of the portion of an I/O operation involving transfer of data or control information between the I/O device and the channel. The condition indicates that the subchannel has become available for use for another operation.

Each I/O operation causes a Channel End condition to be generated, and there is only one Channel End for an operation. The Channel End condition is not generated when programming errors or equipment malfunctions are detected during initiation of the operation. When command chaining takes place, only the Channel End of the last operation of the chain is made available to the program. The Channel End condition is not made available to the program when a chain of commands is prematurely terminated because of an unusual condition indicated with Control Unit End or Device End or during the initiation of a chained command.

The instant within an I/O operation when Channel End is generated depends on the operation and the type of device. For operations such as writing on magnetic tape, the Channel End condition occurs when the block has been written. On devices that verify the writing, Channel End may or may not be delayed until verification is performed, depending on the device. When magnetic tape is being read, the Channel End condition occurs when the gap on tape reaches the read-write head. On devices equipped with buffers, such as the IBM 1443 N1 Printer (bar line printer), the Channel End condition occurs upon completion of data transfer between the channel and the buffer. During control operations, Channel End is generated when the control information has been transferred to the devices, although for short operations the condition may be delayed until completion of the operation. Operations that do not cause any data to be transferred can provide the Channel End condition during the initiation sequence.

A Channel End condition pending in the control unit causes the control unit to appear busy for initiation of new operations. Unless the operation has been performed on the selector channel and has been terminated by HALT I/O, a pending Channel End causes the subchannel to be in the Interruption Pending state.

10.4.4.6 Device End

Device End is caused by the completion of an I/O operation at the device or, on some devices, by manually changing the device from the Not Ready to the Ready state. The condition indicates that the I/O device has become available for use for another operation.

Each I/O operation causes a Device End condition, and there is only one Device End to an operation. The Device End condition is not generated when any programming or equipment malfunction is detected during initiation of the operation. When command chaining takes place, only the Device End of the last operation of the chain is made available to the program unless an unusual condition is detected during the initiation of a chained command.

The Device End condition associated with an I/O operation is generated either simultaneously with the Channel End condition or later. On data-transfer operations on devices such as magnetic tape drives, the

JUNE 1, 1971

device terminates the operation at the time Channel End is generated, and both Device End and Channel End occur together. On buffered devices, such as an IBM 1443 Printer, the Device End condition occurs upon completion of the mechanical operation. For control operations, Device End is generated at the completion of the operation at the device. The operation may be completed at the time Channel End is generated or later.

When command chaining is specified in the subchannel, receipt of the Device End signal, in the absence of any unusual conditions, causes the channel to initiate a new I/O operation.

10.4.4.7 Unit Check

Unit Check indicates that the I/O device or control unit has detected an unusual condition that is detailed by the information available to a Sense command. Unit Check may indicate that a programming or an equipment error has been detected, that the Not Ready state of the device has affected the execution of the command or instruction, or that an exceptional condition other than the one identified by Unit Exception has occurred. The Unit Check bit provides a summary indication of the conditions identified by sense data.

An error condition causes the Unit Check indication only when it occurs during the execution of a command or TEST I/O, or during some activity associated with an I/O operation. Unless the error condition pertains to the activity initiated by a command and is of immediate significance to the program, the condition does not cause the program to be alerted after Device End has been cleared.

The not-ready state causes a unit-check indication when it precludes a satisfactory execution of the command or the command by its nature tests the state of the device. Unit check is not indicated if the command is properly executed even though the device has become not-ready during, or as a result of the operation. An example of the latter is a printer running out of forms. Similarly, unit check is not indicated if the command can be executed with the device not-ready.

Selection of a device in the not-ready state does not cause a unit-check indication when the sense command is issued and whenever an interruption condition is pending for the addressed device at the control unit. On the other hand, when no interruption condition is pending for the addressed device at the control unit, the control unit signals unit-check when TEST I/O or any command other than sense is issued to a not-ready device, including the no-operation control command. In the case of no operation, the command is rejected, and Channel End and Device End do not accompany Unit Check.

If the device detects during the initiation sequence that the command cannot be executed, Unit Check is presented to the channel and appears without Channel End, Control Unit End, or Device End. Such unit status indicates that no action has been taken at the device in response to the command. If the condition precluding proper execution of the operation occurs after execution has been started, Unit Check is accompanied by Channel End, Control Unit End, or Device End, depending on when the condition was detected. Any errors associated with an operation, but detected after Device End has been cleared, are indicated by signaling Unit Check with Attention.

Errors, such as invalid command code or invalid command code parity, do not cause Unit Check when the device is working or contains a pending interruption condition at the time of selection. Under these circumstances, the device responds by providing the Busy bit and indicating the pending interruption condition, if any. The command code invalidity is not indicated.

JUNE 1, 1971

Termination of an operation with the Unit Check indication causes command chaining to be suppressed.

PROGRAMMING NOTE

If a device becomes Not Ready upon completion of a command, the ending interruption condition can be cleared by TEST I/O without generation of Unit Check due to the Not Ready state, but any subsequent TEST I/O issued to the device causes a Unit Check indication.

10.4.4.8 Unit Exception

Unit Exception is caused when the I/O device detects a condition that usually does not occur. Unit Exception includes conditions such as recognition of a tape mark and does not necessarily indicate an error. It has only one meaning for any particular command and type of device.

The Unit Exception condition can be generated only when the device is executing an I/O operation, or when the device is involved with some activity associated with an I/O operation and the condition is of immediate significance to the program. If the device detects during the initiation sequence that the operation cannot be executed, Unit Exception is presented to the channel and appears without Channel End, Control Unit End, or Device End. Such unit status indicates that no action has been taken at the device in response to the command. If the condition precluding normal execution of the operation occurs after the execution has been started, Unit Exception is accompanied by Channel End, Control Unit End, or Device End, depending on when the condition was detected. Any unusual conditions associated with an operation, but detected after Device End has been cleared, is indicated by signaling Unit Exception with Attention.

A command does not cause Unit Exception when the device responds during the initial selection with Busy status to the command.

Termination of an operation with the Unit Exception indication causes command chaining to be suppressed.

10.4.5 CHANNEL STATUS CONDITIONS

The following conditions are detected and indicated by the channel. Except for the conditions caused by equipment malfunctioning, they can occur only while the subchannel is involved with the execution of an I/O operation.

10.4.5.1 Program Controlled Interruption

The Program Controlled Interruption condition is generated when the channel fetches a CCW with the Program Controlled Interruption (PCI) flag on, or, when the PCI bit is turned on in a subchannel by the SET PCI instruction. The interruption due to the PCI flag takes place as soon as possible after the CCW takes control of the operation but may be delayed an unpredictable amount of time because of masking of the channel or other activity in the system. The interruption due to SET PCI causes the CSW to be stored without waiting for the current CCW to be completed, but may also be delayed due to masking or other activity.

Detection of the PCI condition does not affect the progress of the I/O operation.

10.4.5.2 Incorrect Length

Incorrect length occurs when the number of bytes contained in the storage areas assigned for the I/O operation is not equal to the number

JUNE 1, 1971

of bytes requested or offered by the I/O device. Incorrect length is indicated for one of the following reasons:

Long Block on Input: During a read, read-backward, or sense operation, the device attempted to transfer one or more bytes to storage after the assigned storage areas were filled. The extra bytes have not been placed in storage. The count in the CSW is zero.

Long Block on Output: During a write or control operation the device requested one or more bytes from the channel after the assigned storage areas were exhausted. The count in the CSW is zero.

Short Block on Input: The number of bytes transferred during a read, read-backward, or sense operation is insufficient to fill the storage areas assigned to the operation. The count in the CSW is not zero.

Short Block on Output: The device terminated a write or control operation before all information contained in the assigned storage areas was transferred to the device. The count in the CSW is not zero.

The Incorrect Length indication is suppressed when the current CCW has the SLI flag and does not have the CD flag. The indication does not occur for immediate operations and for operations rejected during the initiation sequence.

Presence of the Incorrect Length condition suppresses command chaining unless the SLI flag in the CCW is on or unless the condition occurs in an immediate operation. See Table 10-VI in the chaining section of this manual for the effect of the CD, CC, and SLI flags on the indication of incorrect length.

10.4.5.3 Program Check

Program Check occurs when programming errors are detected by the channel. The condition can be due to the following causes:

Invalid CCW Address Specification: The CAW or the Transfer In Channel command does not designate the CCW on intergral boundaries for double words. The three low-order bits of the CCW address are not zero.

Invalid CCW Address: The channel has attempted to fetch a CCW from a location outside the storage of the particular installation, or outside the configured storage, or storage assigned by the storage address translator for a particular IOCE. An invalid CCW address can occur in the channel because the program has specified an invalid address in the CAW or in the Transfer In Channel command or because on chaining the channel has stepped the address above the highest available location.

Invalid Command Code: The command code in the first CCW designated by the CAW or in a CCW fetched on command chaining has four low-order zeros. The command code is not tested for validity during data chaining.

Invalid Count: A CCW other than a CCW specifying Transfer In Channel contains the value zero in bit positions 48-63.

Invalid Data Address: The channel has attempted to transfer data to or from a location outside the storage of the particular installation, or outside the configured storage, or storage assigned by the storage address translator for the particular IOCE. An invalid data address can occur in the channel because the program has specified an invalid address in the CCW or because the channel has stepped the address above the highest available address or, on reading backward, below zero.

JUNE 1, 1971

Invalid CAW Format: The CAW does not contain zeros in bit positions 4-7.

Invalid CCW Format: A CCW other than a CCW specifying Transfer In Channel does not contain zeros in bit positions 37-39.

Invalid Sequence: The first CCW designated by the CAW specifies Transfer In Channel or the channel has fetched two successive CCWs both of which specify Transfer In Channel.

Detection of the Program Check condition during the initiation of an operation causes execution of the operation to be suppressed. When the condition is detected after the device has been started, the device is signaled to terminate the operation the next time it requests or offers a byte of data. The Program Check condition causes command chaining to be suppressed.

10.4.5.4 Protection check

Protection Check occurs when the channel attempts to place data in or fetch data or a CCW from a portion of main storage that is protected for the current operation on the subchannel. The protection key associated with the I/O operation does not match the key of the addressed main-storage location, and neither of the protection keys is zero.

When the Protection Check condition occurs during the fetching of a CCW that specifies the initiation of an I/O operation, the operation is not initiated. When Protection Check is detected after the device has been started, the device is signaled to terminate the operation the next time it requests or offers a byte of data. The condition causes command chaining to be suppressed.

10.4.5.5 Channel Data Check

Channel Data Check indicates that the channel has detected a parity error in the information transferred to or from main storage during an I/O operation. This information includes the data read or written, as well as the information transferred as data during a sense or control operation. The error may have been detected anywhere inboard the I/O interface in the channel. Channel Data Check may be indicated for parity errors detected in data that are referred to by the channel but do not participate in the operation.

Whenever a parity error on I/O data is indicated by means of Channel Data Check, the channel forces correct parity on all data received over the I/O interface and correct parity is forced on all data placed in storage. On an output operation, the parity of the data is not changed when Channel Data Check is indicated.

A condition indicated as Channel Data Check causes command chaining to be suppressed, but does not affect the execution of the current operation. Data transfer proceeds to normal completion, and an I/O interruption condition is generated when the device presents Channel End. No log-out or reset occurs, and the detection of the error does not affect the state of the channel or device.

10.4.5.6 Channel Control Check

Channel Control Check is caused by any machine malfunction affecting channel controls. The condition includes parity errors on CCW and data addresses and parity errors on the contents of the CCW. Conditions responsible for Channel Control Check may cause the contents of the CSW to be invalid and conflicting. The CSW as generated by the channel has correct parity.

JUNE 1, 1971

Detection of the Channel Control Check condition causes the current operation, if any, to be immediately terminated and causes the channel to perform the Malfunction Reset function.

The presence of the Channel Control Check bit in a stored CSW indicates that a diagnostic procedure has been performed by the IOCE (Chapter 12).

10.4.5.7 Interface Control Check

Interface Control Check is caused by any invalid signal on the I/O interface. The condition is detected by the channel and usually indicates malfunctioning of an I/O device. It can be due to the following reasons:

1. The address or status byte received from a device has invalid parity.
2. A device responded with an address other than the address specified by the channel during initiation of an operation.
3. During command chaining the device appeared not operational or indicated the busy condition without providing any other status bits.
4. A signal from a device occurred at an invalid time or had invalid duration.

Detection of the Interface Control Check condition causes the current operation, if any, to be immediately terminated and causes the channel to perform the Malfunction Reset function.

The presence of the Interface Control Check bit in a stored CSW indicates that a diagnostic procedure has been performed by the IOCE (Chapter 12).

10.4.5.8 Chaining Check

Chaining Check is caused by channel overrun during data chaining on input operations. The condition occurs when the I/O data rate is too high for the particular resolution of data addresses. Chaining Check cannot occur on output operations.

Detection of the Chaining Check condition causes the I/O device to be signaled to terminate the operation. It causes command chaining to be suppressed.

Chaining Check is also caused by an IOCE making a non-PSA storage access to an SE which is in logout-stop state. Detection of the chaining-check condition causes all I/O operations on that subchannel to be terminated.

10.4.6 CONTENT OF CHANNEL STATUS WORD

The content of the CSW depends on the condition causing the storing of the CSW and on the programming method by which the information is obtained. The status portion always identifies the condition that caused storing of the CSW. The protection key, command address, and count fields may contain information pertaining to the last operation or may be set to zero, or the original contents of these fields at location 64 of the CE's preferential-storage area may be left unchanged.

JANUARY 31, 1973

TNL: GN31-3005

10.4.6.1 Information Provided by Channel Status Word

Conditions associated with the execution or termination of an operation at the subchannel cause the whole CSW to be replaced. Such a CSW can be stored only by an I/O interruption or by TEST I/O. Except for conditions associated with command chaining and equipment malfunctioning, the storing can be caused by the PCI or Channel End condition and by the execution of HALT I/O on the selector channel. The contents of the CSW are related to the current values of the corresponding quantities, although the count is unpredictable after Program Check, Protection Check, Chaining Check, and after an interruption due to the PCI flag.

A CSW stored upon the execution of a chain of operation pertains to the last operation the channel executed or attempted to initiate. Information concerning the preceding operations is not preserved and is not made available to the program.

When an unusual condition causes command chaining to be suppressed, the premature termination of the chain is not explicitly indicated in the CSW. A CSW associated with a termination due to a condition occurring at Channel End time contains the Channel End bit and identifies the unusual condition. When the device signals the unusual condition with Control Unit End or Device End, the Channel End indication is not made available to the program, and the channel provides the current protection key, command address, and count, as well as the unusual indication, with the Control Unit End or Device End bit in the CSW. The command address and count fields pertain to the operation that was executed.

When the execution of a chain of commands is terminated by an unusual condition detected during initiation of a new operation, the command address and count fields pertain to the rejected command. Except for conditions caused by equipment malfunctioning, termination at the initiation time can occur because of Attention, Unit Check, Unit Exception, or Program Check, and causes both the Channel End and Device End bits in the CSW to be off.

A CSW associated with conditions occurring after the operation at the subchannel has been terminated contains zeros in the protection key, command address, and count fields, provided the conditions are not cleared by START I/O. These conditions include Attention, Control Unit End, and Device End, and Channel End when it occurs after termination of an operation on the selector channel by HALT I/O.

When the above conditions are cleared by START I/O, only the status portion of the CSW is stored, and the original contents of the protection key, command address and count fields in location 64 of the CE's preferential-storage area are preserved. Similarly, only the status bits of the CSW are changed when the command is rejected or the operation at the subchannel is terminated during the execution of START I/O or whenever HALT I/O causes CSW status to be stored.

Errors detected during execution of the I/O operation do not affect the validity of the CSW unless the Channel Control Check or Interface Control Check conditions are indicated. Channel Control Check indicates that equipment errors may have been detected which can cause any part of the CSW, as well as the address in the PSW identifying the I/O device, to be invalid. Interface Control Check indicates that the address identifying the device or the status bits received from the device may be invalid. The channel forces correct parity on invalid CSW fields.

JUNE 1, 1971

10.4.6.2 Protection Key

A CSW stored to reflect the progress of an operation at the subchannel contains the protection key used in that operation. The content of this field is not affected by programming errors detected by the channel or by the condition causing termination of the operation.

PROGRAMMING NOTE

Since storage protection applies only to main storage references, the protection key has no significance when the command address and data address in the last CCW used both reference MACH storage.

10.4.6.3 Command Address

When the CSW is formed to reflect the progress of the I/O operation at the subchannel, the command address is normally eight higher than the address of the last CCW used in the operation.

The following table lists the contents of the command address field for all conditions that can cause the CSW to be stored. The conditions are listed in order of priority; that is, if two conditions are indicated or occur, the CSW appears as indicated for the condition higher on the list. The programming errors listed in the table refer to conditions included in Program Check.

10.4.6.4 Status

The status bits identify the conditions that have been detected during the I/O operation, that have caused a command to be rejected, or that have been generated by external events.

When the channel detects several error conditions, all conditions may be indicated or only one may appear in the CSW, depending on the conditions. Conditions associated with equipment malfunctioning have precedence, and whenever malfunctioning causes an operation to be terminated, Channel Control Check, Interface Control Check, or Channel Data Check is indicated, depending on the condition. When an operation is terminated by Program Check, Protection Check, or Chaining Check, the channel identifies the condition responsible for the termination and may or may not indicate incorrect length. When a data error has been detected and the operation is terminated prematurely because of a Program Check, Protection Check, or Chaining Check, both Data Check and the programming error are identified.

If the CCW fetched on command chaining contains the PCI flag but a programming error in the contents of the CCW or an unusual condition signaled by the device precludes the initiation of the operation, the PCI bit appears in the CSW associated with the interruption condition. Similarly, if device status or a programming error in the contents of the CCW causes the command to be rejected during execution of START I/O, the CSW stored by START I/O contains the PCI bit. However, when the channel detects a programming error in the CAW or in the first CCW, the PCI bit may unpredictably appear in a CSW stored by START I/O without the PCI flag being on in the first CCW associated with the START I/O.

Conditions detected by the channel are not related to those identified by the I/O device.

The following table summarizes the handling of status bits. The table lists the states and activities that can cause status indications to be created and the methods by which these indications can be placed in the CSW.

JUNE 1, 1971

TABLE 10-XI CONTENT OF CHANNEL STATUS WORD ADDRESS FIELD

CONDITION	CONTENT
Channel Control Check	Unpredictable
Status stored by START I/O	Unchanged
Status stored by HALT I/O	Unchanged
Invalid CCW address spec in Transfer In Channel (TIC)	Address of TIC + 8
Invalid CCW address in TIC	Address of TIC + 8
Invalid CCW address generated	First invalid CCW address + 8
Invalid command code	Address of invalid CCW + 8
Invalid count	Address of invalid CCW + 8
Invalid data address	Address of invalid CCW + 8
Invalid CCW format	Address of invalid CCW + 8
Invalid sequence - 2 TIC's	Address of second TIC + 8
Protection Check	Address of invalid CCW + 8
Chaining Check	Address of last-used CCW + 8
Termination under count control	Address of last-used CCW + 8
Termination by I/O device	Address of last-used CCW + 8
Termination by HALT I/O	Address of last-used CCW + 8
Suppression of command chaining due to Unit Check or Unit Exception with Device End or Control Unit End	Address of last CCW used in the completed operation + 8
Termination on command chaining by Attention, Unit Check, or Unit Exception	Address of CCW specifying the new operation + 8
Program Controlled Interruption	Address of last-used CCW + 8
Interface control check	Address of last-used CCW + 8
Channel End after HALT I/O on selector channel	Zero
Control Unit End	Zero
Device End	Zero
Attention	Zero
Busy	Zero
Status Modifier	Zero

JUNE 1, 1971

TABLE 10-XII TIME AND METHOD OF CREATING AND STORING STATUS INFORMATION

STATUS	WHEN I/O IS IDLE	WHEN SUBCHANNEL WORKING	UPON TERMINATION OF OPERATION			DURING COMMAND CHAINING	BY START I/O	BY TEST I/O	BY HALT I/O	BY I/O INTERRUPT	BY SET PCI
			AT SUBCHANNEL	AT CONTROL UNIT	AT DEVICE						
Attention	C*				C	C*	S	S		S	S
Status Modifier					C	C	CS	CS	CS	S	S
Control Unit End				C*			CS	CS	CS	S	S
Busy						C	CS	CS	CS	S	S
Channel End			C*A	C*H		C*#A	C/S	S		S	S
Device End	C*				C*A	C #	C/S	S		S	S
Unit Check			C	C	C	C*	CS	CS		CS	S
Unit Exception			C	C	C	C*	CS	S		S	S
Program-Controlled Interruption		C*	C*			C*	CS	S		S	S
Incorrect Length		C	C					S		S	S
Program Check		C	C			C*	CS	S		S	S
Protection Check		C	C			C*	CS	S		S	S
Channel Data Check		C	C					S		S	S
Channel Control Check	C*	C*	C*	C*	C*	C*	CS	CS	CS	CS	CS
Interface Control Check	C*	C*	C*	C*	C*	C*	CS	CS	CS	CS	CS
Chaining Check		C	C			C*	CS	S		S	S

NOTES

- C The channel or device can create or present the status condition at the indicated time. A CSW or its status portion is not necessarily stored at this time.

Conditions such as Channel End or Device End are created at the indicated time. Other conditions may have been created previously, but are made accessible to the program only at the indicated time. Examples of such conditions are Program Check and Channel Data Check, which are detected while data are transferred, but are made available to the program only with Channel End, unless the PCI flag or equipment malfunctioning have caused an interruption condition to be generated earlier.
- S The status indication is stored in the CSW at the indicated time.

An S appearing alone indicates that the condition has been created previously. The letter C appearing with the letter S indicates that the status condition did not necessarily exist previously in the form that causes the program to be alerted, and may have been created by the I/O instruction or I/O interruption. For example, equipment malfunctioning may be detected during an I/O interruption, causing Channel Control Check or Interface Control Check to be indicated; or a device may signal the Control Unit Busy condition in response to interrogation by an I/O instruction, causing Status Modifier, Busy, and Control Unit End to be indicated in the CSW.
- * The status condition generates or, in the case of Channel Data Check may generate an interruption condition.

Channel End and Device End do not result in interruption conditions when command chaining is specified and no unusual conditions have been detected.
- # This status indication can be created at the indicated time only by an immediate operation.
- H When an operation on the selector channel has been terminated by HALT I/O, Channel End indicates the termination of the data-handling portion of the operation at the control unit.

JUNE 1, 1971

10.4.6.5 Count

The residual count, in conjunction with the original count specified in the last CCW used, indicates the number of bytes transferred to or from the area designated by the CCW. When an input operation is terminated, the difference between the original count in the CCW and the residual count in the CSW is equal to the number of bytes transferred to storage, on an output operation, the difference is equal to the number of bytes transferred to the I/O device.

The following table lists the contents of the count field for all conditions that can cause the CSW to be stored. The conditions are listed in the order of priority; that is, if two conditions are indicated or occur, the CSW appears as for the condition higher on the list.

TABLE 10-XIII CONTENT OF CHANNEL STATUS WORD COUNT FIELD

CONDITION	CONTENT
Channel Control Check	Unpredictable
Status stored by START I/O	Unchanged
Status stored by HALT I/O	Unchanged
Program Check	Unpredictable
Protection Check	Unpredictable
Chaining Check	Unpredictable
Termination under count control	Correct
Termination by I/O device	Correct
Termination by HALT I/O	Correct
Suppression of command chaining due to Unit Check or Unit Exception with Device End or Control Unit End	Correct, Residual count of last CCW used in the completed operation.
Termination on command chaining by Attention, Unit Check, or Unit Exception	Correct. Original count of CCW specifying the new operation.
Program Controlled Interruption	Unpredictable
Interface control check	Correct
Channel End after HALT I/O on selector channel	Zero
Control Unit End	Zero
Device End	Zero
Attention	Zero
Busy	Zero
Status Modifier	Zero

JUNE 1, 1971

This page
intentionally
left blank.

JUNE 1, 1971

CHAPTER 11: SYSTEM CONTROL FACILITIES

The IBM 9020D System and the IBM 9020E System each have two principal control facilities: the Computing Element Control Panel and the System Console in the 9020D System, the Computing Element Control Panel and the Configuration Console in the 9020E System. The computing element (CE) control panel is identical on both systems. The Configuration Console (CC) provides most of the control facilities available on the system console (SC) plus various status indications for the Display Generators and Radar Keyboard Multiplexors in the display subsystem.

The computing element (CE) control panel (Figure 11-1) contains the switches and lights necessary to operate and control a computing subsystem. A subsystem consists of the CE, storage, I/O control element (IOCE) channels, and on line control units and I/O devices.

Controls are divided into two sections: operator control and intervention, and maintenance control. Maintenance controls may also be available on individual storage, channel, and control-unit frames.

The system console (SC) (Figure 11-2) is designed to be the central monitoring and control position in the IBM 9020D System. It contains the switches and lights necessary to selectively operate and control each CE in the system. Conversely, all critical console functions are duplicated on the control panel of each CE or elsewhere in the system.

The configuration console (CC) (Figure 11-3) is designed to be the central monitoring and control position in the IBM 9020E System. It contains the switches and lights necessary to selectively operate and control each CE in the system. In addition, the CC provides various status indications for the display subsystem. Reference Configuration Console SRL for description of the status function.

Operator controls on the CE control panel, the system console, and the configuration console, which could be inadvertently used to disrupt the 9020D or 9020E programmed operation are activated by key-operated System Interlock switches. A system interlock switch is provided on the system console, configuration console, and on each CE control panel.

Table 11-III summarizes the state-dependent 9020D and 9020E operator control and intervention switches on the system console, configuration console, and on the CE control panel as they relate to each other, to the CE program state, the configuration state, and to the setting of the system interlock switches.

11.1 CONTROL PANEL FUNCTIONS

The main 9020D and 9020E system functions provided by the CE control panel, system console, and the configuration console are the ability to store and display information in storage and in registers; and to load initial program information.

11.1.1 STORE AND DISPLAY

The store-and-display function permits manual intervention in the progress of a program. The store-and-display function may be provided by a supervisor program in conjunction with proper I/O equipment and the interrupt switch.

JUNE 1, 1971

In the absence of an appropriate supervisor program, the operator controls permit the CE to be placed in the stopped state, and subsequently to store and display information in main storage and display storage, in general and floating-point registers, and in the instruction-address part of the PSW. The stopped state is achieved at the end of the current instruction when the stop switch is pressed, when single instruction execution is specified, or when a preset address is reached. (Note that storage protection is ignored during manual store-and-display operations.) Once the desired intervention is completed the CE can be started again.

All basic store-and-display functions can be simulated by a supervisor program. The stopping and starting of the CE in itself does not cause any alteration in program execution other than the time element involved. (The transition from operating to stopped state is described under Stopped State, in Chapter 8).

Machine checks occurring during store-and-display functions do not interrupt or log immediately but may, in some cases, create a pending machine-check interruption. This interruption request can be removed by a CE-reset. Otherwise the interruption, when not masked off, is taken when the CE is again in the operating state.

11.1.2 INITIAL PROGRAM LOADING

Initial program loading (IPL) is provided for the initiation of processing when the contents of storage or the PSW are not suitable for further processing.

The IBM 9020D and 9020E Systems each perform two types of initial program loading: sub-system IPL and system IPL. Subsystem IPL is performed at the CE control panel by pressing the load pushbutton switch with the system interlock switch off. The Load switch on the CE is not enabled in states Three and Two. (See Table 11-III.) Subsystem IPL may also be performed from the 9020E Configuration Console, regardless of the CE state.

System IPL is performed at either the CE control panel, or at the system console or configuration console by pressing the Load switch with the associated system interlock switch on.

System IPL is initiated manually by selecting a CE (if performed at the system or configuration console), an IOCE channel and input device with the Load-Unit switches, a main storage element (SE) with the Main Storage Select switch, and subsequently pressing the Load switch with the associated system Interlock switch on.

Note: On a 9020E System, the Main Storage Select switch positions 6-10 represent display elements (DE) 1-5 and should not be selected during a system IPL since no data paths exist between IOCEs and DEs. Also, the Enable System IPL switch must be held depressed if it is desired to initiate a System IPL from the Configuration Console.

Pressing the Load switch causes the selected CE to issue a system reset to all configurable elements in the system. For the duration of the reset, no element responding to it will accept a signal from any other element in the system. In addition, the load light on the system console or configuration console and the issuing CE's control panel are turned on, and the manual lights are turned off.

When the CE is reset, it suspends all instruction processing, interruptions, and timer updating. The content of the "physical" preferential-storage base address register, and position one of the address translator (ATR) are set to reference the selected SE. Posi-

JUNE 1, 1971

tions 2-10 of the ATR are set to zeros. The content of the "logical" preferential-storage base address register is set to reference the lowest 4096-byte block of the selected SE. Upon the completion of the system-reset, the configuration control register in each major element in the system is set to state Three, and where provided, the inhibit logout-stop bit is set on. A configuration mask is accepted from the selected CE only, and communication paths are established (where applicable) with the CE, the selected SE and the selected IOCE only.

The selected IOCE has its ATR set to correspond to the setting of the ATR in the selected CE. All other CEs and IOCEs, not in Zero state (test switch on), have their ATRs set to zeros.

Upon the completion of the configuration operation, the CE causes the selected IOCE to accept the effective preferential-storage base address, channel and unit address, and to initiate a read operation. The selected IOCE then reads a record from the selected input device into the SE. The first 24 bytes read, or the entire record if less than 24 bytes, are placed in the preferential storage locations 0-23. Storage protection, program controlled interruption, and possible incorrect length indication are ignored. The doubleword read into location 8 is used by the IOCE as the next channel command word (CCW) for a subsequent input command. When chaining is specified in this CCW, the operation continues with the CCW in location 16. The IOCE automatically adds the preferential-storage base address to the address portion of each CCW, permitting the operation to be extended to load the first 128K bytes of the selected SE, if so desired.

After the input operation is performed, the I/O address is stored in bits 20-31 of the first word in the selected storage element. Bits 0-19 remain unchanged.

The system IPL procedure causes the selected elements to be configured into an operational subsystem with all other elements not in the Zero state (with Test switch on) receptive to communications from the subsystem. The selected CE may then configure other elements into the system as needed.

The CE subsequently fetches the doubleword in location 0 of the selected SE as a new PSW and proceeds under control of the new PSW. The load light is turned off. When the I/O operations and PSW loading are not completed satisfactorily, the CE idles, and the load light remains on.

Subsystem IPL is similar to system IPL with the following differences: the subsystem reset generated when the load pushbutton is pressed at the CE (System Interlock switch off) or at the Configuration Console (System Interlock switch on) is issued to all configured IOCEs, the input device selected in the load-unit switches, and to the SE designated in the main storage select switch. The contents of the configuration control registers in these elements are not altered, with the exception of the inhibit logout stop bits, which are set on.

Note: On a 9020E System, the Main Storage Select switch positions 6-10 represent display elements (DE) 1-5 and should not be selected during a subsystem IPL since no data paths exist between IOCEs and DEs.

The content of the ATR in the selected CE remains unchanged. The ATR in the selected IOCE is set to correspond to the setting of the ATR in the selected CE. All other CEs and IOCEs have their ATRs unchanged. The content of the "logical" preferential-storage base address register is set to reference the lowest 4096-byte block in the selected SE. The content of the "physical" preferential-storage base address register is set to reference the selected SE.

JUNE 1, 1971

Subsystem IPL implies that the elements involved have been previously configured into a subsystem, and that the selected CE's ATR has been set prior to initiating the subsystem IPL operation.

PROGRAMMING NOTES

Initial program loading resembles a START I/O that specifies the I/O device selected in the load/unit switches and a zero protection key. The CCW for this START I/O has a read command, zero data address, a byte count of 24, command-chain flag on, suppress-length-indication flag on, program-controlled-interruption flag off, and a virtual command address of zero.

Initial program loading reads new information into the first six words of storage. Since the remainder of the IPL program may be placed in any section of storage, it is possible to preserve such areas of storage as the timer and PSW locations, which may be helpful in program debugging. Initial program loading occurs in the first 128K of storage only.

When the PSW in location 0 of the preferential-storage has bit 14 set to one, the CE is in the wait state after the IPL procedure (the manual, system and load lights are off, and the wait light is on). Interruptions that become pending during IPL are taken before instruction execution.

11.1.3 PROGRAM STATUS WORD RESTART

The PSW restart facility provided on the CE control panel allows the IBM 9020D or 9020E System to restart without performing an IPL operation. An SE is selected at the CE control panel, the system interlock switch is turned on, and the PSW restart switch is pressed. The operation causes all elements in the system not in State 0 (Test) to receive a system reset and to be configured to the selected CE. Position one of the selected CE's ATR and "physical" PSBAR are set to reference the selected SE. "Logical" PSBAR is set to zero, i.e., to reference the lowest 4096-byte block in the selected SE. Since the IOCE designated by the load unit switches does not enter into a PSW restart, its ATR is set to zeros, as are those in other CEs and IOCEs. A preloaded PSW is fetched from location 0 of the selected storage element and processing is resumed. The selected CE may therefore bring the remaining elements back to their former states and thus restart the system.

Note: Since Preferential Storage Areas are not resident in display elements (DE), the Main Storage Select switch should not be positioned at 6-10 for a PSW Restart on a 9020E System, since these positions represent DES 1-5.

Pressing the PSW Restart switch, with the system interlock switch off, initiates a subsystem restart. The contents of the configuration control registers and address translators are not altered. "Logical" PSBAR references the lowest 4096-byte block in the selected SE in the lowest position of ATR in which it appears. "Physical" PSBAR is set to reference the selected SE. Fetching of the PSW and resumption of processing occurs as above.

11.2 OPERATOR CONTROL SECTION

The main 9020D and 9020E System functions provided by the operator controls are the control and indication of power, the indication of subsystem status, operator to machine communication, and initial program loading.

JUNE 1, 1971

Table 11-I lists all operator controls and lights by the names on the panel or controls, and describes their implementation.

Certain controls are only available on the CE control panel, others are only on the system console, or configuration console, and some are on all. Table 11-I and the functional description of each control indicate its location for quick reference by "CE" and/or "SC" and/or "CC" in parentheses opposite each name.

11.2.1 EMERGENCY PULL (SC,CC)

This switch is a red, mechanically latched, pull-type switch. When the switch is pulled, power to all IBM 9020D or 9020E System Elements will be removed within a two-second period. Power will still be present in the power compartments of each system element. This switch must be manually reset at the console by maintenance personnel to restore system power. Power will automatically sequence up on all units if their power switch is on.

11.2.2 ELEMENT MASTER POWER OFF PULL SWITCH (CE,CC)

Pulling the Element Master Power Off (MPO) switch turns off all power beyond the power-entry terminal on the computing element, or configuration console.

The switch latches in the out-position and can be restored to its in-position by maintenance personnel only.

When the element MPO pull switch is in the out-position, the power-on switch is ineffective.

11.2.3 POWER ON/OFF SWITCH (CE, SC, CC)

On the CE control panel, this toggle switch initiates the power-on and power-off sequences for the computing element.

At the completion of the power-on sequence, a power-on reset is performed in such a manner that the CE performs no instructions until explicitly directed. The contents of main storage, including its protection keys, remain preserved during both power-on and power-off sequencing.

On the system console, this lighted pushbutton switch causes the console power to be sequenced on or off.

On the configuration console, three power on/off toggle switches are provided, one for each of the two reconfiguration control units and one for the system console control unit. Each switch controls power to its associated control unit in the CC.

11.2.4 POWER SEQUENCE COMPLETE (CE, SC, CC)

On the CE control panel, the power sequence complete light is on when all DC voltages have been turned on in the CE following power-on.

On the system console, this light is on when all DC voltages have been turned on in the console following power-on by the console power-on/off switch.

On the configuration console, three power sequence complete lights provide indication of power on for each of the associated control units.

JUNE 1, 1971

TABLE 11-I CONTROLS AND LIGHTS ON OPERATOR CONTROL SECTION

NAME	IMPLEMENTATION	SECTION
Emergency Pull (SC,CC)	Pull Switch	11.2.1
Element MPO Pull (CE,CC)	Pull Switch	11.2.2
Power On/Off (CE,SC,CC)	Switches	11.2.3
Power Sequence Complete (CE,SC,CC)	Light	11.2.4
Wait (CE,SC,CC)	Light	11.2.5
Manual (CE,SC,CC)	Light	11.2.6
System (CE)	Light	11.2.7
Test (CE)	Light	11.2.8
Load (CE,SC,CC)	Light	11.2.9
Load Unit (CE,SC,CC)	Three Rotary Switches	11.2.10
Load (CE,SC,CC)	Pushbutton Switch	11.2.11
Main Storage Select (CE,SC,CC)	Rotary Switch	11.2.12
System Interlock (CE,SC,CC)	Key Operated Switch	11.2.13
CE Select (SC,CC)	Rotary Switch	11.2.14
Stop (CE,SC,CC)	Pushbutton Switch	11.2.15
Rate (CE,SC,CC)	Rotary Switch	11.2.16
Start (CE,SC,CC)	Pushbutton Switch	11.2.17
Storage Select (CE,SC,CC)	Lever Switch	11.2.18
Address (CE,SC,CC)	Lever Switches (24)	11.2.19
Data (CE,SC,CC)	Lever Switches (32 on SC,CC), (64 on CE)	11.2.20
Store (CE,SC,CC)	Pushbutton Switch	11.2.21
Display (CE,SC,CC)	Pushbutton Switch	11.2.22
Set IC (CE,SC,CC)	Pushbutton Switch	11.2.23
Interrupt (CE,SC,CC)	Pushbutton Switch	11.2.24
PSW Restart (CE)	Pushbutton Switch	11.2.25
State Three (CE,SC,CC)	Light	11.2.26
State Two (CE,SC,CC)	Light	11.2.27
State One (CE,SC,CC)	Light	11.2.28
State Zero (CE,SC,CC)	Light	11.2.29
Lamp Test (CE,SC,CC)	Pushbutton Switch	11.2.30
Control CE (SC,CC)	Lever Switch (4)	11.2.31
Control CE Activate (SC,CC)	Pushbutton Switch	11.2.32
Address Compare (CE,SC,CC)	Lever or Rotary Switch	11.2.33
Address Compare Enable (SC,CC)	Pushbutton Switch	11.2.34
Sense Switches (SC,CC)	Lever Switches (6)	11.2.35
Reader/Punch Printer Select (SC,CC)	Rotary Switch	11.2.36
Reader/Punch Printer Enable Switch Indicator (SC,CC)	Pushbutton Switch, Lighted	11.2.37
Printer Keyboard Select (SC)	Rotary Switch	11.2.38
Printer Keyboard Enable Switch-indicator (SC)	Pushbutton Switch, Lighted	11.2.39
Console Reset (SC,CC)	Pushbutton Switch	11.2.40
Bell Reset (SC)	Pushbutton Switch	11.2.41
Buzzer Reset (SC)	Pushbutton Switch	11.2.42
All Stop (SC,CC)	Pushbutton Switch	11.2.43
Invalid Selection (SC,CC)	Light	11.2.44
Test Mode (SC,CC)	Light	11.2.45
360 Mode (CE)	Pushbutton Switch, Lighted	11.2.46
Enable System IPL (CC)	Pushbutton Switch	11.2.47
I/O Interface Enable/Disable (SC,CC)	Toggle Switch	11.2.48
I/O Interface Disabled (SC,CC)	Light	11.2.49
Fault Reset (CC)	Pushbutton Switch	11.2.50
Buzzer/Bell Reset (CC)	Pushbutton Switch	11.2.51

JUNE 1, 1971

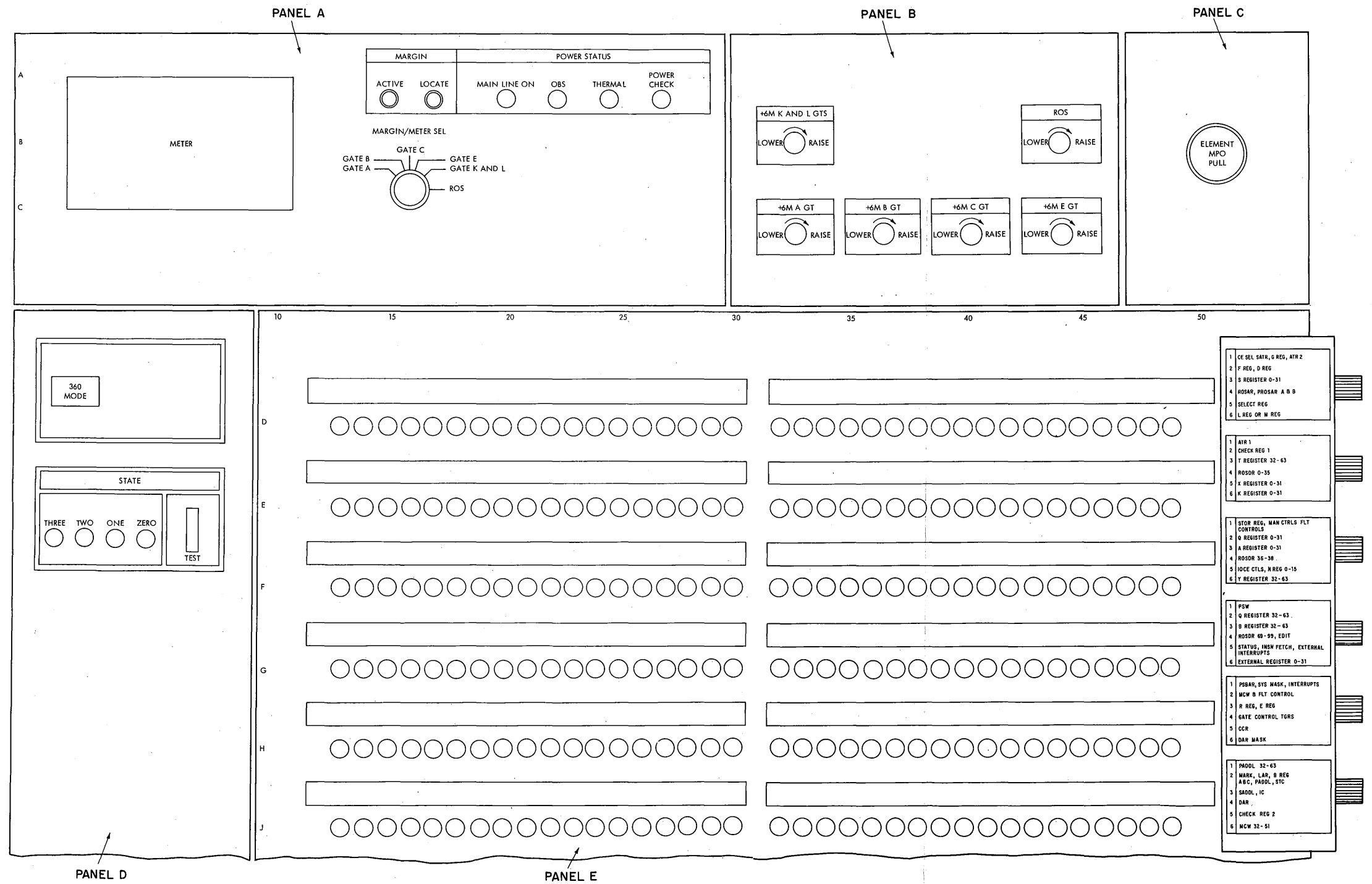


Figure 11-1 Computing Element (CE) Control Panel (1 of 2)

JUNE 1, 1971

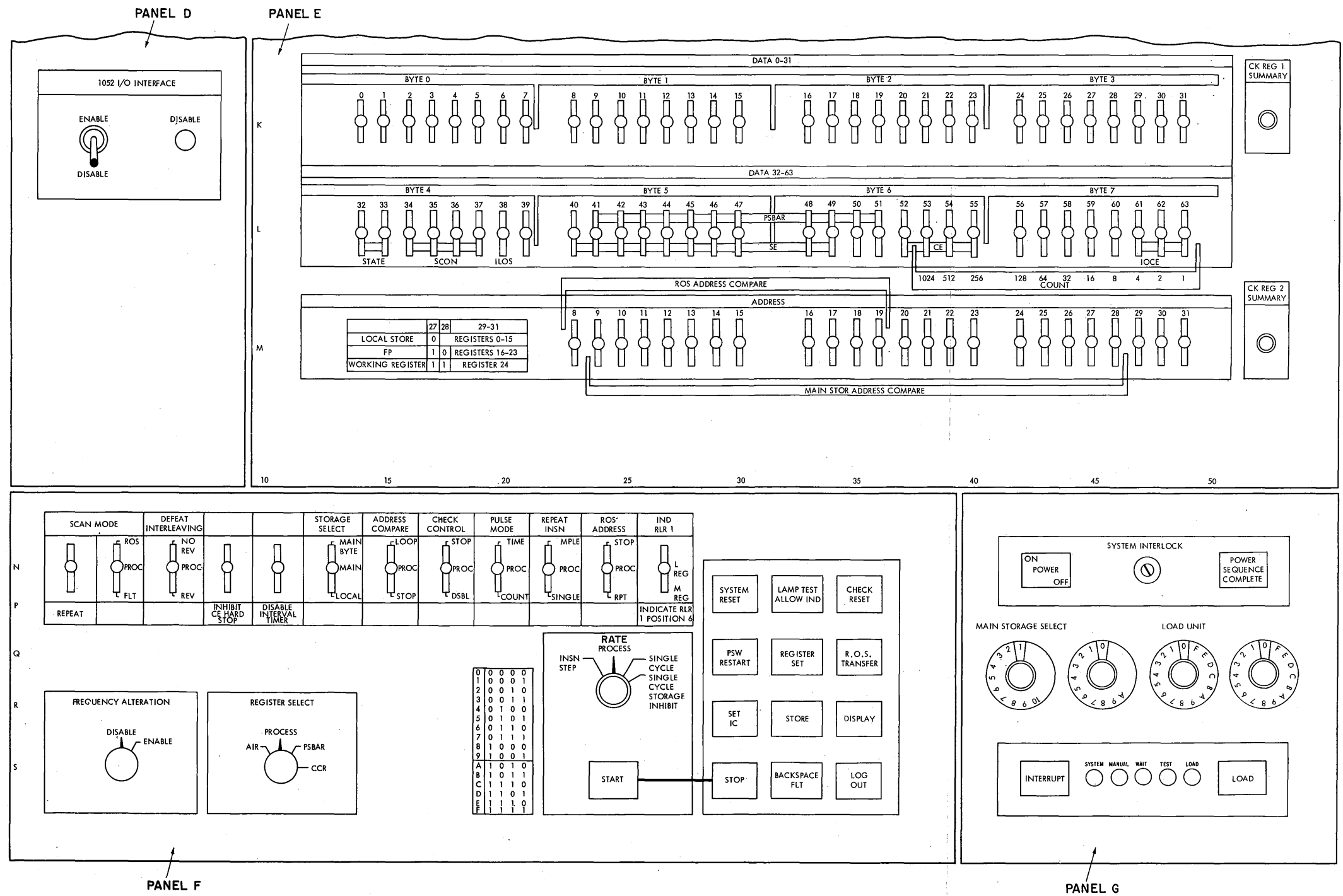
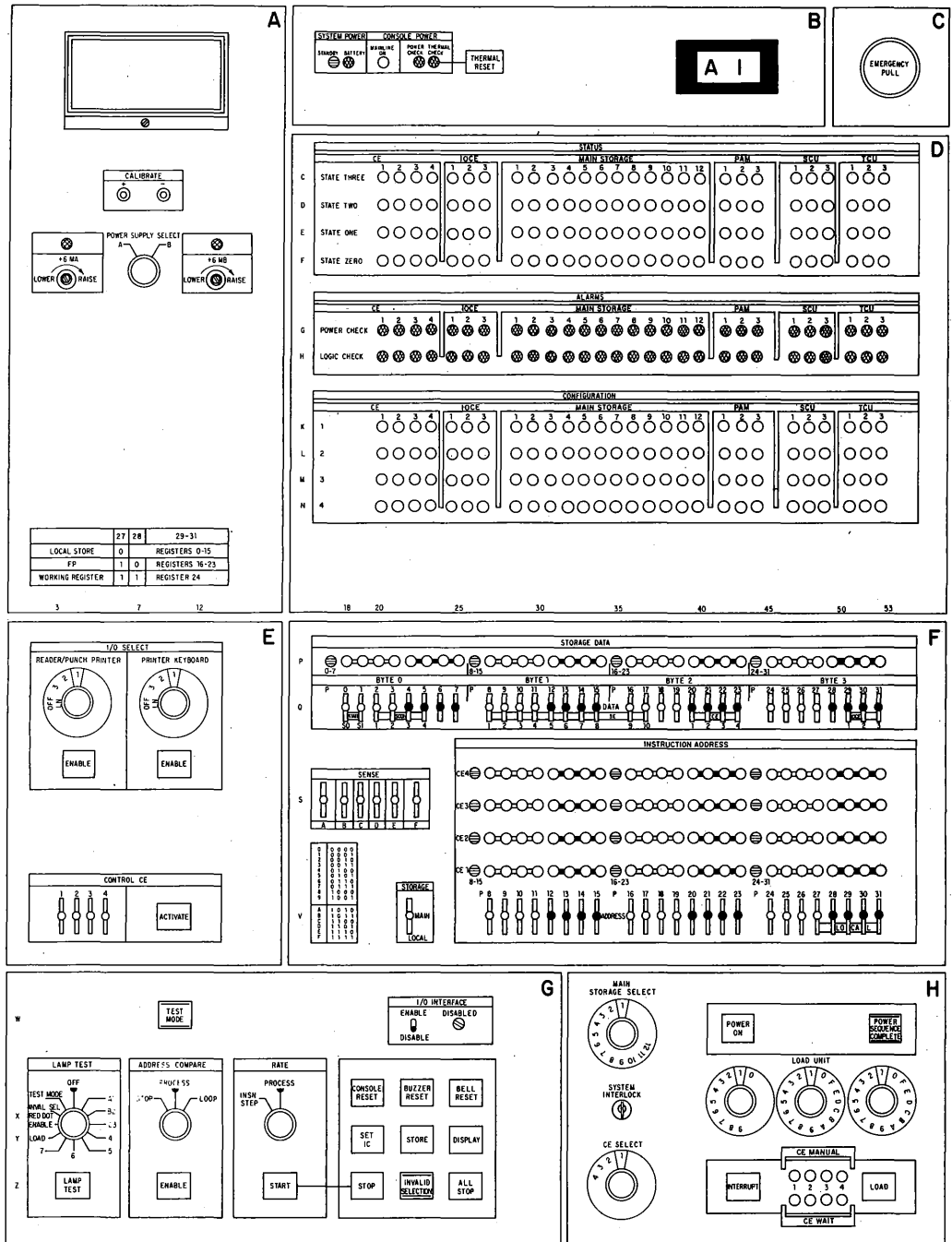


Figure 11-1 Computing Element (CE) Control Panel (2 of 2)

JUNE 1, 1971



| Figure 11-2 9020D System Console (SC) Panel

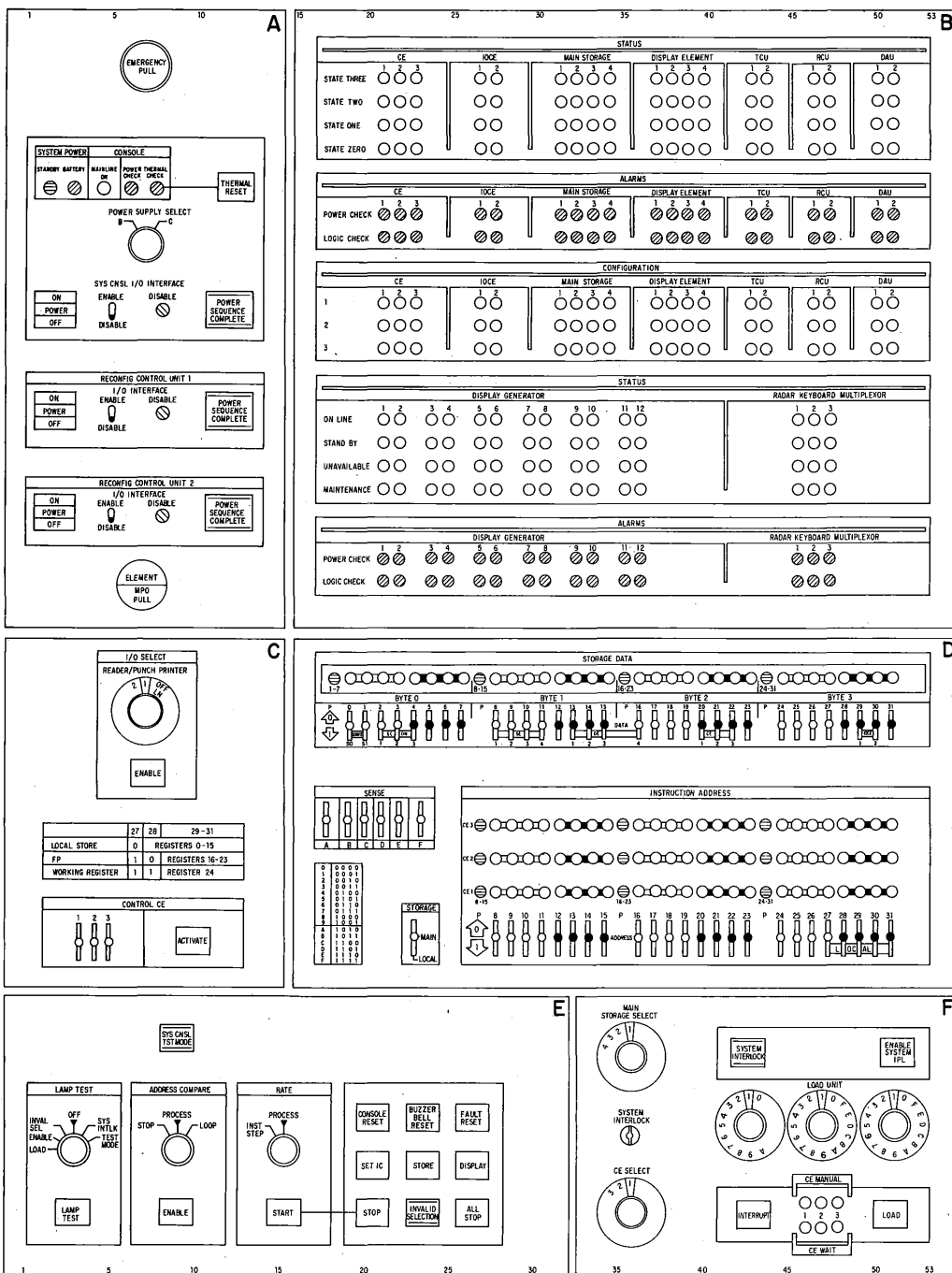


Figure 11-3 9020E Configuration Console (CC) Panel

JUNE 1, 1971

| 11.2.5 WAIT LIGHT (CE, SC, CC)

| The wait light is on when the CE is in the wait state.

11.2.6 MANUAL LIGHT (CE, SC, CC)

The manual light is on when the CE is in the stopped state. Several of the manual controls are effective only when the CE is stopped, i.e., the manual light is on.

11.2.7 SYSTEM LIGHT (CE)

| The system light is on when the CE is operating and running.

PROGRAMMING NOTE

The states indicated by the wait and manual lights are independent of each other; however, the state of the system light is not independent of the state of these two lights. The following table shows the possible conditions.

TABLE 11-II STATES OF WAIT, MANUAL, AND SYSTEM LIGHTS

System Light	Manual Light	Wait Light	CE State
OFF	OFF	OFF	Power Off
OFF	OFF	ON	Operating, Waiting
OFF	ON	OFF	Stopped, Running
OFF	ON	ON	Stopped, Waiting
ON	OFF	OFF	Operating, Running

11.2.8 TEST LIGHT (CE)

Indicates when a manual control is not in its normal position.

11.2.9 LOAD LIGHT (CE, SC, CC)

The load light is on during initial program loading; it is turned on when the load key is pressed, and is turned off after the read operation and the loading of the new PSW is completed successfully.

The load light on the system console and configuration console is combined with the load switch.

11.2.10 LOAD-UNIT SWITCHES (CE, SC, CC)

Three rotary switches provide the 12-bit address of the channel and I/O device to be used for initial program loading.

The left switch has 11 positions labeled 0 to 9, A and selects the channel. The other two switches have 16 positions, each labeled with the hexadecimal characters 0-9, A-F, and select the device.

11.2.11 LOAD SWITCH (CE, SC, CC)

The load switch is pressed to start initial program loading.

The load switch on the system console and the configuration console is activated by the system interlock switch. When pressed on the system console it causes a "system reset" and initiates a "system IPL." A system IPL from the configuration console requires the enable system IPL switch to be held depressed while using the load switch.

The load switch on the CE control panel causes a "subsystem reset" and initiates a "subsystem IPL" when pressed with the system Interlock switch off. When pressed with the system interlock switch on, it causes a "system reset" and initiates a "system IPL". A "subsystem IPL" is initiated from the configuration console when the system interlock switch is on, the selected CE is in State 1 or 0 and the load switch is depressed.

11.2.12 MAIN STORAGE SELECT SWITCH (CE, SC, CC)

This switch provides a capability to select one of up to 10 storage elements at a CE control panel, system console, or configuration console for IPL.

NOTE: On 9020E System only 1-5 are valid storage elements, 6-10 representing display element positions.

11.2.13 SYSTEM INTERLOCK SWITCH (CE, SC, CC)

These switches enable the system console, configuration console, and the CE control panel to perform certain system functions. The functions enabled are shown in Table 11-III. If the system Interlock switches on either the system console or configuration console and the selected CE's control panel are on simultaneously, the system console or configuration console takes precedence.

11.2.14 COMPUTING ELEMENT SELECT (SC, CC)

A rotary switch is provided to select a computing element. The following switches on the system console and configuration console are made available for the selected CE; Interrupt, Load, Rate, Start, Stop, Store, Display, Set IC, Address-Compare, Enable and Control CE Activate.

The Invalid Selection light is also conditioned by the CE select switch.

JUNE 1, 1971

TABLE 11-III STATE DEPENDENT OPERATOR SWITCHES

	CE PROGRAM STATE		SYSTEM CONSOLE OR CONFIGURATION CONSOLE CONTROLS				CE PANEL CONTROLS															
			SYSTEM INTERLOCK SWITCH		SYSTEM INTERLOCK SWITCH		SYSTEM INTERLOCK SWITCH		SYSTEM INTERLOCK SWITCH													
	OFF	ON	OFF	ON	OFF	ON	OFF	ON														
	O	S	CE STATE				CE STATE				CE STATE				CE STATE							
	P	T	3	2	1	0	T	3	2	1	0	T	3	2	1	0	T	3	2	1	0	T
	R	P																				
ACTIVATE		X						X	X	X	X											
ADDRESS COMPARE	X							X	X	X	X		X	X	X		X	X	X	X	X	X
ADDRESS SWITCHES*								X	X	X	X		X	X	X		X	X	X	X	X	X
ALL STOP	X							X	X	X	X											
CE SELECT*								X	X	X	X											
CONTROL CE*								X	X	X	X											
DATA SWITCHES*								X	X	X	X		X	X	X		X	X	X	X	X	X
DISPLAY		X						X	X	X	X		X	X	X		X	X	X	X	X	X
ELEMENT MPO	X	X										X	X	X	X	X	X	X	X	X	X	X
EMERGENCY PULL	X	X	X	X	X	X	X	X	X	X	X	X										
ENABLE (FOR ADDR COMP)	X							X	X	X	X											
ENABLE (FOR I/O SELECT)	X	X	X	X	X	X	X	X	X	X	X	X										
INTERRUPT	X	X						X	X	X	X		X	X	X		X	X	X	X	X	X
I/O SELECT SWITCHES	X	X	X	X	X	X	X	X	X	X	X	X										
LOAD	X	X						X	X	X	X		X	X	X		X	X	X	X	X	X
LOAD UNIT*								X	X	X	X		X	X	X		X	X	X	X	X	X
MAIN STORAGE SELECT*								X	X	X	X		X	X	X		X	X	X	X	X	X
POWER OFF+	X	X	X	X	X	X	X	X	X	X	X	X										X
PSW RESTART	X	X						X	X	X	X		X	X	X		X	X	X	X	X	X
RATE	X	X						X	X	X	X		X	X	X		X	X	X	X	X	X
SENSE SWITCHES	X		X	X	X	X	X	X	X	X	X	X										
SET IC		X						X	X	X	X		X	X	X		X	X	X	X	X	X
START		X						X	X	X	X		X	X	X		X	X	X	X	X	X
STORAGE SELECT SWITCH*								X	X	X	X		X	X	X		X	X	X	X	X	X
STORE		X						X	X	X	X		X	X	X		X	X	X	X	X	X
STOP	X							X	X	X	X		X	X	X		X	X	X	X	X	X
SYSTEM INTERLOCK SWITCH	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
TEST SWITCH	X	X											X	X							X	X
S/360 MODE SWITCH	X	X											X	X	X						X	X

NOTES: X - Yes, Control is Functional + - Not Interlocked on System Console, but
 T - State Zero, Test Switch on System Console Control only turns off
 * - Static, Requires use of System Console Power. Not Interlocked
 on Configuration Console, but Configuration Console Control only turns off
 Power to the System Console Portion of
 the Configuration Console. The Recon-
 figuration Control Units housed within
 the Configuration Console have sepa-
 rate, State-Dependent Power Control
 Switches.-

JUNE 1, 1971

11.2.15 STOP SWITCH (CE, SC, CC)

The stop switch is pressed to cause the selected CE to enter the stopped state.

PROGRAMMING NOTE

Pressing the stop switch has no effect when a continuous string of interruptions is performed or when the CE is unable to complete an instruction because of machine malfunction. The effect of pressing the switch is indicated by the manual light's coming on as the CE enters the stopped state.

The transition from operating state to stopped state occurs at the end of instruction execution, prior to the fetching of the next instruction. When the CE is in the wait state, the transition takes place immediately. All interruptions which are pending and which are not masked off are taken, causing the old PSW to be stored and the new one fetched before entering the stopped state.

11.2.16 RATE SWITCH (CE, SC, CC)

This rotary switch indicates the manner in which instructions are to be performed.

The CE control panel switch has four positions; PROCESS, INSTRUCTION STEP, SINGLE-CYCLE, and SINGLE CYCLE STORAGE INHIBIT. The system console, and configuration console switches do not have the SINGLE-CYCLE position, nor the SINGLE CYCLE STORAGE INHIBIT position. In the process position the CE starts operating at normal speed when the start switch is pressed.

When the start switch is pressed with the rate switch in the INSTRUCTION STEP position one complete instruction is performed, and all pending, not masked interruptions are subsequently taken. The CE next returns to the stopped state.

Any instruction can be executed with the rate switch set to INSTRUCTION STEP. Input/output operations are completed to the interruption point. When the CE is in the wait state, no instruction is performed, but pending interruptions, if any, are taken before the CE returns to the stopped state. Initial program loading is completed with the loading of the new PSW before any instruction is performed. The timer is not updated while the rate switch is set to INSTRUCTION STEP.

In the SINGLE-CYCLE position, the CE advances by its minimum clock amount for each depression of the start switch, returning to the stopped state each time. The timer is not updated while the rate switch is set

JUNE 1, 1971

to single cycle. This position is not provided on the system console switch nor on the configuration console.

In the SINGLE CYCLE STORAGE INHIBIT position, the CE operation will be the same as SINGLE CYCLE except that all storage references will be blocked. This position is not provided on either the system console or configuration console switch.

The test light is on when the rate switch on the CE control panel is not set to PROCESS.

The position of the rate switch, when enabled by the system Interlock switch, should be changed only while the CE is in the stopped state. Otherwise, unpredictable results occur.

11.2.17 START SWITCH (CE, SC, CC)

The start switch is pressed to start instruction execution in the manner defined by the rate switch.

Pressing the start switch after a normal stop causes instruction processing to continue as if no stop had occurred, provided that the rate switch is in the PROCESS or INSTRUCTION-STEP position. Pressing the start switch after a system or subsystem reset causes the instruction designated by the Instruction Counter to be executed.

The switch is effective only while the CE is in the stopped state.

11.2.18 STORAGE-SELECT SWITCH (CE, SC, CC)

The storage area to be addressed by the address switches is selected by the storage-select switch.

The CE control panel switch allows selection of a doubleword in either main or display storage (address specified by ADDRESS switches 8-28), MAIN BYTE (byte address specified by ADDRESS switches 8-31), or a fullword in LOCAL storage (address specified by ADDRESS switches 27-31).

The system console and configuration console switch allows selection of a word in main, display, or local storage.

Note: Main storage switch position refers to both main storage (SE) and display storage (DE) in a 9020E System.

11.2.19 ADDRESS SWITCHES (CE, SC, CC)

The 24 instruction address switches address a location in a storage area and can be manipulated without disrupting CE operations. The address switches, in conjunction with the storage-select switch, permit access to any addressable location. Correct address parity is generated. Addresses generated at the system console or configuration console have their parity checked at the selected CE.

11.2.20 DATA SWITCHES (CE, SC, CC)

The data switches specify the data to be stored in the location specified by the storage-select switch and address switches.

The 32 data switches provided on both the system console and configuration console allow storing of a fullword. Correct data parity

JUNE 1, 1971

is generated at the system console, or configuration console, and is checked at the selected CE.

The 64 data switches provided on each CE panel allow storing of a double word. Parity is automatically assigned.

The switches can be changed without disrupting the CE operation.

11.2.21 STORE SWITCH (CE, SC, CC)

The store switch is pressed to cause the selected CE to store information in the location specified by the storage-select switch and address switches.

The contents of the data switches are placed in the main storage, display storage, general register, or floating-point register location specified. Storage protection is ignored. When the location designated by the address switches and storage-select switch is not available, data is not stored.

The switch is effective only while the CE is in the stopped state.

11.2.22 DISPLAY SWITCH (CE, SC, CC)

The display switch is pressed to cause the selected CE to display information in the location specified by the storage-select switch and address switches.

The data in the main storage, display storage, general register, or floating point register location, or in the instruction-address part of the PSW specified by the address switches and the storage-select switch, are displayed. When the designated location is not available, the displayed information is unpredictable.

The switch is effective only while the CE is in the stopped state.

11.2.23 SET INSTRUCTION COUNTER SWITCH (CE, SC, CC)

This switch is pressed to cause the selected CE to enter the contents of the address switches into the instruction-address part of the current PSW, and fills the Q and R registers with the instructions beginning at the selected address.

The switch is effective only while the CE is in the stopped state.

11.2.24 INTERRUPT SWITCH (CE, SC, CC)

The interrupt switch is pressed to request an external interruption. The switch is operable from the system console, or configuration console, when the system Interlock switch is on, for all states of the selected CE, except Zero with the Test switch on. The switch is operable from the CE control panel for states Three and Two when the system Interlock switch is on; it is always enabled in states One and Zero.

The interruption is taken when not masked off and when the CE is not stopped. Otherwise, the interruption request remains pending. Bit 25 in the interruption- code portion of the current PSW is made one to indicate that the interrupt switch is the source of the external interruption.

JUNE 1, 1971

11.2.25 PROGRAM STATUS WORD RESTART SWITCH (CE)

When the system Interlock switch is off, this switch causes subsystem reset, followed by a load PSW operation from storage location zero of the storage specified by the Storage Select switch. When the system Interlock switch is on, this switch causes a system reset and the configuring of all elements to the selected CE prior to loading the PSW. At completion of the load PSW, the element is changed from a stopped to an operating state. The switch is active only when the element is in the stopped state.

11.2.26 STATE THREE (CE, SC, CC)

This light is on when the state bits in the CCR are set to 1,1. The CE control panel is disabled.

11.2.27 STATE TWO (CE, SC, CC)

This light is on when the state bits in the CCR are set to 1,0. The CE control panel is disabled.

11.2.28 STATE ONE (CE, SC, CC)

This light is on when the state bits in the CCR are set to 0,1. Limited manual controls on the CE control panel are enabled. (See Table 11-III).

11.2.29 STATE ZERO (CE, SC, CC)

This light is on when the state bits in the CCR are set to 0,0. All manual controls on the CE control panel are enabled. (See Table 11-III).

11.2.30 LAMP TEST (CE, SC, CC)

Lamp test switches are provided to permit the operator to test the operator's panel for faulty indicators.

11.2.31 CONTROL COMPUTING ELEMENT SWITCHES (SC, CC)

A switch, corresponding to the scon bit for each CE, is provided to allow manual update of the scon field in the configuration control register of the selected CE. Transfer is accomplished by pressing the control CE Activate Pushbutton. Odd parity is assigned at the console.

11.2.32 CONTROL COMPUTING ELEMENT ACTIVATE SWITCH (SC, CC)

Pressing this pushbutton transfers the contents of the control CE switches to the scon field in the configuration control register of the selected CE. The state bits are also set to 0,0.

11.2.33 ADDRESS-COMPARE SWITCHES (CE, SC, CC)

These switches provide a means of either stopping the selected CE, or causing a branch to a new instruction location on a successful address comparison. The address formed using the content of the address switches (Bits 9-31) is compared against the effective storage address.

JUNE 1, 1971

The address formed using the content of the data switches becomes the effective branch address.

When these switches are set to the STOP position, a successful equal comparison causes the CE to enter the stopped state. The content of the data switches is ignored.

When set to the LOOP position, a successful equal comparison causes the CE to execute the instruction at the effective branch address.

The address-compare switches can be manipulated without disrupting the CE operation, other than by causing the address-comparison stop. When the switch on the CE control panel is set to any position but PROCESS, the test light is on.

PROGRAMMING NOTE

Comparison excludes the three low-order bits of the addresses involved. The comparison is, therefore, made between the addresses of physical storage double words.

Whenever the 12 high-order bit positions of the address switches are zeros, the effective comparison address is the sum of the content of the address switches and the preferential-storage base address. Similarly, whenever the 12 high-order bit positions of the data switches are zeros, the effective branch address is the sum of the content of the data switches and the preferential-storage base address.

When an address not used in the program is selected in the address switches, the CE runs as if the address-compare switches were set to PROCESS.

11.2.34 ADDRESS-COMPARE ENABLE (SC, CC)

Pressing the Enable pushbutton with the System Interlock switch turned on enables the Address Compare switch in the STOP, and LOOP positions. Release of the Enable pushbutton has the affect of placing the Address Compare switch in the PROCESS position.

11.2.35 SENSE SWITCHES (SC, CC)

Six switches are provided for program sense switch operations. The content of the switches is obtained by an I/O Read operation. (See SRL document for the System Console or Configuration Console).

11.2.36 READER/PUNCH-PRINTER SELECT SWITCH (SC, CC)

This switch permits selection of IOCE interfaces designated 1, 2, 3 for interconnection of the Reader/Punch Printer to the selected interface.

11.2.37 READER/PUNCH-PRINTER ENABLE SWITCH-INDICATOR (SC, CC)

Pressing the Enable pushbutton will condition the I/O selection designated by the Reader/Punch-Printer select switch. The indicator will be turned on at the completion of the switching operation and remain on as long as the select switch remains in this position. (Switching occurs upon the completion of the current operation, i.e., after reading or punching a card, or printing a line.) The indicator will turn off if the select switch is repositioned to select another interface. The indicator will be turned on if either the enable

JUNE 1, 1971

pushbutton is pressed or the select switch is returned to the initial selected position.

11.2.38 PRINTER KEYBOARD SELECT SWITCH (SC)

This switch permits selection of the IOCE interface designated 1, 2, 3 for interconnection of the Printer Keyboard to the selected interface.

11.2.39 PRINTER KEYBOARD ENABLE SWITCH-INDICATOR (SC)

Pressing the Enable pushbutton will condition the I/O selection designated by the Printer Keyboard select switch. The indicator will be turned on at the completion of the switching operation and remains on as long as the select switch remains in this position. (Switching occurs upon the completion of the current operation, i.e., after printing or reading a message.) The indicator will turn off if the select switch is repositioned to select another interface. The indicator will be turned on if either the enable pushbutton is pressed or the select switch is returned to the initial selected position.

11.2.40 CONSOLE RESET SWITCH (SC, CC)

Pressing the Console Reset pushbutton resets all console logical functions. The unit status and alarm indicators are not affected.

Pressing the CC Console Reset pushbutton resets those functions associated with the system console control unit.

11.2.41 BELL RESET SWITCH (SC)

Pressing the Bell Reset pushbutton resets the audible bell alarm.

11.2.42 BUZZER RESET SWITCH (SC)

Pressing the Buzzer Reset pushbutton resets the audible buzzer alarm.

11.2.43 ALL STOP SWITCH (SC, CC)

Pressing this pushbutton causes all CEs to enter the stopped state at the completion of their current instructions as defined for STOP.

11.2.44 INVALID SELECTION INDICATOR (SC, CC)

This indicator is turned on by the selected CE whenever a storage address specifies a location outside the available storage for the particular installation, or outside the configured storage, or storage assigned by the storage address translator for the selected CE during manual intervention from the system console or the configuration console.

11.2.45 TEST MODE INDICATOR (SC, CC)

This indicator is turned on by the Test-Operate switch located on the console maintenance and test panel to inform the operator that the test panel controls are operable. The indicator is turned off when the Test-Operate switch is positioned to OPERATE.

JUNE 1, 1971

11.2.46 360-MODE SWITCH-INDICATOR (CE)

Pressing the 360-Mode switch on a CE in State One or Zero enables it to operate in System/360 compatible mode. The indicator is turned on. Pressing the switch when the CE is in "360-mode" returns the CE to "9020 mode", and turns the indicator off. The CE may be returned to "9020-mode" by a power-on reset or an external start.

11.2.47 ENABLE SYSTEM IPL (CC)

This pushbutton switch is used in conjunction with the load pushbutton during an IPL operation. Depressing the load pushbutton with the enable system IPL pushbutton held depressed will initiate a system IPL. If the enable system IPL switch is not depressed, a subsystem IPL is initiated.

11.2.48 I/O INTERFACE ENABLE/DISABLE SWITCH (SC,CC)

The I/O interface enable/disable switch allows the operator to logically disconnect the associated control unit from the IOCE channel(s) prior to turning power off. The SC has one switch (enable/disable) which disconnects all associated control units. The CC has three enable/disable switches, one for each of the two RCU's and one for the system console control unit.

When placed in the disable position, this switch causes the associated control units to logically disconnect from the interconnected IOCE multiplexor channel(s). Any operation in process when the switch is placed in the disable position is completed prior to disconnecting. When the disconnect sequence is completed, the disable light is turned on.

When placed in the enable position, the process is reversed, i.e., the associated control units logically connect to the IOCE channels and the disable light is turned off.

11.2.49 DISABLED INDICATOR (SC,CC)

This green indicator (one indicator for each I/O interface enable/disable switch) is turned on when the interfaces controlled by the switch are disabled. When the disable lamp is on, the operator may change power status of the associated control units without interfering with other channel operations.

11.2.50 FAULT RESET (CC)

Depression of this Fault Reset pushbutton will turn off the SMMC fault signal.

11.2.51 BUZZER/BELL RESET (CC)

Pressing the Buzzer/Bell reset pushbutton resets both the buzzer and the bell audible alarms.

11.3 MAINTENANCE SECTION

This section of the CE control panel, the system console, and the system configuration console contains controls intended for maintenance use.

JUNE 1, 1971

All controls shown in Figures 11-1, 11-2, and 11-3 which are unlisted in Section 11.2, are for maintenance use.

JUNE 1, 1971

CHAPTER 12 -- SYSTEM MONITORING

The IBM 9020D and 9020E Systems possess extensive fault monitoring capability. Each major element in each of the systems -- Computing Element (CE), Input/Output Control Element (IOCE), Storage Element (SE), Display Element (DE), Tape Control Unit (TCU), Storage Control Unit (SCU), Reconfiguration Control Unit (RCU), and Peripheral Adapter Module (PAM) -- checks its own operations, monitors its own power supplies for catastrophic conditions, and senses for an abnormal internal temperature rise. Faults detected are signaled to all Computing Elements either directly or through other controlling units.

12.1 ELEMENT OPERATIONS

Four of the major system elements are described in this manual: Computing Element, Input/Output Control Element, Display Element, and Storage Element. The operation of other system elements and units are covered in separate System Reference Library (SRL) documents.

12.1.1 CE OPERATION

The CE executes instructions stored in SE main storage by means of its internal data flow (which includes a Local Storage), under control of a micro-program in a Read Only Storage (ROS).

The ROS contains micro-instructions which control each computing cycle of the CE. Several micro-instructions are required to execute each instruction; there are four computing cycles per non-interleaved storage cycle. When accessing storage in an interleaved mode, there are two computing cycles per storage cycle. The local storage can cycle at the computing cycle rate, and contains the general purpose registers, floating-point registers, and a working register where partial results are stored.

12.1.2 IOCE OPERATION

Each IOCE contains one multiplexor channel and two selector channels (with capability to add a third selector channel in two of three IOCE's) which are used for input/output operations. Each IOCE also has processing capability. From a user's viewpoint, the IOCE consists of two distinct and independent functional units, IOCE-channel controller and IOCE-processor, sharing common logic and storage units. The IOCE can perform I/O operations and processing operations concurrently. The IOCE may be divided into four types of logical areas.

12.1.2.1 Common Logic Unit (CLU)

The common logic unit (CLU) contains local storage for storing control and data information for the selector channels, and control information for the unit currently being serviced on the multiplexor channel. In addition, the CLU contains a data flow path, which is shared by all channels and the IOCE-processor, and a read-only storage, which controls the CLU and portions of the channel logic.

12.1.2.2 Maintenance and Channel Storage (MACH)

The Maintenance and Channel Storage (MACH) provides a storage for unit control words for all units operating on the multiplexor channel.

JUNE 1, 1971

The IOCE-channel controller also uses MACH storage for I/O data transfers and CCW fetching during I/O operations. MACH storage is also used by the IOCE-processor for storage of instructions and/or data sets. The selection of main or MACH storage is under program control. MACH is also used for off-line diagnostic purposes.

The upper 4096 bytes of MACH are reserved for unit control words and are not directly addressable by the programmer.

12.1.2.3 Multiplexor Channel

The multiplexor channel consists of a minimal data flow and associated control circuitry. The majority of its control comes from the CLU ROS.

12.1.2.4 Selector Channel

Two or three independent selector channels, each consisting of a data flow and controls, are provided in each IOCE and permit semi-independent operation with regard to the CLU. The CLU is primarily used by the selector channels for obtaining references to storage, and for communicating with the controlling CE to initiate and terminate channel operations. The selector channels control the I/O control units and devices, and assemble and disseminate data to these devices independently of the CLU operation (except for fetching and storing storage data).

12.1.3 SE OPERATION

Each SE contains circuitry necessary for data flow, as well as controls to allow interleaving on a doubleword basis, and to allow interconnection of up to seven accessing elements to the main storage arrays. This includes priority and tie breaking circuitry to resolve internal priority conflicts. The SE also contains a Storage Protection Buffer (SPB), and associated controls and compare circuitry to perform the storage protection function.

12.1.4 DE OPERATION

Each DE contains circuitry necessary for data flow and controls to allow interleaving on a doubleword basis. Additionally each DE contains that logic necessary to provide pseudo synchronous data transfers for up to twenty-four display addresses as well as data transfers for up to four CEs. The DEs provide controls for interconnection of up to four CEs and eight display generators (only four of the eight display generators being active at a given time). Also contained in each DE is the priority and tie breaking circuitry for multiple CE simultaneous accesses. A Storage Protection Buffer (SPB) is contained within each DE to perform the storage protection function on CE accesses.

12.1.5 CHECK STOP

A Check-Stop is implemented in each major element of the 9020 System (CE, DE, IOCE, and SE). This causes the element to stop upon detection of a malfunction. In the case of the SE, and DE, the stop occurs upon completion of the current storage cycles in progress i.e., at the end of the interleaved cycles during one of which the malfunction is detected. This effectively freezes the status of the element. By means of the logout facilities of the elements, the machine status at the time of malfunction may be stored in main storage. In the case of the IOCE and CE, the logout is automatically performed under control of CE machine-check masking. In addition, under program control, the SE, DE, Local Storage of the CE, and the normal CE logout information may be logged by

JUNE 1, 1971

use of the DIAGNOSE instruction, or the CE and IOCE may be logged out by use of the WRITE DIRECT instruction.

If a malfunction occurs while the logout of the CE or IOCE is being performed, the element logging out will go into the hard stop state and no further logout will be attempted. When this condition occurs, the element check signal remains up.

An SE, upon detecting a logic malfunction, will remain in a check Stopped state for 2.5 microseconds. If machine checks are masked off or the ILOS bit is set in the accessing element, the SE will return to an operational state at the completion of this time period. If machine checks are not masked and the ILOS bit is set off in the accessing element, the SE will remain Check Stopped until the program controlled logout is complete.

The action of DE upon detection of a malfunction is dependent on whether the malfunction occurred in relation to a CE or display generator access. If related to a CE access, the DE will check stop as described above for the SE. Should the malfunction be related to a display generator access, the DE will interrogate the IDES bit within the CCR and respond accordingly. If the IDES bit is set (Inhibit Stop), the DE will not check stop but will reset the check condition upon completion of the storage cycle and continue in its normal operational state. A pulsed Element Check will be issued to all configured computing elements. If the IDES bit is not set (not Inhibit Stop), the DE will check stop immediately, issue a level Element Check, and remain stopped until the program controlled logout is complete.

When a CE is check Stopped, it will accept no external communication except a reset, a reconfiguration (from a SCON instruction issued by another CE), an external start, an external stop, or a logout signal (Chapter 8) issued by Write Direct instruction. The conditions necessary for a logout are that the CE has an available PSA, or alternate PSA, and is capable of performing the logout. An IOCE will also respond only to a reconfiguration, a reset, or a logout signal. However, the IOCE will always attempt to logout in the latter case.

The SE or DE will respond only to a reset, a reconfiguration, or to a DIAGNOSE instruction specifying the SE or DE (respectively) logout kernel when it is Check Stopped.

PROGRAMMING NOTE

If a reconfiguration is issued to an SE or DE while it is Check Stopped, some of the log data is destroyed, and the content of its CCR is altered. In particular, the request and response latches are changed.

12.1.6 RESETS

There are three types of resets which occur in the 9020D and 9020E Systems (aside from a selective reset to an I/O channel which is issued only as a result of a malfunction detected at the channel -- the interpretation of the reset is part of the specifications for the device). A system reset occurs when the System IPL facility is used, either at the System Console, Configuration Console, or at a CE control panel. It is also issued when the System PSW Restart facility is used at the CE control panel. A subsystem reset occurs when the Subsystem IPL, or Subsystem PSW Restart, facility is used at the CE control panel, System Console, or Configuration Console. A power-on reset causes a system reset in the element for which power is being turned on. This could be the result of either the use of the power-on switch, or the re-energizing of the power line to an element after a power-line failure

JUNE 1, 1971

which lasted beyond the support time of the back-up power system (if available in the element).

A system reset causes all system components not in state zero (with the test switch on) to immediately terminate their current operation and to go into a reset state. In the reset state, the only communication a system element will accept is a configuration mask from any CE, or a signal associated with an IPL operation. All control circuitry is reset in each system element upon receipt of a system reset. The configuration control register (CCR) in each element is set to all zeros with the exception of the scon bits, which are set to all ones. Figure 12-1 is a schematic indication of the initiation and propagation of a system reset to system elements, control units, and devices.

A subsystem reset signal is only accepted by those system elements which are configured into the subsystem of the issuing CE. In particular, an IOCE, a DE or an SE must be configured to the issuing CE, and any configurable I/O control units must be configured to an IOCE(s) controlled by the issuing CE. A non-configurable I/O control unit switched to an IOCE will accept the reset. The subsystem reset does not affect the contents of the CCR or ATR in any element. The system elements involved immediately terminate their current operations and reset their control circuitry. SE's and DE's are issued a "Logout Complete" signal to release them from a possible logout stopped state. Figure 12-2 is a schematic representation of the propagation of the subsystem reset signal to system elements, control units, and devices.

The power-on reset causes the affected system element to execute a system reset. Power-on reset does not propagate from one element to another.

A CE machine reset is caused by an External Start or External Stop and resets all CE error and control logic. CE machine reset does not propagate from one element to another.

JUNE 1, 1971

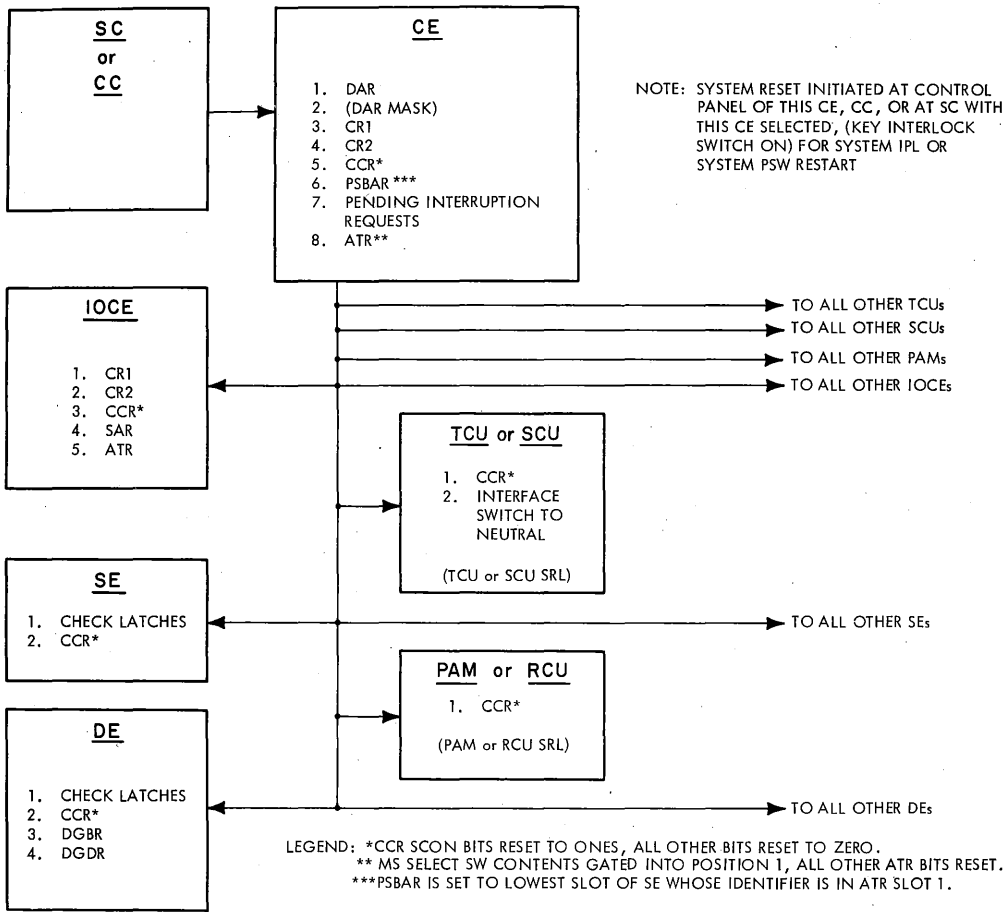


Figure 12-1 IBM 9020D/E System Reset Schematic

JUNE 1, 1971

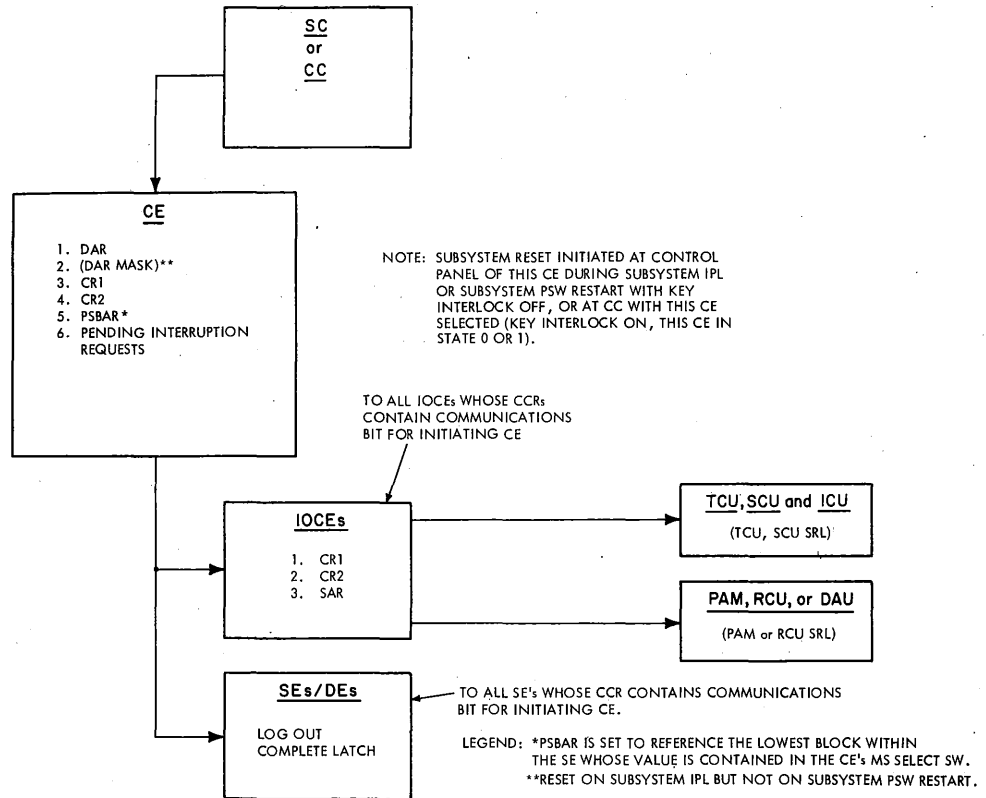


Figure 12-2 IBM 9020D/E Subsystem Reset Schematic

12.2 RECORDING AND HANDLING CHECK CONDITIONS

When a check condition is detected in a CE, IOCE, DE, or SE, the following sequence occurs:

- 1) The check condition is recorded in the source element.
- 2) The detection of the check condition is signaled to other elements in the system.
- 3) The circumstances attending the check condition and the prevailing element status are preserved for analysis.

12.2.1 RECORDING CHECK CONDITIONS

A check condition detected in an IOCE or CE causes an identifier bit to be set on in the check registers provided in these elements. A check condition detected in an SE, or DE causes identifier bit(s) to be set on in the check latches provided in each storage element.

For information on conditions peculiar to recording check conditions in PAM, RCU, SCU, DAU and TCU elements, refer to System Reference Library documents for these elements.

12.2.2 SIGNALING CHECK CONDITIONS

When a check condition is detected in a CE, [with the exception of CE (own) OTC and CE (own) OBS] IOCE, SE, DE, PAM, TCU, SCU, or RCU an abnormal condition signal is presented to each CE in the system. (An abnormal-condition signal, generated by a CE, is not presented to

JUNE 1, 1971

itself.) An abnormal-condition signal may be an element check (ELC), out-of-tolerance (OTC), or an on-battery supply (OBS) signal. Each of the eight major elements can issue an ELC. In addition the IOCE can also issue explicit OTC and OBS signals. (See Table 12-I for a summary of the abnormal-condition signals.)

An abnormal-condition signal (ELC) from an SE, DE, PAM, RCU, SCU, or TCU sets an identifier bit in the diagnose accessible register (DAR) in each CE. Those from an IOCE are encoded in a bit-pair to explicitly represent the ELC, OTC and OBS signals. Computing elements present ELCs to other CEs, but use CE (own) OTC and CE (own) OBS signals internally to cause interrupts.

An ELC signal may be presented to a CE either as a pulse or as a steady level. A pulsed ELC is a signal of relatively short duration. It is presented once per check-condition appearance. The pulsed ELC indicates that the issuing element will attempt to continue operation. A level ELC is a signal of indefinite duration. It is continuously presented to DAR until the check condition is cleared in the sending element. A level ELC indicates that the element can proceed no further and requires external intervention from a computing element, or manual intervention from an operator and/or maintenance personnel. An IOCE issues a level ELC when a machine-check interruption request is not permitted by its controlling CE. An element in power-off state, in the process of logging out, or in check-stop status always presents a level ELC.

The handling of DAR settings and their masking conditions in the computing element are described in Chapter 9.

JANUARY 31, 1973

TNL: GN31-3005

TABLE 12-I ABNORMAL-CONDITION SIGNALS SUMMARY

SOURCE ELEMENT		BITS SET IN DAR	SETTING CONDITION	SIGNAL DURATION	DETAIL REFERENCE	
9020D	9020E					
IOCE-1	IOCE-1	0 1	OTC OBS CCR Parity ELC	Level Level Pulse Pulse-Level	Table 12-IV	
IOCE-2	IOCE-2	0,1 2 3	(As for IOCE-1)			
IOCE-3	IOCE-3	2,3 4 5 4,5	(As for IOCE-1)			
SE-1	SE-1	6	ELC	Pulse-Level		Table 12-V (SE) Table 12-VI (DE)
SE-2	SE-2	7				
SE-3	SE-3	8				
SE-4	SE-4	9				
SE-5	SE-5	10				
SE-6	DE-1	11				
SE-7	DE-2	12	(As for SE-1)			
SE-8	DE-3	13				
SE-9	DE-4	14				
SE-10	DE-5	15				
Spare	Spare	16				
Spare	Spare	17				
PAM-1	RCU1	18	ELC	See Systems Reference Library document covering the PAM and CC		
PAM-2	RCU2	19	ELC			
PAM-3	Spare	20	ELC			
TCU-1	TCU-1	21	ELC	See Systems Reference Library document covering the TCU.		
TCU-2	TCU-2	22	ELC			
TCU-3	TCU-3	23	ELC			
SCU-1	SCU-1	24	ELC	See System Reference Library document covering the SCU.		
SCU-2	SCU-2	25	ELC			
SCU-3	SCU-3	26	ELC			
CE-1	CE-1	27	ELC or OTC(own)	Pulse-Level	Table 12-III	
CE-2	CE-2	28				
CE-3	CE-3	29	(As for CE-1)			
CE-4	CE-4	30				
CE(own) OBS	CE(own) OBS	31	OBS	Level		

PROGRAMMING NOTE

The diagnostic kernel, Store DAR, enables the programmer to transfer the contents of DAR to main storage, and at the same time to reset the

JUNE 1, 1971

register (Chapter 8, DIAGNOSE instruction). If a level abnormal-condition signal is present, this attribute can be inferred by comparing the settings of a particular bit position following two successive applications of the Store DAR kernel. However, if a pulsed signal arrives at the CE simultaneously with the first store DAR execution, the associated DAR bit position may be set a second time. To insure that the pulsed signal is not interpreted as a level signal, three executions of the Store DAR kernel is recommended.

12.2.3 PRESERVATION OF CHECK DATA

Logout is the process of preserving critical CE, IOCE, DE, or SE information for future analysis. Typical information stored includes the contents of check registers, operational registers, counters, and control and status latches. CE and IOCE log data are preserved in fixed locations in the preferential-storage area. SE or DE log data may be placed in any valid main storage area, unless that SE is the element being logged out.

The preferential-storage area assignments are shown in Table 12-II.

The capacity of the preferential-storage area is 4096 bytes of which 512 bytes (128 words) are reserved for special use. The first 32 words are used to hold the various new and old program status words (PSW), Channel Status Word (CSW), Channel Address Word (CAW), the interval timer, and certain channel command words (CCWs) associated with initial program loading (Chapter 9).

The CE logout area occupies 24 doublewords (48 logwords); words 32 through 79 inclusive. When a CE logout occurs, this area receives the information shown in detail in Figure 12-3.

The IOCE logout area occupies 45 words; words 81 through 125 inclusive. When an IOCE logout occurs, this area receives the information shown in detail in Figure 12-4. The IOCE has two modes of logout: Common Logic Unit (CLU) logout, and Selector Channel logout. A CLU logout may be either partial or complete. If a selector channel is not involved in a CLU logout, words 114 through 124, inclusive, are stored with zeros, and the logout is said to be partial. If a selector channel is involved in a CLU logout, all 45 words contain log information, and the logout is said to be complete. If a Selector Channel alone is involved, a Selector Channel logout using words 114 through 124, inclusive, occurs.

Word 80 is reserved as temporary storage during logouts.

Words 126 and 127 are unused.

In every case, the addresses shown in Table 12-II are relative to the current preferential-storage base address for the particular computing element.

Storage element and display element logout is initiated by the DIAGNOSE instruction, and the log data, shown in detail in Figures 12-5 and 12-6, are always received in 6 doublewords of contiguous main storage immediately following the location containing the Maintenance Control Word associated with the DIAGNOSE instruction which initiated the logout (Chapter 8, DIAGNOSE instruction).

JUNE 1, 1971

TABLE 12-II IBM 9020 SYSTEM PREFERENTIAL-STORAGE AREA ASSIGNMENT

ADDRESS (bytes)	WORD No.	PURPOSE																																													
000	000	PSWs, CSW, CAW, Timer, IPL CCWs (32 words) (See Chapter 9 for details.)																																													
124	031																																														
128	032	CE LOGOUT AREA (48 words) (See Figure 12-3)																																													
316	079																																														
320	080	RESERVED																																													
324	081	<table style="border-collapse: collapse; width: 100%;"> <tr> <td style="width: 50%;"></td> <td style="width: 5%; text-align: right;">081</td> <td style="width: 40%;"></td> <td style="width: 5%;"></td> <td style="width: 10%;"></td> </tr> <tr> <td></td> <td></td> <td></td> <td style="border-left: 1px dashed black; border-right: 1px dashed black;">+</td> <td>CLU LOGOUT (partial)</td> </tr> <tr> <td></td> <td></td> <td>IOCE LOGOUT AREA</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td style="border-left: 1px dashed black; border-right: 1px dashed black;">+</td> <td>CLU LOGOUT AREA (complete)</td> </tr> <tr> <td></td> <td></td> <td>(45 words)</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>(See Figure 12-4)</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td style="border-left: 1px dashed black; border-right: 1px dashed black;">+</td> <td>SELECTOR CHANNEL LOGOUT AREA</td> </tr> <tr> <td>496</td> <td>124</td> <td></td> <td></td> <td></td> </tr> <tr> <td>500</td> <td>125</td> <td></td> <td></td> <td>**</td> </tr> </table>		081							+	CLU LOGOUT (partial)			IOCE LOGOUT AREA						+	CLU LOGOUT AREA (complete)			(45 words)					(See Figure 12-4)						+	SELECTOR CHANNEL LOGOUT AREA	496	124				500	125			**
	081																																														
			+	CLU LOGOUT (partial)																																											
		IOCE LOGOUT AREA																																													
			+	CLU LOGOUT AREA (complete)																																											
		(45 words)																																													
		(See Figure 12-4)																																													
			+	SELECTOR CHANNEL LOGOUT AREA																																											
496	124																																														
500	125			**																																											
504	126	UNUSED AREA (2 words)																																													
508	127																																														
512	128	Remainder of PSA (896 words)																																													
4095	1023																																														
** Included in CLU logout (partial)																																															

JUNE 1, 1971

12.3 ELEMENT CHECKING SUMMARY TABLES

Element checking summary tables, Table 12-III, Table 12-IV, Table 12-V, and Table 12-VI, are applicable to the CE, IOCE, SE, and DE respectively. The information tabulated includes:

- 1) The name of the check indicated
- 2) A description of the condition detected, which caused the check
- 3) The check register bit(s) set for a CE or an IOCE, and the Logword bit(s) set for an SE or a DE.
- 4) The class of interruption requested
- 5) The type and duration of the abnormal-condition signal issued
- 6) The elements which receive the signals.

JUNE 1, 1971

TABLE 12-IIIa COMPUTING ELEMENT CHECKING SUMMARY (CHECK REGISTER 1)

CHECK INDICATED	CONDITION DETECTED	BIT SET IN CHECK REGISTER 1	INTER- RUPTION REQUEST	ELEMENT CHECK (ELC)
E Reg Parity	Parity check in E register	0		
PADD Full Sum 4-7	Parallel Adder Latch 4-7 had incorrect parity	1		
PADD Full Sum 8-15	Parallel Adder Latch 8-15 had incorrect parity	2		
PADD Full Sum 16-23	Parallel Adder Latch 16-23 had incorrect parity	3		
PADD Full Sum 24-31	Parallel Adder Latch 24-31 had incorrect parity	4	Machine Check*	Pulse (to other CEs)
PADD Full Sum 32-39	Parallel Adder Latch 32-39 had incorrect parity	5		
PADD Full Sum 40-47	Parallel Adder Latch 40-47 had incorrect parity	6		
PADD Full Sum 48-55	Parallel Adder Latch 48-55 had incorrect parity	7		
PADD Full Sum 56-63	Parallel Adder Latch 56-63 had incorrect parity	8		
PADD Full Sum 64-67	Parallel Adder Latch 64-67 had incorrect parity	9		
MPLR Decode Parity	Parity Check in the Multiply Decoder	10	Machine Check*	Pulse (to other CEs)
PADD Half Sum 4-7	Parallel Adder 4-7 received incorrect parity or failure in 4-7 half-sum hardware	11		
PADD Half Sum 8-15	Parallel Adder 8-15 received incorrect parity or failure in 8-15 half-sum hardware	12		
PADD Half Sum 16-23	Parallel Adder 16-23 received incorrect parity or failure	13		

JUNE 1, 1971

TABLE 12-IIIa COMPUTING ELEMENT CHECKING SUMMARY (CHECK REGISTER 1)
(continued)

CHECK INDICATED	CONDITION DETECTED	BIT SET IN CHECK REGISTER 1	INTER- RUPTION REQUEST	ELEMENT CHECK (ELC)
	in 16-23 half-sum hardware			
PADD Half Sum 24-31	Parallel Adder 24-31 received incorrect parity or failure in 24-31 half-sum hardware)	14		
PADD Half Sum 32-39	Parallel Adder 32-39 received incorrect parity or failure in 32-39 half-sum hardware	15		
PADD Half Sum 40-47	Parallel Adder 40-47 received incorrect parity or failure in 40-47 half-sum hardware	16		
PADD Half Sum 48-55	Parallel Adder 48-55 received incorrect parity or failure in 48-55 half-sum hardware	17		
PADD Half Sum 56-67	Parallel Adder 56-67 received incorrect parity or failure in 56-67 half-sum hardware	18		
PADD Half Sum Check	Parallel Adder received incorrect parity or failure in half-sum hardware	19		
ROS Parity 2-42	Incorrect parity on bits 2-42 of selected ROS word	23		
ROS Parity 43-68	Incorrect parity on bits 43-68 of selected ROS word	24	Machine Check	Pulse (to other CEs)
ROS Parity 69-99	Incorrect parity on bits 69-99 of selected ROS word	25		
SADD Half Sum	Serial Adder received incorrect parity or failure in half-sum hardware	26		
SADD Full Sum	Serial Adder Latch had incorrect parity	27		
<p><u>NOTE</u> * A machine- check interruption request is presented in the originating CE only. However, external (abnormal condition) interruption requests are presented to all other CEs, excluding the originating CE via element checks (ELC).</p>				

JUNE 1, 1971

TABLE 12-IIIb COMPUTING ELEMENT CHECKING SUMMARY (CHECK REGISTER 2)

CHECK INDICATED	CONDITION DETECTED	BIT SET IN CHECK REGISTER 2	INTER- RUPTION REQUEST	ELEMENT CHECK (ELC)
LS Bus Parity	Parity incorrect on local store bus out	17	Machine Check	Pulse
Read Direct Timeout	The CE has detected timeout condition on Read Direct Instruction	27	Machine Check (No Logout)	None
Data Check	SE or DE detected data parity check condition	12	Machine Check**	Pulse (Level during Logout)
CCR Parity	Incorrect parity on the content of the configuration control register	18	Machine Check	Pulse
Address Check	SE or DE detected parity error on address received; or on protection key received or read out, or on marks received; or tag received failed to verify for that SE or DE or an invalid operation was detected, or a multi-accept condition	11	Machine Check**	Pulse (Level during Logout)
Storage Timeout	CE detects that 25ms (nominal) has elapsed after issuance of a select to a non-stopped SE or DE with no Accept received	10	Machine Check**	Pulse Level during Logout)
SE/DE LOS	The CE has issued logout-stop to the SE or DE.	14	None*	None
Fetch Data Check	A parity check is detected on Storage Data Bus Out (SDBO).	13	Machine Check	Pulse
Log ROS	A ROS location outside the ROS logout routine is accessed in ROS mode during logout.	24	None*, +	Level
Log ADR	An access is attempted outside CE logout area during logout.	25	None*, +	Level

JUNE 1, 1971

TABLE 12-IIIb COMPUTING ELEMENT CHECKING SUMMARY (CHECK REGISTER 2)
(continued)

CHECK INDICATED	CONDITION DETECTED	BIT SET IN CHECK REGISTER 2	INTER- RUPTION REQUEST	ELEMENT CHECK (ELC)
CE Logout Request	A logout request is accepted from another CE.	26	Machine Check	None
IOCE Bus Check	A control bus check is signaled from an IOCE	16	Machine Check	Pulse
ATR Parity	A parity check is detected on the content of the Address Translator.	19	Machine Check	Pulse
PSBAR Not Configured	CE cannot find a valid preferential storage area.	21	None*, +	Level
PSBAR Parity	A parity check is detected on the contents of PSBAR (either 'logical' or "physical") or the PSBAR counter.	20	None*, +	Level
Split Logout	CE is attempting to logout into its alternate preferential storage area.	23	None*	None (Level if ILOS on)
PSBAR Alt Check	CE encounters, while in alternate PSA SE, a storage condition which would normally step PSBAR	22	None*, +	Level
SAB Check	A parity check is detected on Storage Address Bus (SAB).	8	Machine Check	Pulse
SDBI Check	A parity check is detected on Storage Data Bus In (SDBI).	9	Machine Check	Pulse
<p>NOTE * "None" indicates that this check causes no machine check interruption request in the originating CE. External (abnormal condition) interruption requests are made to other CE's whenever ELC's are indicated.</p> <p>** Hardstop if during logout</p> <p>+ CE is hardstopped</p> <p>++ Split logout if occurs during logout</p>				

JUNE 1, 1971

TABLE 12-IIIc COMPUTING ELEMENT CHECKING SUMMARY (DAR)

CHECK INDICATED	CONDITION DETECTED	BIT SET IN DAR	INTER- RUPTION REQUEST	ELEMENT CHECK ELC
Out of Tolerance	OTC (own): Temperature has risen to within 10% of shut down value.	27 (CE-1) 28 (CE-2) 29 (CE-3) 30 (CE-4)	other CES External* (Abnormal condition)	None to
On Battery Supply	OBS (own): The CE is operating on its battery power supply	31		None to other CES
None	Power-Off condition	None	None	Level**

NOTES

* OTC (own) and OBS (own) are unique in that they set bits in the DAR of the originating CE only.

** Power-Off condition in a CE causes a level ELC to be issued to all other CEs in the system.

TABLE 12-IVa INPUT/OUTPUT CONTROL ELEMENT CHECKING SUMMARY (CHECK REGISTER 1)

CHECK INDICATED	CONDITION DETECTED	BIT SET IN CHECK REGISTER 1	INTER- RUPTION REQUEST	ELEMENT CHECK (ELC)
Half Sum 0-7	Adder 0-7 received incorrect parity on Byte 0, or failed to generate half-sum.	0		
Half Sum 8-15	Adder 8-15 received incorrect parity on Byte 1, or failed to generate half-sum.	1		
Half Sum 16-23	Adder 12-23 received incorrect parity on Byte 2, or failed to generate half-sum.	2		
Half Sum 24-31	Adder 24-31 received incorrect parity on Byte 3, or failed to generate half-sum.	3		
Full Sum 0-7	Bits 0-7 carry incorrect parity on a shifted sum, register-to-register transfer, storage reference, selector channel data path, or on the unit address from the CE.	4		
Full Sum 8-15	Bits 8-15 carry incorrect parity on a shifted sum, register-to-register transfer, storage reference, selector channel data path, or on the PSBA from CE.	5	Machine-check*	Pulse (to all CE's)
Full Sum 16-23	Bits 16-23 carry incorrect parity on a shifted sum, register-to-register transfer, storage reference, selector channel data path, or on the channel address or the PSBA from the CE.	6		
Full Sum 24-31	Bits 24-31 carry incorrect parity on a shifted sum, register-to-register transfer, storage reference, selector channel data path, or on the I/O operation from the CE.	7		
Group Carry	Failure in adder carry generation circuits.	8		
LB Counter	Incorrect parity on contents of LB counter.	9		
MB Counter	Incorrect parity on contents of MB counter.	10		
MD Counter	Incorrect parity on contents of MD counter.	11		
Length Counter G1	Incorrect parity on contents of G1 counter.	12		

JUNE 1, 1971

TABLE 12-IVa (continued)				
CHECK INDICATED	CONDITION DETECTED	BIT SET IN CHECK REGISTER 1	INTER- RUPTION REQUEST	ELEMENT CHECK (ELC)
Length Counter G2	Incorrect parity on contents of G2 counter	13		
Mover Left Input	Incorrect parity on data entering left side of mover	14		
Mover Right Input	Incorrect parity on data entering right side of mover	15		
Mover output	Incorrect parity generated on output of mover	16		
Storage Address Register (SAR) 8-15	Incorrect parity on byte entering SAR 8-15	18		
Storage Address Register (SAR) 16-23	Incorrect parity on byte entering SAR 16-23	19	Machine-check*	Pulse (to all CE's)
Storage Address Register (SAR) 24-31	Incorrect parity on byte entering SAR 24-31	20		
ROS 1-30	Incorrect parity on bits 1-30 of selected ROS word.	21		
ROS 32-55	Incorrect parity on bits 32-55 of selected ROS word.	22		
ROS 57-89	Incorrect parity on bits 57-89 of selected ROS word.	23		
Log Request	A logout resulting from a check condition which was pending.	24		

NOTE

* A machine-check interruption request is presented in the controlling CE only. External (abnormal condition) interruption requests are presented in all CE's, including the controlling CE, wherever an element check (ELC) is indicated.

TABLE 12-IVb INPUT/OUTPUT CONTROL ELEMENT CHECKING SUMMARY (CHECK REGISTER 2)				
CHECK INDICATED	CONDITION DETECTED	BIT SET IN CHECK REGISTER 2	INTER- RUPTION REQUEST	ELEMENT CHECK (ELC)
CCR Parity	Incorrect parity on the content of the configuration control register.	0	None*	OBS Pulsed
Storage Check	SE detected parity error on data received from IOCE, or on data fetched from storage, or on an address received from the IOCE, or protection key received on read out, or on byte stats received from IOCE; or address received from IOCE fails to verify for that SE; or an invalid operation; or a multi-accept condition.	1	Machine-check (None during logout)	Pulse (Level during logout)
Storage Timeout	An accept has not been received for a non-stopped SE within 16 us, following a request.	2	Machine-check (None during logout)	Pulse (Level during logout)
SE LOS	The IOCE has issued logout-stop to the SE.	3	None*	None
Fetch Data Check	Incorrect parity is detected on the adder out-bus on a data fetch from MACH or Main Storage.	4	None*	None
Invalid ROS Address	The IOCE attempts to access an unused ROS location.	5	Machine-check (None during logout)	Pulse (Level during logout)
Log ROS	Invalid ROS location is accessed in ROS mode during logout.	6	None*	Level
Log ADR	An access is attempted outside IOCE logout area during logout.	7	None*	Level
CE Log Request	A logout request is accepted from a CE.	8	None*	None
Control Bus Check	A parity check is detected on a CE-IOCE control bus data transfer	9	None*	None
ATR Parity	A parity check is detected on the content of the ATR.	10	Machine-check	Pulse
Multiplexor Interface Parity	Incorrect parity is detected on control information or data crossing the Multiplexor interface.	11	Machine-check	Pulse
PSA Lockout	The IOCE fails to gain access to storage on a PSA Reference.	12	Program	None (Level during logout)

JUNE 1, 1971

NOTE FOR TABLE 12-IVb

*Note: "None" indicates that this check causes no machine-check interruption request in the controlling CE. External (abnormal condition) interruption requests are made to all CEs wherever an ELC is indicated.

TABLE 12-IVc INPUT/OUTPUT CONTROL ELEMENT CHECKING SUMMARY (DAR)

CHECK INDICATED	CONDITION DETECTED	BIT SET IN DAR	INTER- RUPTION REQUEST	ELEMENT CHECK (ELC)
Out of tolerance	OTC: Temperature has risen to within 10% of shut down value.	See Table 12-I	External (abnormal Condition)	Level
On Battery Supply	OBS: The IOCE is operating on its battery power supply	See Table 12-I		Level
None	Power-Off condition	None*		Level

NOTE

* Power-Off condition is not indicated by a unique bit in DAR. However, a level ELC is presented to all CEs in the system when power is off on an IOCE.

12.4 COMPUTING ELEMENT CHECKING

12.4.1 CE MACHINE-CHECK HANDLING

When a machine malfunction is detected in a CE, the associated check condition is recorded by setting an identifier bit in a check register (Check Register 1, or Check Register 2), or, in the case of OBS and OTC, in the diagnose accessible register (DAR). In general, machine malfunctions cause a machine-check interruption request to be presented within the CE, and an abnormal-condition signal (pulsed ELC) to be issued to other computing elements in the system (Table 12-III).

When the machine-check mask bit (PSW 13) is one, occurrence of a machine check terminates the current instruction (Chapter 9), delays for approximately 11 ms., initiates a diagnostic logout into the preferential-storage area, and subsequently causes the machine-check interruption to be taken. If the machine-check mask bit is zero, the logout and machine-check interruption both remain pending until machine-check interruptions are again permitted. A pulsed ELC is issued independently of the machine-check masking. Any checks which occur while one is already pending will not cause an ELC to be issued.

The occurrence of most machine checks during a CE logout operation is ignored. However, the occurrence of a Data Check, Storage Timeout, Address Check, a Split Logout with the ILOS bit set on, a Log ROS check, or a Log ADR check, cause the CE to check stop and to issue a level ELC to all other CEs.

JUNE 1, 1971

NOTE

In order to manually reset a Check Stop condition, put the INHIBIT CHECK STOP switch on and depress the CHECK RESET pushbutton.

12.4.2 INTERNAL MACHINE-CHECKS

Those machine-checks which are detected within the CE, and whose occurrence is usually independent of the operation of other elements, are designated internal checks. These include checks on the storage data bus in, storage data bus out, storage address bus, parallel adder, serial adder, E register, read-only storage (ROS), "physical" or "logical" PSBAR, address translator (ATR), and configuration control register (CCR). With the exceptions of OTC and OBS checks, internal checks set identifier bits in check register 1, or in check register 2. Most internal checks are self-explanatory. Only those of particular significance to multi-computing element operation are described in detail.

12.4.2.1 Out-of-Tolerance and On Battery Supply Checks

Two special internal checks are recognized: out-of-tolerance (OTC) condition, and on battery supply (OBS). These checks set identifier bits in the CE's own DAR (bit positions 27-30 and 31 respectively). Provided all masking conditions are satisfied an abnormal-condition interruption is taken: CE (own) OTC, or CE (own) OBS. An ELC signal is not issued to other CE's, nor is a machine-check interruption requested. Consequently, no automatic logout occurs for these check conditions.

12.4.2.2 CCR Parity Check

When the content of the CCR carries improper parity, bit 18 is set on in check register 2, and a pulsed ELC is issued to other CE's. A machine-check interruption is taken. The CE does not modify the contents of the CCR but will ignore the SCON field and accept a SCON from any CE.

12.4.2.3 Fetch Data Check

When data fetched from storage is examined on the storage data bus out and is found to carry improper parity, bit 13 is set on in check register 2, a pulsed ELC is issued to other CE's and a machine-check interruption is requested.

If a fetch data check occurs on an access to the SE which contains the preferential storage area, and the ILOS bit is set off, the CE will step to the alternate preferential storage area SE. Should the ILOS bit be set on, the CE will attempt to logout into the primary PSA. The CE does not issue a logout-stop to the SE.

12.4.2.4 Local Store Bus Check

If while reading Local Store, DAR Mask, CCR, ATR, Ext Reg, PSBAR, or G Reg and improper parity is detected on the local storage out bus, bit 17 is set on in check register 2, a pulsed ELC is issued to other CE's, and a machine check interruption is requested.

12.4.2.5 Log ROS Check

When the CE attempts to access a ROS word outside the logout micro-program during the ROS controlled portion of logout, a Log ROS check is detected. Bit 24 is set on in check register 2 and a level ELC is issued to other CE's. The CE enters a check-stop condition.

JUNE 1, 1971

12.4.2.6 Log ADR Check

When an attempt is made during a logout to store in a location outside bytes 128-319 inclusive, bit 25 is set on in check register 2 and a level ELC is issued to other CEs. The CE enters a check-stop condition.

12.4.2.7 ATR Parity Check

When improper parity is detected on the content of ATR, bit 19 is set on in Check Register 2 a pulse ELC is issued to other CE's, and a machine-check interruption is requested.

12.4.2.8 PSBAR Parity Check

When the content of either the "physical" or "logical" PSBAR or the PSBAR counter carries improper parity, bit 20 is set on in check register 2 and a level ELC is issued to other CE's. The CE enters a check stop condition.

12.4.2.9 SAB Check

When improper parity is detected on the storage address bus before it exits the CE, bit 8 is set on in check register 2, a pulsed ELC is issued to other CE's, and a machine-check interruption is requested.

12.4.2.10 SDBI Check

When improper parity is detected on the storage data bus in before it leaves the CE, bit 9 is set on in check register 2, a pulsed ELC is issued to other CE's, and a machine-check interruption is requested.

12.4.3 EXTERNAL MACHINE-CHECKS

Checks detected within the CE as a result of its operation with other system elements, are designated external checks. With the exception of CE Log Request (bit 26, check register 2), Control Bus check (bit 16, check register 2), and Read Direct Timeout (bit 27, check register 2) external checks are associated with CE-SE accesses or CE-DE accesses.

12.4.3.1 Data Check

A Data Check signal received from an accessed SE or DE causes bit 12 of check register 2 to be set on and a pulsed ELC to be issued to other CEs in the system. Provided machine checks are masked on, and the Inhibit Logout Stop (ILOS) bit is not set on in the CEs CCR, a logout-stop is issued to the malfunctioning SE or DE, and a machine-check interruption is requested.

If a data check condition is detected on an access to the SE which contains the preferential storage area, and the ILOS bit is not set, the CE adjusts its PSA base address to reference its alternate PSA (APSA). If a data check condition occurs on an APSA reference, the CE sets the PSBAR ALT CHECK (bit 22, check register 2), check-stops, and issues a level ELC to other CE's.

A data check condition detected during a CE logout causes a check-stop and a level ELC to be issued to other CEs.

12.4.3.2 Address Check

An Address-Check signal received from an accessed SE or DE causes bit 11 of check register 2 to be set on and a pulsed ELC to be issued to other CE's in the system. Provided machine checks are unmasked, and the

JUNE 1, 1971

Inhibit Logout Stop (ILOS) bit is not set on in the CE's CCR, logout-stop is issued to the malfunctioning SE or DE, and a machine-check interruption is requested.

If an address check is detected on an access to an SE which contains the preferential-storage area (PSA), and the ILOS bit is not set, the CE adjusts its PSA base address to reference its alternate PSA (APSA). If an address check condition occurs on an APSA reference, the CE sets the PSBAR Alt Check bit (bit 22, check register 2), check-stops, and issues a level ELC to other CE's in the system (Chapter 8).

An address check condition detected during a CE logout causes a check-stop, and a level ELC to be issued.

12.4.3.3 Storage Timeout Check

This check, which sets bit 10 on in check register 2, indicates that the CE requested an access from a configured SE or DE, but the access was not serviced within 25ms. The storage timeout check does not cause the logout-stop signal to be issued to the SE or DE. A pulsed ELC is issued to other CE's and a machine-check interruption is requested.

If a storage timeout is encountered while attempting an access to a PSA address and the ILOS bit is not set on, the CE will adjust its PSA base address to reference its alternate PSA (APSA). If a storage timeout is encountered while accessing the APSA, the CE sets the PSBAR ALT CHECK (bit 22, check register 2), check-stops, and issues a level ELC to other CE's.

If a storage timeout is encountered during a CE logout, and the ILOS bit is off, the CE will attempt a Split Logout. Should the ILOS bit be on, the CE will check stop.

12.4.3.4 Read Direct Timeout

When a CE detects a Read Direct Timeout condition, bit 27 is set on in check register 2. A machine check interrupt code of 4 is set, no logout is performed, and the instruction is terminated.

12.4.3.5 IOCE Bus Check

When the CE is notified by a check response signal from an IOCE that an IOCE bus parity check is detected on a CE-to-IOCE data transfer, bit 16 is set on in check register 2, a pulsed ELC is issued to other CE's, and a machine-check interruption is requested.

12.4.3.6 SE/DE Logout-Stop Check (LOS)

Bit 14 is set on in check register 2 when a CE issues a logout-stop signal to an accessed storage element (either an SE or DE) as a result of an address check or data check.

12.4.3.7 CE LOG REQUEST

The handling of a CE log request signal by a CE is described in Chapter 8. Provided the SCON bit for the requesting CE is already set on in the CCR of the receiving CE, a log request signal causes bit 26 in check register 2 to be set on. If machine-check interruptions are masked on, a logout of the CE occurs, and a machine-check interruption is taken.

12.4.3.8 PSBAR Not Configured

The PSBAR Not Configured check, which sets bit 21 on in check register 2, and causes a level ELC to be issued, occurs when the CE

JUNE 1, 1971

cannot find a valid preferential storage area. This condition, which results in a check-stop, can only be cleared by a reset, an IPL operation, PSW restart, reconfiguration to a new PSA, by an external start or an external stop (WRITE DIRECT, Chapter 8).

12.4.3.9 PSBAR Alt. Check

The PSBAR Alternate Check, which sets bit 22 on in check register 2, and causes a level ELC to be issued, occurs when the CE encounters while in the alternate PSA SE, a storage condition which would normally step PSBAR. This condition, which results in a check-stop, can only be cleared by a reset, an IPL operation, PSW Restart, reconfiguration to a new PSA, an external start, or an external stop.

12.4.3.10 Split Logout

When a CE detects a Storage Timeout (25 ms), or an invalid address, or a logout stop during a CE logout, the split-logout bit (bit 23 in check register 2) is turned on. An ELC is not issued to other CE's unless ILOS is set. (If ILOS is set, the CE check-stops and issues a level ELC). A machine-check interruption is not requested.

If a CE detects a Storage Timeout, an Invalid Address, or a Log Out Stopped condition during an APSA logout, the PSBAR ALT CHECK bit is set and the CE check-stops.

PROGRAMMING NOTE 1

Whenever the split-logout bit is found to be set in Word 42 of the CE logout data, a portion of the logout data may be in the primary PSA. This may be determined by examining the contents of the Scan Address Sequencers (bits 8-11 of word 43), which is logged in the APSA. If the logged Address Sequencers reference a doubleword number less than 23, the doubleword number is one less than the last doubleword of logout information contained in primary PSA.

The APSA always contains a complete set of current log data since the log data is reinitiated. However, the log data in those locations down to one greater than where the Address Sequencer points may not be identical to the data in the primary PSA logout. The doubleword of log data indicated by the Address Sequencers will be found in the ST register logwords in the APSA.

12.4.3.11 Storage ID

Bits 0-3 of Check Register 2 contain the ID of the storage unit which caused the check.

12.5 INPUT-OUTPUT CONTROL ELEMENT CHECKING

12.5.1 IOCE MACHINE-CHECK HANDLING

When a machine malfunction is detected in an IOCE, the associated check condition is recorded by setting an identifier bit in a check register (Check Register 1, or Check Register 2 or in the case of "Timeout Check", or "Multiplexor Log Request" which will appear in log words 108 and 111 respectively). The IOCE stops, issues an abnormal-condition interruption request signal (a pulsed ELC) to all CE's in the system, and an IOCE machine-check interruption request to the controlling CE (Table 12-IV).

When the machine-check mask bit (PSW 13) is one in the controlling CE, permission for an IOCE machine-check occurs at the completion of the current CE instruction. The IOCE logs out into the preferential-storage

JUNE 1, 1971

area of the controlling CE while the CE waits. At the completion of a successful logout, the IOCE stores its old machine-check PSW in MACH location 48, sends a response signal to the CE which now takes a machine-check interruption, and enters the wait loop.

NOTE

Storing the old machine-check PSW in MACH location 48 occurs only for machine-checks detected during IOCE-processor operations; the PSW is not stored for machine-checks detected during IOCE-channel controller operations.

Most machine checks are ignored during an IOCE logout. However, the occurrence of a Log ROS check, a Log ADR check, an invalid ROS address, a Storage Check, or a Storage Timeout (16 usec.) check causes the IOCE to check stop and issue a level ELC to all CE's in the system.

12.5.2 INTERNAL MACHINE-CHECKS

Machine checks detected within the IOCE whose occurrence is usually independent of the operation of other elements are designated internal checks. These include checks on the data paths, adder, counters, mover, read-only storage (ROS), address translator (ATR), and configuration control register (CCR). With the exceptions of OTC and OBS checks, internal checks set identifier bits in a check register. Most internal checks are self-explanatory. Only those of particular significance to multi-computing element operation are described in detail.

12.5.2.1 Out-of-Tolerance and On Battery Supply Checks

These checks do not set identifier bits in the check registers, but present abnormal-condition interruption requests (OTC, OBS) to all CEs in the system by setting on bits in their diagnose accessible registers (DAR).

Machine-check interruption is not requested, and no diagnostic logout ensues.

The levels presented remain up until the cause of the check is cleared.

12.5.2.2 CCR Parity Check

When the content of the CCR carries improper parity, bit 0 is set on in check register 2, and a pulsed OBS is issued to all CEs in the system. An ELC signal is not issued, nor is an IOCE machine-check interruption requested. The contents of the CCR remain unchanged and the IOCE will accept a SCON from any CE.

12.5.2.3 Fetch Data Check

When data fetched from storage is examined on the adder out-bus and is found to carry improper parity, bit 4 is set on in check register 2. A machine-check interruption request is presented to the controlling CE and a pulsed ELC is issued to all CE's in the system due to a full sum check which always occurs at the same time (see Table 12-IVa).

12.5.2.4 Invalid ROS Address Check

When an invalid ROS location is accessed, bit 5 is set on in check register 2. A machine-check interruption request is presented to the controlling CE and a pulsed ELC is issued to all CE's in the system. When this check condition is detected during a logout, a check-stop results in a level ELC.

JUNE 1, 1971

12.5.2.5 Log ROS Check

Whenever an invalid read-only storage (ROS) location, i.e., a location not used by logout, is accessed during a logout, bit 6 is set on in check register 2. A check-stop condition results in a level ELC being issued to all CE's in the system.

12.5.2.6 Log ADR Check

Whenever an attempt is made to store in PSA locations outside bytes 320-511 inclusive, bit 7 is set on in check register 2. A check-stop condition results in a level ELC being issued to all CE's in the system.

12.5.2.7 Multiplexor Interface Parity Check

When incorrect parity is detected on control information or data crossing the multiplexor interface, bit 11 is set on in check register 2. A machine-check interruption request is presented to the controlling CE, and a pulsed ELC is issued to all CE's in the system.

12.5.2.8 IOCE Bus Check

When incorrect parity is detected on data received off the CE-to-IOCE control bus, the data is rejected, and bit 9 is set on in check register 2. The controlling CE is automatically alerted to the condition by a check response signal. No machine-check interruption request is presented and an ELC is not issued as a consequence of setting this bit (see section 12.4.3.5).

12.5.2.9 ATR Parity Check

When improper parity is detected on the content of ATR, bit 10 is set on in Check Register 2. A pulsed ELC is issued to all CE's, and a machine-check interruption is requested in its controlling CE.

12.5.3 EXTERNAL MACHINE-CHECKS

Checks detected within an IOCE as a result of its operation with other system elements are designated external checks. External checks are usually connected with IOCE-SE accesses.

12.5.3.1 Storage Check

A Storage Check issued to an IOCE by an accessed SE causes bit 1 of check register 2 to be set on, and a pulsed ELC to be issued to all CE's in the system. Provided the inhibit-logout-stop (ILOS) bit is not set on in the IOCE's CCR, logout stop is issued to the SE, and a machine-check interruption request is presented to the controlling CE. When ILOS is set on the machine-check interruption request is presented but the SE is not stopped.

A storage check during an IOCE logout causes a check-stop condition in the IOCE and a level ELC is issued.

12.5.3.2 Storage Timeout Check

When the IOCE requests an access from a configured SE which is not serviced within 16 usec a storage timeout results.

If the SE is not in logout-stop condition, bit 2 is set on in check register 2, a pulsed ELC is issued to all CE's in the system, and a machine-check interruption is requested in the controlling CE. During IOCE logout, a level ELC is issued and the IOCE check-stops.

JUNE 1, 1971

If the SE is in logout-stop condition, one of the following may occur:

- a) When the storage timeout occurs on a PSA reference, a program interruption on a PSA lockout exception is requested in the controlling CE (12.5.3.5). During IOCE logout a level ELC is also issued and the IOCE check-stops.
- b) When the storage timeout occurs on a non-PSA reference involving an I/O data transfer or a CCW fetch for a busy subchannel, the CSW is stored containing the chaining check bit and an I/O interruption is requested (10.4.5.8). If a CCW fetch for an available subchannel is involved, a condition code 1 is set for the SIO and a CSW is stored containing the chaining check bit.
- c) When the storage timeout occurs on a non-PSA reference by the IOCE-Processor, a program interrupt on a SE stopped exception is requested in the IOCE-Processor. No ELC is issued.

12.5.3.3 SE Logout-Stop Check

Bit 3 is set on in check register 2 when an IOCE issues a logout-stop signal to an accessed storage element. No ELC is issued, and a machine-check interruption request is not presented to the controlling CE as a consequence of setting this bit.

12.5.3.4 CE Log Request Check

Provided the communications bit for the requesting CE is already set on in the CCR of the receiving IOCE, a log-request signal causes bit 8 to be set on in check register 2. A machine-check interruption is not requested.

12.5.3.5 PSA Lockout Check

When an IOCE fails to gain access to its PSA, (that is, on a CAW fetch, I/O interruption, CSW store pertaining to an I/O instruction, or on a store during logout), bit 12 is set on in check register 2. A program interruption is requested in the controlling CE.

During logout a PSA lockout causes a check-stop condition resulting in a level ELC being issued to all CE's in the system.

JUNE 1, 1971

TABLE 12-V STORAGE ELEMENT CHECKING SUMMARY

CHECK INDICATED		CONDITION DETECTED	BIT SET		ELEMENT NOTIFIED		ELEMENT CHECK (ELC)	
IOCE	CE		LOG-WORD	BIT	IOCE	CE		
Storage Check and Element Check	Address Check and Element Check	Tag Mismatch	5	10	Storage Check to accessing CE, IOCE, ELC to all CEs	Address Check to accessing all CEs	Pulse*	
		Normal Operation Check (odd)	11	5				
			(even)	5				5
		Storage Protect Address (odd) or Key Parity Check	11	3				
			(even)	5				3
		Address Parity Check	(odd)	11				1
			(even)	5				1
		Mark Parity Check	(odd)	11				0
			(even)	5				0
	Multi-Accept Check		5	20				
		5	9					
	N/A				N/A			
Data & Element Check	Data Parity Check	(odd)	11	2	Note 1			
		(even)	5	2				
Element Check Only		Thermal Condition	5	17	Element Check to all CEs			
		On Battery Supply (OBS)	5	16				
		CCR Parity Check	5	8				
		Double Cycle Timeout	5	18				
		Logout Stop Condition		None				
	Power Off			None		Level		
Protect Check		Protection Key Mismatch (odd)	11	4	Program Interrupt#	None		
		(even)	5	4				
<p>* - When a logout-stop signal is received from an accessing element in response to either a storage check (IOCE access), address check (CE access), or data check (CE access), the ELC becomes a level.</p> <p># - When the accessing element is a CE or IOCE Processor, a program interrupt occurs. When the accessing element is an IOCE Channel Controller, the CSW stored will contain the protection check bit, and an I/O interruption is requested in the controlling CE.</p> <p>Note 1 - Data Check to accessing CE, ELC to all CE's.</p>								

JUNE 1, 1971

TABLE 12-VI DISPLAY ELEMENT CHECKING SUMMARY

CHECK INDICATED	CONDITION DETECTED	BIT SET LOG-WORD BIT		ELEMENT NOTIFIED	ELEMENT CHECK (ELC)
Address Check and Element Check	Tag Mismatch	10	9	Address Check to accessing CE	Pulse #
	Normal Operation Check	10	13		
	Storage Protect Address Parity Check	3	24		
	Storage Protect Key Parity Check	3	25		
	Address Parity Check (odd) (CE access)	7	5		
	Address Parity Check (even)	3	5		
	Mark Parity Check (odd)	7	6		
	Mark Parity Check (even)	3	6		
	Multi-Accept	10	11		
	Tag Parity Check	10	8		
Data Check and Element Check	Data Parity Check (odd) (CE access)	7	4	Data Check to accessing CE	
	Data Parity Check (even)	3	4		
Element Check Only	Address Parity Check (odd) (DG access)	7	5	Element Check to all CE's	Pulse (level if IDES bit is on)
	Address Parity Check (even)	3	5		
	Data Parity Check (odd) (DG access)	7	4		
	Data Parity Check (even)	3	4		
	Local Storage Parity Check	10	12		
	Refresh Clock Parity Check	10	14		
	DG Register 1 Parity Check	10	0		
	DG Register 2 Parity Check	10	1		
	DG Register 3 Parity Check	10	2		
	DG Register 4 Parity Check	10	3		
DG Register 5 Parity Check	10	4			
DG Register 6 Parity Check	10	5			
DG Register 7 Parity Check	10	6			
DG Register 8 Parity Check	10	7			

JUNE 1, 1971

TABLE 12-VI DISPLAY ELEMENT CHECKING SUMMARY (continued)

CHECK INDICATED	CONDITION DETECTED	BIT SET LOG-WORD BIT		ELEMENT NOTIFIED	ELEMENT CHECK (ELC)
Element Check Only	Thermal Condition	3	28	Element Check to all CE's	Pulse Level
	On Battery Supply (OBS)	3	27		
	CCR Parity Check	10	10		
	Logout Stop Check		None		
None	Power Off		None		
None	Invalid DG Configuration	8	31	None	
Protect Check	Protection Key Mismatch	3	26	Program Interrupt	None

- When a logout-stop signal is received from a CE in response to either a data or address check, the ELC becomes a level.

12.6 STORAGE ELEMENT CHECKING

12.6.1 SE MACHINE-CHECK HANDLING

Three categories of machine checks are originated in the SE. They are:

- 1) Storage Check -- This check is returned to an accessing IOCE when an improper parity condition, invalid operation, multi-accept condition or an invalid address is detected in access servicing for that IOCE.
- 2) Address Check or Data Check -- Either an address check or a data check, or both, is returned to an accessing CE when an improper parity condition, invalid operation, multi-accept condition or an invalid address is detected in access servicing for that CE.
- 3) Element Check -- The ELC signal is issued to all CE's in the system when a power-off condition, thermal condition, CCR parity error, on-battery supply condition, storage check (IOCE access), address check (CE access), double cycle timeout, or data check (CE access) is detected.

When a machine malfunction is detected in an SE, the associated check condition is recorded by setting an identifier bit in a latch. This information becomes the content of Logword 5 or Logword 11 upon an SE logout. Logword 5 pertains to a machine malfunction occurring in the even side of the SE while Logword 11 pertains to the odd side. Malfunctions detected in logic common to both sides of the SE such as Multi-Accept, Double Cycle Timeout, IOCE Storage Bus In (SBI), CCR Parity and Tag checks are indicated in Logword 5.

The SE Checking Summary is given in Table 12-V.

JUNE 1, 1971

With the exception of the power-off condition or logout-stop condition, a pulsed ELC is issued. If the accessing element responds to an address or data check (CE access), or a storage check (IOCE access) with a logout-stop signal, the SE check stops and the ELC becomes a level which remains up until the logout is complete.

12.6.2 SE LOGWORD 5 AND LOGWORD 11

12.6.2.1 Tag Mismatch

When an address (tag) is presented to an SE which does not lie within that SE, bit 10 of Logword 5 is set. An address check (CE access) or storage check (IOCE access) is returned to the accessing element and a pulsed ELC is issued to all CE's.

12.6.2.2 Normal Op Check

When an SE detects a conflict in control signals from an accessing element, i.e., when the normal operation line is active and an operation other than fetch is requested, such as TEST and SET (or conversely), bit 5 is set on in either Logword 5 (even) or Logword 11 (odd). Additionally, a storage check (IOCE access) or address check (CE access) is returned to the accessing element and a pulsed ELC is issued to all CEs and the operation is suppressed.

12.6.2.3 Storage Protect Address or Key Parity Check

When an SE detects improper parity on either the Storage Protect Address or Key, bit 3 of either Logword 5 (even) or Logword 11 (odd) is set on. A storage check (IOCE access) or address check (CE access) is returned to the accessing element and a pulsed ELC is issued to all CE's.

12.6.2.4 Address Parity Check

When an SE detects improper parity in one of the SE Storage Address Registers (SESAR), bit 1 of either Logword 5 (even) or Logword 11 (odd) is set on. A storage check (IOCE access) or address check (CE access) is returned to the accessing element, and a pulsed ELC is issued to all CE's.

12.6.2.5 Mark Parity Check

When an SE detects improper mark or byte stat parity, bit 0 of either Logword 5 (even) or Logword 11 (odd) is set on. A storage check (IOCE access) or address check (CE access) is returned to the accessing element, and a pulsed ELC is issued to all CE's. If a byte stat parity error is detected during an IOCE access, the byte stats will be logged out in either the high or low order four bits of the mark field, depending on which word of the doubleword was being accessed.

12.6.2.6 Multi Accept Check

When an SE detects a multi-accept condition, bit 20 of Logword 5 is set on. A storage check (IOCE access) or an address check (CE access) is returned to the accessing element, or elements, and a pulsed ELC is issued to all CE's. A multi-accept condition arises if an SE detects that it has honored multiple CE's or multiple IOCE's for an even or odd storage cycle, or it has honored requests for both even and odd storage simultaneously from the same CE or IOCE.

JUNE 1, 1971

12.6.2.7 Tag Parity

When an SE detects improper parity on the tag (SE select address), bit 9 of Logword 5 is set on. A storage check (IOCE access) or address check (CE access) is returned to the accessing element, and a pulsed ELC is issued to all CE's.

12.6.2.8 Double Cycle Timeout

When the time between two successive SE accesses by a CE or IOCE executing an OR Immediate, AND Immediate, or Exclusive OR Immediate exceeds 5-10 microseconds, the Double Cycle Timeout bit (bit 18 of logword 5) is set on and a pulsed ELC is issued to all CEs.

12.6.2.9 IOCE SBI Check

When improper data parity is detected on the IOCE SBI during a store, bit 11 of Logword 5 is set on. A storage check is returned to the accessing IOCE, and a pulsed ELC is issued to all CE's. The bad parity word is stored.

12.6.2.10 Data Parity Check

When an SE detects improper parity in one of the SE Storage Data Registers (SESDR), bit 2 of either Logword 5 (even) or Logword 11 (odd) is set on. A storage check (IOCE access) or data check (CE access) is returned to the accessing element, and a pulsed ELC is issued to all CE's.

12.6.2.11 Thermal Condition

When an SE detects that its temperature has risen to within 10% of shut-down value, bit 17 of Logword 5 is set on, and a pulsed ELC is issued to all CE's.

12.6.2.12 On-Battery Supply (OBS)

When an SE is switched to its battery power supply, bit 16 of Logword 5 is set on, and a pulsed ELC is issued to all CE's.

12.6.2.13 CCR Parity Check

When the content of the CCR carries improper parity, bit 8 of Logword 5 is set on and a pulsed ELC is issued to all CE's. The SE will not modify its CCR but will accept a SCON from any CE regardless of the SCON field setting within the CCR.

12.6.2.14 Logout Stop Condition

When the SE receives a logout-stop signal from a CE or an IOCE, a level ELC is issued to all CE's. This level remains until logout is complete. No check bits are set.

12.6.2.15 Power Off

When power is off, a level ELC is issued to all CE's. No check bits are set.

12.7 DISPLAY ELEMENT CHECKING

12.7.1 DE MACHINE-CHECK HANDLING

Two categories of machine checks are originated in the DE. They are:

JUNE 1, 1971

- 1) Storage Data or Storage Address Check (CE) - These checks are returned to an accessing CE when an improper parity or invalid address is detected in access servicing for CE's.
- 2) Element Check - The ELC signal is issued to all CE's in the system when a power-off condition, thermal condition, CCR parity error, on-battery supply condition, CE related storage data or storage address checks, or DG related storage data or storage address checks are detected.

When a machine malfunction is detected in a DE, the associated check condition is recorded by setting an identifier bit in a latch register. This information becomes the content of either Logword 3, Logword 7, or Logword 10. Logword 3 pertains to a machine malfunction in the even side of the DE, Logword 7 pertains to the odd side of the DE, and Logword 3 and 10 contains machine malfunctions considered to have occurred in the common portion of the DE.

The DE Checking Summary is given in Table 12-VI.

With the exception of the power-off condition or a logout-stop condition, a pulsed ELC is issued. If a CE responds to a storage data or storage address check (related to a CE access) with a Logout-Stop Signal or should a storage data or storage address check occur which is related to a DG access and the IDES bit is set off, the DE check stops and issues a level ELC which remains up until the logout is complete.

12.7.2 DE LOGWORD 3, LOGWORD 7, AND LOGWORD 10

12.7.2.1 Tag Mismatch

When an address (tag) presented to the DE by the CE, does not lie within that DE, bit 9 of Logword 10 is set. An Address Check is returned to the accessing CE and a pulsed ELC is issued to all CE's. This check condition usually implies an error in the accessing CE's selection circuitry.

12.7.2.2 Normal Op Check

When a DE detects an inconsistency between the state of the normal operation line and the operation being performed, bit 13 of Logword 10 is set. An Address Check is returned to the accessing CE and a pulsed ELC is issued to all CE's. The normal operation line should accompany only a store or fetch operation.

12.7.2.3 Storage Protect Address Parity

When a DE detects improper parity on the storage protect address, bit 24 of Logword 3 is set. An Address Check is returned to the accessing CE and a pulsed ELC is issued to all CE's if occurring during a CE access.

If this check condition occurs during a DG access, the DE will interrogate the IDES bit within the DE CCR and, if set to one, will issue a pulsed ELC to all CE's, reset the check condition and continue normal operation. If the IDES bit is not set, the DE will issue a level ELC and enter a logout stop condition. This level remains until logout is complete or a reset is received.

12.7.2.4 Storage Protect Key Parity

When a DE detects improper parity on the storage protect key received from the accessing CE or fetched from the storage protect buffer, bit 25

JUNE 1, 1971

of Logword 3 is set. An Address Check is returned to the accessing CE and a pulsed ELC is issued to all CE's, if occurring during a CE access.

If this check condition occurs during a DG access, the DE will interrogate the IDES bit within the DE CCR and if set to one, will issue a pulsed ELC to all CE's, reset the check condition and continue normal operation. If the IDES bit is not set, the DE will issue a level ELC and enter a logout stop condition. This level remains until logout is complete or a reset is received.

12.7.2.5 Address Parity

When a DE detects improper parity in one of the DE storage address registers (DESAR) during a CE access bit 5 of Logword 7 (odd side) or Logword 3 (even side) is set on. An Address Check is returned to the accessing CE and a pulsed ELC is issued to all CE's.

If improper parity in one of the DESAR's is detected during a DG access, bit 5 of Logword 7 (odd side) or Logword 3 (even side) is set on and a pulsed ELC is issued to all CE's. The DE will then interrogate the IDES bit in the DE CCR and if set, will reset the check bits in the logword and continue normal operation. If the IDES bit is not set, the DE will issue a level ELC and enter a logout stop condition. This level remains until logout is complete or a reset is received.

12.7.2.6 Mark Parity

When a DE detects improper parity on the marks, which control regeneration of bytes during a store, bit 6 of Logword 7 (odd side) or Logword 3 (even side) is set on. An Address Check is returned to the accessing CE and a pulsed ELC is issued to all CE's if occurring during a CE access.

If this check condition occurs during a DG access, the DE will interrogate the IDES bit within the DE CCR and, if set to one, will issue a pulsed ELC to all CE's, reset the check condition and continue normal operation. If the IDES bit is not set, the DE will issue a level ELC and enter a logout stop condition. This level remains until logout is complete or a reset is received.

12.7.2.7 Multi-Accept

When a DE detects a multi-accept condition, bit 11 of Logword 10 is set on. An Address Check is returned to the accessing CE's and a pulsed ELC is issued to all CE's. A multi-accept condition arises if the DE detects that it has accepted multiple CE's for an even or odd storage cycle, or it has accepted CE's and/or DG's on both the even and odd storage simultaneously.

12.7.2.8 Tag Parity

When a DE detects improper parity on the tag (element select address), bit 8 of Logword 10 is set. An Address Check is returned to the accessing CE and a pulsed ELC is issued to all CE's.

NOTE: Tag parity check is actually performed on bits 1-5 which consist of the tag (1-4) and the hi/low bit (5). Bit 5 should always be zero when the tag references a DE.

12.7.2.9 Data Parity

When a DE detects improper parity in one of the DE storage data registers (DESDR) during a CE access, bit 4 of Logword 7 (odd side) or Logword 3 (even side) is set on. A Data Check is returned to the accessing CE and a pulsed ELC is issued to all CE's.

JUNE 1, 1971

If improper parity is detected in the DESDR's during a DG access, bit 4 of Logword 7 (odd side) or Logword 3 (even side) is set on and a pulsed ELC is issued to all CE's. The DE will then interrogate the IDER bit in the DE CCR and if set, will reset the check bits in the logword and continue normal operation. If the IDER bit is not set, the DE will issue a level ELC and enter a logout stop condition. This level remains until logout is complete or a reset is received.

12.7.2.10 Local Storage Parity

When a DE detects improper parity on the local storage bus, bit 12 of Logword 10 is set and a pulsed ELC is issued to all CE's. If the IDER bit is set, the check bit will then be reset and the DE will resume normal operation. If the IDER bit is not set, the DE will issue a level ELC to all CE's and enter a logout stop condition. This level remains until logout is complete or a reset is received.

12.7.2.11 DG Data Register Parity

When a DE detects improper parity in any one of the DG registers, bit 0-7 (for DG registers 1-8, respectively) are set in Logword 10 and a pulsed ELC is issued to all CE's. If the IDER bit is set, the check bit will be reset and the DE will resume normal operation. If the DE IDER bit is not set, the DE will issue a level ELC and enter a logout stop condition. This level remains until logout is complete or a reset is received.

12.7.2.12 Thermal Condition

When the DE detects that its temperature has risen to within 10% of shut-down value, bit 28 of Logword 3 is set on and a pulsed ELC is issued to all CE's.

12.7.2.13 On-Battery Supply

When the DE is switched to its battery power supply, bit 27 of Logword 3 is set on and a pulsed ELC is issued to all CE's.

12.7.2.14 CCR Parity

When the content of the CCR carries improper parity, bit 10 of Logword 10 is set on and a pulsed ELC is issued to all CE's. The content of the CCR is not modified but the DE will accept a SCOR from any CE.

12.7.2.15 Logout-Stop Check

When the DE receives a logout-stop signal from a CE or enters a logout stop condition due to the occurrence of a malfunction related to a DG access and the IDER bit is not set, a level ELC is issued to all CE's. This level remains until logout is complete. No check bits are set in the logwords due to the logout stop condition itself.

12.7.2.16 Power-Off Condition

When power is off, a level ELC is issued to all CE's. No check bits are set in the logwords.

12.7.2.17 Invalid DG Configuration

When the DE receives an invalid DG communications field in the CCR, bit 31 of Logword 8 is set on. The CCR will not be modified but the DE will ignore the DG communications field and will not respond to any DG requests (i.e., effectively an all zero DG communication field). Invalid configurations are (1) more than four DG interfaces enabled, (2)

JUNE 1, 1971

more than one interface per DG is enabled or (3) more than one DG interface enabled to the same DG register.

12.7.2.18 Refresh Parity Check

When incorrect parity is detected in the refresh counter, the DE will interrogate the IDES bit within the DE CCR and if set to a one will issue a pulsed ELC to all CE's, reset the Refresh Counter and the check condition, and continue normal operation. If the IDES bit is not set, the DE will issue a level ELC, set bit 14 in Logword 10, and enter a logout stop condition. The level ELC remains until a Logout complete or reset is received.

12.8 DIAGNOSTIC LOGOUT OF THE CE, IOCE, SE AND DE

12.8.1 CE LOGOUT

A diagnostic logout may be initiated in a CE by four different means. Each method of initiating the logout causes a specific identifier bit to be set in the check register of the CE logging out, and causes a machine-check interruption request to be presented. Provided machine-check interruptions are not masked off, the CE delays for approximately 11 ms., the logout is carried out and then a machine-check interruption is taken. If machine-check interruptions are masked off, both the logout and machine-check interruption requests remain pending.

When a CE logs out, the log information is placed in 48 contiguous words of the preferential-storage area, extending from byte location 128 through location 319 inclusive (Table 12-II).

12.8.1.1 Logout Initiated by Another CE

Another CE may initiate the logout by executing WRITE DIRECT with the CE to be logged out specified. Provided the scon bit for the CE initiating the logout is already set on in the CCR of the receiving CE, bit 26 is set in check register 2. If machine-check interruptions are not masked off, the logout is carried out and then the machine-check interruption is taken.

12.8.1.2 Logout Initiated by an Address or Data Check

When an address check or data check is received from a configured SE or DE on any storage reference [other than on an alternate preferential-storage area (APSA) reference in an SE] and provided masking conditions are satisfied, a CE logout occurs. Bit 11 or 12 is set in check register 2 to identify the cause of the logout.

12.8.1.3 Logout Initiated by a Storage Timeout Check

When a storage timeout check is received from a configured SE or DE, bit 10 of check register 2 is set on, and a logout is carried out, subject to the machine-check masking conditions described previously.

12.8.1.4 Logout Initiated by an Internal Check

When an internal check (CE machine check) is detected in the CE, the appropriate check bit is set in check register 1. If machine checks are not masked, the logout is carried out, and a machine-check interruption is taken.

JUNE 1, 1971

12.8.2 IOCE LOGOUT

The IOCE performs two types of diagnostic logouts: the Common Logic Unit (CLU) logout and the Selector Channel logout.

The CLU logout information includes the Common Logic Unit (CLU), Common Channel, Multiplexor Channel, and one Selector Channel, provided the selector channel is involved in the malfunction. When no selector channel is involved, all zeros are stored in the selector channel logout area (logwords 114-124). A CLU logout may be initiated by an IOCE or a CE. CLU logouts initiated by the IOCE request a machine-check interruption in the controlling CE. The logout occurs when the interruption is permitted. CLU logouts initiated by the CE executing WRITE DIRECT do not request machine-check interruptions.

The Selector Channel logout uses words 114-124, and applies only to the selector channel involved. Selector channel logouts are initiated by requesting an I/O interruption in the controlling CE. The logout occurs when the interruption is permitted (Chapter 10).

12.8.2.1 CLU Logout

A CLU logout may be initiated in an IOCE by three different means.

- 1) A CLU check (IOCE machine check) is detected during a normal operation (i.e., I/O operation or IOCE-processor operation), or during a Selector channel logout.
- 2) A selector channel logout-request condition is detected while a TEST CHANNEL instruction is being processed.
- 3) An IOCE logout signal is received from a CE executing WRITE DIRECT with the IOCE to be logged out specified.

When a CLU logout is initiated by (1) or (2), the IOCE issues a pulsed ELC to all CE's in the system, requests a machine-check interruption in the controlling CE, and waits in check-stop condition until the machine-check interruption is allowed. When allowed, the IOCE logs out into the PSA, and ends the operation by issuing a response to the CE. In addition, when the CLU logout is initiated by a CLU check during an IOCE-processor operation, the IOCE stores its old machine-check PSW in MACH 48 at the completion of a successful logout.

The manner in which the CE handles an IOCE machine-check interruption request is as follows:

If an I/O instruction is in process, the CE terminates the operation, storing a condition code 3 in the PSW. If machine-check interruptions are unmasked, the IOCE is allowed to start the logout. If no I/O instruction is in process, and machine-check interruptions are unmasked, the IOCE is allowed to start the logout at completion of the current CE instruction. The CE waits during the logout. Upon its completion a machine-check interruption is taken.

An IOCE logout may be initiated by a CE executing WRITE DIRECT provided the communications bit for the CE is already set in the IOCE's CCR. The IOCE sets bit 8 in check register 2 (12.5.3.4), and proceeds to logout. The CE starts a time down (approx. 120 microseconds), and waits for a logout complete indication from the IOCE. Upon receipt of the indication the CE sets the condition code to zero and fetches its next instruction. Should a time down occur, a condition code of three is set and instruction fetching is resumed as before.

When a CLU logout occurs, IOCE-processor operation ceases and it goes into the stopped state at the completion of the logout.

JANUARY 31,1973

TNL: GN31-3005

12.8.2.2 Check Handling During CLU Logout

When a storage check, PSA lockout, invalid ROS address check, Log ROS check, or a Log ADR check is detected during a CLU logout, the logout is terminated. The IOCE check-stops and issues a level ELC to all CE's in the system. All other IOCE machine-checks are ignored during a CLU logout.

12.8.2.3 Selector Channel Logout

When an IOCE detects a check, other than a Channel Data Check, which is entirely associated with the selector channel hardware, a Selector Channel Logout is initiated. The IOCE terminates the affected channel operation, logs it out and resets it.

- 1) If an I/O instruction other than TEST CHANNEL is in progress, the IOCE logs out the Common Multiplexor and Selector Channels, and sets bits 45 (Channel Control Check), or 46 (Interface Control Check), as is appropriate in the CSW. A response is returned to the controlling CE with condition code 1. The instruction is terminated and the CSW is stored.
- 2) If the TEST CHANNEL instruction is in progress, a full CLU logout is taken.
- 3) If a data transfer is in progress, the IOCE requests an I/O interruption in the controlling CE. When permitted, the selector channel is logged out. Appropriate bits are set in the CSW as in 1 above, and a response is returned to the controlling CE.

If I/O interruptions are masked off, the affected channel waits in stopped state until the interruption is allowed, or the IOCE receives a logout signal from the CE via WRITE DIRECT.

When a Selector Channel logout occurs, a concurrent IOCE-processor operation is suspended during the logout, and then resumed upon completion of the channel logout.

12.8.2.4 Check Handling During Selector Channel Logout

When a check is detected during a selector channel logout, a full CLU logout is initiated. Checks detected during a CLU logout are handled as described in 12.8.2.2.

12.8.3 SE LOGOUT

In order to initiate a logout operation for a storage element, the SE must first be placed in logout-stop status by a Logout-Stop signal (LOS) from a CE or an IOCE. A SE accepts a LOS signal from an issuing element provided its communications bit is set on in the SE's configuration control register and in the case of an IOCE, if the response latch for that IOCE is set in the SE.

12.8.3.1 LOS from CE

Logout stop is automatically issued by a computing element whenever it receives either an address check or data check, provided machine check interruptions are not masked off, a CE logout is not in progress, and the Inhibit Logout Stop (ILOS) bit is not already set on in the CCR of the CE. It is also issued by the logout storage diagnose kernel without regard to the machine check mask bit or the ILOS bit.

JUNE 1, 1971

An SE accepts a LOS signal when the communications bit for the issuing CE is set in the SE's CCR; otherwise it is ignored.

12.8.3.2 LOS from IOCE

Logout stop is automatically issued by an IOCE whenever it receives a storage check provided the ILOS bit is not already set on in the CCR of the IOCE and an IOCE logout is not in progress. Bit 3 in check register 2 in the IOCE is set on to identify the element which stopped the SE. Apulsed ELC is issued to all CE's, and the IOCE requests a machine-check interruption in the controlling CE.

The IOCE will also automatically issue a logout stop to an SE when the IOCE detects a parity error on data fetched provided the ILOS bit is not already set on in the CCR of the IOCE and an IOCE logout is not in progress. Unless the SE also detected a storage check during that IOCE access, the SE information pertinent to the IOCE cycle may be lost.

PROGRAMMING NOTE

Upon receipt of the logout-stop signal, the SE issues a level ELC to all CE's in the system. The ELC remains until the SE receives a Logout Complete signal from a configured CE. A storage element may be issued a logout-stop signal by any configured CE or IOCE and thus be placed in a stopped condition, ready to be logged out. However, only a CE configured to communicate with the particular SE has the ability to effect the logout, and restart the SE by issuing the logout-complete signal.

The transfer of logout information from the SE's check latches to a designated main storage is initiated by a CE executing a DIAGNOSE instruction which specifies the Logout Main Storage Kernel for a designated SE (Chapter 8). The SE will always logout the complete format. The logout information is placed in main storage in 6 contiguous doublewords following the MCW specified by the DIAGNOSE instruction which initiated the logout operation. Logout complete is automatically generated at the end of the diagnostic logout or may be issued by a subsystem reset. The logout data may not be directed to a storage location within the element being logged out.

12.8.4 DE LOGOUT

In order to initiate a logout operation for a display element, the DE must first be placed in a logout-stop status by a Logout-Stop Signal (LOS) from a CE or by the occurrence of either an address or data parity check condition during a DG access while the IDES bit is set off. A DE will accept a LOS signal from an issuing CE provided its communications bit is set on in the DE's configuration control register.

12.8.4.1 LOS from CE

Logout stop is automatically issued by a CE whenever it receives either an address check or data check, provided machine-check interruptions are not masked off and the Inhibit Logout Stop (ILOS) bit is not set on in the CCR of the CE and the CE is not doing a logout. It is also issued by the logout storage diagnose kernel without regard to the machine check mask bit.

A DE accepts a LOS signal when the communications bit for the issuing CE is set in the DE's CCR; otherwise it is ignored.

JUNE 1, 1971

12.8.4.2 LOS from DG Access Parity Check

A logout stop condition is automatically entered into by a DE if the IDES bit is off upon occurrence of an address or data parity check condition during a DG access.

PROGRAMMING NOTE

Upon receipt of a logout-stop signal, or upon entering a logout stop condition due to a check condition related to a DG access, the DE issues a level ELC to all CE's in the system. The ELC remains until the DE receives a Logout Complete Signal from a configured CE. A display element may be issued a logout stop and logged out by any configured CE. Upon completion of the logout, the DE is restarted via a logout-complete signal.

The transfer of logout information from a DE to main storage is initiated by a CE executing a DIAGNOSE instruction which specifies the Logout Main Storage Kernel for a designated DE (Chapter 8). In a 9020E System, those kernels specifying Main Storage 6-10 represent DEs 1-5. The logout information is placed in main storage in six contiguous doublewords following the Maintenance Control Word (MCW) specified by the DIAGNOSE which initiated the operation. Logout-Complete is automatically generated at the end of the diagnostic logout, or may be issued via a subsystem reset.

12.9 COMPUTING ELEMENT LOGWORD FORMATS

Figure 12-3 shows the preferential-storage area location addresses, word numbers, and bit designations for the CE logwords.

12.9.1 CHECK REGISTER LOGWORD FORMATS

Check Registers 1 and 2 -- The content of check register 1 is logged out into words 57 and 58 while check register 2 is logged out into word 42. The purpose of each bit in the check registers is defined in Table 12-III.

Storage-Check Address Register -- The content of this register is logged into bit positions 0-3 of word 42. It contains a binary value which identifies the storage element from which an Address Check or Data Check is received. Identifier values are 1-10, corresponding to SE-1 through SE-10 on a 9020D System or SE-1 through SE-5 (value 1-5) and DE-1 through DE-5 (value of 6-10) on a 9020E System.

12.9.2 GENERAL LOGWORD FORMATS

Word 32 - This word contains the parity bits for the F, D, and Q register and the parity bits for the Diagnose Accessible Register (DAR) mask.

Word 33 - This word contains the parity bits for the Select register, ST register and CCR.

Word 34 - This word contains parity bits for the G register, ATR, Mark, ST register and Parallel Adder Latches. The status of several internal control latches are also presented.

JUNE 1, 1971

Word 35 - This word contains the parity bits for the AB register, R register and E register. Also contained are the contents of several control latches used to communicate with IOCEs (e.g., Permit IOCE interrupt and Logout IOCE).

Word 36 - This word contains the contents of the G register (Write Direct), bits 32-39 of the ATR, and the parity bits for the External register. Additionally, the status of the control latches used to communicate with the IOCE are presented.

Word 37 - This word contains the content of the Select register.

Word 38 - This word contains bits 0-31 of the Address Translation Register (ATR).

Word 39 - This word contains the External Interrupt register and Processor Interrupt register as well as several internal control triggers.

Word 40 - This word contains portions of the PSW plus several internal control triggers.

Word 41 - This word contains the status of CE control triggers.

Word 42 - This word contains the Storage Check Address Register, and Check Register 2.

Word 43 - This word contains the content of the MCW, as well as SCAN and FLT controls.

Word 44 - This word contains the contents of the External register.

Word 45 - This word contains the Mark bits, AB register overflow positions (bits 64-67), Parallel Adder Latches overflow (bits 64-67), ST register counter (STC), AB register counter (ABC), and the Local Store Address Register (LSAR).

Word 46 and 47 - These words contain the contents of the AB register.

Word 48 - This word contains information associated with the Preferential Storage Base Address Register (PSBAR) and portions of the Program Status Word.

Bits 0-3 contain the physical PSBAR. This quantity identifies the SE containing the PSA (or APSA).

Bits 4-7, contain the PSBAR counter plus parity. This counter is used when incrementing the PSBA to access the alternate PSA.

Bits 8-19 contain the logical PSBAR plus parity. All fixed addresses pertinent to logout, status words, etc., are relative to the 4096 byte block of storage referenced by logical PSBAR.

Bit 20, Alternate PSBAR, indicates that the CE is operating in its Alternate Preferential Storage Area (APSA).

Bit 21, PSBAR Configuration Check, indicates that the current PSA address does not reference an SE configured to this CE. This latch is never set during a logout, as such a condition precludes logging out.

Bits 24-31, contain part of the PSW.

JUNE 1, 1971

Word 49 - This word contains bits 0-31 of the Configuration Control Register (CCR).

Word 50 - This word contains the contents of the Diagnose Accessible Register (DAR) mask.

Word 51 - This word contains the contents of the Serial Adder Latches (SADDL) and the Instruction Counter (IC).

Word 52 - This word contains the contents of the R register and E register.

Word 53 - This word contains the contents of the D register and F register.

Word 54 and 55 - These words contain the contents of the Q register.

Word 56 - This word contains the contents of the Diagnose Accessible Register (DAR). It records the pending abnormal condition interruption requests from other system elements, and from itself (e.g., OTC (own) and OBS (own)).

Word 57 - This word contains bits 1-8, 11-18, and 23-27 of Check Register 1.

Word 58 - This word contains bits 0, 9, 10, and 19 of Check Register 1, parity bits for LS Working Register, PSW Register, IC, N Register, Serial Adder Latches, and IOCE communication controls.

Word 59 - This word contains the contents of the Local Store Working Register.

Word 60 - This word is not used.

Word 61 - This word contains the Parallel Adder ingating conditions.

Word 62 - This word is not used.

Word 63 - This word contains a portion (bits 70-99) of the Read Only Storage Data Register (ROSDR) plus several internal control triggers.

Word 64 - This word is not used.

Word 65 - This word contains a portion (bits 37-68) of the ROSDR plus several internal control triggers.

Word 66 - This word is not used.

Word 67 - This word contains a portion (bits 2-35) of the ROSDR.

Word 68 - This word contains the L register and K register parity bits.

Word 69 - This word contains bits 1-11 of the ROS Address Register (ROSAR) and bits 0-11 of the Previous ROS Address Registers A and B (PROSA).

Word 70 - This word contains the content of the N register plus the parity bits for the X, Y, and M registers.

Word 71 - This word contains portions of ROSAR, PROSA, and ROSDR as well as the contents of the Local Store Address Register.

JUNE 1, 1971

Additionally, the status of the S, D, AB and F ingating to the Parallel Adder is contained in bits 16-19.

Word 72 - This word contains the content of the K register.

Word 73 - This word contains the content of the Maintenance Control Word.

Word 74 and 75 - These words contain the contents of the LM register.

Word 76 and 77 - These words contain the contents of the XY register.

Word 78 and 79 - These words contain the contents of the ST register.

JUNE 1, 1971

PSA LOC-ATION	PSA WORD NO.	BIT POSITIONS IN MAIN STORAGE																														COMMENTS																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29		30	31																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
128	32	F REG P 0-7							D REG P 8-15							Q REGISTER P 16-23							DAR MASK P 24-31							PARITY FOR F, D, Q AND DAR MASK																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
132	33	SELECT REG P 0-7							ST REGISTER P 8-15							CCR P 16-23							P 24-31							PARITY FOR SR, K, ST AND CCR																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
136	34	CE SEL SATR 1							G REG P 0-7							ATR P 8-15							IC REQ P 16-23							RPT INSTR ADJ P 24-31							INSTR STEP TGR P 0-7							DISABLE INTER-LEAVE P 8-15							STAT TGR A P 16-23							I FETCH 1 TGR P 24-31							IL NOT AVAIL P 0-7							EXT INTRPT CE4 WDD 28 P 8-15							MCW RTP 0 P 16-23							ADDR SEQ 0 P 24-31							FLT CLOCK 0 P 0-7							MARK P 0-7							ST REG P 32-39							AB REG P 64-67							PADDL P 64-67																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
140	35	EXTERNAL TRIGGERS PERMIT IOCE 2							LOGOUT IOCE 3							BCU OSC INH 1							SELECT IOCE 2							IOCE MC REQ 3							INTRPT GATE TGR P 0-7							EXTERNAL REG P 8-15							R REG P 16-23							E REG P 24-31							P 0-7							P 8-15							P 16-23							P 24-31																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
144	36	CE SEL SATR 2							TIME CLOCK STEP 3							BCU OSC INH 4							SELECT IOCE 1							IOCE MC REQ 2							INTRPT GATE TGR 3							EXTERNAL REG P 0-7							G REG (WRITE DIRECT) P 8-15							ATR-2 P 16-23							P 24-31							P 0-7							P 8-15							P 16-23							P 24-31							G REG, ATR-2																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
148	37	PAM/RCU 1/1							TCU 2/2							SE/DE 3/3							CE 4/4							SCU 5/5							IOCE 6/6							1							2							3							1							2							3							SELECT REG																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
152	38	ADDRESS TRANSLATION REGISTER (ATR-1)																																ATR-1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
156	39	STAT TRIGGERS B							BLOCK 1 FETCH C							I FETCH 2 TGR D							I FETCH 3 TGR E							EXEC IN PROG TGR F							PROG STORE COMP G							I FETCH REQ TGR H							BR INV ADDR TGR							I FETCH INV ADDR TGR							CE1 WDD 20							CE1 RDD 21							CE2 WDD 22							CE2 RDD 23							EXTERNAL INTERRUPT REGISTER TIMER 24							PB 25							CE3 WDD 26							CE3 RDD 27							CE4 RDD 29							PIR 30							DAR 31							1							2							3							TIMING GT TGR																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
160	40	SYSTEM MASK 0							KEY 1							A 12							M 13							W 14							P 15							IC IN LSWR							COND CODE 34							PROGRAM MASK 35							MC INTRPT 36							PROGRAM INTERRUPT CODE 37							INTERRUPT PRIORITY 38							1							2							PSW																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
164	41	D REQ							SCAN REQ							SET KEY							INSERT KEY							3 CYCLE REQ							PAGE 1 LTH							RPT INSTR INIT							PULSE MODE ADJ							PULSE MODE INIT							STOP TGR							BLOCK TGR							PASS PULSE TGR							BLOCK INTRPT							FORCE ADDR TGR							SINGLE CYCLE TGR							CHECK SUM							CLOCK INHIBIT							RPT TEST							FLT TEST							ROS TEST TGR							SCC TGR							SYNC TGR							DISABLE INTER-LEAVE REV STOR ADDR							FLT BKSP							CONSOLE LOG OUT							SOROS							DIAG TGR							FLT REL							TIC							GAP																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
168	42	STORAGE CHECK ADDRESS REGISTER 0							360 MODE 1							TEST MODE 2							SE STPD IN 360 MODE 3							SAB CHK							SDBI CHK							SE/DE T/O							SE/DE ADDR CHK							SE/DE DATA CHK							SDBO LOS TO SE/DE							IOCE BUS CHK							LOCAL STORE BUS CHK							CCR PTY CHK							ATR PTY CHK							PSBAR PTY CHK							PSBAR NOT CONF							PSBAR ALT CHK							SPLIT LOG OUT							LOG ROS CHK							LOG ADDR CHK							CE LOG REQ							RDD T/O							CR2																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
172	43	LACMP 1							RSA FSP 2							RMP 3							MCW REGISTER RSARP 4							LOC 5							DT 20							SCAN ADDRESS SEQUENCER 1							SCAN CYCLE COUNTER 2							3							FLT CLOCK 1							ROS TEST SEQUENCER 0							1							2							3							4							0							1							2							3							4							5							6							7							8							9							10							11							12							13							14							15							16							17							18							19							20							21							22							23							24							25							26							27							28							29							30							31							32							33							34							35							36							37							38							39							40							41							42							43							44							45							46							47							48							49							50							51							52							53							54							55							56							57							58							59							60							61							62							63							64							65							66							67							0							1							2							3							4							5							6							7							8							9							10							11							12							13							14							15							16							17							18							19							20							21							22							23							24							25							26							27							28							29							30							31							32							33							34							35							36							37							38							39							40							41							42							43							44							45							46							47							48							49							50							51							52							53							54							55							56							57							58							59							60							61							62							63							64							65							66							67							0							1							2							3							4							5							6							7							8							9							10							11							12							13							14							15							16							17							18							19							20							21							22							23							24							25							26							27							28							29							30							31							32							33							34							35							36							37							38							39							40							41							42							43							44							45							46							47							48							49							50							51							52							53							54							55							56							57							58							59							60							61							62							63							64							65							66							67							0							1							2							3							4							5							6							7							8							9							10							11							12							13							14							15							16							17							18							19							20							21							22							23							24							25							26							27							28							29							30							31							32							33							34							35							36							37							38							39							40							41							42							43							44							45							46							47							48							49							50							51							52							53							54							55							56							57							58							59							60							61							62							63							64							65							66							67							0							1							2							3							4							5							6							7							8							9							10							11							12							13							14							15							16							17							18							19							20							21							22							23							24							25							26							27							28							29							30							31							32							33							34							35							36							37							38							39							40							41							42							43							44							45							46							47							48							49							50							51							52							53							54							55							56							57							58							59							60							61							62							63							64							65							66							67							0							1							2							3							4							5							6							7							8							9							10							11							12							13							14							15							16							17							18							19							20							21							22							23							24							25							26							27							28							29							30							31							32							33							34							35							36							37							38							39							40							41							42							43							44							45							46							47							48							49							50							51							52							53							54							55							56							57							58							59							60							61							62							63							64							65							66							67							0							1							2							3							4							5							6							7							8							9							10							11							12							13							14							15							16							17							18							19							20							21							22							23							24							25							26							27							28							29							30							31							32							33							34							35							36							37							38							39							40							41							42							43							44							45							46							47							48							49							50							51							52							53							54							55							56							57							58							59							60							61							62							63							64							65							66							67							0							1							2							3							4							5							6							7							8							9							10							11							12							13							14							15							16							17							18							19							20							21							22							23							24							25							26							27							28							29							30							31							32							33							34							35							36							37							38							39							40							41							42							43							44							45							46							47							48							49							50							51							52							53							54							55							56							57							58							59							60							61							62							63							64							65							66							67							0							1							2							3							4							5							6							7							8							9							10							11							12							13							14							15							16							17							18							19							20							21							22							23							24							25							26							27							28							29							30							31							32							33							34							35							36							37							38							39							40							41							42							43							44							45							46							47							48							49							50							51							52							53							54							55							56							57							58							59							60							61							62							63							64							65							66							67							0							1							2							3							4							5							6							7							8							9							10							11							12							13							14							15							16							17							18							19							20							21							22							23							24							25							26							27							28							29							30							31							32							33							34							35							36							37							38							39							40							41							42							43							44							45							46							47							48							49							50							51							52							53							54							55							56							57							58							59							60							61							62							63							64							65							66							67							0							1							2							3							4							5							6							7							8							9							10							11							12							13							14							15							16							17							18							19							20							21							22							23							24							25							26							27							28							29							30							31							32							33							34							35							36							37							38							39							40							41							42							43							44							45							46							47							48							49							50							51							52							53							54							55							56							57							58							59							60							61							62							63							64							65							66							67							0							1							2							3							4							5							6							7							8							9							10							11							12							13							14							15							16							17							18							19							20							21							22							23							24							25							26							27							28							29							30							31							32							33							34							35							36							37							38							39							40							41							42							43							44							45							46							47							48							49							50							51							52							53							54							55							56							57							58							59							60							61							62							63							64							65							66							67							0							1							2							3							4							5							6							7							8							9							10							11							12							13							14							15							16							17							18							19							20							21							22							23							24							25							26							2						

JUNE 1, 1971

PSA LOCATION	PSA WORD NO.	BIT POSITIONS IN MAIN STORAGE																															COMMENTS										
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		31									
192	48	PHYSICAL PSBAR				P PSBAR COUNTER			LOGICAL PSBAR											ALT PSBAR	PROGRAM STATUS WORD SYSTEM MASK				SE STPD	PSA LOCKOUT			PSBAR														
196	49	STATE		SCON			ILOS		SE/DE											CE			IOCE			CCR																	
200	50	IOCE			SE/DE											SPARE		PAM/RCU		TCU			SCU			OTHER CE/OWN OTC				CE OWN OBS	DAR MASK												
204	51	SERIAL ADDER LATCHES (SADDL)							INSTRUCTION COUNTER (IC)																								SADDL, IC										
208	52	R REGISTER															E REGISTER															R REG, E REG											
212	53	F REGISTER (READ DIRECT)															D REGISTER															D REG, F REG											
216	54	Q REGISTER																															Q REG										
220	55	Q REGISTER																															Q REG										
224	56	IOCE1		IOCE2		IOCE3		SE/DE ELC											SPARE		PAM/RCU ELC			TCU ELC			SCU ELC			OTHER CE ELC/CE OWN OTC				CE OWN OBS	DAR								
228	57	PARALLEL ADDER FULL SUM-CHECK															PARALLEL ADDER HALF-SUM CHECK															ROS PARITY CHECK			SADD HALF SUM CHK	SADD FULL SUM CHK	CR1						
232	58	LSWR															CHECK REGISTER 1																IC REG										
236	59	LOCAL STORAGE WORKING REGISTER (LSWR)																															LSWR										
240	60																																										
244	61	T32-63 TO PADDA				T32-47 TO PADDA		T48-63 TO PADDA		K0-31 TO PADDA		D8-31 TO PADDA											FMT0	FMTN	FMTW	IC8-31 TO PADDB	AB4-7 TO PADDB	AB8-31 TO PADDB	AB32-63 TO PADDB	AB64-67 TO PADDB	AB6-31 TO PADDB	AB32-67 TO PADDB	AB64-67 TO PADDB	EX6 TO PADDB	HOT ONE TO PADDB	E8-11 TO PADDB	E12-15 TO PADDB	E8-11 TO PADDB	Q4-15 TO PADDB	Q20-31 TO PADDB	Q36-47 TO PADDB	Q52-63 TO PADDB	
248	62																																										
252	63	READ ONLY STORAGE DATA REGISTER (ROSDR)																												RIGHT DIGIT TGR		S TGR	LEAVE TGR	STEP ABC TGR	ROSDR								

Figure 12-3 - CE Logword Formats (continued)

JUNE 1, 1971

PSA LOC-ATION	PSA WORD NO.	BIT POSITIONS IN MAIN STORAGE																															COMMENTS					
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		31				
256	64																																					
260	65	READ ONLY STORAGE DATA REGISTER (ROSDR)																												INHIBIT LS WRITE	PROSA A LAST	FLT MODE	ROSDR					
264	66																																					
268	67	READ ONLY STORAGE DATA REGISTER (ROSDR)																												P 2-42			ROSDR					
272	68	L REG				K REG																																
276	69	READ ONLY STORAGE ADDRESS REGISTER (ROSAR)											PREVIOUS ROS ADDRESS (A)											PREVIOUS ROS ADDRESS (B)											ROSAR, PROSA, PROSB			
280	70	M REG				X REG				Y REG				N REGISTER													N REG											
284	71	ROSAR	PROSA	SPARE	READ ONLY STORAGE DATA REGISTER (ROSDR)											S 0-31 TO PADDA 32-63	D 8-31 TO PADDA 8-31 C	AB 0-31 TO PADDB 32-63	F 4-7 TO PADDB 60-63	LOCAL STORAGE ADDRESS REG				WR LOCAL STORE	ROSAR, LSAR													
288	72	K REGISTER																															K REG					
292	73	MAINTENANCE CONTROL WORD (MCW)											RNOP	FDGR	DSE 1	RERP	RC	WRAP DE	MCW																			
296	74	LM REGISTER																															LM REG					
300	75	LM REGISTER																															LM REG					
304	76	XY REGISTER																															XY REG					
308	77	XY REGISTER																															XY REG					
312	78	ST REGISTER																															ST REG					
316	79	ST REGISTER																															ST REG					

Figure 12-3 - CE Logword Formats (continued)

JUNE 1, 1971

12.10 INPUT/OUTPUT CONTROL ELEMENT LOGWORD FORMATS

Figure 12-4 shows the preferential-storage area location addresses, word numbers, and bit assignments for the IOCE logwords.

12.10.1 CHECK-CONDITION LOGWORD FORMATS

Check Registers 1 and 2 -- The content of the check registers is logged into words 89 and 90. The purpose of each bit in the check registers is defined in Table 12-IV.

Word 108, Bit 14 - Timeout -- A line is sent to the channels after a timeout, following the issuance of an I/O instruction of an I/O interruption proceed. It is used for checking burst mode on a Multiplexor Channel. If the interface to a Selector Channel is hung up, issuance of Timeout initiates a selector channel logout by causing an I/O interruption request for that channel. If an I/O instruction is being executed (other than TEST CHANNEL), the logout is initiated without an interruption.

Word 108, Bit 15 - Timeout Check -- This indicates that there was no response from a channel after issuance of Timeout. The condition initiates a CLU logout.

Word 108, Bit 16 - START I/O Fault -- When the CLU detects a program check on a START I/O instruction, bit 16 is set on. If the addressed channel was not busy, a program check is indicated in the CSW.

Word 109, Bit 23 - First Cycle Check -- This indicates that an I/O micro-program routine did not start with the correct micro-instruction.

Word 110, Bit 18 -- I/O Check Mode -- This indicates that a channel has control of the CLU. Other checks encountered with this bit set on can then be associated with channel operation.

Word 112, Bits 22 and 23 -- These bits are used to indicate that the multiplexor channel has detected a program or storage protection check (Chapter 10).

Word 118, Bit 25 - Incorrect Length Indication: -- (Chapter 10).

Word 118, Bits 26 and 27 -- These bits indicate a Selector Channel has detected a program check or a Storage Protect Check (Chapter 10).

Word 118, Bit 28 - Channel Data Check -- Occurs when a parity check is detected on either Bus-in or Bus-out during data transmission.

Word 118, Bit 29 - Channel Control Check -- This indicates that one of the following checks is detected:

- 1) Byte Counter Parity Check -- See Logword 118, bits 0-4, and logword 119, bit 28.
- 2) Bus-out parity check while transmitting an address or command.
- 3) End-of-Operation Validity Check -- At the completion of an I/O operation, Logword 123 should be set to all zeros, and Logword 124, bits 28-31, should be set to all ones. This check is made

JUNE 1, 1971

at the completion of each operation. If it is not met the Channel Control Check is set on.

- 4) Routine Response Check -- This check verifies that a channel ROS routine is started at a legal location, and that these starting locations are not subsequently used in a routine.
- 5) CLU Check -- Any check condition in the CLU while the channel is using it will set Channel Control Check.

Items (1) and (2) result in a Selector Channel logout. Items (3) through (5) result in a CLU logout which includes logging the Selector Channel.

Word 118, Bit 30 - Interface Control Check -- This results in a selector channel logout. It is caused by any of the following conditions:

- 1) Bus-in parity check - while receiving an address or a status byte.
- 2) A timeout (word 108, bit 14) - when the I/O control unit fails to respond completely to an interface sequence.
- 3) The detection of an improper I/O interface sequence or presentation of improper status for the sequence by the I/O control unit.
- 4) The receipt of an improper address on the interface during selection. (Word 123, bit 15 is also set on for this condition.)

Word 118, Bit 31 - Chaining Checks -- This indicates that there is an overrun on a read operation due to chaining (Chapter 10) or a non-PSA storage access was made to a Logout Stopped SE.

Word 123, Bit 14, 15 - Compare Equal/Not Equal -- This indicates the results of comparing the Unit Address received with the Unit Address sent over the channel-to-device interface. Compare not equal results in an Interface Control Check condition (Word 118, bit 30).

Word 123, Bit 22 - Channel Stop -- This bit is set on whenever the Selector Channel is logged out.

PROGRAMMING NOTE 1

Word 123, bit 22, may be used to determine whether a Selector Channel was logged out when a CLU logout occurs in an IOCE.

PROGRAMMING NOTE 2

In order to determine whether a CLU logout was initiated by an IOCE-processor or an IOCE-input/output operation, it is necessary to examine the states of two bits in the logout data: I/O Routine Mode (bit 3, word 109), and the IOCE-processor mode bit (bit 17, word 88). The significance of these bits is shown in the following table:

JUNE 1, 1971

PROCESSOR MODE BIT	I/O ROUTINE MODE BIT	IOCE OPERATION AT TIME OF LOGOUT
1	0	IOCE-processor running, IOCE- input/output idle.
0	0	Entire IOCE was idle - no processor or I/O operation occurring.
1	1	IOCE-processor was operating, but is now held off by I/O break-in.
0	1	IOCE-processor was idle. An I/O break-in taking place.

12.10.2 GENERAL LOGWORD FORMATS

Word 81 -- This word contains the storage address register (SAR) and byte stats. The SAR contains the address which is provided to mach or main on a storage access. The byte stats determine whether a byte will be stored or regenerated. A byte stat of 1 indicates a store, a byte stat of 0 a regeneration.

Word 82 -- This word contains the Instruction Address Register (IAR). The IAR holds the address of the storage location from which the next instruction will be fetched.

Word 83 -- This word contains bits 0-30 of the Read-Only Storage Data Register (ROSDR). The ROSDR bits provided the micro-program control for the last machine cycle before logout.

Word 84 -- This word contains ROSDR bits 31-55 and the Mover Function Register. The mover function register provides micro-program control of the logical functions performed in the mover. Functions are defined for both CLU and I/O modes of operation.

Word 85 -- This word contains ROSDR bits 56-87.

Word 86 -- The previous ROS address is the address of the micro-instruction in Read-Only Storage which controlled the next to last machine cycle before logout. This word also contains ROSDR bits 88-89.

Word 87 -- The Storage Protect Key of the current PSW occupies bits 0-4. The current ROS address is the address of the micro-instruction which is logged out in words 83, 84, 85 and 86. The One Syllable OP and Refetch stats; and the FLT End stat check are used in micro-program instruction handling. These are significant only for the IOCE-Processor, as are the Program Status Word bits 32-36. Bits 18 and 19 contain the CE condition code register and bit 21 contains the CE external interrupt request.

Word 88 -- The SE accesses request bits 0-11 indicate which SE was last accessed by the IOCE (during a logout, the SE that is being logged into will be indicated in these positions). Bits 18-31 contain the content of the IOCE's PSBAR. This quantity is bits 8-19 of the "effective" PSBA last received on a PSA operation. Bits

JUNE 1, 1971

12-16 contain the system mask and the A, M, W, P bits of the IOCE PSW. Bit 17 contains the IOCE-processor mode latch bit.

Words 89-90 -- These words contain the content of Check Registers 1 and 2.

Words 91-92 -- These words contain the content of the Address Translator (ATR). Positions 1-6 of ATR are logged into Word 91, and positions 7-12 are logged into Word 92.

Word 93 -- The word contains the IOCE's Configuration Control Register (CCR). State bits 0 and 1 determine whether the IOCE is in State Three, Two, One, or Zero. The scon bits determine from which CE(s) the IOCE will accept a configuration mask. The inhibit logout-stop (ILOS) bit controls the ability of the IOCE to issue logout stop to an SE. SE bits 1-10 show which SE's are configured to the IOCE. CE bits 1-4 show to which CE the IOCE is configured.

Word 94 -- This is a duplicate of word 84 but may have different data content.

Word 95 -- This is a duplicate of Word 87 but may have different data content.

Word 96 -- The I/O Mode, General Purpose, Sign and Carry Stats are used for micro-program control. The timer and interrupt switch portion of the External Interrupt Register, and the No Retry Latch are significant only for the IOCE-Processor. The L and M byte counters control byte selection for the L and M registers. The F, Q, and I/O registers are data path registers. The Wait Loop Trigger indicates that the IOCE-Processor is in its Wait Loop.

JUNE 1, 1971

PSA LOCATION	PSA WORD NO.	BIT POSITIONS IN MAIN STORAGE																														COMMENTS		
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29		30	31
384	96	I/O MODE	I-O REGISTER		TIMER		INTR PB	L BYTE COUNTER		M BYTE COUNTER		F REGISTER			Q REG	WAIT LOOP	GENERAL PURPOSE STATS							L SIGN STAT	R SIGN STAT	CARRY STAT	NO RETRY LATCH	STATUS GROUP						
388	97	LOCAL STORE ADR REG				LS FN REG		J REGISTER			MD COUNTER			G1 COUNTER			G2 COUNTER										LSAR GROUP							
392	98	P 0-7	0	1	2	3	4	5	6	7	P 8-15	8	9	10	11	12	13	14	15	P 16-23	16	17	18	19	20	21	22	23	P 24-31	24	25	26	27	H REG
396	99																															H REG		
400	100	P 0-7	0	1	2	3	4	5	6	7	P 8-15	8	9	10	11	12	13	14	15	P 16-23	16	17	18	19	20	21	22	23	P 24-31	24	25	26	27	M REG
404	101																															M REG		
408	102	P 0-7	0	1	2	3	4	5	6	7	P 8-15	8	9	10	11	12	13	14	15	P 16-23	16	17	18	19	20	21	22	23	P 24-31	24	25	26	27	R REG
412	103																															R REG		
416	104	P 0-7	0	1	2	3	4	5	6	7	P 8-15	8	9	10	11	12	13	14	15	P 16-23	16	17	18	19	20	21	22	23	P 24-31	24	25	26	27	L REG
420	105																															L REG		
424	106	P 0-7	0	1	2	3	4	5	6	7	P 8-15	8	9	10	11	12	13	14	15	P 16-23	16	17	18	19	20	21	22	23	P 24-31	24	25	26	27	SDR
428	107																															SDR		
432	108	I/O INSTRUCTION				CHANNEL SELECT		REPLY REGISTER				I/O INTR RQST	PERMT I/O INTR	TIME OUT	TIME OUT CHECK	FAULT ON START I/O	SPCIB FULL	RTNE RQST BFP POS X								COMMON CHANNEL								
436	109	RTNE RCVD	PCI ENABL	BREAK IN	I/O RTNE MODE	EARLY FIRST CYCLE	FIRST CYCLE	FIRST CYCLE CHAIN	LOCAL STORE ACCESS		CHAL DTC	ALCH DTC	CHAIN REG	LAST CYCLE	BREAK OUT	MEMORY BYTE CONTROL REGISTER			TIMING SIGNALS		CONTROL SIGNALS		FIRST CYCLE CHECK					COMMON CHANNEL						
440	110	CHANNEL BUFFER 1				CHANNEL BUFFER 2			CHANNEL BUFFER 3			I/O STATS				I/O CHECK MODE	CHANNEL LOG RQSTS			GATE STATUS	LOG RESET CHANL					COMMON CHANNEL								

Figure 12-4 - IOCE Logword Formats (continued)

JUNE 1, 1971

PSA LOCATION	PSA WORD NO.	BIT POSITIONS IN MAIN STORAGE																														COMMENTS		
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29		30	31
444	111	DATA BUFFER 1															DATA BUFFER 2															START LOG-OUT	MPX CHANNEL	
448	112	MULTIPLEX CHANNEL INTERFACE TAGS												MPX BUS OUT							CHECKS			MPX CHANNEL										
452	113	CONTROLLED EMIT				ROUTINE REQUEST REG				PRIORITY REG			CONTROL TRIGGERS				MPX I/O STATS					MPX CHANNEL												
456	114	B REGISTER																																SELECT CHANNEL
460	115																													B REGISTER PARITY				SELECT CHANNEL
464	116																													C REGISTER PARITY				SELECT CHANNEL
468	117	C REGISTER																																SELECT CHANNEL
472	118	BYTE COUNTERS				LAST WORD AND EOR CONTROL				DATA BUFFER CONTROL				READ INSTRUCTIONS				WRITE INSTRUCTIONS				CHANNEL CHECKS				SELECT CHANNEL								
476	119																													SELECTR CHAN GRP 3				SELECT CHANNEL
480	120																													SELECTR CHAN GRP 4				SELECT CHANNEL
484	121	ROUTINE TYPE REGISTER				CYCLE COUNTER				A CLOCK			PRIORITY RQST LATCHES				ROUTINE REQUEST ADDRESS CONTROLS				PRIORITY DETECT LATCHES				SELECT CHANNEL									
488	122	GENERAL PURPOSE REGISTER							FLAGS				RECORD TRANSFER CTRLS				MULTIPURPOSE LATCHES				INTERFACE CONTROLS				SELECT CHANNEL									
492	123	A CLOCK LATCHES				SPL PURPOSE LATCHES		INTERFACE CONTROLS												INTERFACE TAGS				SELECT CHANNEL										
496	124																													SELECTR CHAN				SELECT CHANNEL
500	125	MPLX CHANNEL UNIT ADDRESS																MPLX CHANNEL UNIT ADDRESS																

Figure 12-4 - IOCE Logword Formats (continued)

JUNE 1, 1971

Word 97 -- The Local Storage Address Register (LSAR) determines the word which will be accessed from Local Storage. The Local Storage Function Register is a source of bits 0 and 1 of LSAR, and determines the quadrant of the Local Storage to be accessed. The J Register is a data path register. MD, G₁, and G₂ are data path counters used for internal byte counting by the IOCE-Processor.

Words 98-105 -- These words contain the content of the data path registers H,M,R, and L and their parities.

Words 106, 107 -- These words contain the content of the Storage Data Register (SDR). Information sent to and received from Main Storage and MACH Storage is held in this register.

Words 108, 109, 110 -- These words contain the content of registers, status indicators, and control bits used in the Common Channel.

Words 111, 112, and 113 -- These words contain the content of registers, status indicators and control bits used in the multiplexor channel. Of particular interest is Word 112 which contains the status of Channel-to-Device interface lines and the status of the data bus from the channel to the device. Word 111, Buffer 2 holds bytes that are sent to the I/O interface. Buffers 1 and 2 receive bytes from the I/O interface.

Words 114, 115, 116, and 117 -- These words contain the content of the Selector Channel B and C Registers. The C register is used to supply bytes to, or receive them from the I/O interface. The B register is connected to the C register, and is used for communicating with the data path which stores or fetches I/O main or MACH storage data.

Words 118-124 -- These words contain Internal Counters, Registers, and Status and Control Bits for Selector Channel Operation. Bits 28 and 29 of word 122, and Bits 23-31 of word 123, contain the status of the Channel to Device interface lines.

Word 125 -- This word contains the current unit address and the subchannel address of the device with which the multiplexor channel is working. Because of hardware design, these addresses are identical.

12.11 STORAGE ELEMENT LOGWORD FORMATS

Execution of a DIAGNOSE instruction which specifies the Logout Main Storage Kernel causes the logout information from the SE designated to be placed in the six doublewords of contiguous main storage following the Maintenance Control Word (MCW) designated by the DIAGNOSE instruction which initiated the operation (Chapter 8). Figure 12-5 shows the SE logwords.

12.11.1 SE LOGWORDS

Words 0 and 1 - These words contain the contents of one of the SE's (even) Storage Data Register (SESDR) as specified by bit 5 of the storage address. These words contain the data doubleword fetched (fetch operation) from the even addressed storage or the data doubleword being stored (store operation) from the accessing element on the last even storage cycle.

JUNE 1, 1971

Word 2 - This word contains the even marks (or byte stats) and the content of the Tag plus the SE Storage Address Register (SESAR). If the last even storage cycle serviced a CE, this word contains the marks which control those bytes to be regenerated on a store, the tag, the high/low storage bit (bit 5 of the address), and the content of the even SESAR (address bits 6-19).

If the last even storage cycle serviced an IOCE, the mark field contains the byte stats in either the high or low order four bits, as designated by bit 21 of the IOCE SAB.

If the SE had encountered an Address Check on an IOCE access to the even side, the IOCE address bits 20-23 are placed in the address field.

Word 3 - This register contains the content of the SE's Configuration Control Register (CCR), and the even request and response latches. The request latches indicate which elements were requesting service to the even side of Storage when the check condition was detected. The response latches indicate which element was serviced in the last SE even cycle. The existence of both IOCE and CE or more than one CE or more than one IOCE response latch set would indicate a multi/accept check condition.

Word 4 - This word contains the SE ID, the type of operation last performed on the even side, IOCE SAB parity bits, and the even mark and address parity bits.

The SE ID is a plugged identification which the SE compares against the tag received from an accessing element.

The operation field indicates which operation was last performed on either the even side of storage or a common SE Storage Protect operation as follows:

ISK	-	Insert Storage Key (common to both sides)
SSK	-	Set Storage Key (common to both sides)
T&S	-	Test and Set (even)
Store	-	Store operation (even)
Fetch	-	Fetch operation (even)
Suppress Log Chk.	-	Active during a portion of either a CE or IOCE logout.

The IOCE SAB parity bits indicate the contents of the parities for bytes 1, 2, and 3 of the IOCE addresses received from the IOCE.

The Mark and Address parity bits are associated with the corresponding logout fields contained in Logword 2. The address parity bits correspond to either a CE or IOCE access.

Word 5 - This word contains both even and common logic SE malfunction indicators and the parity bits for CCR and the SESDR that is contained in Logwords 0 and 1.

The check conditions are as follows:

Mark Parity Chk.	-	Improper parity on even Marks
Addr Parity Chk.	-	Improper parity on even SESAR
Data Parity Chk.	-	Improper parity on even SESDR
SP Parity Chk.	-	Improper parity on Storage Protection In Key Register (SPIKR) or Storage Protect Out Key Register (SPOKR) or Storage Protect Address Register (SPAR) during even storage cycle.
Key	-	Storage Protect Key Mismatch on even side

JUNE 1, 1971

Normal Op Chk	-	Inconsistency between control lines and operation requested
CCR Parity	-	Improper parity on CCR
Tag Parity	-	Improper parity on Tag (Address 1-5)
Tag	-	Tag mismatch: address not within range of SE
SBI Chk (IOCE)	-	Improper parity on data being stored by the IOCE
OBS	-	On Battery Signal
Thermal	-	Over temperature indication
Double Cycle Timeout	-	A CE or IOCE did not return with the second request to the SE on a Double cycle operation.
Multi-Accept Chk	-	Two user requests or an Odd and Even request from the same CE have been granted priority by the SE.

Words 6 and 7 - Same as Logword 0 and 1 but for the odd side.

NOTE: IOCE SBO is never contained in these Logwords.

Word 8 - Same as Logword 2 but for the odd side.

Word 9 - This word contains the response and request latches for the odd side of the SE. See Logword 3 for discussion.

Word 10 - This word contains the SPIKR, SPOKR, type of operation during last odd storage cycle, and the parity bits for IOCE SAB, odd marks and odd address.

The SPIKR contains the Storage Protect Key received from the accessing element while the SPOKR contains the Storage Protect Key fetched from the Storage Protect Buffer.

Word 11 - This word contains check condition indications from the odd side of the SE, the content of the Storage Protect Address Register (SPAR) and the parity bits for the SESDR contained in Logwords 6 and 7.

12.12 DISPLAY STORAGE LOGWORD FORMATS

Execution of a DIAGNOSE instruction which specifies the Logout Main/Display Storage Kernel causes the logout information from the DE designated to be placed in the six doublewords of contiguous main storage following the MCW designated by the DIAGNOSE instruction which initiated the operation (Chapter 8). Figure 12-6 shows the DE Logwords.

12.12.1 DE LOGWORDS

Words 0 and 1 - These words contain the content of the Display Elements (even) Storage Data Register (DESDR). Following a fetch, they contain the data doubleword fetched from the even addressed storage for either a CE or DG. Following a store, they contain the data doubleword as received from the accessing CE for storage into the even addressed storage.

Word 2 - This word contains the even marks, the parity bits for the DESDR contained in Logwords 0 and 1, and the contents of the Display Element's (even) Storage Address Register (DESAR). The marks and the content of the DESAR indicate information regarding the last even storage cycle by either a CE or DG.

Word 3 - This word contains the Storage Protect In Key Register (SPIKR), Storage Protect Out Key Register (SPOKR), the parity bit for even marks, the type of operation last performed on the even side of storage, and the check condition (if any) detected on the last even storage cycle or the check condition (if any) detected in common logic.

The SPIKR contains the storage protect key received from the accessing CE on the last storage cycle. The SPOKR contains the storage protect key fetched from the storage protect buffer for the last CE or DG access. These fields are significant on Storage Protect errors only.

The operation bits indicate the type of operation last performed on the even side as follows:

- ISK - Insert Storage Key
- SSK - Set Storage Key
- T & S - Test and Set
- Store - Store operation
- Fetch - Fetch operation

The check conditions indicate the type of malfunction detected (if any) on the last even storage cycle as follows:

- SP Addr Parity Chk - Improper parity on Storage Protect Address
- SP Key Parity Chk - Improper parity on Storage Protect Key
- Key - Storage Protect Key mismatch
- Data Parity Chk - Improper parity on DESDR
- Addr Parity Chk - Improper parity on DESAR
- Mark Parity Chk - Improper parity on Marks
- Cancel Latch - This latch indicates that even storage forced a regeneration of existing storage data. It will normally be set on tag parity, tag mismatch and multi-accept type malfunctions.

JUNE 1, 1971

WORD NO.	BIT POSITIONS IN MAIN STORAGE																																COMMENTS	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
0	SESDR (EVEN)																																SESDR (EVEN)	
1	SESDR (EVEN)																																SESDR (EVEN)	
2	MARKS (EVEN) 0-7 8-15 16-23 24-31 32-39 40-47 48-53 54-63								TAG 1 2 3 4				HI/LO 5	SESAR 1-14 (EVEN) 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23														IOCE SAB 24 25 26			MARKS, SESAR (EVEN)			
3	CCR(8-15) 1 2 3 4				CCR 12		IOCE 1 2 3			REQUEST (EVEN) 1 2 3 4				IOCE 1 2 3			RESPONSE (EVEN) 1 2 3 4 1 2 3						CCR(0-7) STATE: S ₀ S ₁ 1 2 3 4 SCON: 5 6 7			CCR, REQUEST AND RESPONSE LATCHES (EVEN)								
4					SE ID 1 2 3 4				OPERATION ISK SSK		OPERATION (EVEN) T & S STORE FETCH			SUPP LOG CHK				IOCE SAB PARITY P ₀ P ₁ P ₂			MARK PARITY (EVEN)	ADDR PARITY P _T P _A P _B			TYPE OF OP (EVEN)									
5	CHECK CONDITION (EVEN) MARK PARITY CHK ADDR PARITY CHK DATA PARITY CHK SP PARITY CHK KEY					NORMAL OP CHK	CHECK CONDITIONS (COMMON) CCR TAG PARITY CHK TAG CMPR IOCE SBI CHK				OBS		OTC THERMAL	DOUBLE CYCLE T/O	MULTI ACCEPT CHK	CCR PARITY 0-7 8-15		SESDR PARITY (EVEN) 0-7 8-15 16-23 24-31 32-39 40-47 48-55 56-63										CHECKS (EVEN AND COMMON), SESDR (EVEN)						
6	SESDR (ODD)																																SESDR (ODD)	
7	SESDR (ODD)																																SESDR (ODD)	
8	MARKS (ODD) 0-7 8-15 16-23 24-31 32-39 40-47 48-55 56-63								TAG 1 2 3 4				HI/LO 5	SESAR 1-14 (ODD) 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23														IOCE SAB 24 25 26			MARKS, SESAR (ODD)			
9					REQUEST (ODD) 1 2 3 4				IOCE 1 2 3			RESPONSE (ODD) 1 2 3 4 1 2 3									REQUEST AND RESPONSE LATCHES (ODD)													
10	SPIKR P 0-4 0 1 2 3 4					SPOKR P 0-4 0 1 2 3 4						OPERATION (ODD) T & S STORE FETCH			SUPP LOG CHK				IOCE SAB PARITY P ₀ P ₁ P ₂			MARK PARITY (ODD)	ADDR PARITY P _T P _A P _B			SPIKR, SPOKR, TYPE OF OP (ODD)								
11	CHECK CONDITION (ODD) MARK PARITY CHK ADDR PARITY CHK DATA PARITY CHK SP PARITY CHK KEY					NORMAL OP CHK	SPAR P X Y 2 3 4 5 6 7																	SESDR PARITY (ODD) 0-7 8-15 16-23 24-31 32-39 40-47 48-55 56-63										CHECKS (ODD), SPAR, SESDR (ODD)

Figure 12-5 SE Logword Format

JUNE 1, 1971

WORD NO.	BIT POSITIONS IN MAIN STORAGE																															COMMENTS				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		31			
0	DESDR (EVEN)																															DESDR (EVEN)				
1	DESDR (EVEN)																															DESDR (EVEN)				
2	MARKS (EVEN)							DESDR PARITY (EVEN)							DESDR 1-14 (EVEN)																	MARKS, DESDR, DESAR (EVEN)				
3	OPERATION (EVEN)				CHECK CONDITIONS (EVEN)				SPIKR							SPOKR				OPERATION		CHECK CONDITION (COMMON)				OPERATION, CHECKS (EVEN AND COMMON), SPIKR, SPOKR										
	MARK PARITY	T & S	STORE	FETCH	DATA CHK	ADDR CHK	MARK PARITY CHK	CANCEL LATCH	P 0-4	0	1	2	3	4	P 0-4	0	1	2	3	4	ISK	SSK	SP ADDR CHK	SP KEY MIS-MATCH	OBS	OTC										
4	DESDR (ODD)																															DESDR (ODD)				
5	DESDR (ODD)																															DESDR (ODD)				
6	MARKS (ODD)							DESDR PARITY (ODD)							DESDR 1-14 (ODD)																	MARKS, DESDR, DESAR (ODD)				
7	OPERATION (ODD)				CHECK CONDITIONS (ODD)				SPIKR							SPOKR				OPERATION		CHECK CONDITION (COMMON)				OPERATION, CHECKS (ODD)										
8	CCR																															CCR				
	P ₀	1A	1B	2B	2C	3C	3D	4D	4A	P ₁	5A	5B	6B	6C	7C	7D	8D	8A	P ₄	S ₀	S ₁	1	2	3	4	IDES	P ₇	1	2	3	4	CCR INVALID				
9	CE CHECK				CVG OPERATION												ADDRESS REG				OPERATION LATCHES, DG ADDR REG ID															
	1	2	3	4	1	2	3	A	4	5	6	7	8	9	B	10	11	12	13	14	15	C	16	17	18	19	20	21	D	22	23	24	A	B	C	D
10	DG REGISTER CHECK								CHECK CONDITIONS (COMMON)							DG ADDRESS REGISTER														CHECKS, DG ADDR REG						
	1	2	3	4	5	6	7	8	TAG PARITY CHK	TAG	CCR PARITY CHK	MULTI ACCEPT	LS PARITY CHK	NORMAL OP	REFRESH PARITY CHECK	P 1-7	1	2	3	4	5	6	7	P 8-14	8	9	10	11	12	13	14					
11	TAG					DE ID				HALFWORD COUNTER				DG DATA REGISTER																DG DATA REG						
	P 1-5	1	2	3	4	5	1	2	3	4	8	4	2	1	P 0-7	0	1	2	3	4	5	6	7	P 8-15	8	9	10	11	12	13	14	15				

Figure 12-6 DE Logword Format

JUNE 1, 1971

OBS - On Battery Signal
OTC - over Temperature Condition

Words 4-7 - Same as words 0-3 (odd side of storage).

Word 8 - This word contains the content of the DE Configuration Control Register (CCR), and the CCR Invalid bit (invalid DG communications field).

Word 9 - This word contains the content of the response latches and the DG Address Register identification bits. The CE portion of the response latches and the address register identification bits indicate which element (CE or DG) was being serviced when the malfunction occurred. The CVG field further identifies which CVG's were in the process of a data transfer when the malfunction occurred. A Storage check condition, DG Register Check, or local storage check condition may be related to a CVG via interrogation of the Address Register identification, CCR DG communication bits, and the CVG field of the response latches.

Word 10 - This word contains the DG Register Checks, common DE check conditions, and the contents of the DG Address Register.

The common DE check conditions are as follows:

Tag Parity Chk	- Improper parity on Tag (Address 1-4)
Tag	- Tag mismatch; both Tag received and DE ID are contained in Logword 11.
CCR Parity Chk	- Improper parity on CCR
Local Storage Parity Chk	- Improper parity on DG access address being stored into Local Storage.
Normal Operation	- Inconsistency between Normal Operation line and operations requested. (CE only)
Multi-Accept	- Multi-accept condition, e.g., two elements accepted for same storage cycle.

The DG Address Register (one of four available) contains the address last accessed by a DG if the malfunction detected was either a LS Parity Check, Address Parity Check or DG Register Check during a DG access; otherwise, this field is all ones. The specific register may be determined by interrogating both the response latches and the DG communications field of the CCR.

Word 11 - This word contains the Tag, DE ID, Halfword Counter, and the DG Data Register.

The tag field indicates the tag last received from a CE and is significant only on a tag mismatch or tag parity check.

The DE ID field indicates the DE "number" (in hexadecimal); 6-A representing DG 1-5, respectively.

The Halfword Counter indicates which halfword is contained in the DG Data Register and is significant only on a DG register check condition. A count of 8-F represents halfwords 1-8, respectively.

The DG Data Register contains the content of the particular DG Data Register which had a DG Register Check (Logword 10) and is significant only when such a check condition exists; being all zeroes otherwise. This field represents the contents of one of eight DG Data Registers, the specific register being specified by the DG Register Check field.

JUNE 1, 1971

CHAPTER 13: INSTRUCTION TIMING

The actual time for execution of a particular instruction depends upon a great many factors. The times stated are average values, based on the assumptions stated below. In general, the effects of data format and indexing are stated explicitly, while variations due to operand values and instruction location are included in the averages. Unless stated otherwise, the following assumptions have been made in computing average instruction times.

- 1) Instructions start on even and odd halfwords with equal probability.
- 2) In data fields (except decimal) each bit position has equal probability for values of one or zero, and is independent of other bit positions.
- 3) Decimal data can contain digit values 0-9 in each digit position with equal probability. When either the multiplier of a "multiply decimal" instruction or a divisor of a "divide decimal" instruction contains the digits 5 or 6 in each position, a slower than average instruction time (worst case) will result. An incremental decrease in instruction time is realized as the digits descend from 5 to 0 or ascend from 6 to 9.
- 4) Positive and negative operands are equally probable.
- 5) No instruction refetch occurs. An instruction refetch occurs if destination address equals Instruction Counter address ± 16 .
- 6) Interruptions are not reflected in these timings.
- 7) The time required for indexing by a base register is included in the times given. For those instructions (RX format) that can be double indexed (indicated by one or two asterisks in the following instruction tables) an additional 0.15 usec (one asterisk) or 0.20 usec (two asterisks) must be added to the times given in the table.
- 8) No delays in accessing shared storage are caused by references to the same storage unit by other CES or IOCES.
- 9) No interval timer updating occurs.
- 10) Addresses for unsuccessful branches are valid, but are unprotected storage locations.
- 11) All timings provided include both decoding and execution times for instructions.
- 12) Cable delays experienced within the CE/SE or CE/DE data paths which affect CE instruction times are dependent on both instruction format and relative CE/SE or CE/DE physical relationship. As an aid in determining the average cable delay for a given instruction executed in a particular CE/SE or CE/DE combination, Figure 13-1 and Table 13-I are presented. Figure 13-1 provides a sample floor plan for CE/SE and CE/DE physical relationships and their designated identifications. Using the indicated CE, SE and DE identifications, Table 13-I may then be referenced for that particular CE/SE or CE/DE combination under

JUNE 1, 1971

the appropriate instruction format group. The cable delay time thus referenced should then be added to the base instruction times presented in this chapter.

For example:

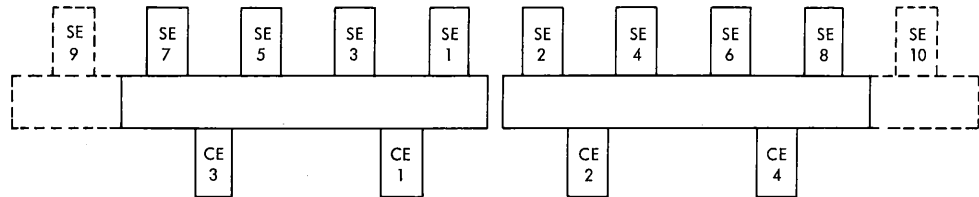
Calculating the average effective instruction time for an AND (SS format) with the overlapping fields and a one byte operand (N=1), we have:

$$\begin{aligned} \text{Base time} &= 3.80 + 0.2N \\ &= 3.80 + 0.2(1) \\ &= 4.0 \text{ usec} \end{aligned}$$

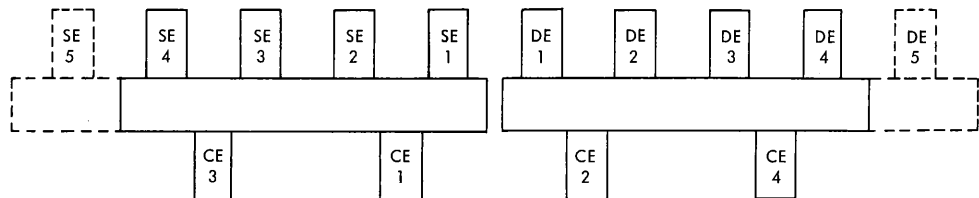
Assuming the instruction is executed by CE3 and both the instruction and operands (fetch two, store one) are in SE3, we have:

$$\begin{aligned} \text{Time} &= (\text{base time}) + (\text{I fetch cable delay}) + (\text{number of} \\ &\quad \text{operand accesses})(\text{operand cable delay}) \\ &= 4.0 + (0.024) + 3(0.035) \\ &= 4.129 \end{aligned}$$

- 13) Those instructions which fetch and store data from or to a display element will experience an average delay of 400 nanoseconds per access due to the synchronous operation of the element. Additional delay due to interference is dependent on access loads. Each lost cycle, due to interference, will add 800 nanoseconds to the execution time.



(a) 9020D System



(b) 9020E System

Figure 13-1 Physical Placement - Cable Delay Relationship

TABLE 13-I INSTRUCTION TIME AVERAGE CABLE DELAYS

9020D		SE 1	SE 2	SE 3	SE 4	SE 5	SE 6	SE 7	SE 8	SE 9	SE 10
RR per Instruction	CE1	0	25	0	34	9	42	18	51	26	60
	CE2	25	0	34	0	42	9	51	18	60	26
	CE3	18	42	9	51	0	60	0	69	9	78
	CE4	42	18	51	9	60	0	69	0	78	9
RX,RS,SI per Instruction	CE1	0	50	0	68	18	85	35	103	53	120
	CE2	50	0	68	0	85	18	103	35	120	53
	CE3	35	85	18	103	0	120	0	138	18	155
	CE4	85	35	103	18	120	0	138	0	155	18
SS per Instruction	CE1	0	75	0	101	26	128	53	154	79	180
	CE2	75	0	101	0	128	26	154	53	180	79
	CE3	53	128	26	154	0	180	0	206	26	233
	CE4	128	53	154	26	180	0	206	0	233	26
Each Operand Access	CE1	0	100	0	135	35	170	70	205	105	240
	CE2	100	0	135	0	170	35	205	70	240	105
	CE3	70	170	35	205	0	240	0	275	35	310
	CE4	170	70	205	35	240	0	275	0	310	35
9020E		SE 1	DE 1	SE 2	DE 2	SE 3	DE 3	SE 4	DE 4	SE 5	DE 5

Note:

- a) All times presented in this table are expressed in nanoseconds.
b) This table is presented for explanatory purposes and presents worst case values.

JUNE 1, 1971

13.1 FIXED-POINT ARITHMETIC INSTRUCTIONS

TABLE 13-II FIXED-POINT ARITHMETIC INSTRUCTIONS

INSTRUCTION	FORMAT	MNEMONIC	TIME (usec)
LOAD	RR	LR	0.69
LOAD ¹	RX	L	1.43
LOAD HALFWORD ¹	RX	LH	1.63
LOAD AND TEST	RR	LTR	0.69
LOAD COMPLEMENT	RR	LCR	0.69
LOAD POSITIVE	RR	LPR	0.99
LOAD NEGATIVE	RR	LNR	0.99
LOAD MULTIPLE	RS	LM	
CASE 1			1.63
CASE 2			0.88+0.38GR
CASE 3			1.28+0.38GR
CASE 4			1.08+0.38GR
ADD	RR	AR	0.69
ADD ¹	RX	A	1.63
ADD HALFWORD ¹	RX	AH	2.03
ADD LOGICAL	RR	ALR	0.69
ADD LOGICAL ¹	RX	AL	1.63
SUBTRACT	RR	SR	0.69
SUBTRACT ¹	RX	S	1.63
SUBTRACT HALFWORD ¹	RX	SH	2.03
SUBTRACT LOGICAL	RR	SLR	0.69
SUBTRACT LOGICAL ¹	RX	SL	1.63
COMPARE	RR	CR	0.69
COMPARE ¹	RX	C	1.63
COMPARE HALFWORD ¹	RX	CH	2.03
MULTIPLY	RR	MR	4.49
MULTIPLY ¹	RX	M	5.03
MPY HALFWORD ¹	RX	MH	5.23
DIVIDE	RR	DR	8.49+0.15G ₁
DIVIDE ¹	RX	D	8.93+0.15G ₁
STORE ¹	RX	ST	1.16
STORE HALFWORD ¹	RX	STH	1.96
STORE MULTIPLE	RS	STM	
CASE 1			1.56
CASE 2			0.61+0.28GR
CASE 3			1.41+0.28GR
CASE 4			1.01+0.28GR
CONVERT TO DECIMAL ¹	RX	CVD	8.88+0.40G ₃
CONVERT TO BINARY ¹	RX	CVB	7.83+0.20G ₃ + 0.20G ₁

¹ Add 0.15 usec when double indexed.

Where:
 GR = Number of general purpose registers loaded or stored.
 G₁ = 1 if an overflow interruption occurs (PSW bit 36=1) or fixed point divide interruption occurs; 0 otherwise.
 G₃ = 0 if operand to be converted is positive; 1 otherwise.

Case 1 is used if the number of registers is 2, and if the operand lies on double word boundaries.

Case 2 is used if the number of registers is greater than 2 and even, and if the operand lies on double word boundaries.

Case 3 is used if the number of registers is even, and the operand does not lie on double word boundaries.

Case 4 is used if the number of registers is odd.

13.2 SHIFT INSTRUCTIONS

Shifts are accomplished by a number of one-bit shifts followed by a number of four-bit shifts. The shift amount is stated in the instruction as a six-bit quantity. The high-order four bits (q4) of this quantity determine the number of four-bit shifts. The low-order two bits (Sx) determine the number of one-bit shifts.

TABLE 13-III SHIFT INSTRUCTIONS

INSTRUCTION	MNEMONIC	TIME (usec)
SHIFT LEFT SINGLE (logical)	SLL	$0.78+0.2q_4+0.2S_1$
SHIFT LEFT DOUBLE (logical)	SLLD	$0.98+0.4q_4+0.2S_3$
SHIFT RIGHT SINGLE (logical)	SRL	$0.98+0.2q_4+0.2S_2$
SHIFT RIGHT DOUBLE (logical)	SRDL	$0.98+0.4q_4+0.2S_4$
SHIFT LEFT SINGLE (algebraic)	SIA	$0.78+0.2q_4+0.2S_1$
SHIFT LEFT DOUBLE (algebraic)	SLDA	$0.98+0.4q_4+0.2S_3$
SHIFT RIGHT SINGLE (algebraic)	SRA	$0.98+0.2q_4+0.2S_2$
SHIFT RIGHT DOUBLE (algebraic)	SRDA	$0.98+0.4q_4+0.2S_4$

Where:

q4 = Quotient found by dividing by 4 the number of positions to be shifted

r4 = Remainder after dividing by 4 the number of positions to be shifted

S1 = 1 if r4=3 or if q4=0
 = 2 if r4=3 and q4=0
 = 0 otherwise

S2 = -1 if r4=0
 = 1 if r4=1 and q4=0
 = 0 otherwise

S3 = 0 if r4=0 and q4=0
 = 1 if r4=0 and q4=0

S3 = 3 if r4=1
 = 5 if r4=2 or 3

S4 = 0 if r4=0
 = 4 if q4=0 and r4=1,
 or if q4≠0 and r4=2
 = 3 if q4=0 and r4=2,
 or if q4≠0 and r4=3
 = 2 if q4=0 and r4=3
 = 5 if q4≠0 and r4=1

JUNE 1, 1971

13.3 FLOATING-POINT ARITHMETIC INSTRUCTIONS

The instruction times for floating point instructions depend on both the number of hexadecimal digits that are preshifted and post shifted and on the number of times the result is recomplemented. The floating point instruction times given below are a weighted average of these variables.

TABLE 13-IV FLOATING-POINT ARITHMETIC INSTRUCTIONS

INSTRUCTION	FORMAT	MNEMONIC	TIME (usec)
LOAD (short)	RR	LER	0.69
LOAD (short) ¹	RX	LE	1.43
LOAD (long)	RR	LDR	1.09
LOAD (long) ¹	RX	LD	1.63
LOAD POSITIVE (short)	RR	LPER	0.89
LOAD POSITIVE (long)	RR	LPDR	1.09
LOAD NEGATIVE (short)	RR	LNER	0.89
LOAD NEGATIVE (long)	RR	LNDR	1.09
LOAD AND TEST (short)	RR	LTER	0.89
LOAD AND TEST (long)	RR	LTDR	1.09
LOAD COMPLEMENT (short)	RR	LCER	0.89
LOAD COMPLEMENT (long)	RR	LCDR	1.09
STORE (short) ¹	RX	STE	1.16
STORE (long) ¹	RX	STD	1.16
ADD NORMALIZED (short)	RR	AER	1.72
ADD NORMALIZED (short) ¹	RX	AE	2.66
ADD NORMALIZED (long)	RR	ADR	1.76
ADD NORMALIZED (long) ¹	RX	AD	2.68
ADD UNNORMALIZED (short)	RR	AUR	1.68
ADD UNNORMALIZED (short) ¹	RX	AU	2.61
ADD UNNORMALIZED (long)	RR	AWR	1.69
ADD UNNORMALIZED (long) ¹	RX	AW	2.63
SUBTRACT NORMALIZED (short)	RR	SER	1.72
SUBTRACT NORMALIZED (short) ¹	RX	SE	2.66
SUBTRACT NORMALIZED (long)	RR	SDR	1.76
SUBTRACT NORMALIZED (long) ¹	RX	SD	2.68
SUBTRACT UNNORMALIZED (short)	RR	SUR	1.68
SUBTRACT UNNORMALIZED (short) ¹	RX	SU	2.61
SUBTRACT UNNORMALIZED (long)	RR	SWR	1.69
SUBTRACT UNNORMALIZED (long) ¹	RX	SW	2.63
COMPARE (short)	RR	CER	1.28
COMPARE (short) ¹	RX	CE	2.21
COMPARE (long)	RR	CDR	1.30
COMPARE (long) ¹	RX	CD	2.23
MULTIPLY (short)	RR	MER	4.09
MULTIPLY (short) ¹	RX	ME	4.63
MULTIPLY (long)	RR	MDR	7.29
MULTIPLY (long) ¹	RX	MD	7.83
DIVIDE (short)	RR	DER	6.59
DIVIDE (short) ¹	RX	DE	7.53
DIVIDE (long)	RR	DDR	13.39
DIVIDE (long) ¹	RX	DD	14.33
HALVE (short)	RR	HER	1.09
HALVE (long)	RR	HDR	1.29

¹ Add 0.15 usec when double indexed

13.4 LOGICAL INSTRUCTIONS

The instruction times stated here are approximations adequate for normal use. Although reasonably accurate in the average, individual cases may differ from these approximations by as much as 20 percent.

TABLE 13-V LOGICAL INSTRUCTIONS

INSTRUCTION	FORMAT	MNEMONIC	TIME (usec)
INSERT CHARACTER ¹	RX	IC	1.63
LOAD ADDRESS	RX	LA	1.13
MOVE	SS	MVC	* Ref. 13.7.7
MOVE	SI	MVI	1.56
MOVE NUMERIC	SS	MVN	* Ref. 13.7.8
MOVE ZONE	SS	MVZ	* Ref. 13.7.8
COMPARE LOGICAL	SS	CLC	* Ref. 13.7.12
COMPARE LOGICAL	SI	CLI	1.63
COMPARE LOGICAL	RR	CLR	0.69
COMPARE LOGICAL ¹	RX	CL	1.63
AND	SS	NC	* Ref. 13.7.8
AND	SI	NI	1.96
AND	RR	NR	1.29
AND ¹	RX	N	2.23
OR	SS	OC	* Ref. 13.7.8
OR	SI	OI	1.96
OR	RR	OR	1.29
OR ¹	RX	O	2.23
EXCLUSIVE OR	SS	XC	* Ref. 13.7.8
EXCLUSIVE OR	SI	XI	1.96
EXCLUSIVE OR	RR	XR	1.29
EXCLUSIVE OR ¹	RX	X	2.23
STORE CHARACTER ¹	RX	STC	1.56
TRANSLATE	SS	TR	* Ref. 13.7.9
TRANSLATE AND TEST	SS	TRT	* Ref. 13.7.10
EDIT	SS	ED	* Ref. 13.7.11
EDIT AND MARK	SS	EDMK	* Ref. 13.7.11
TEST UNDER MASK	SI	TM	1.83

¹ Add 0.15 usec where double indexed
 * Reference Detailed VFL Instruction Timing Section in this Chapter.

JUNE 1, 1971

13.5 DECIMAL ARITHMETIC INSTRUCTIONS

The instruction times stated here are approximations adequate for normal use. Although reasonably accurate in the average, individual cases may differ from these approximations by as much as 20 percent. All times are in microseconds.

TABLE 13-VI DECIMAL ARITHMETIC INSTRUCTIONS

INSTRUCTION	FORMAT	MNEMONIC	TIME (usec)
ZERO AND ADD	SS	ZAP	* Ref. 13.7.1
ADD DECIMAL	SS	AP	* Ref. 13.7.2
SUBTRACT DECIMAL	SS	SP	* Ref. 13.7.2
COMPARE DECIMAL	SS	CP	* Ref. 13.7.3
MULTIPLY DECIMAL	SS	MP	$3.89+3.4N^1-2.9N^2+1.0N^2(N^1-N^2)$
DIVIDE DECIMAL	SS	DP	$6.07+4.5N^1-4.7N^2+2.2N^2(N^1-N^2)+0.6T^6+1.35T^7Qs$
MOVE WITH OFFSET	SS	MVO	* Ref. 13.7.4
PACK	SS	PACK	* Ref. 13.7.5
UNPACK	SS	UNPK	* Ref. 13.7.6

Where:

N^1 = Total number of bytes in the first operand

N^2 = Total number of bytes in the second operand

Qs = Smaller of N^1-8 or N^1-N^2

T^6 = 0 if $N^2 \leq 4$

= 1 otherwise

T^7 = 0 if $N^1 \leq 8$

= 1 otherwise

* Reference Detailed VFL Instruction Timing Section in this Chapter.

13.6 BRANCHING AND STATUS-SWITCHING INSTRUCTIONS

TABLE 13-VII BRANCHING AND STATUS SWITCHING INSTRUCTIONS

INSTRUCTION	FORMAT	MNEMONIC	TIME (usec)
BRANCH ON CONDITION	RR	BCR	0.74+0.55F1
BRANCH ON CONDITION ²	RX	BC	0.88+0.45F1
BRANCH ON COUNT	RR	BCTR	1.02+0.32F1
BRANCH ON COUNT ²	RX	BCT	1.38
BRANCH AND LINK	RR	BALR	1.24
BRANCH AND LINK ²	RX	BAL	1.43
BRANCH ON INDEX HIGH	RS	BXH	1.68-0.05F1
BRANCH ON INDEX LOW EQUAL	RS	BXLE	1.68-0.05F1
EXECUTE ¹	RX	EX	
CASE 1			1.68+E
CASE 2			2.83+E+0.4T ²
SET SYSTEM MASK	SI	SSM	2.27* 2.94**
SET PROGRAM MASK	RR	SPM	0.89
SUPERVISOR CALL	RR	SVC	3.79
LOAD PSW	SI	LPSW	2.43
SET STORAGE KEY	RR	SSK	5.00
INSERT STORAGE KEY	RR	ISK	3.19
READ DIRECT	SI	RDD	2.93+ED
WRITE DIRECT	SI	WRD	2.43

¹ Add 0.15 usec when double indexed
² Add 0.20 usec when double indexed
* If 2 byte operand is within doubleword
** If 2 byte operand straddles doubleword boundary

Where:

E = Time for the subject instruction which is executed
ED = External delay
F1 = 1 if the branch operation is successful
= 0 otherwise
T² = 1 if R1 field is not zero
= 0 otherwise

Case 1 is used when subject instruction is successful branch
Case 2 is used when subject instruction is not a successful branch

13.7 DETAILED VARIABLE-FIELD-LENGTH INSTRUCTION TIMING

The microprogram sequences used to execute the variable field length instructions are quite complex, and because of the variety of relative field positions within words, the execution time for these instructions is dependent upon the addresses of the fields, as well as their lengths. The execution times of these instructions are also affected by such data conditions as overflow and recomplementation.

The following timing formulas are for VFL instructions (i.e., those instructions that contain an "L" field). All times are given in terms of word boundary crossovers and operand addresses. The term "word boundary" specifies the boundary between two physical words. A physical word is the amount of information fetched in a single storage cycle (this is 64 bits for the 7201-02 CE). Thus, the number of word boundary

JUNE 1, 1971

crossovers is one less than the number of doublewords spanned by the field.

The following abbreviations are used:

DB	Total number of bytes in the first operand (destination),
SB	Total number of bytes in the second operand (source).
NWBL1	Number of doubleword boundary crossovers for first operands.
NWBL2	Number of doubleword boundary crossovers for second operands.
NWBL1L2	Number of doubleword boundary crossovers for that part of the first operand which consists of SB bytes of high order zeros.
SDA	Address of first operand
SBA	Address of second operand.
W	Total number of doublewords in the first operand.
N	Total number of bytes in the first operand for those instructions with a single length field.
NWBB1	Number of doubleword boundary crossovers for that part of the first operand processed.
NWBB2	Number of doubleword boundary crossovers for that part of the second operand processed.
B	Total number of bytes of the first operand which are processed. It applies to instructions with a single length field.
WF	Number of doublewords in operand.
NB	Number of bytes in non-full doubleword.
K1	1 if $NWBL1 \geq 2$.

13.7.1 ZERO AND ADD

$$\text{Time (usec)} = a + bNWBL1 + cNWBL2$$

Values of coefficients:	a	4.50
	b	0.97
	c	0.97
If SB greater than DB add per SB		0.20
If SB less than DB add per DB		0.20
If fields overlap, add		0.72
If the result field is zero, add		1.35

13.7.2 ADD DECIMAL AND SUBTRACT DECIMAL

$$\text{Time (usec)} = a + bNWBL1 + cNWBL2$$

Values of coefficients:	a	4.50
	b	1.74
	c	0.97
If SB greater than DB, add per SB		0.20
If SB less than DB, add per DB		0.20
If the result field is zero, add		(1.17 + 0.2K1)
For ADD Decimal,		
If the result field is recomplemented (i.e., changes sign), add		2.2+2.0DB+1.75NWBL1
For Subtract Decimal, if the result field is recomplemented (i.e., changes sign), add		2.2+0.75NWBL1

JUNE 1, 1971

13.7.3 COMPARE DECIMAL

$$\text{Time (usec)} = a + b\text{NWBL1} + c\text{NWBL2}$$

Values of coefficients:	a	4.30
	b	1.17
	c	0.97
If SB greater than DB, add per SB		0.20
If SB less than DB, add per DB		0.20

13.7.4 MOVE WITH OFFSET

$$\text{Time (usec)} = a + b\text{NWBL1} + c\text{NWBL2}$$

Values of coefficients:	a	4.00
	b	0.77
	c	0.97
If $DB \leq SB$, add per DB		0.40
If DB greater than SB, add per SB plus per fill byte		0.40
If $SBA \geq SDA$ and the fields overlap, add per each overlapped byte		0.20
		0.40

13.7.5 PACK

$$\text{Time (usec)} = a + b\text{NWBL1} + c\text{NWBL2} + d\text{DB} + e\text{SB}$$

Values of coefficients:	a	3.88
	b	0.77
	c	0.97
	d	0.40
	e	0.10

13.7.6 UNPACK

$$\text{Time (usec)} = a + b\text{NWBL1} + c\text{NWBL2} + d\text{DB}$$

Values of coefficients:	a	3.80
	b	1.77
	c	1.77
	d	0.20
If SB less than $0.5(\text{DB}+1)$, add		0.20

JUNE 1, 1971

| 13.7.7 MOVE

| I. If first and second operand fields start and end on doubleword boundaries:

$$\text{Time (usec)} = a + bW$$

Values of coefficients: a 2.42
b 1.55

| II. If first and second operand fields start at corresponding word addresses within doublewords but do not lie on doubleword boundaries:

$$\text{Time (usec)} = a + bNWBL1 + cNWBL2$$

Values of coefficients: a 3.48
b 0.95
c 0.96

If $SBA \geq SDA$ and fields overlap, add 0.20 per each overlapped byte

| III. If first and second operand fields do not start at corresponding byte addresses within doublewords or if the total number of bytes in the first operand is less than 8:

$$\text{Time (usec)} = a + bNWBL1 + cNWBL2 + dN$$

Values of coefficients: a 3.80
b 0.77
c 0.97
d 0.20

If $SBA \geq SDA$ and fields overlap, add 0.20 per each overlapped byte

| IV. If both operands start on doubleword boundaries but end on byte boundaries:

$$\text{Time (usec)} = 4.0 + 1.54WF + 0.2NB$$

| 13.7.8 MOVE NUMERIC, MOVE ZONE, AND, OR, AND EXCLUSIVE OR

$$\text{Time (usec)} = a + bNWBL1 + cNWBL2 + dN$$

Values of coefficients: a 3.80
b 1.54
c 0.97
d 0.20

If $SBA \geq SDA$ and field overlaps, add 0.20 per each overlapped byte

| 13.7.9 TRANSLATE

$$\text{Time (usec)} = a + bNWBL1 + cN$$

Values of coefficients: a 2.87
b 1.55
c 1.77

JUNE 1, 1971

| 13.7.10 TRANSLATE AND TEST

Time (usec) = a + bNWBB1 + cN

Values of coefficients:	a	6.04
	b	0.97
	c	1.17

If no nonzero function byte is found, subtract		1.00
---------------------------------------------------	--	------

| 13.7.11 EDIT AND EDIT AND MARK

EDIT

Time (usec) = a + bNWBL1 + cNWBL2 + dN

Values of coefficients:	a	3.69
	b	1.75
	c	0.95
	d	0.43

EDIT AND MARK

Time (usec) = a + bNWBL1 + cNWBL2 + dN

Values of coefficients:	a	3.68
	b	1.75
	c	1.35
	d	0.43

For each time the mark address is stored, add		1.20
--------------------------------------------------	--	------

| 13.7.12 COMPARE LOGICAL

Time (usec) = a + bNWBB1 + cNWBB2 + dB

Values of coefficients:	a	3.80
	b	0.97
	c	0.97
	d	0.20

13.8 INPUT/OUTPUT INSTRUCTIONS

The execution time for the I/O instructions is dependent upon the current activity in the addressed IOCE, the current activity and type of the addressed channel, response time of the addressed control unit and the current activity, if any, of the addressed device. Therefore, it is not possible to delineate any explicit timings for the I/O instructions. The I/O instructions are listed below:

START I/O,
HALT I/O,
TEST I/O,
TEST CHANNEL, and
SET PCI.

JUNE 1, 1971

13.9 MULTIPLE COMPUTING ELEMENT INSTRUCTIONS

TABLE 13-VIII MULTIPLE COMPUTING ELEMENT INSTRUCTIONS

INSTRUCTION	FORMAT	MNEMONIC	TIME (usec)
DELAY	RR	DLY	$0.89+256n+0.2n$ (note1)
INSERT ADDRESS TRANSLATOR	RR	IATR	1.69
SET ADDRESS TRANSLATOR	RR	SATR	54.00
SET CONFIGURATION	RR	SCON	$13.8+0.8c$
STORE PSBA	SI	SPSB	1.76
TEST AND SET	SI	TS	2.03
LOAD PSBA	SI	LPSB	5.30
LOAD IDENTITY	RR	LI	0.89
MOVE WORD	SS	MVW	
Case 1			$6.0+1.37W$
Case 2			$6.0+1.77W$
Case 3			$6.0+0.57W$ (note2)
START I/O PROC	SI	SIOP	*

* See Section 13.8
 Note 1: Delay is increased by interval timer update cycles
 Note 2: Average case of storage availability

Where:

c = 1 if any IOCE is selected by the select mask
 = 0 otherwise
 W = Total number of doublewords in the first operand
 n = Delay count

Case 1 is used if first and second operand fields start on corresponding word or doubleword boundaries.
 Case 2 is used if first and second operand fields do not start at corresponding word addresses within doublewords.
 Case 3 is a special case for store only on word or doubleword overlap where the destination address is greater than the source address by one or two words.

13.10 DISPLAY INSTRUCTIONS

TABLE 13-IX DISPLAY INSTRUCTIONS

INSTRUCTION	FORMAT	MNEMONIC	TIME(usec)
LOAD CHAIN	RX	LC	
Case 1			2.3
Case 2			2.1
CONVERT AND SORT SYMBOLS	RR	CSS	*
CONVERT WEATHER LINES	RR	CVWL	**
REPACK SYMBOLS	RR	RPSB	***

JUNE 1, 1971

Case 1: Second operand on doubleword boundary
 Case 2: Second operand on odd word boundary

* CONVERT AND SORT SYMBOLS

Primary Radar and Single Symbol

	<u>Time (usec)</u>	<u>SE accesses</u>
Instruction Overhead	5.3	5 1/4
Header	2.3	1
Search (Geographic filter per symbol)	2.3	1/2
Process (Per Symbol including search)	8.9	1 1/2

Sample Threads

Total time for 2 primary accepts (header plus left and right data word)	20.1	4
Total time for 2 primary rejects (header plus left and right data word, words rejected on geographic filter)	6.9	2

Beacon

Instruction Overhead	4.6	4 1/4
Search (Fail Alt and type, SB off; per symbol)	2.7	2
Search (Fail Alt and type, SB on, IX off; per symbol)	3.6	3
Search (Fail Alt and type, SB on, IX on, Fail Geographic per symbol)	5.6	4
Process (Fail Alt and type, SB on, IX on;	12.2	5
Paging	1.3	-

Sample Threads

Total time for 1 beacon accept (Header plus data word)	12.2	5
Total time for 1 beacon reject (Header, Fail Alt and type, SB on, IX on, Fail Geographic)	5.6	4

** CONVERT WEATHER LINES

	<u>Time (usec)</u>	<u>SE accesses</u>	<u>DE accesses</u>
Instruction Overhead	2.8	1 1/4	-
Header	2.3	1	-
Search (per weather line)			
Pt. in SA 3	4.3	1	-
Pt. 2 in SA 3, Pt. 1 passes all filters	6.9	1	-
Pt. 1 and 2 fail Geographic filter	5.1	1	-
Process (per weather line)			
Pt. 1 ON; Pt. 2 ON	15.2	1	1
Pt. 1 ON; Pt. 2 OFF (one successful truncation)	16.0	1	1
Pt. 1 OFF; Pt. 2 ON (one successful truncation)	16.4	1	1
Truncation	1.4	-	-

JANUARY 31, 1973

TNL: GN31-3005

Paging	1.3	1	-
<u>Sample Threads</u>			
Total time for 1 weather line accept (Pt. 1 OFF; Pt. 2 ON. One truncation cycle, no header, no overhead)	16.4	1	1
Total time for 1 weather line reject (reject Pt. 1 on SA 3. No header, no overhead)	4.3	1	-
*** REPACK SYMBOLS			
	<u>Time (usec)</u>	<u>SE accesses</u>	<u>DE accesses</u>
Overhead (per instruction with one class/type)	24.0	variable	variable
Delete N2 per descriptor	1.5	1/4	-
Delete N1 per descriptor	1.5	1/4	-
Delete			
No Match (per WCT entry)	1.6	1/4	-
Match (per ODT descriptor)	2.5	1/2	-
Move			
History or Current Descriptor (per ODT descriptor)	1.9	1/2	-
Current Symbols (per symbol)	1.1	-	1
Modify			
No Match (per WCT entry)	1.2	1/4	-
Match (per ODT descriptor)	3.0	1 1/2	-
Move History or Modified Current Symbols (per symbol)	1.1	-	1
Insert			
Descriptor (per descriptor)	5.8	3	-
Symbol (per symbol)	1.0	1/2	1/2
Paging	1.3	1	-

NOTE: All CSS, CVWL, and RPSB times exclude storage interference, paging delay, synchronous delay, and external cable delay. These times must be adjusted for external cable delay due to the quantity of accesses indicated in the 'SE ACCESSES' and 'DE ACCESSES' columns (according to assumption 12 at the beginning of this chapter). Factors to account for storage interference and DE synchronous delay are given in assumption 13.

13.11 MAINTENANCE INSTRUCTIONS

The DIAGNOSE instruction has such varied use that no explicit time can be given.

13.12 IOCE-CE TO SE COMMUNICATIONS

Each SE has built-in priority circuitry to grant service to one element in cases of access request ties, and to distribute its

JUNE 1, 1971

availability where multiple like elements require continuous service. IOCEs have higher priority than CEs since some of the I-O devices with which they communicate may overrun if they are not serviced in time.

The 7251-09 SEs, which provide interleaved accesses, may be conceptually visualized as consisting of two separate storage elements with respect to priority circuitry; one set of circuitry granting accesses to odd doubleword addresses, the other set of circuitry granting service to even doubleword addresses. Each set of priority circuitry operates independently of the other with respect to granting access priority to a particular IOCE or CE. Consequently, the following discussion represents the characteristics of either an even or odd set of priority circuitry, the other set having identical characteristics.

As previously discussed, the IOCEs have higher priority than CEs due to the overrun potential of some I/O devices. The internal cycle time for the IOCE when accessing the SE is 2.5 usec while the actual time required in the SE for an IOCE storage cycle is 800 ns. Because of this difference in cycle time the SE honors the IOCE request when the storage cycle can be utilized by the IOCE. In the case of a fetch request from an IOCE, the SE honors the IOCE request immediately provided the storage is available and priority is granted. However, if the IOCE is storing data, the SE does not allow the IOCE request to content for priority until approximately 750 ns after the IOCE request. This allows other users to utilize the storage between the time of the IOCE request and the time the IOCE actually use the storage cycle.

For example, if three IOCEs and two CEs simultaneously request service from an SE and both IOCE's are fetching data, the sequence of storage cycles is IOCE-1, IOCE-2, CE-1, CE-2, CE-3. In the event that only the first IOCE is fetching data and IOCE-2 is storing data, the sequence is the same. However, if IOCE-2 is fetching and IOCE-1 is storing, the sequence is IOCE-2, IOCE-1, CE-1, CE-2, CE-3. When both IOCEs are storing, the sequence is CE-1, IOCE-1, IOCE-2, CE-2, CE-3. The latter two cases are examples of how the IOCE store operation is delayed, allowing another user to obtain a storage cycle during the delay period.

To see how the SE distributes its availability among like elements, consider the case where three CEs simultaneously require a storage cycle from an SE. CE-1 is serviced first, then CE-2 is serviced. Even if CE-1 again requests service while the cycle for CE-2 is in progress, the following cycle will still be granted to CE-3. Then CE-1 will be granted service again. If during the service of CE-2 an IOCE fetch request is received by the SE, the IOCE will gain priority ahead of CE-3 issuing IOCE priority over CEs. This system assures that no CEs or IOCEs will be stopped because other CEs or IOCEs are using the same SE.

13.13 CE TO DE COMMUNICATIONS

The IBM 7289-04 Display Element, unlike the SE, is a synchronous or "slotted" storage device. This characteristic, which is required to provide a synchronous environment to the display subsystem, manifests itself as a free running storage, independent of the occurrence of access requests.

A DE grants service to both CEs and display generators; IOCEs do not communicate with DEs. The display generators have a fetch capability only, while the CEs are allowed to either fetch or store. Each fetch request by a display generator is interpreted by the DE as a quadword (sixteen bytes) request and the DE accordingly allocates two interleaved storage accesses. This double storage access consists of an odd-even or even-odd pair of contiguous doublewords. CEs access data on a doubleword basis and are serviced accordingly.

JUNE 1, 1971

The allocation of storage accesses within a DE is cyclical in nature in order to assure each display address the availability of a synchronous storage access. Basically, the free running DE allocates eight doublewords (four quadword) cycles to the display subsystem and one doubleword access to the CEs. Each of the eight display subsystem storage cycles are assigned to a particular display address. The assignment of these display subsystem storage cycles are not affected by the existence of pending CE requests. However, should the display address assigned to a particular storage cycle not require an access at the initiation of its quadword "slot" the DE will make those storage cycles available to any CEs which have a pending request. Consequently, the DE access availability to CEs over and above the minimum of one of nine storage cycles is dependent on both the data loads and data rates of the display subsystem.

Priority determination for those storage cycles which are always allocated to CEs as well as those storage cycles which are made available to the CEs by default (i.e., the display address assigned to the particular slot did not request data), is identical to the SE. That is, the DE distributes its CE storage cycle availability in a manner which assures that no CE will be stopped because of other CEs which are using the same DE. As in the SE, the DE independently determines CE priority on an odd or even doubleword basis.

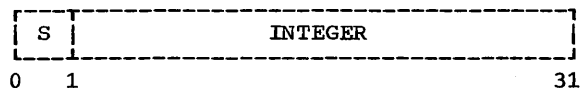
JUNE 1, 1971

APPENDIX A. FORMATS AND TABLES

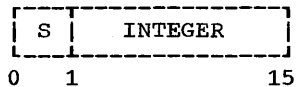
DATA FORMATS

Fixed-Point Numbers

Fullword Fixed-Point Number

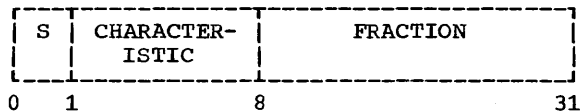


Halfword Fixed-Point Number

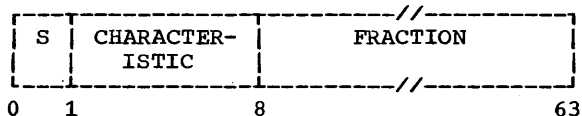


Floating-Point Numbers

Short Floating-Point Number

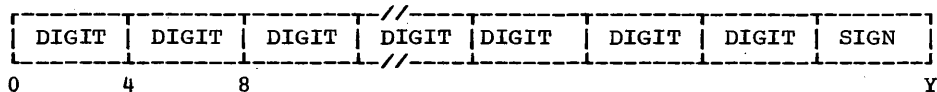


Long Floating-Point Number



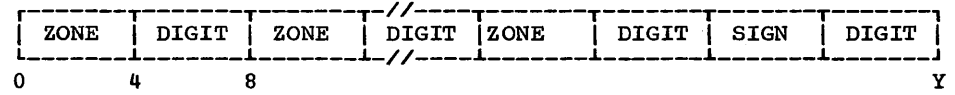
Decimal Numbers

Packed Decimal Number



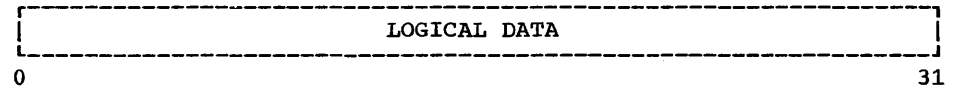
JUNE 1, 1971

Zoned Decimal Number

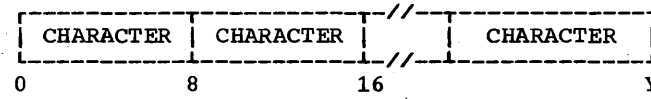


Logical Information

Fixed-Length Logical Information



Variable-Length Logical Information



EXTENDED BINARY-CODED-DECIMAL- INTERCHANGE CODE

Second Hexadecimal Digit Digit Punctures		00				01				10				11				Bit Positions 0,1						
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	Bit Positions 2,3						
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	First Hexadecimal Digit						
Zone Punctures		12	11	0	0	12	11	0	0	12	11	0	0	12	11	0	0							
Digit Punctures		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0000	0	8-1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	8-1	0					
0001	1	1	SOH	DC1	SOS					a	i			A	J		1	1	1					
0010	2	2	STX	DC2	FS	SYN				b	k	s		B	K	S	2	2	2					
0011	3	3	ETX	TM					c	l	t		C	L	T	3	3	3						
0100	4	4	PF	RES	BYP	PN				d	m	u		D	M	U	4	4	4					
0101	5	5	HT	NL	LF	RS				e	n	v		E	N	V	5	5	5					
0110	6	6	LC	BS	ETB	UC				f	o	w		F	O	W	6	6	6					
0111	7	7	DEL	IL	ESC	EOT				g	p	x		G	P	X	7	7	7					
1000	8	8		CAN												H	Q	Y	8	8	8			
1001	9	8-1		EM												I	R	Z	9	9	9			
1010	A	8-2	SMM	CC	SM	c	l	15	:													8-2	0	
1011	B	8-3	VT	CU1	CU2	CU3	\$,	#													8-3	0	
1100	C	8-4	FF	IFS		DC4	<	*	%	@													8-4	0
1101	D	8-5	CR	IGS	ENQ	NAK	()	-	'													8-5	0
1110	E	8-6	SO	IRS	ACK		+	;	>	=													8-6	0
1111	F	8-7	SI	IUS	BEL	SUB		~	?	"													8-7	0
Zone Punctures		12	11	0	0	12	11	0	0	12	11	0	0	12	11	0	0							
Digit Punctures		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							

Card Hole Patterns

- ① 12-0-9-8-1
- ② 12-11-9-8-1
- ③ 11-0-9-8-1
- ④ 12-11-0-9-8-1
- ⑤ No Punctures
- ⑥ 12
- ⑦ 11
- ⑧ 12-11-0
- ⑨ 12-0
- ⑩ 11-0
- ⑪ 0-8-2
- ⑫ 0
- ⑬ 0-1
- ⑭ 11-0-9-1
- ⑮ 12-11

Control Character Representations

ACK	Acknowledge	EOT	End of Transmission
BEL	Bell	ESC	Escape
BS	Backspace	ETB	End of Transmission Block
BYP	Bypass	ETX	End of Text
CAN	Cancel	FF	Form Feed
CC	Cursor Control	FS	Field Separator
CR	Carriage Return	HT	Horizontal Tab
CU1	Customer Use 1	IFS	Interchange File Separator
CU2	Customer Use 2	IGS	Interchange Group Separator
CU3	Customer Use 3	IL	Idle
DC1	Device Control 1	IRS	Interchange Record Separator
DC2	Device Control 2	IUS	Interchange Unit Separator
DC4	Device Control 4	LC	Lower Case
DEL	Delete	LF	Line Feed
DLE	Data Link Escape	NAK	Negative Acknowledge
DS	Digit Select	NL	New Line
EM	End of Medium	NUL	Null
ENQ	Enquiry		

Special Graphic Characters

¢	Cent Sign	-	Minus Sign, Hyphen
.	Period, Decimal Point	/	Slash
<	Less-than Sign	,	Comma
(Left Parenthesis	%	Percent
+	Plus Sign	_	Underscore
	Logical OR	>	Greater-than Sign
&	Ampersand	?	Question Mark
!	Exclamation Point	:	Colon
\$	Dollar Sign	#	Number Sign
*	Asterisk	@	At Sign
Space	Space	'	Prime, Apostrophe
;	Semicolon	=	Equal Sign
~	Logical NOT	"	Quotation Mark

JUNE 1, 1971

UNITED STATES OF AMERICA STANDARD CODE FOR INFORMATION INTERCHANGE

Hexadecimal Digit	00				01				10				11			
	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL	DLE			SP	0				@	P			\	p
0001	1	SOH	DC1			! (1)	1				A	Q			a	q
0010	2	STX	DC2			"	2				B	R			b	r
0011	3	ETX	DC3			#	3				C	S			c	s
0100	4	EOT	DC4			\$	4				D	T			d	t
0101	5	ENQ	NAK			%	5				E	U			e	u
0110	6	ACK	SYN			&	6				F	V			f	v
0111	7	BEL	ETB			'	7				G	W			g	w
1000	8	BS	CAN			(8				H	X			h	x
1001	9	HT	EM)	9				I	Y			i	y
1010	A	LF	SUB			*	:				J	Z			j	z
1011	B	VT	ESC			+	;				K	[k	
1100	C	FF	FS			,	<				L	\			l	
1101	D	CR	GS			-	=				M]			m	
1110	E	SO	RS			.	>				N	^ (2)			n	~
1111	F	SI	US			/	?				O	_			o	DEL

- ① If IBM equipment implementing USASCII-8 is provided, the graphic | (Logical OR) will be used instead of ! (Exclamation Point).
- ② If IBM equipment implementing USASCII-8 is provided, the graphic ¬ (Logical NOT) will be used instead of ^ (Circumflex).

NOTE: Current activities in committees under the auspices of the United States of America Standards Institute may result in changes to the characters and/or structure of the eight-bit representation of USASCII devised by the Institute. Such changes may cause the eight-bit representation of USASCII implemented in System/360 (USASCII-8) to be different from a future USA Standard. Since a difference of this nature may eventually lead to a modification of System/360, it is recommended that users avoid: (1) operation with PSW bit 12 set to 1, and (2) the use of any sign codes in decimal data other than those preferred for EBCDIC.

Control Character Representations

NUL	Null	DLE	Data Link Escape (CC)
SOH	Start of Heading (CC)	DC1	Device Control 1
STX	Start of Text (CC)	DC2	Device Control 2
ETX	End of Text (CC)	DC3	Device Control 3
EOT	End of Transmission (CC)	DC4	Device Control 4
ENQ	Enquiry (CC)	NAK	Negative Acknowledge (CC)
ACK	Acknowledge (CC)	SYN	Synchronous Idle (CC)
BEL	Bell	ETB	End of Transmission Block (CC)
BS	Backspace (FE)	CAN	Cancel
HT	Horizontal Tabulation (FE)	EM	End of Medium
LF	Line Feed (FE)	SUB	Substitute
VT	Vertical Tabulation (FE)	ESC	Escape
FF	Form Feed (FE)	FS	File Separator (IS)
CR	Carriage Return (FE)	GS	Group Separator (IS)
SO	Shift Out	RS	Record Separator (IS)
SI	Shift In	US	Unit Separator (IS)
(CC)	Communication Control	DEL	Delete
(FE)	Format Effector		
(IS)	Information Separator		

Special Graphic Characters

SP	Space	<	Less Than
	Exclamation Point	=	Equals
	Logical OR	>	Greater Than
"	Quotation Marks	?	Question Mark
#	Number Sign	@	Commercial At
\$	Dollar Sign	[Opening Bracket
%	Percent	\	Reverse Slant
&	Ampersand]	Closing Bracket
'	Apostrophe	^	Circumflex
(Opening Parenthesis	¬	Logical NOT
)	Closing Parenthesis	_	Underline
*	Asterisk	~	Grave Accent
+	Plus	{	Opening Brace
,	Comma		Vertical Line (This graphic is stylized to distinguish it from Logical OR)
-	Hyphen (Minus)	}	Closing Brace
.	Period (Decimal Point)	~	Tilde
/	Slant		
:	Colon		
;	Semicolon		

JUNE 1, 1971

HEXADECIMAL REPRESENTATION			
HEXADECIMAL CODE	PRINTED GRAPHIC	EBCDIC CODE *	USASCII-8 CODE †
0000	0	1111 0000	0101 0000
0001	1	1111 0001	0101 0001
0010	2	1111 0010	0101 0010
0011	3	1111 0011	0101 0011
0100	4	1111 0100	0101 0100
0101	5	1111 0101	0101 0101
0110	6	1111 0110	0101 0110
0111	7	1111 0111	0101 0111
1000	8	1111 1000	0101 1000
1001	9	1111 1001	0101 1001
1010	A	1100 0001	1010 0001
1011	B	1100 0010	1010 0010
1100	C	1100 0011	1010 0011
1101	D	1100 0100	1010 0100
1110	E	1100 0101	1010 0101
1111	F	1100 0110	1010 0110

LEGEND:

* Extended Binary-Coded-Decimal Interchange Code.

† United States of America Standard Code for Information Interchange

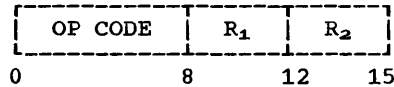
BIT POSITIONS

0	1	2	3	4	5	6	7	EBCDIC
8	7	6	5	4	3	2	1	USASCII-8

JUNE 1, 1971

INSTRUCTIONS BY FORMAT TYPE

RR Format



FIXED POINT

LOAD
 LOAD AND TEST
 LOAD COMPLEMENT
 LOAD POSITIVE
 LOAD NEGATIVE
 ADD
 ADD LOGICAL
 SUBTRACT
 SUBTRACT LOGICAL
 COMPARE
 MULTIPLY E
 DIVIDE E

FLOATING POINT

LOAD S/L
 LOAD AND TEST S/L
 LOAD COMPLEMENT S/L
 LOAD POSITIVE S/L
 LOAD NEGATIVE S/L
 ADD NORMALIZED S/L
 ADD UNNORMALIZED S/L
 SUBTRACT NORMALIZED S/L
 SUBTRACT UNNORMALIZED S/L
 COMPARE S/L
 HALVE S/L
 MULTIPLY S/L
 DIVIDE S/L

LOGICAL

COMPARE
 AND
 OR
 EXCLUSIVE OR
BRANCHING
 BRANCH ON CONDITION 1
 BRANCH AND LINK
 BRANCH ON COUNT

STATUS SWITCHING

SET PROGRAM MASK 2
 SUPERVISOR CALL 3
 SET STORAGE KEY
 INSERT STORAGE KEY

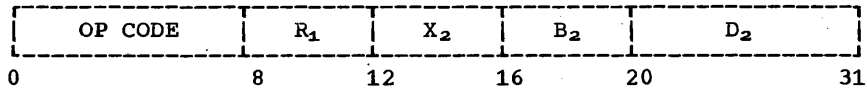
MULTI-COMPUTE ELEMENT

SET ADDR. TRANSLATOR
 INSERT ADDR. TRANSLATOR
 DELAY
 LOAD IDENTITY
 SET CONFIGURATION

DISPLAY

CONV. AND SORT SYMBOLS 6,7
 CONV. WEATHER LINES 7,8
 REPACK SYMBOLS 2,8

RX Format



FIXED POINT

LOAD H/F
 ADD H/F
 ADD LOGICAL
 SUBTRACT H/F
 SUBTRACT LOGICAL
 COMPARE H/F
 MULTIPLY H
 MULTIPLY F E
 DIVIDE F E
 CONVERT TO BINARY
 CONVERT TO DECIMAL
 STORE H/F

FLOATING POINT

LOAD S/L
 ADD NORMALIZED S/L
 ADD UNNORMALIZED S/L
 SUBTRACT NORMALIZED S/L
 SUBTRACT UNNORMALIZED S/L
 COMPARE S/L
 MULTIPLY S/L
 STORE S/L
 DIVIDE S/L

LOGICAL

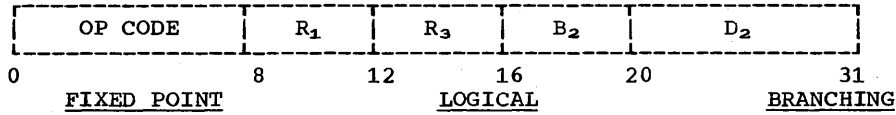
COMPARE
 LOAD ADDRESS
 INSERT CHARACTER
 STORE CHARACTER
 AND
 OR
 EXCLUSIVE OR
BRANCHING
 BRANCH ON CONDITION 1
 BRANCH AND LINK
 BRANCH ON COUNT
 EXECUTE

DISPLAY

LOAD CHAIN

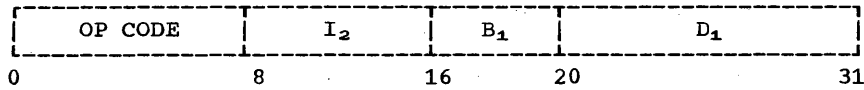
JUNE 1, 1971

RS Format



LOAD MULTIPLE	SHIFT LEFT SINGLE	2	BRANCH ON INDEX HIGH
STORE MULTIPLE	SHIFT RIGHT SINGLE	2	BRANCH ON INDEX LOW-EQ
SHIFT LEFT SINGLE	SHIFT LEFT DOUBLE	E2	
SHIFT RIGHT SINGLE	SHIFT RIGHT DOUBLE	E2	
SHIFT LEFT DOUBLE		E2	
SHIFT RIGHT DOUBLE		E2	

SI Format

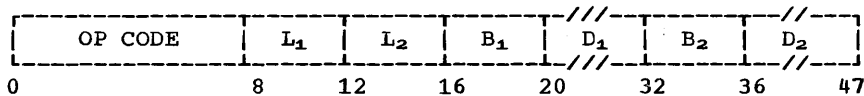


START I/O	4	MOVE	LOAD PSW	4
TEST I/O	4	COMPARE	SET SYSTEM MASK	4
HALT I/O	4	AND	WRITE DIRECT	
TEST CHANNEL	4	OR	READ DIRECT	
SET PCI	4	EXCLUSIVE OR	DIAGNOSE	
		TEST UNDER MASK		

MULTI-COMPUTE ELEMENT

START I/O PROCESSOR	
TEST AND SET	
LOAD PS BASE ADDR.	4
STORE PS BASE ADDR.	4

SS Format



PACK	MOVE	MOVE WORD 5
UNPACK	MOVE NUMERIC	5
MOVE WITH OFFSET	MOVE ZONE	5
ZERO AND ADD	COMPARE	5
ADD	AND	5
SUBTRACT	OR	5
COMPARE	EXCLUSIVE OR	5
MULTIPLY	TRANSLATE	5
DIVIDE	TRANSLATE AND TEST	5
	EDIT	5
	EDIT AND MARK	5

JUNE 1, 1971

Legend

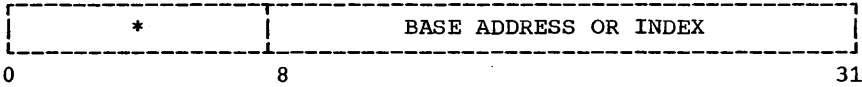
- | E R_1 must be even
- | F Fullword
- | H Halfword
- | L Long
- | S Short

- | 1 R_1 used as mask M_1
- | 2 R_2 or R_3 ignored
- | 3 R_1 and R_2 used as immediate information
- | 4 I_2 ignored
- | 5 L_1 and L_2 used as eight-bit L field.
- | 6 R_1 field when zero, specifies primary radar and/or
Single Symbol data; when two (binary 10), specifies beacon data
- | 7 R_2 field must contain a constant 9 (binary 1001)
- | 8 R_1 field ignored

JUNE 1, 1971

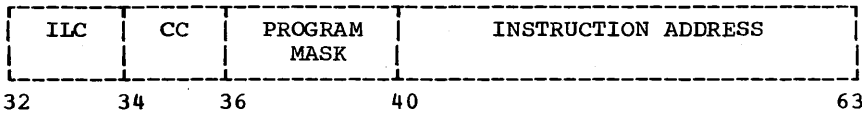
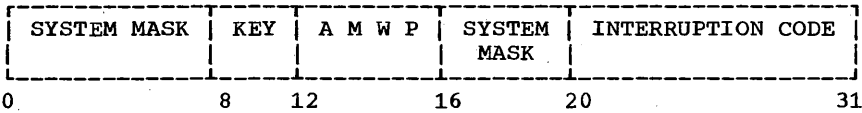
CONTROL WORD FORMATS

Base and Index Registers



- 0 - 7 Ignored
- 8 - 31 Base address or index

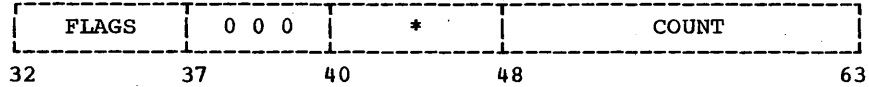
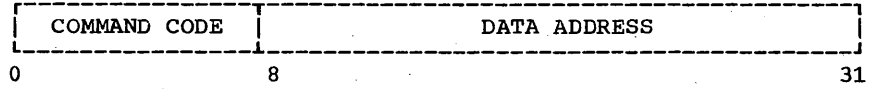
Program Status Word



- 0 - 7 System mask
 - 0 Multiplexor channel A mask
 - 1 Selector channel 1A mask
 - 2 Selector channel 2A mask
 - 3 Selector channel 3A mask
 - 4 Multiplexor channel B mask
 - 5 Selector channel 1B mask
 - 6 Selector channel 2B mask
 - 7 External mask
- 8 - 11 Protection key
 - 12 USASCII-8 mode (A)
 - 13 Machine check mask (M)
 - 14 Wait state (W)
 - 15 Problem state (P)
- 16 - 19 System mask
 - 16 Selector channel 3B mask
 - 17 Multiplexor channel C mask
 - 18 Selector channel 1C mask
 - 19 Selector channel 2C mask
- 20 - 31 Interruption code
- 32 - 33 Instruction length code (ILC)
- 34 - 35 Condition code (CC)
- 36 - 39 Program mask
 - 36 Fixed-point overflow mask
 - 37 Decimal overflow mask
 - 38 Exponent underflow mask
 - 39 Significance mask
- 40 - 63 Instruction address

JUNE 1, 1971

Channel Command Word



- 0 - 7 Command code
- 8 - 31 Data address
- 32 - 36 Command flags
 - 32 Chain data flag
 - 33 Chain command flag
 - 34 Suppress length indication flag
 - 35 Skip flag
 - 36 Program-controlled interruption flag
- 37 - 39 Zero
- 40 - 47 Ignored
- 48 - 63 Count

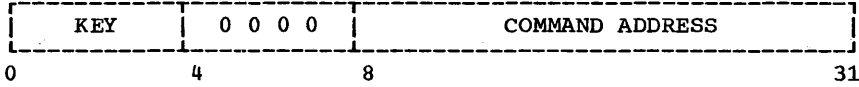
COMMAND CODE ASSIGNMENT			
NAMES	FLAGS	CODE	SECTION
Invalid		xxxx 0000	10.3.4
Write	CD CC SLI PCI	mmmm mm01	10.3.9.1
Read	CD CC SLI SKIP PCI	mmmm mm10	10.3.9.2
Read Backward	CD CC SLI SKIP PCI	mmmm 1100	10.3.9.3
Control	CD CC SLI PCI	mmmm mm11	10.3.9.4
Sense	CD CC SLI SKIP PCI	mmmm 0100	10.3.9.5
Transfer in Channel		xxxx 1000	10.3.9.6
Search	CD CC SLI PCI	mmmm mm01	10.3.9.7

Legend

- CD = Chain data
- CC = Chain command
- SLI = Suppress length indication
- SKIP = Skip
- PCI = Program-controlled interruption
- m = modifier bits
- x = ignored

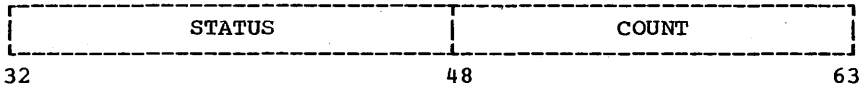
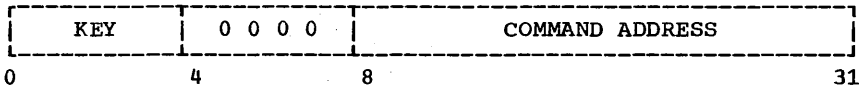
JUNE 1, 1971

Channel Address Word



- 0 - 3 Protection key
- 4 - 7 Zero
- 8 - 31 Command address

Channel Status Word



- 0 - 3 Protection key
- 4 - 7 Zero
- 8 - 31 Command address
- 32 - 47 Status
 - 32 Attention
 - 33 Status modifier
 - 34 Control unit end
 - 35 Busy
 - 36 Channel end
 - 37 Device end
 - 38 Unit check
 - 39 Unit exception
 - 40 Program-controlled interruption
 - 41 Incorrect length
 - 42 Program check
 - 43 Protection check
 - 44 Channel data check
 - 45 Channel control check
 - 46 Interface control check
 - 47 Chaining check
- 48 - 63 Count

JUNE 1, 1971

OPERATION CODES				
FORMAT	RR DISPLAY, BRANCHING AND STATUS SWITCHING	RR FIXED-POINT FULLWORD, AND LOGICAL	RR FLOATING-POINT LONG	RR FLOATING-POINT SHORT
CLASS	0000xxxx	0001xxxx	0010xxxx	0011xxxx
0000		LOAD POSITIVE	LOAD POSITIVE	LOAD POSITIVE
0001	SET CONFIG.	LOAD NEGATIVE	LOAD NEGATIVE	LOAD NEGATIVE
0010	CONV & SORT SYMB	LOAD AND TEST	LOAD AND TEST	LOAD AND TEST
0011	CONV WEATH LINES	LOAD COMPLEMENT	LOAD COMPLEMENT	LOAD COMPLEMENT
0100	SET PROGRAM MASK	AND	HALVE	HALVE
0101	BRANCH AND LINK	COMPARE LOGICAL		
0110	BRANCH ON COUNT	OR		
0111	BRANCH/CONDITION	EXCLUSIVE OR		
1000	SET KEY	LOAD	LOAD	LOAD
1001	INSERT KEY	COMPARE	COMPARE	COMPARE
1010	SUPERVISOR CALL	ADD	ADD N	ADD N
1011	DELAY	SUBTRACT	SUBTRACT N	SUBTRACT N
1100	LOAD IDENTITY	MULTIPLY	MULTIPLY	MULTIPLY
1101	SET ADR. XLR	DIVIDE	DIVIDE	DIVIDE
1110	INSERT ADR. XLR	ADD LOGICAL	ADD U	ADD U
1111	REPACK SYMBOLS	SUBTRACT LOGICAL	SUBTRACT U	SUBTRACT U
FORMAT	RX FIXED-POINT HALFWORD AND BRANCHING	RX FIXED-POINT FULLWORD AND LOGICAL	RX FLOATING-POINT LONG	RX FLOATING-POINT SHORT
CLASS	0100xxxx	0101xxxx	0110xxxx	0111xxxx
0000	STORE HALFWORD	STORE	STORE LONG	STORE SHORT
0001	LOAD ADDRESS			
0010	STORE CHARACTER	LOAD CHAIN		
0011	INSERT CHARACTER			
0100	EXECUTE	AND		
0101	BRANCH AND LINK	COMPARE LOGICAL		
0110	BRANCH ON COUNT	OR		
0111	BRANCH/CONDITION	EXCLUSIVE OR		
1000	LOAD HALFWORD	LOAD	LOAD LONG	LOAD SHORT
1001	COMPARE HALFWORD	COMPARE	COMPARE LONG	COMPARE SHORT
1010	ADD HALFWORD	ADD	ADD LONG N	ADD SHORT N
1011	SUBTRACT HALFWORD	SUBTRACT	SUBTRACT LONG N	SUBTRACT SHORT N
1100	MULTIPLY HALFWORD	MULTIPLY	MULTIPLY LONG	MULTIPLY SHORT
1101		DIVIDE	DIVIDE LONG	DIVIDE SHORT
1110	CONVERT-DECIMAL	ADD LOGICAL	ADD LONG U	ADD SHORT U
1111	CONVERT-BINARY	SUBTRACT LOGICAL	SUBTRACT LONG U	SUBTRACT SHORT U

FORMAT CLASS	RS, SI BRANCHING, STATUS SWITCHING AND SHIFTING	RS, SI FIXED-POINT, LOGICAL, AND INPUT/OUTPUT	SI STATUS SWITCHING	
XXXX	1000XXX	1001XXXX	1010XXXX	1011XXXX
0000	SET SYSTEM MASK	STORE MULTIPLE	STORE PS BASE	
0001		TEST UNDER MASK	LOAD PS BASE	
0010	LOAD PSW	MOVE I		
0011	DIAGNOSE	TEST AND SET		
0100	WRITE DIRECT	AND I		
0101	READ DIRECT	COMPARE LOGICAL I		
0110	BRANCH/HIGH	OR I		
0111	BRANCH/LOW-EQUAL	EXCLUSIVE OR I		
1000	SHIFT RIGHT SL	LOAD MULTIPLE		
1001	SHIFT LEFT SL			
1010	SHIFT RIGHT S	START I/O PROC.		
1011	SHIFT LEFT S	SET PCI		
1100	SHIFT RIGHT DL	START I/O		
1101	SHIFT LEFT DL	TEST I/O		
1110	SHIFT RIGHT D	HALT I/O		
1111	SHIFT LEFT D	TEST CHANNEL		
FORMAT CLASS		SS LOGICAL		SS DECIMAL
XXXX	1100XXXX	1101XXXX	1110XXXX	1111XXXX
0000		MOVE NUMERIC		MOVE W OFFSET
0001		MOVE C		PACK
0010		MOVE ZONE		UNPACK
0011		AND C		
0100		COMPARE LOGICAL C		
0101		OR C		
0110		EXCLUSIVE OR C		
0111				
1000		MOVE WORD		ZERO AND ADD
1001				COMPARE
1010				ADD
1011				SUBTRACT
1100		TRANSLATE		MULTIPLY
1101		TRANSLATE AND TEST		DIVIDE
1110		EDIT		
1111		EDIT AND MARK		
Legend				
N = Normalized		U = Unnormalized		
SL = Single logical		S = Single		
DL = Double logical		D = Double		
I = Immediate		C = Character		

JUNE 1, 1971

PREFERENTIAL-STORAGE ASSIGNMENT

A number of addresses are automatically generated by the CE for interruption or updating purposes. These are assigned to an area of storage called the preferential-storage area. In systems in which a CE shares all of its storage with one or more other CEs, the location of the preferential-storage area is determined by a base address in each CE, thus allowing different areas to be assigned for each CE. All addresses generated automatically by the CE are constructed using the current value of the preferential-storage base address.

The following table lists the main-storage locations in the preferential-storage area. The addresses shown are relative to the preferential-storage base address.

	ADDRESS	LENGTH	PURPOSE
0	0000 0000	double word	Initial program loading PSW
8	0000 1000	double word	Initial program loading CCW1
16	0001 0000	double word	Initial program loading CCW2
24	0001 1000	double word	External old PSW
32	0010 0000	double word	Supervisor call old PSW
40	0010 1000	double word	Program old PSW
48	0011 0000	double word	Machine old PSW
56	0011 1000	double word	Input/output old PSW
64	0100 0000	double word	Channel status word
72	0100 1000	word	Channel address word
76	0100 1100	word	Unused
80	0101 0000	word	Timer
84	0101 0100	word	Unused
88	0101 1000	double word	External new PSW
96	0110 0000	double word	Supervisor call new PSW
104	0110 1000	double word	Program new PSW
112	0111 0000	double word	Machine new PSW
120	0111 1000	double word	Input/output new PSW
128	1000 0000		Diagnostic logout area*

* The diagnostic logout area extends through byte location 511.

CONDITION CODE SETTING				
	0	1	2	3
Fixed-Point Arithmetic				
ADD H/F	zero	< zero	> zero	overflow
ADD LOGICAL	zero	not zero	zero	not zero
	no carry	no carry	carry	carry
COMPARE H/F	equal	low	high	--
LOAD AND TEST	zero	< zero	> zero	--
LOAD COMPLEMENT	zero	< zero	> zero	overflow
LOAD NEGATIVE	zero	< zero	--	--
LOAD POSITIVE	zero	--	> zero	overflow
SHIFT LEFT DOUBLE	zero	< zero	> zero	overflow
SHIFT LEFT SINGLE	zero	< zero	> zero	overflow
SHIFT RIGHT DOUBLE	zero	< zero	> zero	--
SHIFT RIGHT SINGLE	zero	< zero	> zero	--
SUBTRACT H/F	zero	< zero	> zero	overflow
SUBTRACT LOGICAL	--	not zero, no carry	zero, carry	not zero, carry
Decimal Arithmetic				
ADD DECIMAL	zero	< zero	> zero	overflow
COMPARE DECIMAL	equal	low	high	--
SUBTRACT DECIMAL	zero	< zero	> zero	overflow
ZERO AND ADD	zero	< zero	> zero	overflow
Floating-Point Arithmetic				
ADD NORMALIZED S/L	zero	< zero	> zero	--
ADD UNNORMALIZED S/L	zero	< zero	> zero	--
COMPARE S/L	equal	low	high	--
LOAD AND TEST S/L	zero	< zero	> zero	--
LOAD COMPLEMENT S/L	zero	< zero	> zero	--
LOAD NEGATIVE S/L	zero	< zero	--	--
LOAD POSITIVE S/L	zero	--	> zero	--
SUBTRACT NORMALIZED S/L	zero	< zero	> zero	--
SUBTRACT UNNORMALIZED S/L	zero	< zero	> zero	--
Logical Operations				
AND	zero	not zero	--	--
COMPARE LOGICAL	equal	low	high	--
EDIT	zero	< zero	> zero	--
EDIT AND MARK	zero	< zero	> zero	--
EXCLUSIVE OR	zero	not zero	--	--
OR	zero	not zero	--	--
TEST UNDER MASK	zero	mixed	--	one
TRANSLATE AND TEST	zero	incomplete	complete	--

JUNE 1, 1971

CONDITION CODE SETTING (continued)				
	0	1	2	3
Status-Switching				
WRITE DIRECT	oper	wait	redundant	timeout
Input-Output Operations				
HALT I/O	not working	halted	stopped	not oper
START I/O	available	CSW stored	busy	not oper
TEST CHANNEL	available	CSW ready	working	not oper
TEST I/O	available	CSW stored	working	not oper
SET PCI	available	CSW stored	set	not oper
Multi-Compute Element				
TEST AND SET	zero	one	--	--
SET ADDRESS TRANSLATOR	accepted	rejected	not zero	zero
SET CONFIGURATION	zero	--	> zero	--
START I/O PROCESSOR	proceeding	invalid	--	not oper
Display Operations				
LOAD CHAIN	zero	--	--	one
CONVERT AND SORT SYMBOLS	finished, no data stored	finished, data stored	boundary	inv form
REPACK SYMBOLS	finished	--	boundary	--

JUNE 1, 1971

Legend

accepted	Storage element assignment mask accepted
available	Unit and channel available
busy	Unit or channel busy
carry	A carry out of the sign position occurs
complete	Last result byte nonzero
CSW ready	Channel status word ready for test or interruption
CSW stored	Channel status word stored
equal	Operands compare equal
F	Fullword
H	Halfword
halted	Data transmission stopped. Unit in halt-reset mode
high	First operand compares high
L	Long precision
incomplete	Nonzero result byte; not last
low	First operand compares low
mixed	Selected bits are both zero and one
no carry	No carry occurs
not working	Unit or channel not working
not zero	Result is not all zero
one	Selected bits are one
overflow	Result overflows
rejected	Storage element assignment mask rejected because of configuration error
S	Short precision
set	PCI flag set
stopped	Data transmission stopped
working	Unit or channel working
zero	Result or selected bits are zero
invalid	Invalid PSW loaded into IOCE-processor
not oper	Unit or channel (or IOCE-processor) not operational
proceeding	Successful loading of PSW in IOCE-processor
redundant	Redundant operation
timeout	IOCE operation not completed; timeout occurred
wait	IOCE-processor start/stop operation completed; to/from Wait state
oper	IOCE-processor start/stop operation or logout completed
finished	Processing complete
boundary	Page boundary encountered in the sort bin
inv form	Invalid beacon format

Note

The condition also may be changed by LOAD PSW, SET SYSTEM MASK, DIAGNOSE, and by an interruption.

JUNE 1, 1971

INTERRUPTION ACTION					
INTERRUPTION SOURCE IDENTIFICATION	INTERRUPTION CODE PSW BITS 20-31		MASK BITS	ILC SET	INSTRUCTION EXECUTION
Input/Output (old PSW 56, new PSW 120, priority 4)					
Multiplexor channel A	0000	aaaaaaaa	0	x	completed
Selector channel 1A	0001	aaaaaaaa	1	x	completed
Selector channel 2A	0010	aaaaaaaa	2	x	completed
Selector channel 3A	0011	aaaaaaaa	3	x	completed
Multiplexor channel B	0100	aaaaaaaa	4	x	completed
Selector channel 1B	0101	aaaaaaaa	5	x	completed
Selector channel 2B	0110	aaaaaaaa	6	x	completed
Selector channel 3B	0111	aaaaaaaa	16	x	completed
Multiplexor channel C	1000	aaaaaaaa	17	x	completed
Selector channel 1C	1001	aaaaaaaa	18	x	completed
Selector channel 2C	1010	aaaaaaaa	19	x	completed
Program (old PSW 40, new PSW 104, priority 2)					
Operation	0000	00000001		1,2,3	suppressed
Privileged operation	0000	00000010		1,2	suppressed
Execute	0000	00000011		2	suppressed
Protection	0000	00000100		0,2,3	suppressed/ terminated
Addressing	0000	00000101		0,1,2,3	suppressed/ terminated
Specification	0000	00000110		1,2,3	suppressed/ terminated
Data	0000	00000111		2,3	terminated
Fixed-point overflow	0000	00001000	36	1,2	completed
Fixed-point divide	0000	00001001		1,2	suppressed/ completed
Decimal overflow	0000	00001010	37	3	completed
Decimal divide	0000	00001011		3	suppressed
Exponent overflow	0000	00001100		1,2	completed
Exponent underflow	0000	00001101	38	1,2	completed
Significance	0000	00001110	39	1,2	completed
Floating-point divide	0000	00001111		1,2	suppressed
IOCE-3 PSA lockout	0000	00010000		1,2,3	terminated/ completed
IOCE-2 PSA lockout	0000	00100000		1,2,3	terminated/ completed
IOCE-1 PSA lockout	0000	01000000		1,2,3	terminated/ completed
SE stopped	0000	10000000		1,2,3	suppressed/ terminated

JUNE 1, 1971

INTERRUPTION ACTION (continued)					
INTERRUPTION SOURCE IDENTIFICATION	INTERRUPTION CODE PSW BITS 20-31		MASK BITS	ILC SET	INSTRUCTION EXECUTION
Supervisor Call (old PSW 32, new PSW 96, priority 2)					
Instruction bits	0000	rrrrrrrr		1	completed
External (old PSW 24, new PSW 88, priority 3)					
DAR	xxxx	xxxxxxxx1	7	x	completed
PIR	xxxx	xxxxxxxx1x	7	x	completed
CE4 Write Direct	xxxx	xxxxx1xx	7	x	completed
CE4 Read Direct	xxxx	xxxx1xxx	7	x	completed
CE3 Write Direct	xxxx	xxx1xxxx	7	x	completed
CE3 Read Direct	xxxx	xx1xxxxx	7	x	completed
Interrupt switch	xxxx	x1xxxxxx	7	x	completed
Timer	xxxx	1xxxxxxx	7	x	completed
CE2 Write Direct	xxx1	xxxxxxxx	7	x	completed
CE2 Read Direct	xx1x	xxxxxxxx	7	x	completed
CE1 Write Direct	x1xx	xxxxxxxx	7	x	completed
CE1 Read Direct	1xxx	xxxxxxxx	7	x	completed
Machine Check (old PSW 48, new PSW 112, priority 1)					
CE malfunction	0000	00000000	13	x	terminated
IOCE-1 malfunction	0000	00000001	13	x	completed
IOCE-2 malfunction	0000	00000010	13	x	/terminated
IOCE-3 malfunction	0000	00000011	13	x	completed
Read Direct timeout	0000	00000100	13	x	/terminated
					terminated
<u>Legend</u>					
a	Device address bits				
r	Bits of R ₁ and R ₂ field of SUPERVISOR CALL				
x	Unpredictable				
PSA	Preferential Storage Area				
PIR	Processor Interrupt Register				
DAR	Diagnose Accessible Register				

JUNE 1, 1971

PROGRAM INTERRUPTIONS

1. Operation (OP)

The operation code is not assigned.

The operation is suppressed.

The instruction-length code is 1, 2, or 3.

2. Privileged Operation (M)

A privileged instruction is encountered in the problem state.

The operation is suppressed. The instruction-length code is 1 or 2.

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
DIAGNOSE		SI	M P A S	83	8.5.1.9
HALT I/O	HIO	SI	C M	9E	10.2.7.3
INSERT STORAGE KEY	ISK	RR	M A S	09	8.5.1.6
LOAD PS BASE ADDRESS	LPSB	SI	M P A S	A1	8.4.5.3
LOAD PSW	LPSW	SI	L M P A S	82	8.5.1.1
READ DIRECT	RDD	SI	M P A S	85	8.5.1.8
SET ADDRESS TRANSLATOR	SATR	RR	C M S	0D	8.4.5.7
SET CONFIGURATION	SCON	RR	C M S	01	8.4.5.4
SET PCI	SPCI	RS	C M	9B	10.2.7.5
SET STORAGE KEY	SSK	RR	M A S	08	8.5.1.5
SET SYSTEM MASK	SSM	SI	M P A	80	8.5.1.3
START I/O	SIO	SI	C M	9C	10.2.7.1
START I/O PROCESSOR	SIOP	SI	M S	9A	8.4.5.10
STORE PS BASE ADDRESS	SPSB	SI	M P A S	A0	8.4.5.5
TEST CHANNEL	TCH	SI	C M	9F	10.2.7.4
TEST I/O	TIO	SI	C M	9D	10.2.7.2
WRITE DIRECT	WRD	SI	C M P A S	84	8.5.1.7

3. Execute (EX)

The subject instruction of EXECUTE in another EXECUTE.

The operation is suppressed. The instruction-length code is 2.

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION	NOTE
EXECUTE	EX	RX	P A S EX	44	7.4.6	SPR

4. Protection (P)

The storage key of an accessed location does not match the protection key in the PSW. They are said to match if either is zero.

The operation is suppressed on a store-protection violation, except in the case of STORE MULTIPLE, READ DIRECT, and variable-length operations, which are terminated. Zeros may be stored in the destination location for variable-length operations. The operation is terminated on a fetch-protection violation.

The instruction-length code is 0, 2, or 3.

JUNE 1, 1971

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION	NOTE
ADD	A	RX	C P A S IF	5A	3.5.8	TRM
ADD DECIMAL	AP	SS	C P A D DF	FA	4.5.1	TRM
ADD HALFWORD	AH	RX	C P A S IF	4A	3.5.9	TRM
ADD LOGICAL	AL	RX	C P A S	5E	3.5.10	TRM
ADD NORMALIZED LONG	N AD	RX	C P A S U E LS	6A	5.6.6	TRM
ADD NORMALIZED SHORT	N AE	RX	C P A S U E LS	7A	5.6.6	TRM
ADD UNNORMALIZED LONG	AW	RX	C P A S E LS	6E	5.6.7	TRM
ADD UNNORMALIZED SHORT	AU	RX	C P A S E LS	7E	5.6.7	TRM
AND	NI	SI	C P A	94	6.4.5	SPR
AND	NC	SS	C P A	D4	6.4.5	TRM
AND	N	RX	C P A S	54	6.4.5	TRM
COMPARE	C	RX	C P A S	59	3.5.14	TRM
COMPARE DECIMAL	CP	SS	C P A D	F9	4.5.4	TRM
COMPARE HALFWORD	CH	RX	C P A S	49	3.5.15	TRM
COMPARE LOGICAL	CL	RX	C P A S	55	6.4.4	TRM
COMPARE LOGICAL	CLI	SI	C P A	95	6.4.4	TRM
COMPARE LOGICAL	CLC	SS	C P A	D5	6.4.4	TRM
COMPARE LONG	CD	RX	C P A S	69	5.6.10	TRM
COMPARE SHORT	CE	RX	C P A S	79	5.6.10	TRM
CONVERT AND SORT SYMBOLS	CSS	RR	C P A S	02	8.6.1.2	TRM
CONVERT TO BINARY	CVB	RX	P A S D IK	4F	3.5.19	TRM
CONVERT TO DECIMAL	CVD	RX	P A S	4E	3.5.20	SPR
CONVERT WEATHER LINES	CVWL	RR	P A S	03	8.6.1.3	TRM
DIAGNOSE		SI	M P A S	83	8.5.1.9	TRM
DIVIDE DECIMAL	DP	SS	P A S D DK	FD	4.5.6	TRM
DIVIDE	D	RX	P A S IK	5D	3.5.18	TRM
DIVIDE LONG	N DD	RX	P A S U E FK	6D	5.6.13	TRM
DIVIDE SHORT	N DE	RX	P A S U E FK	7D	5.6.13	TRM
EDIT	ED	SS	C P A D	DE	6.4.14	TRM
EDIT AND MARK	EDMK	SS	C P A D	DF	6.4.15	TRM
EXCLUSIVE OR	XI	SI	C P A	97	6.4.7	SPR
EXCLUSIVE OR	XC	SS	C P A	D7	6.4.7	TRM
EXCLUSIVE OR	X	RX	C P A S	57	6.4.7	TRM
EXECUTE	EX	RX	P A S EX	44	7.4.6	TRM
INSERT CHARACTER	IC	RX	P A	43	6.4.9	TRM
LOAD	L	RX	P A S	58	3.5.1	TRM
LOAD CHAIN	LC	RX	C P A S	52	8.6.1.1	TRM
LOAD HALFWORD	LH	RX	P A S	48	3.5.2	TRM
LOAD LONG	LD	RX	P A S	68	5.6.1	TRM
LOAD MULTIPLE	LM	RS	P A S	98	3.5.7	TRM
LOAD PS BASE ADDRESS	LPSB	SI	M P A S	A1	8.4.5.3	TRM
LOAD PSW	LPSW	SI	L M P A S	82	8.5.1.1	TRM
LOAD SHORT	LE	RX	P A S	78	5.6.1	TRM

JUNE 1, 1971

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION	NOTE
MOVE	MVI	SI	P A	92	6.4.1	SPR
MOVE	MVC	SS	P A	D2	6.4.1	TRM
MOVE NUMERICS	MVN	SS	P A	D1	6.4.2	TRM
MOVE WITH OFFSET	MVO	SS	P A	F1	4.5.9	TRM
MOVE WORD	MVW	SS	P A S	D8	8.4.5.9	TRM
MOVE ZONES	MVZ	SS	P A	D3	6.4.3	TRM
MULTIPLY DECIMAL	MP	SS	P A S D	FC	4.5.5	TRM
MULTIPLY	M	RX	P A S	5C	3.5.16	TRM
MULTIPLY HALFWORD	MH	RX	P A S	4C	3.5.17	TRM
MULTIPLY LONG	N MD	RX	P A S U E	6C	5.6.12	TRM
MULTIPLY SHORT	N ME	RX	P A S U E	7C	5.6.12	TRM
OR	OI	SI	C P A	96	6.4.6	SPR
OR	OC	SS	C P A	D6	6.4.6	TRM
OR	O	RX	C P A S	56	6.4.6	TRM
PACK	PACK	SS	P A	F2	4.5.7	TRM
READ DIRECT	RDD	SI	M P A S	85	8.5.1.8	TRM
REPACK SYMBOLS	RPSB	RR	C P A S	0F	8.6.1.4	TRM
SET SYSTEM MASK	SSM	SI	M P A	80	8.5.13	TRM
STORE	ST	RX	P A S	50	3.5.21	SPR
STORE CHARACTER	STC	RX	P A	42	6.4.10	SPR
STORE HALFWORD	STH	RX	P A S	40	3.5.22	SPR
STORE LONG	STD	RX	P A S	60	5.6.14	SPR
STORE MULTIPLE	STM	RS	P A S	90	3.5.23	TRM
STORE PS BASE ADDRESS	SPSB	SI	M P A S	A0	8.4.5.5	SPR
STORE SHORT	STE	RX	P A S	70	5.6.14	SPR
SUBTRACT	S	RX	C P A S IF	5B	3.5.11	TRM
SUBTRACT DECIMAL	SP	SS	C P A D DF	FB	4.5.2	TRM
SUBTRACT HALFWORD	SH	RX	C P A S IF	4B	3.5.12	TRM
SUBTRACT LOGICAL	SL	RX	C P A S	5F	3.5.13	TRM
SUBTRACT NORMALIZED LONG	N SD	RX	C P A S U E LS	6B	5.6.8	TRM
SUBTRACT NORMALIZED SHORT	N SE	RX	C P A S U E LS	7B	5.6.8	TRM
SUBTRACT UNNORMALIZED LONG	SW	RX	C P A S E LS	6F	5.6.9	TRM
SUBTRACT UNNORMALIZED LONG	SU	RX	C P A S E LS	7F	5.6.9	TRM
TEST AND SET	TS	SI	C P A	93	8.4.5.6	SPR
TEST UNDER MASK	TM	SI	C P A	91	6.4.8	TRM
TRANSLATE	TR	SS	P A	DC	6.4.12	TRM
TRANSLATE AND TEST	TRT	SS	C P A	DD	6.4.13	TRM
UNPACK	UNPK	SS	P A	F3	4.5.8	TRM
WRITE DIRECT	WRD	SI	C M P A S	84	8.5.17	TRM
ZERO AND ADD	ZAP	SS	C P A D DF	F8	4.5.3	TRM

Note

SPR = Operation suppressed
 TRM = Operation terminated

JUNE 1, 1971

5. Addressing (A)

An address specifies any part of data, instructions or control words outside the available storage for the particular installation, or outside the configured storage, or storage assigned by the storage address translator for the particular CE or IOCE. The operation is terminated or suppressed. Data in storage remain unchanged, except when designated by valid addresses. The instruction-length code normally is 1, 2, or 3; but may be 0, in the case of a data address.

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION	NOTE
ADD	A	RX	C P A S IF	5A	3.5.8	TRM
ADD DECIMAL	AP	SS	C P A D DF	FA	4.5.1	TRM
ADD HALFWORD	AH	RX	C P A S IF	4A	3.5.9	TRM
ADD LOGICAL	AL	RX	C P A S	5E	3.5.10	TRM
ADD NORMALIZED LONG	N AD	RX	C P A S U E LS	6A	5.6.6	TRM
ADD NORMALIZED SHORT	N AE	RX	C P A S U E LS	7A	5.6.6	TRM
ADD UNNORMALIZED LONG	AW	RX	C P A S E LS	6E	5.6.7	TRM
ADD UNNORMALIZED SHORT	AU	RX	C P A S E LS	7E	5.6.7	TRM
AND	N	RX	C P A S	54	6.4.5	TRM
AND	NI	SI	C P A	94	6.4.5	SPR
AND	NC	SS	C P A	D4	6.4.5	TRM
COMPARE	C	RX	C P A S	59	3.5.14	TRM
COMPARE DECIMAL	CP	SS	C P A D	F9	4.5.4	TRM
COMPARE HALFWORD	CH	RX	C P A S	49	3.5.15	TRM
COMPARE LOGICAL	CL	RX	C P A S	55	6.4.4	TRM
COMPARE LOGICAL	CLI	SI	C P A	95	6.4.4	TRM
COMPARE LOGICAL	CLC	SS	C P A	D5	6.4.4	TRM
COMPARE LONG	CD	RX	C P A S	69	5.6.10	TRM
COMPARE SHORT	CE	RX	C P A S	79	5.6.10	TRM
CONVERT AND SORT SYMBOLS	CSS	RR	C P A S	02	8.6.1.2	TRM
CONVERT TO BINARY	CVB	RX	P A S D IK	4F	3.5.19	TRM
CONVERT TO DECIMAL	CVD	RX	P A S	4E	3.5.20	SPR
CONVERT WEATHER LINES	CVWL	RR	P A S	03	8.6.1.3	TRM
DIAGNOSE		SI	M P A S	83	8.5.1.9	TRM
DIVIDE	D	RX	P A S IK	5D	3.5.18	TRM
DIVIDE DECIMAL	DP	SS	P A S D DK	FD	4.5.6	TRM
DIVIDE LONG	N DD	RX	P A S U E FK	6D	5.6.13	TRM
DIVIDE SHORT	N DE	RX	P A S U E FK	7D	5.6.13	TRM
EDIT	ED	SS	C P A D	DE	6.4.14	TRM
EDIT AND MARK	EDMK	SS	C P A D	DF	6.4.15	TRM
EXCLUSIVE OR	X	RX	C P A S	57	6.4.7	TRM
EXCLUSIVE OR	XI	SI	C P A	97	6.4.7	SPR
EXCLUSIVE OR	XC	SS	C P A	D7	6.4.7	TRM
EXECUTE	EX	RX	P A S EX	44	7.4.6	SPR
INSERT CHARACTER	IC	RX	P A	43	6.4.9	TRM
INSERT STORAGE KEY	ISK	RR	M P A S	09	8.5.1.6	SPR
LOAD	L	RX	P A S	58	3.5.1	TRM
LOAD CHAIN	LC	RX	C P A S	52	8.6.1.1	TRM
LOAD HALFWORD	LH	RX	P A S	48	3.5.2	TRM
LOAD LONG	LD	RX	P A S	68	5.6.1	TRM
LOAD MULTIPLE	LM	RS	P A S	98	3.5.7	TRM
LOAD PS BASE ADDRESS	LPSB	SI	M P A S	A1	8.4.5.3	SPR
LOAD PSW	LPSW	SI	L M P A S	82	8.5.1.1	TRM
LOAD SHORT	LE	RX	P A S	78	5.6.1	TRM

ADDRESSING INTERRUPTION (continued)						
NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION	NOTE
MOVE	MVI	SI	P A	92	6.4.1	SPR
MOVE	MVC	SS	P A	D2	6.4.1	TRM
MOVE NUMERICS	MVN	SS	P A	D1	6.4.2	TRM
MOVE WITH OFFSET	MVO	SS	P A	F1	4.5.9	TRM
MOVE WORD	MVW	SS	P A S	D8	8.4.5.9	TRM
MOVE ZONES	MVZ	SS	P A	D3	6.4.3	TRM
MULTIPLY	M	RX	P A S	5C	3.5.16	TRM
MULTIPLY DECIMAL	MP	SS	P A S D	FC	4.5.5	TRM
MULTIPLY HALFWORD	MH	RX	P A S	4C	3.5.17	TRM
MULTIPLY LONG	N MD	RX	P A S U E	6C	5.6.12	TRM
MULTIPLY SHORT	N ME	RX	P A S U E	7C	5.6.12	TRM
OR	O	RX	C P A S	56	6.4.6	TRM
OR	OI	SI	C P A	96	6.4.6	SPR
OR	OC	SS	C P A	D6	6.4.6	TRM
PACK	PACK	SS	P A	F2	4.5.7	TRM
READ DIRECT	RDD	SI	M P A S	85	8.5.1.8	TRM
REPACK SYMBOLS	RPSB	RR	C P A S	0F	8.6.1.4	TRM
SET STORAGE KEY	SSK	RR	M A S	08	8.5.1.5	SPR
SET SYSTEM MASK	SSM	SI	M P A	80	8.5.1.3	TRM
STORE	ST	RX	P A S	50	3.5.21	SPR
STORE CHARACTER	STC	RX	P A	42	6.4.10	SPR
STORE HALFWORD	STH	RX	P A S	40	3.5.22	SPR
STORE LONG	STD	RX	P A S	60	5.6.14	SPR
STORE MULTIPLE	STM	RS	P A S	90	3.5.23	TRM
STORE PS BASE ADDRESS	SPSB	SI	M P A S	A0	8.4.5.5	SPR
STORE SHORT	STE	RX	P A S	70	5.6.14	SPR
SUBTRACT	S	RX	C P A S IF	5B	3.5.11	TRM
SUBTRACT DECIMAL	SP	SS	C P A D DF	FB	4.5.2	TRM
SUBTRACT HALFWORD	SH	RX	C P A S IF	4B	3.5.12	TRM
SUBTRACT LOGICAL	SL	RX	C P A S	5F	3.5.13	TRM
SUBTRACT NORMALIZED LONG	N SD	RX	C P A S U E LS	6B	5.6.8	TRM
SUBTRACT NORMALIZED SHORT	N SE	RX	C P A S U E LS	7B	5.6.8	TRM
SUBTRACT UNNORMALIZED LONG	SW	RX	C P A S E LS	6F	5.6.9	TRM
SUBTRACT UNNORMALIZED SHORT	SU	RX	C P A S E LS	7F	5.6.9	TRM
TEST AND SET	TS	SI	C P A	93	8.4.5.6	SPR
TEST UNDER MASK	TM	SI	C P A	91	6.4.8	TRM
TRANSLATE	TR	SS	P A	DC	6.4.12	TRM
TRANSLATE AND TEST	TRT	SS	C P A	DD	6.4.13	TRM
UNPACK	UNPK	SS	P A	F3	4.5.8	TRM
WRITE DIRECT	WRD	SI	C M P A S	84	8.5.1.7	TRM
ZERO AND ADD	ZAP	SS	C P A D DF	F8	4.5.3	TRM

The addressing interruption can occur in normal sequential operation following branching, loading a PSW, interruption, or manual operation. Instruction execution is suppressed.

Note

SPR = Operation is suppressed
 TRM = Operation is terminated

JUNE 1, 1971

6. Specification (S)

A data, instruction or control-word address does not specify an integral boundary for the unit of information.

Or the R1 field of an instruction specifies an odd register address for a pair of general registers which contain a 64-bit operand.

Or a floating-point register address other than 0, 2, 4, or 6 is specified.

Or the multiplier or divisor in decimal arithmetic exceeds 15 digits and sign.

Or the first operand field is shorter than or equal to the second operand field in decimal multiplication or division.

Or the block address specified in SET STORAGE KEY or INSERT STORAGE KEY has the four low-order bits not all zero.

Or SET CONFIGURATION is attempted by a CE whose own scon bit is off in its configuration control register; or whose state bits are either set to One or Two.

Or configuration mask has its scon-field bits all zeros, or does not reference at least one CE available to the system.

Or a selection mask has an IOCE selection bit set when two or more CE communications bits are set in the configuration mask.

Or the operand address of LOAD PREFERENTIAL-STORAGE BASE ADDRESS, or STORE PREFERENTIAL-STORAGE BASE ADDRESS does not have the two low-order bits both zero.

Or bit positions 8-19 of the operand addressed by LOAD PREFERENTIAL-STORAGE BASE ADDRESS, interpreted as the twelve high-order bits of a storage address, do not specify a location within a storage element configured to communicate with the CE executing this instruction (must reference an SE ATR slot in a 9020E System).

Or WRITE DIRECT does not select an element; or selects more than one element, or an invalid operation is specified by bits 8-11 of the I₂ field.

Or READ DIRECT does not select a computing element; or selects more than one computing element.

Or SET ADDRESS TRANSLATOR is attempted by a CE whose own scon bit is off in its configuration control register; or whose state bits are either set to One or Two.

Or the storage element assignment mask does not reference a valid storage module available to the particular installation, or assigns an SE module to a position reserved for a DE module in the particular installation, or assigns a DE module to other than positions 6-10 in a 9020E System.

Or the maintenance control word for DIAGNOSE is not on an integral boundary for a double word, or specifies an operation whose execution is not permitted for a CE in state Three, Two or One.

Or START I/O PROCESSOR does not select an IOCE, or selects more than one IOCE, or the first operand address does not reference a double-word storage location or a key of F (hexadecimal) was given.

JUNE 1, 1971

Or in CSS, CVWL, and RPSB instruction, the data chain address contained in bytes 5-7 of the doubleword following a 512-byte page is not on a doubleword boundary.

Or the second operand address of the LOAD CHAIN instruction is not on a word boundary.

Or in the RPSB instruction, the NEXT OLD REFRESH MEMORY ADDR (ORMA) in GPR 4 or the NEXT NEW REFRESH MEMORY ADDR (NRMA) in GPR 5 is not initially on a doubleword boundary.

In all of these cases the operation is suppressed with the exception of SET ADDRESS TRANSLATOR, SET CONFIGURATION, CONVERT and SORT SYMBOLS, CONVERT WEATHER LINES, and REPACK SYMBOLS which are terminated.

The instruction-length code is 1, 2, or 3.

JUNE 1, 1971

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION	NOTE
ADD	A	RX	C P A S IF	5A	3.5.8	4
ADD HALFWORD	AH	RX	C P A S IF	4A	3.5.9	2
ADD LOGICAL	AL	RX	C P A S	5E	3.5.10	4
ADD NORMALIZED LONG	N ADR	RR	C S U E LS	2A	5.6.6	3
ADD NORMALIZED LONG	N AD	RX	C P A S U E LS	6A	5.6.6	3 8
ADD NORMALIZED SHORT	N AER	RR	C S U E LS	3A	5.6.6	3
ADD NORMALIZED SHORT	N AE	RX	C P A S U E LS	7A	5.6.6	3 4
ADD UNNORMALIZED LONG	AWR	RR	C S E LS	2E	5.6.7	3
ADD UNNORMALIZED LONG	AW	RX	C P A S E LS	6E	5.6.7	3 8
ADD UNNORMALIZED SHORT	AUR	RR	C S E LS	3E	5.6.7	3
ADD UNNORMALIZED SHORT	AU	RX	C P A S E LS	7E	5.6.7	3 4
AND	N	RX	C P A S	54	6.4.5	4
COMPARE	C	RX	C P A S	59	3.5.14	4
COMPARE HALFWORD	CH	RX	C P A S	49	3.5.15	2
COMPARE LOGICAL	CL	RX	C P A S	55	6.4.4	4
COMPARE LONG	CDR	RR	C S	29	5.6.10	3
COMPARE LONG	CD	RX	C P A S	69	5.6.10	3 8
COMPARE SHORT	CER	RR	C S	39	5.6.10	3
COMPARE SHORT	CE	RX	C P A S	79	5.6.10	3 4
CONVERT AND SORT SYMBOLS	CSS	RR	C P A S	02	8.6.1.2	9
CONVERT TO BINARY	CVB	RX	P A S D IK	4F	3.5.19	8
CONVERT TO DECIMAL	CVD	RX	P A S	4E	3.5.20	8
CONVERT WEATHER LINES	CVWL	RR	P A S	03	8.6.1.3	9
DIAGNOSE		SI	M P A S	83	8.5.1.9	
DIVIDE	DR	RR	S IK	1D	3.5.18	1
DIVIDE	D	RX	P A S IK	5D	3.5.18	1 4
DIVIDE DECIMAL	DP	SS	P A S D DK	FD	4.5.6	5
DIVIDE LONG	N DDR	RR	S U E FK	2D	5.6.13	3
DIVIDE LONG	N DD	RX	P A S U E FK	6D	5.6.13	3 8
DIVIDE SHORT	N DER	RR	S U E FK	3D	5.6.13	3
DIVIDE SHORT	N DE	RX	P A S U E FK	7D	5.6.13	3 4
EXCLUSIVE OR	X	RX	C P A S	57	6.4.7	4
EXECUTE	EX	RX	P A S EX	44	7.4.6	2
HALVE LONG	N HDR	RR	S U	24	5.6.11	3
HALVE SHORT	N HER	RR	S U	34	5.6.11	3
INSERT STORAGE KEY	ISK	RR	M A S	09	8.5.1.6	7
LOAD	L	RX	P A S	58	3.5.1	4
LOAD AND TEST LONG	LTDR	RR	C S	22	5.6.2	3
LOAD AND TEST SHORT	LTER	RR	C S	32	5.6.2	3
LOAD CHAIN	LC	RX	C P A S	52	8.6.1.1	8
LOAD COMPLEMENT LONG	LCDR	RR	C S	23	5.6.3	3
LOAD COMPLEMENT SHORT	LCER	RR	C S	33	5.6.3	3
LOAD HALFWORD	LH	RX	P A S	48	3.5.2	2
LOAD LONG	LDR	RR	S	28	5.6.1	3
LOAD LONG	LD	RX	P A S	68	5.6.1	3 8
LOAD MULTIPLE	LM	RS	P A S	98	3.5.7	4
LOAD NEGATIVE LONG	LNDR	RR	C S	21	5.6.5	3
LOAD NEGATIVE SHORT	LNER	RR	C S	31	5.6.5	3
LOAD POSITIVE LONG	LPDR	RR	C S	20	5.6.4	3
LOAD POSITIVE SHORT	LPER	RR	C S	30	5.6.4	3
LOAD PS BASE ADDRESS	LPSB	SI	M P A S	A1	8.4.5.3	4 7
LOAD PSW	LPSW	SI	L M P A S	82	8.5.1.1	8
LOAD SHORT	LER	RR	S	38	5.6.1	3
LOAD SHORT	LE	RX	P A S	78	5.6.1	3 4

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION	NOTE
MOVE WORD	MVW	SS	P A S	D8	8.4.5.9	4
MULTIPLY	MR	RR	P A S	1C	3.5.16	1
MULTIPLY	M	RX	P A S	5C	3.5.16	1 4
MULTIPLY DECIMAL	MP	SS	P A S D	FC	4.5.5	5
MULTIPLY HALFWORD	MH	RX	P A S	4C	3.5.17	2
MULTIPLY LONG	N MDR	RR	P A S U E	2C	5.6.12	3
MULTIPLY LONG	N MD	RX	P A S U E	6C	5.6.12	3 8
MULTIPLY SHORT	N MER	RR	P A S U E	3C	5.6.12	3
MULTIPLY SHORT	N ME	RX	P A S U E	7C	5.6.12	3 4
OR	O	RX	C P A S	56	6.4.6	4
READ DIRECT	RDD	SI	M P A S	85	8.5.1.8	6
REPACK SYMBOLS	RPSB	RR	C P A S	0F	8.6.1.4	9 10
SET ADDRESS TRANSLATOR	SATR	RR	C M S	0D	8.4.5.7	TRM
SET CONFIGURATION	SCON	RR	C M S	01	8.4.5.4	
SET PCI	SPCI	RS	C M	9B	10.2.7.5	
SET STORAGE KEY	SSK	RR	M A S	08	8.5.1.5	7
SHIFT LEFT DOUBLE	SLDA	RS	C S	8F	3.5.26	1
SHIFT LEFT DOUBLE LOGICAL	SLDL	RS	S	8D	6.4.18	1
SHIFT RIGHT DOUBLE	SRDA	RS	C S	8E	3.5.27	1
SHIFT RIGHT DOUBLE LOGICAL	SRDL	RS	S	8C	6.4.19	1
START I/O PROCESSOR	SIOP	SI	M S	9A	8.4.5.10	SPR
STORE	ST	RX	P A S	50	3.5.21	4
STORE HALFWORD	STH	RX	P A S	40	3.5.22	2
STORE LONG	STD	RX	P A S	60	5.6.14	3 8
STORE MULTIPLE	STM	RS	P A S	90	3.5.23	4
STORE PS BASE ADDRESS	SPSB	SI	M P A S	A0	8.4.5.5	2
STORE SHORT	STE	RX	P A S	70	5.6.14	3 4
SUBTRACT	S	RX	C P A S IF	5B	3.5.11	4
SUBTRACT HALFWORD	SH	RX	C P A S IF	4B	3.5.12	2
SUBTRACT LOGICAL	SL	RX	C P A S	5F	3.5.13	4
SUBTRACT NORMALIZED LONG	N SDR	RR	C S U E L S	2B	5.6.8	3
SUBTRACT NORMALIZED LONG	N SD	RX	C P A S U E L S	6B	5.6.8	3 8
SUBTRACT NORMALIZED SHORT	N SER	RR	C S U E L S	3B	5.6.8	3
SUBTRACT NORMALIZED SHORT	N SE	RX	C P A S U E L S	7B	5.6.8	3 4
SUBTRACT UNNORMALIZED LONG	SWR	RR	C S E L S	2F	5.6.9	3
SUBTRACT UNNORMALIZED LONG	SW	RX	C P A S E L S	6F	5.6.9	3 8
SUBTRACT UNNORMALIZED SHORT	SUR	RR	C S E L S	3F	5.6.9	3
SUBTRACT UNNORMALIZED SHORT	SU	RX	C P A S E L S	7F	5.6.9	3 4
WRITE DIRECT	WRD	SI	C M P A S	84	8.5.1.7	6

The specification interruption can occur in normal sequential operation following branching, LOAD PSW, interruption, or manual operation (Note 1).

The specification interruption can occur during an interruption (Note 6).

JUNE 1, 1971

Note

- 1 Even register specification
- 2 Two-byte unit of information specification
- 3 Floating-point register specification
- 4 Four-byte unit of information specification
- 5 Decimal multiplier or divisor size specification
- 6 Single element selection specification
- 7 Block address specification
- 8 Eight-byte unit of information specification
- 9 Chain address (during paging) is not in doubleword boundary
- 10 ORMA (GP4 4) or NRMA (GPR 5) is not initially on a doubleword boundary.
- SPR Operation suppressed
- TRM Operation terminated

7. Data (D)

The sign or digit codes of operands in decimal arithmetic, or editing operations, or CONVERT TO BINARY, are incorrect.

Or fields in decimal arithmetic overlap incorrectly.

Or the decimal multiplicand has too many high-order significant digits.

The operation is terminated in all three cases.

The instruction-length code is 2 or 3.

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION	NOTE
ADD DECIMAL	AP	SS	C P A D DF	FA	4.5.1	1
COMPARE DECIMAL	CP	SS	C P A D	F9	4.5.4	1
CONVERT TO BINARY	CVB	RX	P A S D IK	4F	3.5.19	
DIVIDE DECIMAL	DP	SS	P A S D DK	FD	4.5.6	1
EDIT	ED	SS	C P A D	DE	6.4.14	
EDIT AND MARK	EDMK	SS	C P A D	DF	6.4.15	
MULTIPLY DECIMAL	MP	SS	P A S D	FC	4.5.5	1 2
SUBTRACT DECIMAL	SP	SS	C P A D DF	FB	4.5.2	1
ZERO AND ADD	ZAP	SS	C P A D DF	F8	4.5.3	1

Note

- 1 Overlapping fields
- 2 Multiplicand length

8. Fixed-Point Overflow (IF)

A high-order carry occurs or high-order significant bits are lost in fixed-point addition, subtraction, shifting, or sign-control operations.

The operation is completed by ignoring the information placed outside the register. The interruption may be masked by PSW bit 36.

The instruction-length code is 1 or 2.

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
ADD	AR	RR	C	IF	1A 3.5.8
ADD	A	RX	C P A S	IF	5A 3.5.8
ADD HALFWORD	AH	RX	C P A S	IF	4A 3.5.9
LOAD COMPLEMENT	LCR	RR	C	IF	13 3.5.4
LOAD POSITIVE	LPR	RR	C	IF	10 3.5.5
SHIFT LEFT DOUBLE	SLDA	RS	C S	IF	8F 3.5.26
SHIFT LEFT SINGLE	SLA	RS	C	IF	8B 3.5.24
SUBTRACT	SR	RR	C	IF	1B 3.5.11
SUBTRACT	S	RX	C P A S	IF	5B 3.5.11
SUBTRACT HALFWORD	SH	RX	C P A S	IF	4B 3.5.12

9. Fixed-Point Divide (IK)

The quotient exceeds the register size in fixed-point division, including division by zero.

Or the result of CONVERT TO BINARY exceeds thirty one bits.

Division is suppressed. Conversion is completed by ignoring the information placed outside the register.

The instruction-length code is 1 or 2.

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
CONVERT TO BINARY	CVB	RX	P A S D	IK	4F 3.5.19
DIVIDE	DR	RR	S	IK	1D 3.5.18
DIVIDE	D	RX	P A S	IK	5D 3.5.18

JUNE 1, 1971

10. Decimal Overflow (DF)

The destination field is too small to contain the result field in decimal operations.

The operation is completed by ignoring the overflow information. The interruption may be masked by PSW bit 37.

The instruction-length code is 3.

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
ADD DECIMAL	AP	SS	C P A D DF	FA	4.5.1
SUBTRACT DECIMAL	SP	SS	C P A D DF	FB	4.5.2
ZERO AND ADD	ZAP	SS	C P A D DF	F8	4.5.3

11. Decimal Divide (DK)

The quotient exceeds the specified data field. The operation is suppressed.

The instruction-length code is 3.

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
DIVIDE DECIMAL	DP	SS	P A S D DK	FD	4.5.6

12. Exponent Overflow (E)

When the result characteristic in floating-point addition, subtraction, multiplication, or division exceeds 127 and the result is not zero, an exponent overflow exception is recognized. The operation is completed. The fraction is normalized, and the sign and fraction of the result remain correct. The result characteristic is made 128 smaller than the correct characteristic.

The instruction-length code is 1 or 2.

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
ADD NORMALIZED LONG	N ADR	RR	C S U E LS	2A	5.6.6
ADD NORMALIZED LONG	N AD	RX	C P A S U E LS	6A	5.6.6
ADD NORMALIZED SHORT	N AER	RR	C S U E LS	3A	5.6.6
ADD NORMALIZED SHORT	N AE	RX	C P A S U E LS	7A	5.6.6
ADD UNNORMALIZED LONG	AWR	RR	C S E LS	2E	5.6.7
ADD UNNORMALIZED LONG	AW	RX	C P A S E LS	6E	5.6.7
ADD UNNORMALIZED SHORT	AUR	RR	C S E LS	3E	5.6.7
ADD UNNORMALIZED SHORT	AU	RX	C P A S E LS	7E	5.6.7
DIVIDE LONG	N DDR	RR	S U E FK	2D	5.6.13
DIVIDE LONG	N DD	RX	P A S U E FK	6D	5.6.13
DIVIDE SHORT	N DER	RR	S U E FK	3D	5.6.13
DIVIDE SHORT	N DE	RX	P A S U E FK	7D	5.6.13
HALVE LONG	N HDR	RR	S U	24	5.6.11
HALVE SHORT	N HER	RR	S U	34	5.6.11
MULTIPLY LONG	N MDR	RR	S U E	2C	5.6.12
MULTIPLY LONG	N MD	RX	P A S U E	6C	5.6.12
MULTIPLY SHORT	N MER	RR	S U E	3C	5.6.12
MULTIPLY SHORT	N ME	RX	P A S U E	7C	5.6.12
SUBTRACT NORMALIZED LONG	N SDR	RR	C S U E LS	2B	5.6.8
SUBTRACT NORMALIZED LONG	N SD	RX	C P A S U E LS	6B	5.6.8
SUBTRACT NORMALIZED SHORT	N SER	RR	C S U E LS	3B	5.6.8
SUBTRACT NORMALIZED SHORT	N SE	RX	C P A S U E LS	7B	5.6.8
SUBTRACT UNNORMALIZED LONG	SWR	RR	C S E LS	2F	5.6.9
SUBTRACT UNNORMALIZED LONG	SW	RX	C P A S E LS	6F	5.6.9
SUBTRACT UNNORMALIZED SHORT	SUR	RR	C S E LS	3F	5.6.9
SUBTRACT UNNORMALIZED SHORT	SU	RX	C P A S E LS	7F	5.6.9

JUNE 1, 1971

13. Exponent Underflow (U)

When the result characteristic in floating-point addition, subtraction, multiplication, halving, or division is less than zero and the result fraction is not zero, an exponent-underflow exception is recognized. The operation is completed.

The setting of the exponent-underflow mask (PSW bit 38) affects the results of the operation. When the mask bit is zero, the sign, characteristic, and fraction are set to zero, making the result a true zero. When the mask bit is one, the fraction is normalized, the characteristic is made 128 larger than the correct characteristic, and the sign and fraction remain correct.

The instruction-length code is 1 or 2.

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
ADD NORMALIZED LONG	N ADR	RR	C S U E LS	2A	5.6.6
ADD NORMALIZED LONG	N AD	RX	C P A S U E LS	6A	5.6.6
ADD NORMALIZED SHORT	N AER	RR	C S U E LS	3A	5.6.6
ADD NORMALIZED SHORT	N AE	RX	C P A S U E LS	7A	5.6.6
DIVIDE LONG	N DDR	RR	S U E FK	2D	5.6.13
DIVIDE LONG	N DD	RX	P A S U E FK	6D	5.6.13
DIVIDE SHORT	N DER	RR	S U E FK	3D	5.6.13
DIVIDE SHORT	N DE	RX	P A S U E FK	7D	5.6.13
MULTIPLY LONG	N MDR	RR	S U E	2C	5.6.12
MULTIPLY LONG	N MD	RX	P A S U E	6C	5.6.12
MULTIPLY SHORT	N MER	RR	S U E	3C	5.6.12
MULTIPLY SHORT	N ME	RX	P A S U E	7C	5.6.12
SUBTRACT NORMALIZED LONG	N SDR	RR	C S U E LS	2B	5.6.8
SUBTRACT NORMALIZED LONG	N SD	RX	C P A S U E LS	6B	5.6.8
SUBTRACT NORMALIZED SHORT	N SER	RR	C S U E LS	3B	5.6.8
SUBTRACT NORMALIZED SHORT	N SE	RX	C P A S U E LS	7B	5.6.8

14. Significance (LS)

The result of a floating-point addition or subtraction has an all-zero fraction.

The operation is completed. The interruption may be masked by PSW bit 39. The manner in which the operation is completed is determined by the mask bit.

The instruction-length code is 1 or 2.

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
ADD NORMALIZED LONG	N ADR	RR	C S U E LS	2A	5.6.6
ADD NORMALIZED LONG	N AD	RX	C P A S U E LS	6A	5.6.6
ADD NORMALIZED SHORT	N AER	RR	C S U E LS	3A	5.6.6
ADD NORMALIZED SHORT	N AE	RX	C P A S U E LS	7A	5.6.6
ADD UNNORMALIZED LONG	AWR	RR	C S E LS	2E	5.6.7
ADD UNNORMALIZED LONG	AW	RX	C P A S E LS	6E	5.6.7
ADD UNNORMALIZED SHORT	AUR	RR	C S E LS	3E	5.6.7
ADD UNNORMALIZED SHORT	AU	RX	C P A S E LS	7E	5.6.7
SUBTRACT NORMALIZED LONG	N SDR	RR	C S U E LS	2B	5.6.8
SUBTRACT NORMALIZED LONG	N SD	RX	C P A S U E LS	6B	5.6.8
SUBTRACT NORMALIZED SHORT	N SER	RR	C S U E LS	3B	5.6.8
SUBTRACT NORMALIZED SHORT	N SE	RX	C P A S U E LS	7B	5.6.8
SUBTRACT UNNORMALIZED LONG	SWR	RR	C S E LS	2F	5.6.9
SUBTRACT UNNORMALIZED LONG	SW	RX	C P A S E LS	6F	5.6.9
SUBTRACT UNNORMALIZED SHORT	SUR	RR	C S E LS	3F	5.6.9
SUBTRACT UNNORMALIZED SHORT	SU	RX	C P A S E LS	7F	5.6.9

15. Floating-Point Divide (FK)

Division by a floating-point number with zero fraction is attempted. The operation is suppressed.

The instruction-length code is 1 or 2.

NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
DIVIDE LONG	N DDR	RR	S U E FK	2D	5.6.13
DIVIDE LONG	N DD	RX	P A S U E FK	6D	5.6.13
DIVIDE SHORT	N DER	RR	S U E FK	3D	5.6.13
DIVIDE SHORT	N DE	RX	P A S U E FK	7D	5.6.13

SYSTEM CONTROL FACILITIES

OPERATOR CONTROLS		
NAME	IMPLEMENTATION	SECTION
EMERGENCY PULL*	Pull switch	11.2.1
ELEMENT MPO PULL**	Pull switch	11.2.2
POWER ON/OFF	Toggle switch	11.2.3
POWER SEQUENCE COMPLETE	Light	11.2.4
WAIT	Light	11.2.5
MANUAL	Light	11.2.6
SYSTEM**	Light	11.2.7
TEST**	Light	11.2.8
LOAD	Light	11.2.9
LOAD UNIT	Three rotary switches	11.2.10
LOAD	Switch	11.2.11
MAIN STORAGE SELECT	Rotary switch	11.2.12
SYSTEM INTERLOCK	Key operated switch	11.2.13
CE SELECT*	Rotary switch	11.2.14
STOP	Switch	11.2.15
RATE	Rotary switch	11.2.16
START	Switch	11.2.17
STORAGE SELECT	Lever switch	11.2.18
ADDRESS	Lever switches	11.2.19
DATA	Lever switches	11.2.20
STORE	Switch	11.2.21
DISPLAY	Switch	11.2.22
SET IC	Switch	11.2.23
INTERRUPT	Switch	11.2.24
PSW RESTART**	Switch	11.2.25
STATE THREE	Light	11.2.26
STATE TWO	Light	11.2.27
STATE ONE	Light	11.2.28
STATE ZERO	Light	11.2.29
LAMP TEST	Lever switch	11.2.30
CONTROL CE*	Lever switch (4)	11.2.31
CONTROL CE ACTIVATE*	Switch	11.2.32
ADDRESS-COMPARE	Lever switches or rotary	11.2.33
ADDRESS-COMPARE ENABLE*	Switch	11.2.34
SENSE SWITCHES*	Lever switches	11.2.35
READER/PUNCH-PRINTER SELECT*	Rotary switch	11.2.36
READER/PUNCH-PRINTER ENABLE*	Switch, lighted	11.2.37
PRINTER KEYBOARD SELECT***	Rotary switch	11.2.38
PRINTER KEYBOARD ENABLE***	Switch, lighted	11.2.39
CONSOLE RESET***	Switch	11.2.40
BELL RESET*	Switch	11.2.41
BUZZER RESET*	Switch	11.2.42
ALL STOP*	Switch	11.2.43
INVALID SELECTION*	Switch	11.2.44
TEST MODE*	Light	11.2.45
360 MODE**	Switch, lighted	11.2.46
ENABLE SYSTEM IPL	Pushbutton	11.2.47
I/O INTERFACE ENABLE/DISABLE	Toggle Switch	11.2.48
I/O INTERFACE DISABLED	Light	11.2.49
FAULT RESET	Switch	11.2.50
BUZZER/BELL RESET	Switch	11.2.51

Legend

The term "switch" without a modifier denotes a pushbutton switch.

* These functions are provided on the system console and configuration console only.

** These functions are provided on the CE control panel only.

*** These functions are provided on the system console only.

**** These functions are provided on the configuration console.

All other functions are provided on the system console, configuration console and the CE control panel.

JUNE 1, 1971

STATES OF WAIT, SYSTEM AND MANUAL LIGHTS			
SYSTEM LIGHT	MANUAL LIGHT	WAIT LIGHT	CE STATE
off	off	off	Power off
off	off	on	Operating, Waiting
off	on	off	Stopped, Running
off	on	on	Stopped, waiting
on	off	off	Operating, Running

INPUT-OUTPUT ADDRESS ASSIGNMENT				
0000	0000	xxxx	xxxx	Devices on multiplexor channel A
0000	0001	xxxx	xxxx	Devices on selector channel 1A
0000	0010	xxxx	xxxx	Devices on selector channel 2A
0000	0011	xxxx	xxxx	Devices on selector channel 3A
0000	0100	xxxx	xxxx	Devices on multiplexor channel B
0000	0101	xxxx	xxxx	Devices on selector channel 1B
0000	0110	xxxx	xxxx	Devices on selector channel 2B
0000	0111	xxxx	xxxx	Devices on selector channel 3B
0000	1000	xxxx	xxxx	Devices on multiplexor channel C
0000	1001	xxxx	xxxx	Devices on selector channel 1C
0000	1010	xxxx	xxxx	Devices on selector channel 2C
0000	1011	0000	0000	Invalid
		to		
1111	1111	1111	1111	

INPUT/OUTPUT STATES	
NAME	ABBREVIATIONS AND DEFINITION
<u>I/O Devices</u>	
Available	A None of the following states
Working	W Device executing an operation
Not operational	N Device not operational
Interruption pending	I Interruption condition pending in device
<u>Channel</u>	
Available	A None of the following states
Interruption pending	I Interruption immediately available from channel
Working	W Channel operating in burst mode
Not operational	N Channel not operational
<u>Subchannel</u>	
Available	A None of the following states
Interruption pending	I Information for CSW available in subchannel
Working	W Subchannel executing an operation
Not operational	N Subchannel not operational

JUNE 1, 1971

CONDITIONS	CONDITION CODE FOR:					
	START	TEST	HALT	SET	TEST	
	I/O	I/O	I/O	PCI	CHANNEL	
Available	AAA	0,1*	0	1*	0	0
Interruption pending in device	AAI	1*	1*	1*	0	0
Device working	AAW	1*	1*	1*	0	0
Device not operational	AAN	3	3	3	0	0
Interruption pending in subchannel	AIX#					
For the addressed device		2	1*	0	2	0
For another device		2	2	0	2	0
Subchannel working	awx#	2	2	1*	2	0
Subchannel not operational	ANX#	3	3	3	3	0
Interruption pending in channel	IXX	see note below				1
Channel working	WXX#	2	2	2	2	2
Channel not operational	NXX#	3	3	3	3	3
Error						
Channel equipment error		1*	1*	1*	1*	-
Channel programming error		1*	-	-	-	-
Device error		1*	-*	-	-	-

Legend

* The CSW or its status portion is stored at location 64 during execution of the instruction.

The symbol X stands for A, I, W, and N, and indicates that the state of the corresponding component is not significant. As an example, AIX denotes the state AIA, AII, AIW, and AIN, while IXX represents a total of 16 states, some of which do not occur.

- The condition cannot be identified during execution of the instruction.

Note: For the purpose of executing START I/O, TEST I/O, SET PCI, and HALT I/O, a channel containing a pending interruption condition appears the same as an available channel, and the condition code setting depends upon the states of the subchannel and device. The condition codes for the IXX state are the same as for the AXX states, where the Xs represent the states of the subchannel and the device. As an example, the condition code for the IAA state is the same as for the AAA state, and the condition code for the IAW state is the same as for the AAW state.

JUNE 1, 1971

CONTENT OF CHANNEL STATUS WORD ADDRESS FIELD	
CONDITION	CONTENT
Channel control check	Unpredictable
Status stored by START I/O	Unchanged
Status stored by HALT I/O	Unchanged
Invalid CCW address spec. in TIC	Address of TIC + 8
Invalid CCW address in TIC	Address of TIC + 8
Invalid CCW address generated	Address first invalid CCW + 8
Invalid command code	Address of invalid CCW + 8
Invalid count	Address of invalid CCW + 8
Invalid data address	Address of invalid CCW + 8
Invalid CCW format	Address of invalid CCW + 8
Invalid sequence -- 2 TICs	Address of second TIC + 8
Protection check	Address of invalid CCW + 8
Chaining check	Address of last-used CCW + 8
Termination under count control	Address of last-used CCW + 8
Termination by I/O device	Address of last used CCW + 8
Termination by HALT I/O	Address of last-used CCW + 8
Suppression of command chaining due to Unit check or Unit exception with Device end or Control unit end	Address of last CCW used in the completed operation + 8
Termination on command chaining by Attention by Unit check by Unit exception	Address of CCW specifying the new operation + 8
Program-controlled interruption	Address of last-used CCW + 8
Interface control check	Address of last-used CCW + 8
Ch. end after HIO on sel. ch.	Zero
Control unit end	Zero
Device end	Zero
Attention	Zero
Busy	Zero
Status modifier	Zero

JUNE 1, 1971

CONTENT OF CHANNEL STATUS WORD COUNT FIELD	
CONDITION	CONTENT
Channel control check	Unpredictable
Status stored by START I/O	Unchanged
Status stored by HALT I/O	Unchanged
Program check	Unpredictable
Protection check	Unpredictable
Chaining check	Unpredictable
Termination under count control	Correct
Termination by I/O device	Correct
Termination by HALT I/O	Correct
Suppression of command chaining due to Unit check or Unit exception with Device end or Control unit end	Correct. Residual count of last CCW used in the completed operation.
Termination of command chaining by Attention by Unit check by Unit exception	Correct. Original count of CCW specifying the new operation
Program-controlled interruption	Unpredictable
Interface Control Check	Correct
Ch. end after HIO on sel. ch.	Zero
Control unit end	Zero
Device end	Zero
Attention	Zero
Busy	Zero
Status Modifier	Zero

JUNE 1, 1971

TIME AND METHOD OF CREATING AND STORING STATUS INDICATION					
STATUS	WHEN I/O IDLE	WHEN SUBCHANNEL WORKING	UPON TERMINATION OF OPERATION		
			AT SUBCHANNEL	AT CONTROL UNIT	AT DEVICE
ATTENTION	C*				C
STATUS MODIFIER					C
CONTROL UNIT END				C*	
BUSY					
CHANNEL END			C*A	C*H	
DEVICE END	C*				C*A
UNIT CHECK			C	C	C
UNIT EXCEPTION			C	C	C
PROGRAM-CONTROLLED INTERRUPTION		C*	C*		
INCORRECT LENGTH		C	C		
PROGRAM CHECK		C	C		
PROTECTION CHECK		C	C		
CHANNEL DATA CHECK		C	C		
CHANNEL CONTROL CHECK	C*	C*	C*	C*	C*
INTERFACE CONTROL CHECK	C*	C*	C*	C*	C*
CHAINING CHECK		C	C		

JUNE 1, 1971

TIME AND METHOD (continued)						
STATUS	DURING COMMAND CHAINING	BY START I/O	BY TEST I/O	BY HALT I/O	BY I/O INTER- RUPTION	BY SET PCI
ATTENTION	C*	S	S		S	S
STATUS MODIFIER	C	C S	C S	C S	S	S
CONTROL UNIT END		C S	C S	C S	S	S
BUSY	C	C S	C S	C S	S	S
CHANNEL END	C*#A	C*S	S		S	S
DEVICE END	C#	C*S	S		S	S
UNIT CHECK	C*	C S	C S		C S	S
UNIT EXCEPTION	C*	C S	S		S	S
PROGRAM-CONTROLLED INTERRUPTION	C*	C S	S		S	S
INCORRECT LENGTH			S		S	S
PROGRAM CHECK	C*	C S	S		S	S
PROTECTION CHECK	C*	C S	S		S	S
CHANNEL DATA CHECK			S		S	S
CHANNEL CONTROL CHECK	C*	C S	C S	C S	C S	C S
INTERFACE CONTROL CHECK	C*	C S	C S	C S	C S	C S
CHAINING CHECK	C*	C S	S		S	S

JUNE 1, 1971

Legend

C the channel or the device can create or present the status condition at the indicated time. A CSW or its status portion is not necessarily stored at this time.

Conditions such as Channel End and Device End are created at the indicated time. Other conditions may have been created previously, but are made accessible to the program only at the indicated time. Examples of such conditions are Program Check and Channel Data Check, which are detected while data are transferred, but are made available to the program only with Channel End, unless the PCI flag or equipment malfunctioning have caused the interruption condition to be generated earlier.

S the status indication is stored in the CSW at the indicated time.

An S appearing alone indicates that the condition has been created previously. The letter C appearing with the S indicates that the status condition did not necessarily exist previously in the form that causes the program to be altered, and may have been created by the I/O instruction or I/O interruption. For example, equipment malfunctioning may be detected during an I/O interruption, thus causing Channel Control Check or Interface Control Check to be indicated; or a device may signal the control-unit-busy condition in response to interrogation by an I/O instruction, thus causing Status Modifier, Busy, and Control Unit End to be indicated in the CSW.

* the status condition generates or, in the case of Channel Data Check, may generate an interruption condition.

A Channel End and Device End do not result in interruption conditions when command chaining is specified and no unusual conditions have been detected.

≠ this status indication can be created at the indicated time only by an immediate operation.

H when an operation on the selector channel has been terminated by HALT I/O, Channel End indicates the termination of the data handling portion of the operation at the control unit.

FLAGS IN CURRENT CCW			IMMEDIATE OPERATION	REGULAR OPERATION		
			Count not exhausted and channel end from device	Count exhausted, end of block at device not reached	Count exhausted and channel end from device	Count not exhausted and channel end from device
CD	CC	SLI				
0	0	0	End, -	Stop, IL	End, -	End, IL
0	0	1	End, -	Stop, -	End, -	End, -
0	1	0	Chain command	Stop, IL	Chain command	End, IL
0	1	1	Chain command	Chain command	Chain command	Chain command
1	0	0	End, -	Chain data	*	End, IL
1	0	1	End, -	Chain data	*	End, IL**
1	1	0	End, -	Chain data	*	End, IL
1	1	1	End, -	Chain data	*	End, IL**

Explanation:

End - Operation is terminated and an interruption condition is generated in the subchannel.

Stop - Device is signalled to terminate data transfer, but subchannel remains in the working state until channel end is received.

IL - Incorrect length is indicated with the interruption condition.

Chain command - The channel performs command chaining upon receipt of device end.

Chain data - The channel immediately fetches a new CCW for the same operation.

* The situation where the count is zero but data chaining is indicated at the time the device provides channel end cannot validly occur. When data chaining is indicated, the channel fetches the new CCW after transferring the last byte of data designated by the current CCW but before the device provides the next request for data or status transfer. As a result, the channel recognizes the channel end from the device only after it has fetched the new CCW, which cannot contain a count of zero unless a programming error has been made.

** Note that the SLI flag does not suppress the incorrect length indication when the CD flag is on.

JUNE 1, 1971

INDICATION OF THE BUSY CONDITION IN THE CHANNEL STATUS WORD

The table lists the condition when the busy bit (B) appears in the CSW and when it is accompanied by the status-modifier bit (SM). A dash (-) indicates that the busy bit is off; an asterisk (*) indicates that CSW status is not stored or an I/O interruption cannot occur; the (cl) indicates that the interruption condition is cleared and the status appears in the CSW. The abbreviation DE stands for Device End, while CU stands for Control Unit.

CONDITION	CSW STATUS STORED BY:			
	START I/O	TEST I/O	HALT I/O	I/O INT.
Subchannel available				
DE or Attention in device	B,cl	-,cl	*	-,cl
Device working, CU available	B	B	*	*
CU End or Channel End in CU				
for the addressed device	B,cl	-,cl	*	-,cl
for another device	B,SM	B,SM	*	-,cl
CU working	B,SM	B,SM	*	*
Interruption pend. in subchannel				
for the addressed device due to				
chaining terminated by attention	*	B,cl	*	B,cl
(NOTE)				
other type of termination	*	-,cl	*	-,cl
Subchannel working				
CU available	*	*	-	*
CU working	*	*	B,SM	*

NOTE:

The busy bit is included in the status associated with a pending interruption condition from the subchannel only when a chain of commands has been prematurely terminated due to Attention and no interruption was pending in the channel at the time of chaining.

HANDLING OF INCORRECT LENGTH				
FLAGS			ACTION AND INDICATION	
CD	CC	SLI	REGULAR OPERATION	IMMEDIATE OPERATION
0	0	0	Stop,IL	Stop,--
0	0	1	Stop,--	Stop,--
0	1	0	Stop,IL	Chain command
0	1	1	Chain command	Chain command
1	0	0	Stop,IL	Stop,--
1	0	1	Stop,IL	Stop,--
1	1	0	Stop,IL	Stop,--
1	1	1	Stop,IL	Stop,--

ALPHABETIC LIST OF INSTRUCTIONS					
NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
ADD	AR	RR	C	IF	1A 3.5.8
ADD	A	RX	C	P A S IF	5A 3.5.8
ADD DECIMAL	AP	SS	C	P A S D DF	FA 4.5.1
ADD HALFWORD	AH	RX	C	P A S IF	4A 3.5.9
ADD LOGICAL	ALR	RR	C		1E 3.5.10
ADD LOGICAL	AL	RX	C	P A S	5E 3.5.10
ADD NORMALIZED LONG	N ADR	RR	C	S U E LS	2A 5.6.6
ADD NORMALIZED LONG	N AD	RX	C	P A S U E LS	6A 5.6.6
ADD NORMALIZED SHORT	N AER	RR	C	S U E LS	3A 5.6.6
ADD NORMALIZED SHORT	N AE	RX	C	P A S U E LS	7A 5.6.6
ADD UNNORMALIZED LONG	AWR	RR	C	S E LS	2E 5.6.7
ADD UNNORMALIZED LONG	AW	RX	C	P A S E LS	6E 5.6.7
ADD UNNORMALIZED SHORT	AUR	RR	C	S E LS	3E 5.6.7
ADD UNNORMALIZED SHORT	AU	RX	C	P A S E LS	7E 5.6.7
AND	NR	RR	C		14 6.4.5
AND	N	RX	C	P A S	54 6.4.5
AND	NI	SI	C	P A	94 6.4.5
AND	NC	SS	C	P A	D4 6.4.5
BRANCH AND LINK	BALR	RR			05 7.4.2
BRANCH AND LINK	BAL	RX			45 7.4.2
BRANCH ON CONDITION	BCR	RR			07 7.4.1
BRANCH ON CONDITION	BC	RX			47 7.4.1
BRANCH ON COUNT	BCTR	RR			06 7.4.3
BRANCH ON COUNT	BCT	RX			46 7.4.3
BRANCH ON INDEX HIGH	BXH	RX			86 7.4.4
BRANCH ON INDEX LOW OR EQUAL	BXLE	RS			87 7.4.5
COMPARE	CR	RR	C		19 3.5.14
COMPARE	C	RX	C	P A S	59 3.5.14
COMPARE DECIMAL	CP	SS	C	P A S D	F9 4.5.4
COMPARE HALFWORD	CH	RX	C	P A S	49 3.5.15
COMPARE LOGICAL	CLR	RR	C		15 6.4.4
COMPARE LOGICAL	CL	RX	C	P A S	55 6.4.4
COMPARE LOGICAL	CLI	SI	C	P A	95 6.4.4
COMPARE LOGICAL	CLC	SS	C	P A	D5 6.4.4
COMPARE LONG	CDR	RR	C	S	29 5.6.10
COMPARE LONG	CD	RX	C	P A S	69 5.6.10
COMPARE SHORT	CER	RR	C	S	39 5.6.10
COMPARE SHORT	CE	RX	C	P A S	79 5.6.10
CONVERT AND SORT SYMBOLS	CSS	RR	C	P A S	02 8.6.1.2
CONVERT TO BINARY	CVB	RX		P A S D IK	4F 3.5.19
CONVERT TO DECIMAL	CVD	RX		P A S	4E 3.5.20
CONVERT WEATHER LINES	CVWL	RR		P A S	03 8.6.1.3
DELAY	DLY	RR			0B 8.4.5.1
DIAGNOSE		SI		M P A S	83 8.5.1.9
DIVIDE	DR	RR		S IK	1D 3.5.18
DIVIDE	D	RX		P A S IK	5D 3.5.18
DIVIDE DECIMAL	DP	SS		P A S D DK	FD 4.5.6
DIVIDE LONG	N DDR	RR		S U E FK	2D 5.6.13
DIVIDE LONG	N DD	RX		P A S U E FK	6D 5.6.13
DIVIDE SHORT	N DER	RR		S U E FK	3D 5.6.13
DIVIDE SHORT	N DE	RX		P A S U E FK	7D 5.6.13

JUNE 1, 1971

ALPHABETIC LIST OF INSTRUCTIONS (continued)						
NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION	
EDIT	ED	SS	C P A D	DE	6.4.14	
EDIT AND MARK	EDMK	SS	C P A D	DF	6.4.15	
EXCLUSIVE OR	XR	RR	C	17	6.4.7	
EXCLUSIVE OR	X	RX	C P A S	57	6.4.7	
EXCLUSIVE OR	XI	SI	C P A	97	6.4.7	
EXCLUSIVE OR	XC	SS	C P A	D7	6.4.7	
EXECUTE	EX	RX	P A S EX	44	7.4.6	
HALT I/O	HIO	SI	C M	9E	10.2.7.3	
HALVE LONG	N HDR	RR	S U	24	5.6.11	
HALVE SHORT	N HER	RR	S U	34	5.6.11	
INSERT ADDRESS TRANSLATOR	IATR	RR		0E	8.4.5.8	
INSERT CHARACTER	IC	RX	P A	43	6.4.9	
INSERT STORAGE KEY	ISK	RR	M A S	09	8.5.1.6	
LOAD	LR	RR		18	3.5.1	
LOAD	L	RX	P A S	58	3.5.1	
LOAD ADDRESS	LA	RX		41	6.4.11	
LOAD AND TEST	LTR	RR	C	12	3.5.3	
LOAD AND TEST LONG	LTDR	RR	C S	22	5.6.2	
LOAD AND TEST SHORT	LTER	RR	C S	32	5.6.2	
LOAD CHAIN	LC	RX	C P A S	52	8.6.1.1	
LOAD COMPLEMENT	LCR	RR	C IF	13	3.5.4	
LOAD COMPLEMENT LONG	LCDR	RR	C S	23	5.6.3	
LOAD COMPLEMENT SHORT	LCER	RR	C S	33	5.6.3	
LOAD HALFWORD	LH	RX	P A S	48	3.5.2	
LOAD IDENTITY	LI	RR		0C	8.4.5.2	
LOAD LONG	LDR	RR	S	28	5.6.1	
LOAD LONG	LD	RX	P A S	68	5.6.1	
LOAD MULTIPLE	LM	RS	P A S	98	3.5.7	
LOAD NEGATIVE	LNR	RR	C	11	3.5.6	
LOAD NEGATIVE LONG	LNDR	RR	C S	21	5.6.5	
LOAD NEGATIVE SHORT	LNER	RR	C S	31	5.6.5	
LOAD POSITIVE	LPR	RR	C IF	10	3.5.5	
LOAD POSITIVE LONG	LPDR	RR	C S	20	5.6.4	
LOAD POSITIVE SHORT	LPER	RR	C S	30	5.6.4	
LOAD PS BASE ADDRESS	LPSB	SI	M P A S	A1	8.4.5.3	
LOAD PSW	LPSW	SI	L M P A S	82	8.5.1.1	
LOAD SHORT	LER	RR	S	38	5.6.1	
LOAD SHORT	LE	RX	P A S	78	5.6.1	
MOVE	MVI	SI	P A	92	6.4.1	
MOVE	MVC	SS	P A	D2	6.4.1	
MOVE NUMERICS	MVN	SS	P A	D1	6.4.2	
MOVE WITH OFFSET	MVO	SS	P A	F1	4.5.9	
MOVE WORD	MVW	SS	P A S	D8	8.4.5.9	
MOVE ZONES	MVZ	SS	P A	D3	6.4.3	
MULTIPLY	MR	RR	P A S	1C	3.5.16	
MULTIPLY	M	RX	P A S	5C	3.5.16	
MULTIPLY DECIMAL	MP	SS	P A S D	FC	4.5.5	
MULTIPLY HALFWORD	MH	RX	P A S	4C	3.5.17	
MULTIPLY LONG	N MDR	RR	S U E	2C	5.6.12	
MULTIPLY LONG	N MD	RX	P A S U E	6C	5.6.12	
MULTIPLY SHORT	N MER	RR	S U E	3C	5.6.12	
MULTIPLY SHORT	N ME	RX	P A S U E	7C	5.6.12	

JUNE 1, 1971

ALPHABETIC LIST OF INSTRUCTIONS (continued)						
NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION	
OR	OR	RR	C	16	6.4.6	
OR	O	RX	C P A S	56	6.4.6	
OR	OI	SI	C P A	96	6.4.6	
OR	OC	SS	C P A	D6	6.4.6	
PACK	PACK	SS	P A	F2	4.5.7	
READ DIRECT	RDD	SI	M P A S	85	8.5.1.8	
REPACK SYMBOLS	RPSB	RR	C P A S	0F	8.6.1.4	
SET ADDRESS TRANSLATOR	SATR	RR	C M S	0D	8.4.5.7	
SET CONFIGURATION	SCON	RR	C M S	01	8.4.5.4	
SET PCI	SPCI	SI	C M	9B	10.2.7.5	
SET PROGRAM MASK	SPM	RR	L	04	8.5.1.2	
SET STORAGE KEY	SSK	RR	M A S	08	8.5.1.5	
SET SYSTEM MASK	SSM	SI	M P A	80	8.5.1.3	
SHIFT LEFT DOUBLE	SLDA	RS	C S	IF 8F	3.5.26	
SHIFT LEFT DOUBLE LOGICAL	SLDL	RS	S	8D	6.4.18	
SHIFT LEFT SINGLE	SLA	RS	C	IF 8B	3.5.24	
SHIFT LEFT SINGLE LOGICAL	SLL	RS		89	6.4.16	
SHIFT RIGHT DOUBLE	SRDA	RS	C S	8E	3.5.27	
SHIFT RIGHT DOUBLE LOGICAL	SRDL	RS	S	8C	6.4.19	
SHIFT RIGHT SINGLE	SRA	RS	C	8A	3.5.25	
SHIFT RIGHT SINGLE LOGICAL	SRL	RS		88	6.4.17	
START I/O	SIO	SI	C M	9C	10.2.7.1	
START I/O PROCESSOR	SIOP	SI	C M S	9A	8.4.5.10	
STORE	ST	RX	P A S	50	3.5.21	
STORE CHARACTER	STC	RX	P A	42	6.4.10	
STORE HALFWORD	STH	RX	P A S	40	3.5.22	
STORE LONG	STD	RX	P A S	60	5.6.14	
STORE MULTIPLE	STM	RS	P A S	90	3.5.23	
STORE PS BASE ADDRESS	SPSB	SI	M P A S	A0	8.4.5.5	
STORE SHORT	STE	RX	P A S	70	5.6.14	
SUBTRACT	SR	RR	C	IF 1B	3.5.11	
SUBTRACT	S	RX	C P A S	IF 5B	3.5.11	
SUBTRACT DECIMAL	SP	SS	C P A D	DF FB	4.5.2	
SUBTRACT HALFWORD	SH	RX	C P A S	IF 4B	3.5.12	
SUBTRACT LOGICAL	SLR	RR	C	1F	3.5.13	
SUBTRACT LOGICAL	SL	RX	C P A S	5F	3.5.13	
SUBTRACT NORMALIZED LONG	N SDR	RR	C S U E LS	2B	5.6.8	
SUBTRACT NORMALIZED LONG	N SD	RX	C P A S U E LS	6B	5.6.8	
SUBTRACT NORMALIZED SHORT	N SER	RR	C S U E LS	3B	5.6.8	
SUBTRACT NORMALIZED SHORT	N SE	RX	C P A S U E LS	7B	5.6.8	
SUBTRACT UNNORMALIZED LONG	SWR	RR	C S E LS	2F	5.6.9	
SUBTRACT UNNORMALIZED LONG	SW	RX	C P A S E LS	6F	5.6.9	
SUBTRACT UNNORMALIZED SHORT	SUR	RR	C S E LS	3F	5.6.9	
SUBTRACT UNNORMALIZED SHORT	SU	RX	C P A S E LS	7F	5.6.9	
SUPERVISOR CALL	SVC	RR		0A	8.5.1.4	

JUNE 1, 1971

ALPHABETIC LIST OF INSTRUCTIONS (continued)					
NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	SECTION
TEST AND SET	TS	SI	C P A	93	8.4.5.6
TEST CHANNEL	TCH	SI	C M	9F	10.2.7.4
TEST I/O	TIO	SI	C M	9D	10.2.7.2
TEST UNDER MASK	TM	SI	C P A	91	6.4.8
TRANSLATE	TR	SS	P A	DC	6.4.12
TRANSLATE AND TEST	TRT	SS	C P A	DD	6.4.13
UNPACK	UNPK	SS	P A	F3	4.5.8
WRITE DIRECT	WRD	SI	C M P A S	84	8.5.1.7
ZERO AND ADD	ZAP	SS	C P A D DF	F8	4.5.3

Legend

- A Addressing exception
- C Condition code is set
- D Data exception
- DF Decimal-overflow exception
- DK Decimal-divide exception
- E Exponent-overflow exception
- EX Execute exception
- FK Floating-point divide exception
- IF Fixed-point overflow exception
- IK Fixed-point divide exception
- L New condition code loaded
- LS Significance exception
- M Privileged-operation exception
- N Normalized operation
- P Protection exception
- S Specification exception
- U Exponent-underflow exception

JUNE 1, 1971

INSTRUCTION FORMATS BY OPERATION CODE

SCON RR	01	R1	R2
CSS RR	02	R1	09
CVWL RR	03	////	09
SPM RR	04	R1	////
BALR RR	05	R1	R2
BCTR RR	06	R1	R2
BCR RR	07	M1	R2
SSK RR	08	R1	R2
ISK RR	09	R1	R2
SVC RR	0A	R1	R2
DLY RR	0B	N	
LI RR	0C	R1	////
SATR RR	0D	R1	R2
IATR RR	0E	R1	R2
RPSB RR	0F	////	////
LPR RR	10	R1	R2

JUNE 1, 1971

LNR	RR	11	R1	R2
LTR	RR	12	R1	R2
LCR	RR	13	R1	R2
NR	RR	14	R1	R2
CLR	RR	15	R1	R2
OR	RR	16	R1	R2
XR	RR	17	R1	R2
LR	RR	18	R1	R2
CR	RR	19	R1	R2
AR	RR	1A	R1	R2
SR	RR	1B	R1	R2
MR	RR	1C	R1	R2
DR	RR	1D	R1	R2
ALR	RR	1E	R1	R2
SLR	RR	1F	R1	R2
LPDR	RR	20	R1	R2

(long operands)

JUNE 1, 1971

LNDR RR	21	R1	R2	(long operands)
LTDR RR	22	R1	R2	(long operands)
LCDR RR	23	R1	R2	(long operands)
HDR RR	24	R1	R2	(long operands)
LDR RR	28	R1	R2	(long operands)
CDR RR	29	R1	R2	(long operands)
ADR RR	2A	R1	R2	(long operands)
SDR RR	2B	R1	R2	(long operands)
MDR RR	2C	R1	R2	(long operands)
DDR RR	2D	R1	R2	(long operands)
AWR RR	2E	R1	R2	(long operands)
SWR RR	2F	R1	R2	(long operands)
LPER RR	30	R1	R2	(short operands)
LNER RR	31	R1	R2	(short operands)
LTER RR	32	R1	R2	(short operands)
LCER RR	33	R1	R2	(short operands)

JUNE 1, 1971

HER	RR	34	R1	R2					(short operands)
LER	RR	38	R1	R2					(short operands)
CER	RR	39	R1	R2					(short operands)
AER	RR	3A	R1	R2					(short operands)
SER	RR	3B	R1	R2					(short operands)
MER	RR	3C	R1	R2					(short operands)
DER	RR	3D	R1	R2					(short operands)
AUR	RR	3E	R1	R2					(short operands)
SUR	RR	3F	R1	R2					(short operands)
STH	RX	40	R1	X2	B2	D2			
LA	RX	41	R1	X2	B2	D2			
STC	RX	42	R1	X2	B2	D2			
IC	RX	43	R1	X2	B2	D2			
EX	RX	44	R1	X2	B2	D2			
BAL	RX	45	R1	X2	B2	D2			
BCT	RXT	46	R1	X2	B2	D2			

JUNE 1, 1971

BC	RX	47	M1	X2	B2	D2
LH	RX	48	R1	X2	B2	D2
CH	RX	49	R1	X2	B2	D2
AH	RX	4A	R1	X2	B2	D2
SH	RX	4B	R1	X2	B2	D2
MH	RX	4C	R1	X2	B2	D2
CVD	RX	4E	R1	X2	B2	D2
CVB	RX	4F	R1	X2	B2	D2
ST	RX	50	R1	X2	B2	D2
LC	RX	52	R1	X2	B2	D2
N	RX	54	R1	X2	B2	D2
CL	RX	55	R1	X2	B2	D2
O	RX	56	R1	X2	B2	D2
X	RX	57	R1	X2	B2	D2
L	RX	58	R1	X2	B2	D2
C	RX	59	R1	X2	B2	D2

JUNE 1, 1971

A	RX	5A	R1	X2	B2	D2	
S	RX	5B	R1	X2	B2	D2	
M	RX	5C	R1	X2	B2	D2	
D	RX	5D	R1	X2	B2	D2	
AL	RX	5E	R1	X2	B2	D2	
SL	RX	5F	R1	X2	B2	D2	
STD	RX	60	R1	X2	B2	D2	(long operands)
LD	RX	68	R1	X2	B2	D2	(long operands)
CD	RX	69	R1	X2	B2	D2	(long operands)
AD	RX	6A	R1	X2	B2	D2	(long operands)
SD	RX	6B	R1	X2	B2	D2	(long operands)
MD	RX	6C	R1	X2	B2	D2	(long operands)
DD	RX	6D	R1	X2	B2	D2	(long operands)
AW	RX	6E	R1	X2	B2	D2	(long operands)
SW	RX	6F	R1	X2	B2	D2	(long operands)
STE	RX	70	R1	X2	B2	D2	(short operands)

JUNE 1, 1971

LE RX

78	R1	X2	B2	D2
----	----	----	----	----

 (short operands)

CE RX

79	R1	X2	B2	D2
----	----	----	----	----

 (short operands)

AE RX

7A	R1	X2	B2	D2
----	----	----	----	----

 (short operands)

SE RX

7B	R1	X2	B2	D2
----	----	----	----	----

 (short operands)

ME RX

7C	R1	X2	B2	D2
----	----	----	----	----

 (short operands)

DE RX

7D	R1	X2	B2	D2
----	----	----	----	----

 (short operands)

AU RX

7E	R1	X2	B2	D2
----	----	----	----	----

 (short operands)

SU RX

7F	R1	X2	B2	D2
----	----	----	----	----

 (short operands)

SSM SI

80	////////////////	B1	D1
----	------------------	----	----

LPSW SI

82	////////////////	B1	D1
----	------------------	----	----

SI

83	I2	B1	D1
----	----	----	----

 (diagnose)

WRD SI

84	I2	B1	D1
----	----	----	----

RDD SI

85	I2	B1	D1
----	----	----	----

BXH RS

86	R1	R3	B2	D2
----	----	----	----	----

BXLE RS

87	R1	R3	B2	D2
----	----	----	----	----

SRL RS

88	R1	////////	B2	D2
----	----	----------	----	----

JUNE 1, 1971

SLL	RS	89	R1	////	B2	D2
SRA	RS	8A	R1	////	B2	D2
SLA	RS	8B	R1	////	B2	D2
SRDL	RS	8C	R1	////	B2	D2
SLDL	RS	8D	R1	////	B2	D2
SRDA	RS	8E	R1	////	B2	D2
SLDA	RS	8F	R1	////	B2	D2
STM	RS	90	R1	R3	B2	D2
TM	SI	91	I2		B1	D1
MVI	SI	92	I2		B1	D1
TS	SI	93	////////		B1	D1
NI	SI	94	I2		B1	D1
CLI	SI	95	I2		B1	D1
OI	SI	96	I2		B1	D1
XI	SI	97	I2		B1	D1
LM	RS	98	R1	R3	B2	D2

JUNE 1, 1971

SIOP	SI	9A	I2	B1	D1		
SPCI	SI	9B	////	B1	D1		
SIO	SI	9C	////	B1	D1		
TIO	SI	9D	////	B1	D1		
HIO	SI	9E	////	B1	D1		
TCH	SI	9F	////	B1	D1		
SPSB	SI	A0	////	B1	D1		
LPSB	SI	A1	////	B1	D1		
MVN	SS	D1	L	B1	D1	B2	D2
MVC	SS	D2	L	B1	D1	B2	D2
MVZ	SS	D3	L	B1	D1	B2	D2
NC	SS	D4	L	B1	D1	B2	D2
CLC	SS	D5	L	B1	D1	B2	D2
OC	SS	D6	L	B1	D1	B2	D2
XC	SS	D7	L	B1	D1	B2	D2
MVW	SS	D8	L	B1	D1	B2	D2

JUNE 1, 1971

TR	SS	DC	L	B1	D1	B2	D2	
TRT	SS	DD	L	B1	D1	B2	D2	
ED	SS	DE	L	B1	D1	B2	D2	
EDMK	SS	DF	L	B1	D1	B2	D2	
MVO	SS	F1	L1	L2	B1	D1	B2	D2
PACK	SS	F2	L1	L2	B1	D1	B2	D2
UNPK	SS	F3	L1	L2	B1	D1	B2	D2
ZAP	SS	F8	L1	L2	B1	D1	B2	D2
CP	SS	F9	L1	L2	B1	D1	B2	D2
AP	SS	FA	L1	L2	B1	D1	B2	D2
SP	SS	FB	L1	L2	B1	D1	B2	D2
MP	SS	FC	L1	L2	B1	D1	B2	D2
DP	SS	FD	L1	L2	B1	D1	B2	D2

JUNE 1, 1971

INSTRUCTION FORMATS BY MNEMONIC

MNEMONIC

CODE AND FIELDS

SI		83	I2	B1	D1			(diagnose)
A	RX	5A	R1	X2	B2	D2		
AD	RX	6A	R1	X2	B2	D2		(long operands)
ADR	RR	2A	R1	R2				(long operands)
AE	RX	7A	R1	X2	B2	D2		(short operands)
AER	RR	3A	R1	R2				(short operands)
AH	RX	4A	R1	X2	B2	D2		
AL	RX	5E	R1	X2	B2	D2		
ALR	RR	1E	R1	R2				
AP	SS	FA	L1	L2	B1	D1	B2	D2
AR	RR	1A	R1	R2				
AU	RX	7E	R1	X2	B2	D2		(short operands)
AUR	RR	3E	R1	R2				(short operands)
AW	RX	6E	R1	X2	B2	D2		(long operands)
AWR	RR	2E	R1	R2				(long operands)

JUNE 1, 1971

BAL	RX	45	R1	X2	B2	D2	
BALR	RR	05	R1	R2			
BC	RX	47	M1	X2	B2	D2	
BCR	RR	07	M1	R2			
BCT	RX	46	R1	X2	B2	D2	
BCTR	RR	06	R1	R2			
BXH	RS	86	R1	R3	B2	D2	
BXLE	RS	87	R1	R3	B2	D2	
C	RX	59	R1	X2	B2	D2	
CD	RX	69	R1	X2	B2	D2	(long operands)
CDR	RR	29	R1	R2			(long operands)
CE	RX	79	R1	X2	B2	D2	(short operands)
CER	RR	39	R1	R2			(short operands)
CH	RX	49	R1	X2	B2	D2	
CL	RX	55	R1	X2	B2	D2	
CLC	SS	D5	L	B1	D1	B2	D2

JUNE 1, 1971

CLI	SI	95	I2	B1	D1		
CLR	RR	15	R1	R2			
CP	SS	F9	L1	L2	B1	D1	B2 D2
CR	RR	19	R1	R2			
CSS	RR	02	R1	09			
CVB	RX	4F	R1	X2	B2	D2	
CVD	RX	4E	R1	X2	B2	D2	
CVWL	RR	03	/////	09			
D	RX	5D	R1	X2	B2	D2	
DD	RX	6D	R1	X2	B2	D2	(long operands)
DDR	RR	2D	R1	R2			(long operands)
DE	RX	7D	R1	X2	B2	D2	(short operands)
DER	RR	3D	R1	R2			(short operands)
DLY	RR	0B	N				
DP	SS	FD	L1	L2	B1	D1	B2 D2
DR	RR	1D	R1	R2			

JUNE 1, 1971

ED	SS	DE	L	B1	D1	B2	D2	
EDMK	SS	DF	L	B1	D1	B2	D2	
EX	RX	44	R1	X2	B2	D2		
HDR	RR	24	R1	R2				(long operands)
HER	RR	34	R1	R2				(short operands)
HIO	SI	9E	//////////	B1	D1			
IATR	RR	0E	R1	R2				
IC	RX	43	R1	X2	B2	D2		
ISK	RR	09	R1	R2				
L	RX	58	R1	X2	B2	D2		
LA	RX	41	R1	X2	B2	D2		
LC	RX	52	R1	X2	B2	D2		
LCDR	RR	23	R1	R2				(long operands)
LCER	RR	33	R1	R2				(short operands)
LCR	RR	13	R1	R2				
LD	RX	68	R1	X2	B2	D2		(long operands)

JUNE 1, 1971

LDR RR

28	R1	R2
----	----	----

 (long operands)

LE RX

78	R1	X2	B2	D2
----	----	----	----	----

 (short operands)

LER RR

38	R1	R2
----	----	----

 (short operands)

LH RX

48	R1	X2	B2	D2
----	----	----	----	----

LI RR

0C	R1	////
----	----	------

LM RS

98	R1	R3	B2	D2
----	----	----	----	----

LNDR RR

21	R1	R2
----	----	----

 (long operands)

LNER RR

31	R1	R2
----	----	----

 (short operands)

LNR RR

11	R1	R2
----	----	----

LPDR RR

20	R1	R2
----	----	----

 (long operands)

LPER RR

30	R1	R2
----	----	----

 (short operands)

LPR RR

10	R1	R2
----	----	----

LPSB SI

A1	////////	B1	D1
----	----------	----	----

LPSW SI

82	////////	B1	D1
----	----------	----	----

LR RR

18	R1	R2
----	----	----

LTDR RR

22	R1	R2
----	----	----

 (long operands)

JUNE 1, 1971

LTER	RR	32	R1	R2															(short operands)	
LTR	RR	12	R1	R2																
M	RX	5C	R1	X2	B2	D2														
MD	RX	6C	R1	X2	B2	D2														(long operands)
MDR	RR	2C	R1	R2																(long operands)
ME	RX	7C	R1	X2	B2	D2														(short operands)
MER	RR	3C	R1	R2																(short operands)
MH	RX	4C	R1	X2	B2	D2														
MP	SS	FC	L1	L2	B1	D1	B2	D2												
MR	RR	1C	R1	R2																
MVC	SS	D2	L		B1	D1	B2	D2												
MVI	SI	92	I2		B1	D1														
MVN	SS	D1	L		B1	D1	B2	D2												
MVO	SS	F1	L1	L2	B1	D1	B2	D2												
MVW	SS	D8	L		B1	D1	B2	D2												
MVZ	SS	D3	L		B1	D1	B2	D2												

JUNE 1, 1971

N	RX	54	R1	X2	B2	D2		
NC	SS	D4	L	B1	D1	B2	D2	
NI	SI	94	I2	B1	D1			
NR	RR	14	R1	R2				
O	RX	56	R1	X2	B2	D2		
OC	SS	D6	L	B1	D1	B2	D2	
OI	SI	96	I2	B1	D1			
OR	RR	16	R1	R2				
PACK	SS	F2	L1	L2	B1	D1	B2	D2
RDD	SI	85	I2	B1	D1			
RPSB	RR	0F	////	////				
S	RX	5B	R1	X2	B2	D2		
SATR	RR	0D	R1	R2				
SCON	RR	01	R1	R2				
SD	RX	6B	R1	X2	B2	D2	(long operands)	
SDR	RR	2B	R1	R2	(long operands)			

JUNE 1, 1971

SE	RX	7B	R1	X2	B2	D2	(short operands)	
SER	RR	3B	R1	R2			(short operands)	
SH	RX	4B	R1	X2	B2	D2		
SIO	SI	9C	//////////		B1	D1		
SIOP	SI	9A	I2		B1	D1		
SL	RX	5F	R1	X2	B2	D2		
SLA	RS	8B	R1	/////	B2	D2		
SLDA	RS	8F	R1	/////	B2	D2		
SLDL	RS	8D	R1	/////	B2	D2		
SLL	RS	89	R1	/////	B2	D2		
SLR	RR	1F	R1	R2				
SP	SS	FB	L1	L2	B1	D1	B2	D2
SPCI	SI	9B	//////////		B1	D1		
SPM	RR	04	R1	/////				
SPSB	SI	A0	//////////		B1	D1		
SR	RR	1B	R1	R2				

JUNE 1, 1971

	SRA	RS	8A	R1	/////	B2	D2	
	SRDA	RS	8E	R1	/////	B2	D2	
	SRDL	RS	8C	R1	/////	B2	D2	
	SRL	RS	88	R1	/////	B2	D2	
	SSK	RR	08	R1	R2			
	SSM	SI	80	//////////	B1	D1		
	ST	RX	50	R1	X2	B2	D2	
	STC	RX	42	R1	X2	B2	D2	
	STD	RX	60	R1	X2	B2	D2	(long operands)
	STE	RX	70	R1	X2	B2	D2	(short operands)
	STH	RX	40	R1	X2	B2	D2	
	STM	RS	90	R1	R3	B2	D2	
	SU	RX	7F	R1	X2	B2	D2	(short operands)
	SUR	RR	3F	R1	R2			(short operands)
	SVC	RR	0A	R1	R2			
	SW	RX	6F	R1	X2	B2	D2	(long operands)

JUNE 1, 1971

SWR	RR	2F	R1	R2				(long operands)
TCH	SI	9F	////	B1	D1			
TIO	SI	9D	////	B1	D1			
TM	SI	91	I2	B1	D1			
TR	SS	DC	L	B1	D1	B2	D2	
TRT	SS	DD	L	B1	D1	B2	D2	
TS	SI	93	////	B1	D1			
UNPK	SS	F3	L1	L2	B1	D1	B2	D2
WRD	SI	84	I2	B1	D1			
X	RX	57	R1	X2	B2	D2		
XC	SS	D7	L	B1	D1	B2	D2	
XI	SI	97	I2	B1	D1			
XR	RR	17	R1	R2				
ZAP	SS	F8	L1	L2	B1	D1	B2	D2

APPENDIX B. INSTRUCTION USE EXAMPLES

The following examples illustrate the use of many System/360 instructions. Before studying one of these examples, the reader should first consult the instruction description in this manual for the particular instruction of interest to him. Note that each instruction description contains the System/360 assembly language mnemonic op code and symbolic operand designation as well as the machine instruction format.

For clarity and for ease in programming, each example in this section presents the instruction both as it is written in an assembly-language statement and as it appears when assembled in storage (hexadecimal machine format). As a rule, all numerical operands are written in hexadecimal format unless otherwise specified. Hexadecimal operands are shown converted into binary and/or decimal if such conversion helps to clarify the example for the reader. Storage addresses are also given in hexadecimal. In the assembly-language statements, registers, lengths, and masks are all presented in decimal, but displacements may be in hexadecimal or decimal. A hexadecimal displacement is indicated by X 'a number', where the number can range from 000-FFF₁₆. Immediate operands are normally shown in hexadecimal. Whenever the value in a register or storage location is referred to as "not significant," this value is replaced during the execution of the instruction.

When writing ss format instructions in System/360 assembly language, lengths are given as the total number of bytes in the field. This differs from the machine definition regarding lengths which states that the length is the number of bytes to be added to the field address to obtain the address of the last byte of the field. Thus the machine length is one less than the assembly-language length. The assembly program automatically subtracts one from the length specified when the instruction is assembled.

Branching

Branch On Condition (BC, BCR)

The BRANCH ON CONDITION instructions test the condition code to see whether a branch should or should not be taken. The branch is taken only if the condition code is as specified by a mask.

MASK VALUE	CONDITION CODE
8	0
4	1
2	2
1	3

For example, assume that an add (A, AR) operation has been performed and that you wish to branch to address 6050 if the sum is zero or less (condition code = 0 or 1). Also assume:

Register 10 contains 00 00 50 00
 Register 11 contains 00 00 10 00

The RX form of the instruction performs the required test (and branch, if necessary) when written as:

Machine Format					Assembler Format			
OP CODE	M1	X2	B2	D2	OP CODE	M1	D2	X2 B2
47	C	B	A	050	BC	12,X'50'	(11,10)	

A mask of 15 indicates a branch on any condition (an unconditional branch). A mask of zero indicates that no branch is to occur (a no-operation).

Branch and Link (BAL, BALR)

The BRANCH AND LINK instructions are commonly used to branch to a subroutine with the option of later returning to the main instruction sequence. For example, assume that you wish to branch to a subroutine at storage address 1160. Also assume:

The contents of register 2 are not significant
 Register 5 contains 00 00 11 50
 There is a BAL instruction at address 00 00 C6
 (PSW bits 40-63 will contain 00 00 CA after execution of BAL)

The format of the BAL instruction is:

Machine Format					Assembler Format			
OP CODE	R1	X2	B2	D2	OP CODE	R1	D2	X2 B2
45	2	0	5	010	BAL	2,X'10'	(0,5)	

After the instruction is executed:

Register 2 (bits 8-31) contains 00 00 CA
 PSW bits 40-63 contain 00 11 60

The programmer can return to the main instruction sequence at any time with a BRANCH ON CONDITION (BCR) instruction that specifies register 2 and a mask of 15₁₀, provided that register 2 has not meanwhile been disturbed.

The BALR instruction with the R₂ field equal to zero may be used to load a register for use as a base

register. For example, in the assembly language, the sequence of statements:

```
BALR    15,0
USING   *,15
```

tells the assembly program that register 15 is to be used as the base register in assembling this program segment and that when the program is executed, the address of the next sequential instruction following the BALR will be placed in the register. (The USING statement is an "assembler instruction" and is thus not a part of the object program.)

At any time the condition code may be preserved for future inspection with BALR R₁,0. Bits 2 and 3 of the register (R₁) contain the condition code.

Branch On Count (BCT, BCTR)

The BRANCH ON COUNT instructions are often used to execute a program loop for a specified number of times. For example, assume that the following represents some lines of coding in an assembly language program:

```

.
.
LUPE    AR      8,1
.
.
BACK    BCT     6,LUPE
.
.

```

where register 6 contains 00 00 00 03 and the address of LUPE is 6826. Also assume that register 10 contains 00 00 68 00.

The format of the BCT instruction is:

Machine Format					Assembler Format (alternate form to above)				
OP CODE	R1	X2	B2	D2	OP CODE	R1	D2	X2	B2
46	6	0	A	026	BCT	6,X'26'	(0,10)		

The effect of the coding shown above is to execute three times the loop defined by locations LUPE and BACK.

Branch On Index High (BXH)

The BRANCH ON INDEX HIGH instruction is an index-incrementing and loop-controlling instruction that causes a branch whenever the sum of an index value and an increment value is greater than some comparand. For example, assume that:

Register 4 contains 00 00 00 8A = 138₁₀ = the index
 Register 6 contains 00 00 00 02 = 2₁₀ = the increment
 Register 7 contains 00 00 00 AA = 170₁₀ = the comparand
 Register 10 contains 00 00 71 30 = the branch address

The format of the instruction is:

Machine Format					Assembler Format		
OP CODE	R1	R3	B2	D2	OP CODE	R1R3D2	B2
86	4	6	A	000	BXH	4,6,0	(10)

When the instruction is executed, first the contents of register 6 are added to register 4, second the sum is compared with the contents of register 7, and third the decision to branch or not to branch is made. After execution:

Register 4 contains 00 00 00 8C = 140₁₀
 Registers 6 and 7 are unchanged

Since the new value in register 4 is not greater than the value in register 7, the branch to address 7130 is not taken.

When the register used to contain the increment is odd, that register also becomes the comparand register. The following assembly-language subroutine illustrates how this feature may be used to search a table.

Table

ARG1	FUNCT1
ARG2	FUNCT2
ARG3	FUNCT3
ARG4	FUNCT4
ARG5	FUNCT5
ARG6	FUNCT6

2 bytes 2 bytes

Assume that:

Register 0 contains the search argument
 Register 1 contains the width of the table in bytes (00 00 00 04)
 Register 2 contains the length of the table in bytes (00 00 00 18)
 Register 3 contains the starting address of the table
 Register 14 contains the return address in the main program

As the following subroutine is executed, the argument in register 0 is successively compared with the arguments in the table. If an equality is found, the corresponding function replaces the argument in register 0. If an equality is not found, FF₁₆ replaces the argument in register 0.

```

SEARCH   LNR      1,1
NOTEQUAL BXH     2,1,LOOP
NOTFOUND LA      0,X'FF'
         BCR     15,14
LOOP     CH      0,0(2,3)
         BC      7,NOTEQUAL
         LH     0,2(2,3)
         BCR     15,14

```

Branch On Index Low or Equal (BXLE)

This instruction is similar to BRANCH ON INDEX HIGH except that the branch is successful when the sum is low or equal compared to the comparand.

Execute (EX)

The EXECUTE instruction causes one instruction in main storage to be executed out of sequence without actual branching to the object instruction. For example, as-

sume that a MOVE (SI) instruction is located at address 3820, with format as follows:

Machine Format				Assembler Format		
OP CODE	I ₂	B ₁	D ₁	OP CODE	D ₁	B ₁ I ₂
92	66	C	003	MVI	3(12),X'	66
0	7 8	15 16	19 20	31		

where register 12 contains 00 00 89 16.

Further assume that at storage address 5000, the following EXECUTE instruction is located:

Machine Format					Assembler Format		
OP CODE	R ₁	X ₂	B ₂	D ₂	OP CODE	R ₁ D ₂ X ₂	B ₂
44	1	0	A	000	EX	1,0(0,10)	
0	7 8	11 12	15 16	19 20	31		

where register 10 contains 00 00 38 20, and register 1 contains 00 0F F0 99.

When the instruction at 5000 is executed, bits 24-31 of register 1 are or'ed inside the CPU with bits 8-15 of the instruction at 3820:

Bits 8-15: 0110 0110₂ = 66
 Bits 24-31: 1001 1001₂ = 99

Result: 1111 1111₂ = FF

causing the instruction at 3820 to be executed as if it originally were:

Machine Format				Assembler Format		
OP CODE	I ₂	B ₁	D ₁	OP CODE	D ₁	B ₁ I ₂
92	FF	C	003	MVI	3(12),X'	FF
0	7 8	15 16	19 20	31		

However, after execution:

Register 1 is unchanged
 The instruction at 3820 is unchanged
 Storage location 8919 contains FF
 The CPU next executes the instruction at address 5004
 (PSW bits 40-63 contain 00 50 04)

Fixed-Point Arithmetic

Load (L, LR)

The LOAD instructions place, unchanged, the contents of a word in storage or of a register into another register. For example, assume that the four bytes starting with location 21004 (a full-word boundary) are to be loaded into register 10. Initially:

Register 5 contains 00 02 00 00
 Register 6 contains 00 00 10 04
 The contents of register 10 are not significant
 Storage locations 21004-21007 contain 00 00 AB CD

To load register 10, the RX form of the instruction can be used:

Machine Format					Assembler Format		
OP CODE	R ₁	X ₂	B ₂	D ₂	OP CODE	R ₁ D ₂ X ₂	B ₂
58	A	5	6	000	L	10,0(5,6)	

After the instruction is executed, register 10 contains 00 00 AB CD.

Load Halfword (LH)

The LOAD HALFWORD instruction places unchanged the contents of a halfword in storage into the right half of

a register. The left half of the register is replaced by zeros or ones to reflect the sign (leftmost bit) of the halfword.

For example, assume that the two bytes in storage locations 1802-1803 are to be loaded into register 6. Also assume:

Register 6 contains 7F 12 34 56
 Register 14 contains 00 00 18 02
 Locations 1802-1803 contain 00 20

The instruction required to load the register is:

Machine Format					Assembler Format		
OP CODE	R ₁	X ₂	B ₂	D ₂	OP CODE	R ₁ D ₂ X ₂	B ₂
48	6	0	E	000	LH	6,0(0,14)	

After the instruction is executed, register 6 contains 00 00 00 20. If 1802-1803 contained a negative number, for example A7 B6, the sign bit would again be propagated to the left, giving FF FF A7 B6 as the final result in register 6.

Add Halfword (AH)

The ADD HALFWORD instruction algebraically adds the halfword contents of a storage location to the contents of a register. The halfword storage operand is expanded to 32 bits after it is fetched and before it is used in the add operation. The expansion consists of propagating the leftmost (sign) bit 16 positions to the left. For example, assume that the contents of storage locations 2000-2001 are to be added to register 5. Initially:

Register 5 contains 00 00 00 19 = 25₁₀
 Storage locations 2000-2001 contain FF FE = -2₁₀
 Register 12 contains 00 00 18 00
 Register 13 contains 00 00 01 50

The format of the required instruction is:

Machine Format					Assembler Format			
OP CODE	R ₁	X ₂	B ₂	D ₂	OP CODE	R ₁	D ₂	X ₂ B ₂
4A	5	D	C	6B0	AH	5,X'	6B0'	(13,12)

After the instruction is executed, register 5 contains 00 00 00 17 = 23₁₀

Compare Halfword (CH)

The COMPARE HALFWORD instruction compares a halfword in storage with the contents of a register. For example, assume that:

Register 4 contains FF FF 80 00 = -32,768₁₀
 Register 13 contains 00 01 60 50
 Storage locations 16080-16081 contain 8000 = -32,768₁₀

When the instruction:

Machine Format					Assembler Format			
OP CODE	R ₁	X ₂	B ₂	D ₂	OP CODE	R ₁	D ₂	X ₂ B ₂
49	4	0	D	030	CH	4,X'	30'	(0,13)

is executed, the contents of locations 16080-16081 are fetched, expanded to 32 bits (the sign bit is propagated to the left), and compared with the contents

of register 4. Because the two numbers are equal, the condition code is set to 0.

Multiply (M, MR)

Assume that a number in register 5 is to be multiplied by the contents of a word at address 3750. Initially:

The contents of register 4 are not significant
 Register 5 contains 00 00 00 9A = 154₁₀ = the multiplicand
 Register 11 contains 00 00 30 00
 Register 12 contains 00 00 06 00
 Storage locations 3750-3753 contain 00 00 00 83 = 131₁₀ = the multiplier

The instruction required to perform the multiplication is:

Machine Format					Assembler Format				
OP CODE	R1	X2	B2	D2	OP CODE	R1	D2	X2	B2
5C	4	B	C	150	M	4	X'150'	(11,12)	

After the instruction is executed:

Register 4 contains 00 00 00 00
 Register 5 contains 00 00 4E CE = 20,174₁₀ } product
 Storage locations 3750-3753 are unchanged

The RR format of the instruction can be used to square a number in a register. Assume that register 7 contains 00 00 00 10 = 16₁₀. The instruction:

Machine Format			Assembler Format		
OP CODE	R1	R2	OP CODE	R1R2	
1C	6	7	MR	6,7	

multiplies the number in register 7 by itself. The product, 00 00 00 00 00 01 00 = 256₁₀, appears in registers 6 and 7.

Multiply Halfword (MH)

The MULTIPLY HALFWORD instruction is used to multiply a register by a halfword in storage. For example, assume that:

Register 11 contains 00 00 00 15 = 21₁₀ = the multiplicand
 Register 14 contains 00 00 01 00
 Register 15 contains 00 00 20 00
 Storage locations 2102-2103 contain FF D9 = -39₁₀ = the multiplier

The instruction:

Machine Format					Assembler Format				
OP CODE	R1	X2	B2	D2	OP CODE	R1	D2	X2	B2
4C	B	E	F	002	MH	11,2	(14,15)		

multiplies the two numbers. The product, FF FF FC CD = -819₁₀, replaces the original contents of register 11.

Only the low-order 32 bits of a product are stored in a register; any higher-order bits are lost. No program interruption occurs upon overflow.

Divide (D, DR)

The DIVIDE instruction divides a dividend in an even/odd register pair by a divisor in a register or in stor-

age. Since the dividend is assumed to be 64 bits long, it is important that the proper sign is first affixed. For example, assume that:

Storage locations 3550-3553 contain 00 00 08 D7 = 2270₁₀ = the dividend
 Storage locations 3554-3557 contain 00 00 00 32 = 50₁₀ = the divisor

Register 6 does not contain all zeros
 The initial contents of register 7 are not significant
 Register 8 contains 00 00 35 50

The following assembly language statements load the registers properly and perform the divide operation:

		COMMENTS
L	6,0(0,8)	Places 00 00 08 D7 into register 6.
SRDA	6,32(0)	Shifts 00 00 08 D7 into register 7. Register 6 is filled with zeros (sign bits).
D	6,4(0,8)	Performs the division.

The machine format of the above DIVIDE instruction is:

OP CODE	R1	X2	B2	D2
5D	6	0	8	004

After all the above instructions are executed:

Register 6 contains 00 00 00 14 = 20₁₀ = the remainder
 Register 7 contains 00 00 00 2D = 45₁₀ = the quotient

Note that if the dividend had not been first placed in register 6 and shifted into register 7, register 6 would not have been filled with the proper sign bits (zeros in this example) and the DIVIDE instruction would not have given the expected results.

Convert to Binary (CVB)

The CONVERT TO BINARY instruction converts an eight-byte, signed, packed-decimal number into a signed binary number and loads the result into a general register. After the conversion operation is completed, the number is in the proper form for use as an operand in fixed-point arithmetic. For example, assume:

Storage locations 7608-760F contain 00 00 00 00 25 59 4C, a positive packed-decimal number
 The contents of register 7 are not significant
 Register 13 contains 00 00 76 00

The format of the conversion instruction is:

Machine Format					Assembler Format				
OP CODE	R1	X2	B2	D2	OP CODE	R1	D2	X2	B2
4F	7	0	D	008	CVB	7,8	(0,13)		

After the instruction is executed, register 7 contains 00 00 63 FA = + 25, 594₁₀.

Convert to Decimal (CVD)

The CONVERT TO DECIMAL instruction performs functions exactly opposite to those of the CONVERT TO BINARY instruction. CVD converts a binary number in a

register to packed decimal and stores the result in a double word. For example, assume:

Register 1 contains 00 00 0F 0F = 3855₁₀
 Register 13 contains 00 00 76 00
 PSW bit 12 = 0 (EBCDIC mode)

The format of the conversion instruction is:

Machine Format					Assembler Format		
OP CODE	R ₁	X ₂	B ₂	D ₂	OP CODE	R ₁ D ₂ X ₂	B ₂
4E	1	0	D	008	CVD	1,8(0,13)	

After the instruction is executed, location 7608-760F contain 00 00 00 00 00 03 85 5+. The plus sign generated is the standard EBCDIC plus sign, 1100₂.

Shift Left Single (SLA)

Because the sign bit remains unchanged during an SLA operation, this instruction performs an algebraic shift. For example, if the contents of register 2 are:

00 7F 0A 72 = 0000 0000 0111 1111 0000 1010 0111 0010₂

the instruction:

Machine Format					Assembler Format		
OP CODE	R ₁	R ₃	B ₂	D ₂	OP CODE	R ₁ D ₂ B ₂	
8B	2		0	008	SLA	2,8(0)	

results in register 2 being shifted left 8 places so that its new contents are:

7F 0A 72 00 = 0111 1111 0000 1010 0111 0010 0000 0000₂

If a left shift of 9 places had been specified, a significant bit would have been shifted out of position 1, and a fixed-point overflow interruption might have occurred (unless PSW bit 36 equaled 0).

Note that register 0 does not participate in the operation and that the contents of the R₃ field are ignored.

Shift Left Double (SLDA)

The SHIFT LEFT DOUBLE instruction is similar to SHIFT LEFT SINGLE except that SLDA shifts the 63 bits (not including the sign) of an even/odd register pair. The R₁ field of this instruction must be even. For example, if the contents of registers 2 and 3 are:

00 7F 0A 72 FE DC BA 98 =
 0000 0000 0111 1111 0000 1010 0111 0010 1111 1110 1101
 1100 1011 1010 1001 1000₂

the instruction:

Machine Format					Assembler Format		
OP CODE	R ₁	R ₃	B ₂	D ₂	OP CODE	R ₁ D ₂ B ₂	
8F	2		0	01F	SLDA	2,31(0)	

results in registers 2 and 3 both being left-shifted 31 places, so that their new contents are:

7F 6E 5D 4C 00 00 00 00 =
 0111 1111 0110 1110 0101 1101 0100 1100 0000 0000 0000
 0000 0000 0000 0000 0000₂

In this case, a significant bit is shifted out of position 1, and a fixed-point overflow interruption occurs (unless PSW bit 36 equals 0).

Store Multiple (STM)

Assume that the contents of general registers 14, 15, 0, and 1 are to be stored in consecutive words starting with storage location 4050 and that:

Register 14 contains 00 00 25 63
 Register 15 contains 00 01 27 36
 Register 0 contains 12 43 00 62
 Register 1 contains 73 26 12 57
 Register 6 contains 00 00 40 00
 The initial contents of locations 4050-405F are not significant

The STORE MULTIPLE instruction allows the use of just one instruction to store the contents of the four registers when it is written as:

Machine Format					Assembler Format				
OP CODE	R ₁	R ₃	B ₂	D ₂	OP CODE	R ₁	R ₃	D ₂	B ₂
90	E	1	6	050	STM	14,1,X'50'	(6)		

After the instruction is executed:

Locations 4050-4053 contain 00 00 25 63
 Locations 4054-4057 contain 00 01 27 36
 Locations 4058-405B contain 12 43 00 62
 Locations 405C-405F contain 73 26 12 57

Logical Operations

Move (MVI, MVC)

Move Immediate (MVI)

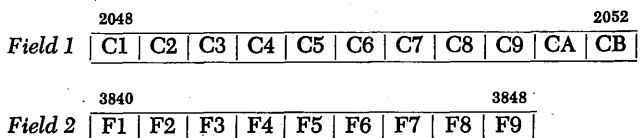
The MOVE IMMEDIATE instruction can place one byte of information from the instruction stream into any designated location in storage. For example, if the instruction:

Machine Format				Assembler Format		
OP CODE	I ₂	B ₁	D ₁	OP CODE	D ₁	B ₁ I ₂
92	FA	0	055	MVI	85(0)	X'FA'

is executed, bits 8-15 of the instruction (1111 1010₂) are copied in storage location 85₁₀.

Move Characters (MVC)

The MVC instruction can be used to move a data field from one location in storage to another. For example, assume that the following two fields are in storage:



Also assume:

Register 1 contains 00 00 20 48
 Register 2 contains 00 00 38 40

With the following instruction the first eight bytes of field 2 replace the first eight bytes of field 1:

Machine Format

OP CODE	L	B1	D1	B2	D2
D2	07	1	000	2	000

Assembler Format

OP CODE	D1	L	B1	D2	B2
MVC	0(8,1)	0(2)			

After the instruction is executed, field 1 becomes:

2048	2052
Field 1	F1 F2 F3 F4 F5 F6 F7 F8 C9 CA CB

Field 2 is unchanged.

As indicated in the programming note in the MOVE instruction description, MVC can be used to propagate one character through a field by starting the first operand field one byte to the right of the second operand field. For example, suppose that an area in storage starting with address 358 contains the following data:

358	360
	00 F1 F2 F3 F4 F5 F6 F7 F8

With the following MVC instruction, the zeros in location 358 can be propagated throughout the entire field (assume that register 11 contains 00 00 03 58):

Machine Format

OP CODE	L	B1	D1	B2	D2
D2	07	B	001	B	000

Assembler Format

OP CODE	D1	L	B1	D2	B2
MVC	1(8,11)	0(11)			

Because the MVC handles one byte at a time, the above instruction essentially takes the byte at address 358 and stores it at 359 (359 now contains 00), takes the byte at 359 and stores it at 35A, etc., until the entire field is filled with zeros. Note that an MVI instruction could have originally been used to place the byte of zeros in location 358.

NOTES:

1. Although the field occupying locations 358-360 contains nine bytes, the length coded in the assembler format is equal to the number of moves (one less than the field length).
2. The order of operands is important even though only one field is involved.

Move Numerics (MVN)

To illustrate the operation of the MOVE NUMERICS instruction, assume that the following two fields are in storage:

7090	7097
Field 1	C1 C2 C3 C4 C5 C6 C7 C8

7041	7049
Field 2	F0 F1 F2 F3 F4 F5 F6 F7 F8

Also assume:

Register 14 contains 00 00 70 90
Register 15 contains 00 00 70 40

After the instruction:

Machine Format

OP CODE	L	B1	D1	B2	D2
D1	03	F	001	E	000

Assembler Format

OP CODE	D1	L	B1	D2	B2
MVN	1(4,15)	0(14)			

is executed, field 2 becomes:

7041	7049
	F1 F2 F3 F4 F4 F5 F6 F7 F8

The numeric portions of locations 7090-7093 have been stored in the numeric portions of locations 7041-7044. The contents of locations 7090-7097 and 7045-7049 are unchanged.

Move Zones (MVZ)

The MOVE ZONES instruction, similar to MVC and MVN, can operate on overlapping or nonoverlapping fields. (See the examples for MVC and MVN.) When operating on nonoverlapping fields, MVZ works similar to the MVN instruction in the previous example, except that the MVZ moves the high-order four bits of each byte. To illustrate the use of MVZ with overlapping fields, assume that the following data field is in storage:

800	805
	F1 C2 F3 C4 F5 C6

Also assume that register 15 contains 00 00 08 00. The instruction:

Machine Format

OP CODE	L	B1	D1	B2	D2
D3	04	F	001	F	000

Assembler Format

OP CODE	D1	L	B1	D2	B2
MVZ	1(5,15)	0(15)			

propagates the zone from the byte at address 800 through the data field, so that the field becomes:

800	805
	F1 F2 F3 F4 F5 F6

Compare Logical (CL, CLR, CLI, CLC)

The COMPARE LOGICAL instructions differ from the algebraic compare instructions (C, CR) in that all quantities are handled as if unsigned.

Compare Logical Registers (CLR)

Assume that:

Register 1 contains 00 00 00 01
 Register 2 contains FF FF FF FF

Execution of the instruction:

Machine Format			Assembler Format	
OP CODE	R1	R2	OP CODE	R1R2
15	1	2	CLR	1,2

sets the condition code to 1. A condition code of 1 indicates that the first operand is lower than the second. However, if an *algebraic* compare instruction had been executed, the condition code would have been set to 2, indicating that the first operand is higher. During algebraic comparison, the contents of register 1 are interpreted as +1 and the contents of register 2 as -1. During logical comparison, the leftmost byte of register 2 is compared with the leftmost byte of register 1; each byte is interpreted as a binary number. In this case:

Leftmost byte of register 1: 0000 0000₂ = 0₁₀
 Leftmost byte of register 2: 1111 1111₂ = 255₁₀

If the two leftmost bytes are equal, the next two bytes will be compared, etc., until either an inequality is discovered or the contents of the registers are exhausted.

Compare Logical Immediate (CLI)

The CLI instruction logically compares a byte from the instruction stream with a byte from storage. For example, assume that:

Register 10 contains 00 00 17 00
 Storage location 1703 contains 7E

Execution of the instruction:

Machine Format				Assembler Format		
OP CODE	I2	B1	D1	OP CODE	D1	I2
95	AF	A	003	CLI	3(10),X'AF'	

sets the condition code to 1, indicating that the first operand (the quantity in main storage) is lower than the second (immediate) operand.

Compare Logical Characters (CLC)

The COMPARE LOGICAL CHARACTERS instruction can be used to perform the logical comparison of storage fields up to 256 bytes in length. For example, assume that the following two fields of data are in storage:

Field 1

1886	1891
D1 D6 C8 D5 E2 D6 D5 6B C1 4B C2 4B	

Field 2

1900	190B
D1 D6 C8 D5 E2 D6 D5 6B C1 4B C3 4B	

Also assume:

Register 6 contains 00 00 18 80
 Register 7 contains 00 00 19 00.

Execution of the instruction:

Machine Format					
OP CODE	L	B1	D1	B2	D2
D5	0B	6	006	7	000

Assembler Format					
OP CODE	D1	L	B1	D2	B2
CLC	6(12,6),0(7)				

sets the condition code to 1, indicating that field 1 is lower than field 2.

Because CLC compares bytes on an unsigned binary basis, the instruction can be used to collate fields composed of characters from the EBCDIC code. For example, in EBCDIC, the above two data fields are:

Field 1 JOHNSON,A.B.
 Field 2 JOHNSON,A.C.

The condition code of 1 tells us that A. B. Johnson precedes A. C. Johnson, thus placing the names in the correct alphabetic order.

AND (N, NR, NI, NC)

When the Boolean operator AND is applied to two bits, the result is 1 when both bits are 1; otherwise, the result is 0. When two bytes are AND'ed in System/360, each pair of bits is handled separately; there is no connection from one bit position to another.

AND (NI)

A frequent use of the AND instruction is to set a particular bit to zero. For example, assume that storage location 4891 contains 0100 0011₂. To set the eighth (rightmost) bit of this byte to 0 without affecting the other bits, the following instruction can be used (assume that register 8 contains 00 00 48 90):

Machine Format				Assembler Format			
OP CODE	I2	B1	D1	OP CODE	D1	B1	I2
94	FE	8	001	NI	1(8),X'FE'		

When this instruction is executed, the byte in storage is AND'ed with the immediate byte:

Location 4891: 0100 0011₂
 Immediate byte: 1111 1110₂
 Result: 0100 0010₂

The resulting byte with bit seven set to 0 is stored in location 4891. The condition code is set to 1.

OR (O, OR, OI, OC)

When the Boolean operator OR is applied to two bits, the result is 1 when either bit is 1; otherwise, the result is 0. When two bytes are OR'ed in System/360, each pair of bits is handled separately; there is no connection from one bit position to another.

OR (OI)

A frequent use of the OR instruction is to set a particular bit to 1. For example, assume that storage location 4891 contains 0100 0010₂. To set the eighth (rightmost) bit of this byte to 1 without affecting the other bits, the following instruction can be used (assume that register 8 contains 00 00 48 90):

Machine Format				Assembler Format			
OP CODE	I ₂	B ₁	D ₁	OP CODE	D ₁	B ₁	I ₂
96	01	8	001	OI	1(8),X'01'		

When this instruction is executed, the byte in storage is or'ed with the immediate byte:

Location 4891: 0100 0010₂
 Immediate byte: 0000 0001₂
 Result: 0100 0011₂

The resulting byte with bit seven set to 1 is stored in location 4891. The condition code is set to 1.

Exclusive OR (X, XR, XI, XC)

When the Boolean operator exclusive OR is applied to two bits, the result is 1 when one, and only one, of the two bits is 1; otherwise, the result is 0. When two bytes are exclusive or'ed in System/360, each pair of bits is handled separately; there is no connection from one bit position to another.

Exclusive OR (XI)

A frequent use of the EXCLUSIVE OR (XI) instruction is to invert a bit (change a 0 bit to a 1 or a 1 bit to a 0). For example, assume that storage location 8082 contains 0110 1001₂. To set the leftmost bit to 1 and the rightmost bit to 0 without affecting any of the other bits, the following instruction can be used (assume that register 9 contains 00 00 80 80):

Machine Format				Assembler Format			
OP CODE	I ₂	B ₁	D ₁	OP CODE	D ₁	B ₁	I ₂
97	81	9	002	XI	2(9),X'81'		

When the instruction is executed, the byte in storage is exclusive or'ed with the immediate byte:

Location 8082: 0110 1001₂
 Immediate byte: 1000 0001₂
 Result: 1110 1000₂

The resulting byte with the leftmost and rightmost bits inverted is stored in location 8082. The condition code is set to 1.

Exclusive OR (XC)

The EXCLUSIVE OR (XC) instruction can be used to exchange the contents of two areas in storage without the use of an intermediate storage area. For example, assume that two words are in storage:

	358		35B
Word 1	00	00	17 90
	360		363
Word 2	00	00	14 01

Execution of the instruction (assume that register 7 contains 00 00 03 58):

Machine Format					
OP CODE	L	B ₁	D ₁	B ₂	D ₂
D7	03	7	000	7	008

Assembler Format					
OP CODE	D ₁	L	B ₁	D ₂	B ₂
XC	0(4,7),8(7)				

exclusive OR's word 1 with word 2 as follows:

Word 1: 0000 0000 0000 0000 0001 0111 1001 0000₂ = 00 00 17 90
 Word 2: 0000 0000 0000 0000 0001 0100 0000 0001₂ = 00 00 14 01
 Result: 0000 0000 0000 0000 0000 0011 1001 0001₂ = 00 00 03 91

The result replaces the former contents of word 1.

Now, execution of the instruction:

Machine Format					
OP CODE	L	B ₁	D ₁	B ₂	D ₂
D7	03	7	008	7	000

Assembler Format					
OP CODE	D ₁	L	B ₁	D ₂	B ₂
XC	8(4,7),0(7)				

produces the following result:

Word 1: 0000 0000 0000 0000 0000 0011 1001 0001₂ = 00 00 03 91
 Word 2: 0000 0000 0000 0000 0001 0100 0000 0001₂ = 00 00 14 01
 Result: 0000 0000 0000 0000 0001 0111 1001 0000₂ = 00 00 17 90

The result of this operation replaces the former contents of word 2. Word 2 now contains the original value of word 1.

Lastly, execution of the instruction:

Machine Format					
OP CODE	L	B ₁	D ₁	B ₂	D ₂
D7	03	7	000	7	008

Assembler Format					
OP CODE	D ₁	L	B ₁	D ₂	B ₂
XC	0(4,7),8(7)				

produces the following result:

Word 1: 0000 0000 0000 0000 0000 0011 1001 0001₂ = 00 00 03 91
 Word 2: 0000 0000 0000 0000 0001 0111 1001 0000₂ = 00 00 17 90
 Result: 0000 0000 0000 0000 0001 0100 0000 0001₂ = 00 00 14 01

The result of this operation replaces the former contents of word 1. Word 1 now contains the original value of word 2.

NOTES:

1. With the xc instruction, fields up to 256 bytes in length can be exchanged.
2. With the xr instruction, the contents of two registers can be exchanged.
3. Because the x instruction operates storage to register only, an exchange cannot be made solely by the use of x.

Now, assume that starting at storage location 2100 there is a sequence of 20₁₀ EBCDIC characters to be translated to USASCII-8:

Locations 2100-2113: JOHNbJONESb257bW.b95

Also assume:

Register 12 contains 00 00 21 00
Register 15 contains 00 00 10 00

As the instruction:

Machine Format

OP CODE	L	B1	D1	B2	D2
DC	13	C	000	F	000

Assembler Format

OP CODE D1 L B1 D2 B2
TR 0(20,12),0(15)

is executed, the binary value of each argument byte is added to the starting address of the table, and the resulting address is used to fetch a function byte:

Table starting address: 1000
First argument byte (J): D1
Address of function byte: 10D1

Because the table is arranged so that every EBCDIC character is replaced by the corresponding USASCII-8 character, the result is:

Locations 2100-2113: JOHNbJONESb257bW.95

NOTE: To verify that this example is correct, find in Appendix F the hexadecimal values for the remaining EBCDIC characters and add them to the starting address of the table (1000). The sums should be the addresses within the table of the corresponding USASCII-8 characters.

Translate and Test (TRT)

The TRANSLATE AND TEST instruction is used to scan a data field (the argument bytes) for characters with special meaning. To indicate which characters have special meaning, first set up a table similar to the one used for the TRANSLATE instruction. (See the preceding example.) Once again the table must be in order by the binary sequence of the code of the argument bytes. This time, however, put zeros in the table to indicate characters without any special meaning and nonzero values to indicate characters with special meaning.

This example deals with EBCDIC characters; the characters with special meaning in the argument field are a selected set of punctuation marks. The translate and test table that follows has been set up accordingly.

Now, assume that starting at storage location 3000 you have the following sequence of 30₁₀ EBCDIC characters:

Locations 3000-301D:
bbbbUNPKbbbbPROUT(9),WORD(5)

Translate and Test Table

	200F															
2000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2040	00	00	00	00	00	00	00	00	00	00	00	00	10	20	25	00
2050	90	00	00	00	00	00	00	00	00	00	00	30	35	40	45	00
2060	80	85	00	00	00	00	00	00	00	00	00	50	55	00	00	00
2070	00	00	00	00	00	00	00	00	00	00	00	60	65	70	75	00
2080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Note: If the character codes in the statement being translated occupy a range smaller than 00₁₆ through FF₁₆, a table less than 256 bytes can be used.

Also assume:

Register 1 contains 00 00 30 00
Register 2 contains 00 00 00 00
Register 15 contains 00 00 20 00

As the instruction:

Machine Format

OP CODE	L	B1	D1	B2	D2
DD	1D	1	000	F	000

Assembler Format

OP CODE D1 L B1 D2 B2
TRT 0(30,1),0(15)

is executed, the value of the first argument byte, a blank, is added to the starting address of the table to produce the address of the function byte to be examined:

Table starting address 2000
First argument byte (blank) 40
Address of function byte 2040

Because zeros were originally placed in storage location 2040, no special action occurs, and the operation continues with the second argument byte. The operation will thus continue until it reaches the symbol ((left parenthesis) in location 3013. When this symbol is reached, its value is added to the starting address of the table, as usual:

Table starting address 2000
Argument byte (left parenthesis) 4D
Address of function byte 204D

Because location 204D contains a nonzero value, the following actions occur:

1. The address of the argument byte, 003013, is placed in the low-order 24 bits of register 1.
2. The function byte, 20, is placed in the low-order eight bits of register 2.
3. The condition code is set to 1 (scan not completed).

In general, TRANSLATE AND TEST is executed by use of an EXECUTE instruction, which supplies the length specification from a general register. In this way a complete statement scan can be performed with a single TRANSLATE AND TEST instruction repeated over and over by means of EXECUTE. In the example, after the first execution of TRT, register 1 contains the address of the last argument byte translated. It is then a simple matter to subtract this address from the address of the last argument byte (301D) to produce a length specification. This length minus one is placed in the register that is referenced as the R₁ field of the EXECUTE instruction. (Because the length code in the machine format is one less than the total number of bytes in the field, one must be subtracted from the computed length.) The branch address part of the EXECUTE instruction points to the TRANSLATE AND TEST instruction, which must now appear in the following format:

Machine Format

OP CODE	L	B1	D1	B2	D2
DD	00	1	000	F	000

Assembler Format

OP CODE	D1	LB1	D2	B2
TRT	0(0,1),0(15)			

Now the entire argument field can be scanned, stopping to examine those characters of special interest, without having to modify any of the instructions already written. After a stop is made to examine a character, only a new length and starting address need be computed before continuing the scan.

Edit (ED)

Because the decimal feature instructions operate only on packed decimal data, it is necessary to convert the data to the zoned format before a legible report can be printed. Moreover, if the report is to be useful to a great many people, certain punctuation marks, such as commas and decimal points, should be inserted in appropriate places. The highly flexible EDIT instruction performs these two functions in a single execution.

This example shows step-by-step one way in which EDIT can be used. The field to be edited (the source) is four bytes long; it is edited against a pattern 13 bytes long. The following symbols are used:

SYMBOL	MEANING
b (hexadecimal 40)	blank character
((hexadecimal 21)	significance starter
d (hexadecimal 20)	digit selector

Assume that the source and pattern fields are:

Source

1200			1203
02	57	42	6C

↑

Pattern

1000			100C
40	20	20	6B
20	20	21	4B
20	20	40	C3
D9			

b d d , d d (. d d b C R

Execution of the instruction (assume that register 12 contains 00 00 10 00):

Machine Format

OP CODE	L	B1	D1	B2	D2
DE	0C	C	000	C	200

Assembler Format

OP CODE	D1	L	B1	D2	B2
ED	0(13,12),X'200'	(12)			

alters the pattern as follows:

PATTERN	DIGIT	SIGNIFICANCE INDICATOR		RULE	LOCATION 1000-100C
		before/after			
b		off/off		leave(1)	bdd,dd(.ddbCR
d	0	off/off		fill	bbd,dd(.ddbCR
d	2	off/on (2)		digit	bb2,dd(.ddbCR
,		on/on		leave	same
d	5	on/on		digit	bb2,5d(.ddbCR
d	7	on/on		digit	bb2,57(.ddbCR
(4	on/on		digit	bb2,574.ddbCR
.		on/on		leave	same
d	2	on/on		digit	bb2,574.2dbCR
d	6+	on/off (3)		digit	bb2,574.26bCR
b		off/off		fill	same
C		off/off		fill	bb2,574.26bbR
R		off/off		fill	bb2,574.26bbb

Notes:

1. This character becomes the fill character.
2. First nonzero decimal source digit turns on significance indicator.
3. Plus sign in the four low-order bits of the byte turns off significance indicator.

Thus, after the instruction is executed:

Source Unchanged

Pattern

1000			100C
40	40	F2	6B
F5	F7	F4	4B
F2	F6	40	40
40	40	40	

b b 2 , 5 7 4 2 6 b b b

Condition code = 2: result greater than zero.

When printed, the pattern field, which now contains the result, appears as:

2,574.26

If the number in the source field is changed to 00 00 02 6D, a negative number, and the original pattern is used, the edited result becomes:

Pattern

1000										100C			
40	40	40	40	40	40	40	40	4B	F2	F6	40	C3	D9
b b b b b b b							2 6		b C R				

Condition code = 1; result less than zero

The significance starter forces the significance indicator to the on state and hence causes the decimal point to be left unchanged. Because the minus sign does not change the significance indicator, the CR symbol is also preserved.

Edit and Mark (EDMK)

After an edit-and-mark operation, a symbol (such as a dollar sign) can be inserted at the appropriate position in the edited result. Usually a currency symbol is inserted to the immediate left of the first significant digit in the amount; however, if a decimal point appears in an amount less than one, the currency symbol must be inserted to the immediate left of the decimal point. A typical operation would leave no blank between the currency symbol and the amount, thus protecting against one form of alteration when the result is printed on a check.

If significance is not forced by the significance starter, the edit-and-mark operation inserts into general register I an address one more than the address at which a currency symbol would normally be inserted. After one is subtracted from the value in general register I (for example, by using a BRANCH ON COUNT instruction with R₁ set to one and R₂ set to zero), a MOVE instruction (MVI) may be used to position the symbol in main storage.

<i>Machine Format</i>				<i>Assembler Format</i>			
OP CODE	I ₂	B ₁	D ₁	OP CODE	D ₁	B ₁	I ₂
92	5D	1	100	MVI	0(1),	C	\$

If significance is forced, general register I remains unchanged. Therefore, the address of the character following the significance starter should be placed in that register before the EDIT AND MARK instruction is performed.

Decimal Arithmetic

Add Decimal (AP)

Assume that the signed, packed-decimal field at storage locations 500-503 is to be added to the signed, packed-decimal field at locations 2000-2002. Also assume:

- Register 12 contains 00 00 20 00
- Register 13 contains 00 00 04 FD
- Storage locations 2000-2002 contain 38 46 0D (a neg number)
- Storage locations 500-503 contain 01 12 34 5C (a pos number)

After the instruction:

Machine Format

OP CODE	L ₁	L ₂	B ₁	D ₁	B ₂	D ₂
FA	2	3	C	000	D	003

Assembler Format

OP CODE	D ₁	L ₁ B ₁	D ₂	L ₂ B ₂
AP	0(3,12),	3(4,13)		

is executed, the storage locations 2000-2002 contain 73 88 5C; the condition code is set to 2 to indicate that the sum is positive. Note that:

1. Although the second operand field is larger than the first operand field, no overflow interruption occurs because the result can be entirely contained within the first operand field.
2. Because the two numbers had different signs, they were in effect subtracted.

Zero and Add (ZAP)

Assume that the signed, packed-decimal field at storage locations 4500-4502 is to be moved to locations 4000-4004 with four leading zeros in the result field. Also assume:

- Register 9 contains 00 00 40 00
- Storage locations 4000-4004 contain 12 34 56 78 90
- Storage locations 4500-4502 contain 38 46 0D

After the instruction:

Machine Format

OP CODE	L ₁	L ₂	B ₁	D ₁	B ₂	D ₂
F8	4	2	9	000	9	500

Assembler Format

OP CODE	D ₁	L ₁ B ₁	D ₂	L ₂ B ₂
ZAP	0(5,9),	X'500'	(3,9)	

is executed, the storage locations 4000-4004 contain 00 00 38 46 0D; the condition code is set to 1 to indicate a negative result. Note that because the first operand is not checked for valid sign and digit codes, it may contain any combination of hexadecimal digits.

Compare Decimal (CP)

Assume that the signed, packed-decimal contents of storage locations 700-703 are to be algebraically compared with the signed, packed-decimal contents of locations 500-503. Also assume:

- Register 12 contains 00 00 06 00
- Register 13 contains 00 00 04 00
- Storage locations 700-703 contain 17 25 35 6D
- Storage locations 500-503 contain 06 72 14 2D

After the instruction:

Machine Format

OP CODE	L ₁	L ₂	B ₁	D ₁	B ₂	D ₂
F9	3	3	C	100	D	100

Assembler Format

OP CODE	D1	L1B1	D2	L2B2
CP	X'100'	(4,12)	X'100'	(4,13)

is executed, the condition code is set to 1, indicating that the first operand (the contents of locations 700-703) is lower than the second.

Multiply Decimal (MP)

Assume that the signed, packed-decimal field in storage locations 1202-1204 (the multiplicand) is to be multiplied by the signed, packed-decimal field in locations 500-501 (the multiplier):

Multiplicand

1202	1204
38	46 0D

Multiplier

500	501
32	1D

Because there are a total of eight significant digits in the multiplier and multiplicand, a field at least five bytes in length must be reserved for the signed result. As indicated in the programming note for MULTIPLY DECIMAL, a ZERO AND ADD into a larger field can provide the required space. If it is assumed:

Register 4 contains 00 00 12 00
Register 6 contains 00 00 05 00

then execution of the assembler instruction:

ZAP X'100'(5,4),2(3,4)

sets up a new multiplicand in storage locations 1300-1304:

Multiplicand (new)

1300	1304
00 00	38 46 0D

Now, after the instruction:

Machine Format

OP CODE	L1	L2	B1	D1	B2	D2
FC	4	1	4	100	6	000

Assembler Format

OP CODE	D1	L1B1	D2	L2B2
MP	X'100'	(5,4)	0(2,6)	

is executed, storage locations 1300-1304 contain the product: 01 23 45 66 0C.

Divide Decimal (DP)

Assume that the signed, packed-decimal field at storage locations 2000-2004 (the dividend) is to be divided by the signed, packed-decimal field at locations 3000-3001 (the divisor). Also assume:

Register 12 contains 00 00 20 00
Register 13 contains 00 00 30 00
Storage locations 2000-2004 contain 01 23 45 67 8C
Storage locations 3000-3001 contain 32 1D

After the instruction:

Machine Format

OP CODE	L1	L2	B1	D1	B2	D2
FD	4	1	C	000	D	000

Assembler Format

OP CODE	D1	L1B1	D2	L2B2
DP	0(5,12)	0(2,13)		

is executed, the dividend field is entirely replaced by the signed quotient and remainder fields, as follows:

Locations 2000-2004

2000	2004
38 46 0D	01 8C
quotient remainder	

NOTES:

1. Because the signs of the dividend and divisor are different, the quotient receives a negative sign.
2. The remainder receives the sign of the dividend and the length of the divisor.
3. If an attempt is made to divide the dividend by the one-byte field at location 3001, the quotient will be too long to fit within the four bytes allotted to it. A decimal-divide exception exists, causing a program interruption.

Pack (PACK)

Assume that storage locations 1000-1004 contain the following zoned-decimal field that is to be converted to a packed-decimal field and left in the same location:

Zoned Field

1000	1004
F1 F2 F3 F4	C5

Also assume that register 12 contains 00 00 10 00. After the instruction:

Machine Format

OP CODE	L1	L2	B1	D1	B2	D2
F2	4	4	C	000	C	000

Assembler Format

OP CODE	D1	L1B1	D2	L2B2
PACK	0(5,12)	0(5,12)		

is executed, the field in locations 1000-1004 is in the packed-decimal format:

Packed Field

1000	1004
00 00	12 34 5C

NOTES:

1. This example illustrates the operation of PACK when the first and second operand fields overlap completely.
2. During the operation, the second operand was extended with high-order zeros.

Unpack (UNPK)

Assume that storage locations 2501-2503 contain a signed, packed-decimal field that is to be unpacked and placed in storage locations 1000-1004. Also assume:

Register 12 contains 00 00 10 00
 Register 13 contains 00 00 25 00
 Storage locations 2501-2503 contain 12 34 5D
 The initial contents of storage locations 1000-1004 are not significant

PSW bit 12 = 0 (EBCDIC mode)

After the instruction:

Machine Format

OP CODE	L1	L2	B1	D1	B2	D2
F3	4	2	C	000	D	001

Assembler Format

OP CODE	D1	L1	B1	D2	L2	B2
UNPK	0(5,12)	1(3,13)				

is executed, the storage locations 1000-1004 contain F1 F2 F3 F4 D5. Because the CPU was in EBCDIC mode, the zone 1111₂ = F₁₆ was attached to all digits except the digit occupying the same byte as the sign.

Move with Offset (MVO)

Assume that the unsigned three-byte field in storage locations 4500-4502 is to be moved to locations 5600-5603 and given the sign of the one-byte field located at 5603. Also assume:

Register 12 contains 00 00 56 00
 Register 15 contains 00 00 45 00
 Storage locations 5600-5603 contain 77 88 99 0C
 Storage locations 4500-4502 contain 12 34 56

After the instruction:

Machine Format

OP CODE	L1	L2	B1	D1	B2	D2
F1	3	2	C	000	F	000

Assembler Format

OP CODE	D1	L1	B1	D2	L2	B2
MVO	0(4,12)	0(3,15)				

is executed, the storage locations 5600-5603 contain 01 23 45 6C. Note that the second operand was extended with one high-order zero to fill out the first operand field.

NOTE: The section "Shifting of Decimal Fields" shows how MOVE WITH OFFSET can be used in shifting a decimal field an odd number of places.

Shifting of Decimal Fields

No instructions have been specifically provided to perform shifting of decimal fields in storage. However, various combinations of System/360 instructions may be used to accomplish in effect this type of shift. The following assembly-language examples illustrate some of the methods for shifting decimal numbers. These

examples additionally illustrate how the assembly language facilitates coding with symbolic operands.

Decimal Right Shift (Even Number of Places)

Assume that symbolic storage location SOURCE contains 12 34 56 78 9C, and you wish to shift the contents of SOURCE two places to the right (to drop the rightmost two digits, thereby dividing SOURCE by 100₁₀). The MOVE NUMERIC (MVN) instruction can be used to accomplish this:

MVN SOURCE+3(1),SOURCE+4

After the MVN instruction is executed, SOURCE contains 12 34 56 7C.9C. Instructions referencing SOURCE should now use a length of 4 instead of 5.

Decimal Right Shift (Odd Number of Places)

Assume that symbolic storage location SOURCE contains 12 34 56 78 9C, and you wish to shift the contents of SOURCE three places to the right (to drop the rightmost three digits, thereby dividing SOURCE by 1000₁₀). The MOVE WITH OFFSET (MVO) instruction can be used to accomplish this:

MVO SOURCE(5),SOURCE(3)

After this instruction is executed, SOURCE contains 00 01 23 45 6C.

Decimal Left Shift (Even Number of Places)

Assume that symbolic location ZERO contains 00 00 and that SOURCE contains 12 34 56 78 9C. The contents of SOURCE can be shifted four places to the left by using the following group of instructions:

	SOURCE
MVC SOURCE+5(2),ZERO	12 34 56 78 9C 00 00
MVN SOURCE+6(1),SOURCE+4	12 34 56 78 9C 00 0C
NI SOURCE+4,240	12 34 56 78 90 00 0C

Note that the number 240₁₀ in the AND (NI) instruction provides a mask of 1111 0000₂, which is used to make the old sign position zero.

Decimal Left Shift (Odd Number of Places)

Assume that symbolic location ZERO contains 00 00 and that SOURCE contains 12 34 56 78 9C. The contents of SOURCE can be shifted three places to the left by using the following group of instructions:

	SOURCE
MVC SOURCE+5(2),ZERO	12 34 56 78 9C 00 00
MVN SOURCE+6(1),SOURCE+4	12 34 56 78 9C 00 0C
NI SOURCE+4,240	12 34 56 78 90 00 0C
MVO SOURCE(6),SOURCE(5)	01 23 45 67 89 00 0C

Floating-Point Arithmetic

In this section, the abbreviations FPR0, FPR2, FPR4, and FPR6 stand for floating-point registers 0, 2, 4, and 6, respectively.

Add Normalized (AE, AER, AD, ADR)

The ADD NORMALIZED instructions perform the addition of two floating-point numbers and place the normalized result in a floating-point register. Neither of the two numbers to be added must necessarily be normalized before addition occurs. For example, assume that:

FPR6 contains 43 08 21 00 00 00 00 00 = $82.11_8 \approx 130.06_{10}$
 (unnormalized)
 Storage locations 2000-2007 contain 41 12 34 56 00 00 00 00 = $1.23456_{16} \approx 1.13_{10}$ (normalized)
 Register 13 contains 00 00 20 00

The instruction:

Machine Format					Assembler Format		
OP CODE	R1	X2	B2	D2	OP CODE	R1D2	X2B2
7A	6	0	D	000	AE	6,0(0,13)	

can be used to perform the short-precision addition of the two operands. In this example the instruction operates as follows:

The characteristics of the two numbers are compared. Since the number in storage has a characteristic that is smaller by 2, it is right-shifted after fetching until the characteristics agree. The two numbers are then added:

			GUARD DIGIT
FPR6:	43 08 21 00		
Shifted number from storage:	43 00 12 34	5	
Intermediate sum:	43 08 33 34	5	

Because the intermediate sum is unnormalized, it is left-shifted to form the normalized floating-point number 42 83 33 45 ($=83.3345_{16} = 131.2_{10}$). This number replaces the high-order portion of FPR6. The low-order portion of FPR6 and the contents of storage locations 2000-2007 are unchanged.

If the long-precision instruction AD is used, the result in FPR6 will be 42 83 33 45 60 00 00 00. Note that in this case, the use of the long-precision instruction provides one additional hexadecimal digit of precision.

Add Unnormalized (AU, AUR, AW, AWR)

The ADD UNNORMALIZED instructions operate identically to the ADD NORMALIZED instructions, except that the final result is not normalized when ADD UNNORMALIZED is used. For example, using the same operands as in the example for ADD NORMALIZED, when the short-precision instruction:

Machine Format					Assembler Format		
OP CODE	R1	X2	B2	D2	OP CODE	R1D2	X2B2
7E	6	0	D	000	AU	6,0(0,13)	

is executed, the two numbers are added as follows:

			GUARD DIGIT
FPR6:	43 08 21 00		
Shifted number from storage:	43 00 12 34	5	
Sum:	43 08 33 34	5	

The guard digit participates in the addition but is discarded. The unnormalized sum replaces the high-order portion of FPR6.

If the result in FPR6 is converted to a normalized number (42 83 33 40 00 00 00 00) and is compared to the result in FPR6 when ADD NORMALIZED was used (42 83 33 45 00 00 00 00), in this case it is apparent that the use of ADD NORMALIZED (with the retention of the guard digit) has preserved some additional significance in the result.

Compare (CE, CER, CD, CDR)

Assume that FPR4 contains 43 00 00 00 00 00 00 00 (= 0), and FPR6 contains 34 12 34 56 78 9A BC DE (a positive number). The contents of the two registers are to be compared with the following long-precision instruction:

Machine Format			Assembler Format	
OP CODE	R1	R2	OP CODE	R1R2
29	4	6	CDR	4,6

When this instruction is executed, the number with the smaller characteristic is taken from the register and right-shifted until the two characteristics agree. The shifted contents of the FPR6 are:

		GUARD DIGIT
FPR6:	43 00 00 00 00 00 00 00	0

Therefore, when the two numbers are compared the condition code is set to 0, indicating an equality.

As the above example implies, when floating-point numbers are compared, more than two numbers may compare equally if one of the numbers is unnormalized. For example, the unnormalized floating-point number 41 00 12 34 56 78 9A BC compares equally with all numbers of the form 3F 12 34 56 78 9A BC 0X (X represents any hexadecimal number). When the COMPARE instruction is executed, the two low-order digits are shifted right two places; the 0 becomes the guard digit, and the X does not participate in the comparison.

Note, however, that when two normalized floating-point numbers are compared, the relationship between numbers that compare equally is unique: each digit in one number must be identical to the corresponding digit in the other number.

Status Switching

Supervisor Call (SVC)

The SUPERVISOR CALL instruction allows a program that is operating in the problem state to switch the CPU to the supervisor state. At the same time, the problem program can make a byte of information available to the supervisor program. For example, the instruction:

Machine Format

OP CODE	I
0A	01

Assembler Format

OP CODE	I
SVC	1

causes a supervisor-call interruption. The byte of information (0000 0001₂) is placed in the interruption-code field of the SUPERVISOR CALL old PSW (logical address 23₁₆), and a new PSW is fetched from logical address 60₁₆. The information byte may indicate, for example, that certain conditions encountered during processing require further attention (e.g., the job has been completed and a printout of storage is desired).

Set Storage Key (SSK)

Assume that the storage block corresponding to addresses 800-FFF is to have bits 11110 set into its storage key (that is, only programs with a protection key of 0 or 15₁₀ can store data in this block, but any program can fetch data). Also assume that:

Register 5 contains 00 00 0A 60
 Register 6 contains 00 00 00 F0

When the instruction:

Machine Format		
OP CODE	R1	R2
08	6	5

Assembler Format		
OP CODE	R1R2	
SSK	6,5	

is executed, bits 8-20 of register 5 are examined; their value indicates which block of 2,048₁₀ bytes is to have its key set:

Register 5 (bits 8-20): 0000 0000 0000 1

In this case register 5 indicates that the "second" block (addresses 800-FFF) is the block being addressed. Note that register 5 will contain all zeros if the block containing addresses 000-7FF is being addressed. Also note that it is not necessary for R₂ to contain the exact address of the first byte in the block (i.e., 00 00 08 00) because only bits 8-20 of R₂ are examined.

The key setting for the storage block indicated by register 5 is obtained from bits 24-28 of register 6:

Register 6 (bits 24-28) 1111 0

If the fetch protection feature is installed, and it is desired to prevent fetching as well as storing of data in locations 800-FFF, the low-order bit of the storage

key must be set to 1. This bit can be set to 1 if bit 28 of register 6 is set to 1 before the execution of ssk. (The register could contain 00 00 00 F8, for example.)

Insert Storage Key (ISK)

Assume that the key of the storage block containing addresses 800-FFF is to be inspected and that:

Register 5 contains 00 00 08 00
 Register 6 contains FF FF FF FF

Execution of the instruction:

Machine Format		
OP CODE	R1	R2
09	6	5

Assembler Format		
OP CODE	R1R2	
ISK	6,5	

changes the contents of register 6 to:

1111 1111 1111 1111 1111 1111 MMMM M000 where MMMMM represents the five-bit storage key. Note that the last M is set to 0 if fetch protection is not installed.

Test and Set (TS)

The TEST AND SET instruction can be used to control the sharing of a storage area that is used in common by more than one program. Assume that the convention has been established that when the leftmost bit of an indicator byte is set to 1, it is a signal to all other programs not to attempt to access the storage area. When a program has finished using the storage area, it can then set the leftmost bit of the indicator byte to 0, indicating that other programs may now access the area.

For example, assume that register 10 contains the address of the indicator byte (00 00 34 56) and that the indicator byte itself initially contains the bits 0000 0000. After the instruction:

Machine Format		
OP CODE	B1	D1
93	A	000

Assembler Format		
OP CODE	D1 B1	
TS	0(10)	

is executed:

The indicator byte (location 3456) contains bits 1111 1111. The condition code is set to zero (indicating that the test revealed the leftmost bit of the indicator byte was zero).

APPENDIX C. FIXED-POINT AND TWO'S-COMPLEMENT NOTATION

A fixed-point number is a signed value, recorded as a binary integer. It is called fixed point because the programmer determines the fixed positioning of the binary point.

Fixed-point operands may be recorded in halfword (16 bits) or word (32 bits) lengths. In both lengths, the first bit position (0) holds the sign of the number, with the remaining bit positions (1-15 for halfwords and 1-31 for fullwords) used to designate the magnitude of the number.

Positive fixed-point numbers are represented in true binary form with a zero sign bit. Negative fixed-point numbers are represented in two's complement notation with a one bit in the sign position. In all cases, the bits between the sign bit and the leftmost significant bit of the integer are the same as the sign bit (i.e. all zeros for positive numbers, all ones for negative numbers).

Negative fixed-point numbers are formed in two's complement notation by inverting each bit of the positive binary number and adding one. For example, the true binary form of the decimal value (plus 26) is made negative (minus 26) in the following manner:

	S	INTEGER	
+26	0	00000000 00011010	
Invert	1	11111111 11100101	
Add 1		<u> 1</u>	
-26	1	11111111 11100110	(Two's complement form)

This is equivalent to subtracting the number: 00000000 00011010 from 1 00000000 00000000.

The following addition examples illustrate two's complement arithmetic. Only eight bit positions are used. All negative numbers are in two's complement form.

		COMMENTS
+57	=	00111001
+35	=	00100011
+92	=	01011100
+57	=	90111001
-35	=	11011101
+22	=	00010110
		No overflow
		Ignore carry - carry into high order position and carry out.
+35	=	00100011
-57	=	11000111
-22	=	11101010
		Sign change only; no carry.
-57	=	11000111
-35	=	11011101
-92	=	10100100
		No overflow
		Ignore carry - carry into high order position and carry out.
-57	=	11000111
-92	=	10100100
-149	=	*01101011
		*Overflow - no carry into high order position but carry out.
+57	=	00111001
+92	=	01011100
149	=	*10010101
		*Overflow - carry into high order position, no carry out.

The following are 16-bit fixed-point numbers. The first is the largest positive number and the last, the largest negative number.

NUMBER	DECIMAL	S	INTEGER
2 ¹⁵ -1	= 32,767	=0	11111111 11111111
2 ⁰	= 1	=0	00000000 00000001
0	= 0	=0	00000000 00000000
-2 ⁰	= -1	=1	11111111 11111111
-2 ¹⁵	= -32,768	=1	00000000 00000000

The following are 32-bit fixed-point numbers. The first is the largest positive number that can be represented by 32 bits, and the last is the largest negative number.

NUMBER	DECIMAL	S	INTEGER
2 ³¹ -1	= 2 147 483 647	=0	11111111 11111111 11111111 11111111
2 ¹⁵	= 65 536	=0	00000000 00000001 00000000 00000000
2 ⁰	= 1	=0	00000000 00000000 00000000 00000001
0	= 0	=0	00000000 00000000 00000000 00000000
-2 ⁰	= -1	=1	11111111 11111111 11111111 11111111
-2 ¹	= -2	=1	11111111 11111111 11111111 11111110
-2 ¹⁵	= -65 536	=1	11111111 11111111 00000000 00000000
-2 ³¹ +1	= -2 147 483 647	=1	00000000 00000000 00000000 00000001
-2 ³¹	= -2 147 483 648	=1	00000000 00000000 00000000 00000000

APPENDIX D. FLOATING-POINT ARITHMETIC

Floating-point arithmetic simplifies programming by automatically maintaining binary point placement (scaling) during computations in which the range of values used vary widely or are unpredictable.

The key to floating-point data representation is the separation of the significant digits of a number from the size (scale) of the number. Thus, the number is expressed as a fraction times a power of 16.

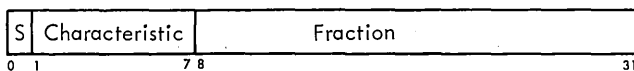
A floating-point number has two associated sets of values. One set represents the significant digits of the number and is called the fraction. The second set specifies the power (exponent) to which 16 is raised and indicates the location of the binary point of the number.

These two numbers (the fraction and exponent) are recorded in a single word or double word.

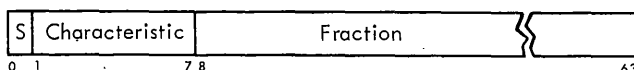
Since each of these two numbers is signed, some method must be employed to express two signs in an area that provides for a single sign. This is accomplished by having the fraction sign use the sign associated with the word (or double word) and expressing the exponent in excess 64 arithmetic; that is, the exponent is added as a signed number to 64. The resulting number is called the characteristic. Since 64 uses 7 bits, the characteristic can vary from 0 to 127, permitting the exponent to vary from -64 through 0 to +63. This provides a decimal range of $n \times 10^{76}$ to $n \times 10^{-78}$.

Floating-point data in the System/360 may be recorded in short or long formats, depending on the precision required. Both formats use a sign bit in bit position 0, followed by a characteristic in bit positions 1-7. Short-precision floating-point data operands contain the fraction in bit positions 8-31; long-precision operands have the fraction in bit positions 8-63.

Short-Precision Floating-Point Format



Long-Precision Floating-Point Format



The sign of the fraction is indicated by a zero or one bit in bit position 0 to denote a positive or negative fraction, respectively.

Within a given fraction length (24 or 56 bits), a floating-point operation will provide the greatest precision if the fraction is normalized. A fraction is normalized when the high-order digit (bit positions 8, 9, 10 and 11) is not zero. It is unnormalized if the high-order digit contains all zeros.

If normalization of the operand is desired, the floating-point instructions that provide automatic normalization are used. This automatic normalization is accomplished by left-shifting the fraction (four bits per shift) until a nonzero digit occupies the high-order digit position. The characteristic is reduced by one for each digit shifted.

Conversion Example

Convert the decimal number 149.25 to a short-precision floating-point operand. (Appendix E provides tables for conversion of hexadecimal and decimal integers and fractions.)

1. The number is decomposed into a decimal integer and a decimal fraction.

$$149.25 = 149 \text{ plus } 0.25$$

2. The decimal integer is converted to its hexadecimal representation.

$$149_{10} = 95_{16}$$

3. The decimal fraction is converted to its hexadecimal representation.

$$0.25_{10} = 0.4_{16}$$

4. Combine the integral and fractional parts and express as a fraction times a power of 16 (exponent).

$$95.4_{16} = (0.954 \times 10^2)_{16}$$

5. The characteristic is developed from the exponent and converted to binary.

$$\begin{array}{rcl} \text{base} + \text{exponent} & = & \text{characteristic} \\ 64 + 2 & = & 66 = 1000010 \end{array}$$

6. The fraction is converted to binary and grouped hexadecimally.

$$.954_{16} = .100101010100$$

7. The characteristic and the fraction are stored in short precision format. The sign position contains the sign of the fraction.

S	Char	Fraction
0	1000010	100101010100000000000000

The following are sample normalized short floating-point numbers. The last two numbers represent the smallest and the largest positive normalized numbers.

NUMBER	POWERS OF 16	S CHAR	FRACTION
1.0	$= +1/16 \times 16^1$	$= 0$	1000001 0001 0000 0000 0000 0000 0000
0.5	$= +8/16 \times 16^0$	$= 0$	1000000 1000 0000 0000 0000 0000 0000
1/64	$= +4/16 \times 16^{-1}$	$= 0$	0111111 0100 0000 0000 0000 0000 0000
0.0	$= +0 \times 16^{-64}$	$= 0$	0000000 0000 0000 0000 0000 0000 0000
-15.0	$= -15/16 \times 16^1$	$= 1$	1000001 1111 0000 0000 0000 0000 0000
5.4×10^{-79}	$= +1/16 \times 16^{-64}$	$= 0$	0000000 0001 0000 0000 0000 0000 0000
7×10^{75}	$= (1-16^{-6}) \times 16^{63}$	$= 0$	1111111 1111 1111 1111 1111 1111 1111

APPENDIX E. POWERS OF TWO TABLE

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25

APPENDIX F. HEXADECIMAL-DECIMAL CONVERSION TABLE

The following tables aid in converting hexadecimal values to decimal values, or the reverse.

Direct Conversion Table

This table provides direct conversion of decimal and hexadecimal numbers in these ranges:

HEXADECIMAL	DECIMAL
000 to FFF	0000 to 4095

For numbers outside the range of the table, add the following values to the table figures:

HEXADECIMAL	DECIMAL
1000	4096
2000	8192
3000	12288
4000	16384
5000	20480
6000	24576
7000	28672
8000	32768
9000	36864
A000	40960
B000	45056
C000	49152
D000	53248
E000	57344
F000	61440

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00_	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
01_	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
02_	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
03_	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
04_	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
05_	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
06_	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
07_	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
08_	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
09_	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A_	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B_	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C_	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D_	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E_	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F_	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255
10_	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
11_	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
12_	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
13_	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
14_	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335
15_	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
16_	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
17_	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
18_	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
19_	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A_	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B_	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C_	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D_	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E_	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F_	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20_	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527
21_	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543
22_	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559
23_	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575
24_	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591
25_	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607
26_	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623
27_	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
28_	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655
29_	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671
2A_	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687
2B_	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703
2C_	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719
2D_	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735
2E_	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751
2F_	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767
30_	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783
31_	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799
32_	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815
33_	0816	0817	0818	0819	0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831
34_	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847
35_	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863
36_	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879
37_	0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895
38_	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911
39_	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927
3A_	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939	0940	0941	0942	0943
3B_	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959
3C_	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975
3D_	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991
3E_	0992	0993	0994	0995	0996	0997	0998	0999	1000	1001	1002	1003	1004	1005	1006	1007
3F_	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
40_	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
41_	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
42_	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
43_	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087
44_	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
45_	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
46_	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
47_	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151
48_	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
49_	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
4A_	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
4B_	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
4C_	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
4D_	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E_	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
4F_	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
50_	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
51_	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
52_	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
53_	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
54_	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
55_	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
56_	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
57_	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
58_	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
59_	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A_	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B_	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C_	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D_	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E_	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F_	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
60_	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
61_	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
62_	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583
63_	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
64_	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
65_	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
66_	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
67_	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663
68_	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
69_	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A_	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B_	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C_	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D_	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E_	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F_	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791
70_	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
71_	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
72_	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
73_	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
74_	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
75_	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
76_	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
77_	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919
78_	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
79_	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
7A_	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B_	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C_	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D_	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E_	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F_	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
80_	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
81_	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
82_	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
83_	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111
84_	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
85_	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
86_	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
87_	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
88_	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
89_	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A_	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B_	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C_	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D_	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E_	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F_	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303
90_	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319
91_	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335
92_	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351
93_	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367
94_	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383
95_	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
96_	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415
97_	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
98_	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447
99_	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463
9A_	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479
9B_	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495
9C_	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511
9D_	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527
9E_	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
9F_	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A0_	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575
A1_	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591
A2_	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607
A3_	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623
A4_	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639
A5_	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655
A6_	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669	2670	2671
A7_	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687
A8_	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703
A9_	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719
AA_	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735
AB_	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751
AC_	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767
AD_	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
AE_	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798	2799
AF_	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815
B0_	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B1_	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B2_	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B3_	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B4_	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B5_	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B6_	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B7_	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B8_	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B9_	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA_	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB_	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC_	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD_	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
BE_	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BF_	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C0_	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C1_	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C2_	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C3_	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C4_	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C5_	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C6_	3168	3169	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C7_	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C8_	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C9_	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA_	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB_	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC_	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD_	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE_	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF_	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327
D0_	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D1_	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D2_	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D3_	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D4_	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D5_	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D6_	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D7_	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455
D8_	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D9_	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA_	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB_	3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519
DC_	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
DD_	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE_	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF_	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E0_	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E1_	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E2_	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E3_	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E4_	3648	3649	3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663
E5_	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E6_	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E7_	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711
E8_	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E9_	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA_	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EB_	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
EC_	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED_	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE_	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EF_	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839
F0_	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F1_	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871
F2_	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F3_	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903
F4_	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F5_	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F6_	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F7_	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F8_	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983
F9_	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA_	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FB_	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC_	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047
FD_	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FE_	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076	4077	4078	4079
FF_	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095

JUNE 1, 1971

APPENDIX G. SYSTEM/360 MODE OF OPERATION

SUBSYSTEM DEFINITION

One simplex subsystem whose operation is compatible with IBM System/360 architecture (as defined in IBM System/360 Principles of Operation, Form A22-6821), may be configured from any IBM 9020D or 9020E System. Such a subsystem contains:

- ... One IBM 7201-02 Computing Element (CE)
- ... One IBM 7231-02 Input/Output Control Element (IOCE-1)
- ... One or more IBM 7251-09 Storage Element (SE)

The subsystem may include the following peripheral elements, assigned to IOCE-1 through an IBM 7265-02 System Console (SC) on a 9020D System or IBM 7265-03 Configuration Console (CC) on a 9020E System:

- ... IBM 2821-1 Control Unit
- ... IBM 2540 Card Read/Punch
- ... IBM 1403-2 Printer
- ... IBM 1052 System Console Printer Keyboard

A subsystem may also include IBM 2803-01 Tape Control Units and the available IBM 2401-2/3 Magnetic Tape Units, and up to three IBM 2314 Storage Control Units and the available IBM 2312 or 2318 Disk Storage Units.

MODE SWITCHING

Any CE in state One or Zero may be switched from "9020-mode" operation to "360-mode" by pressing a back-lighted mode change switch provided on its control panel (Chapter 11). A CE in "360-mode" is returned to "9020-mode" by pressing the mode change switch, by a power-on reset, by receipt of an external start signal from another CE, or by placing it in state Two or Three.

PROGRAMMING NOTE

A CE in "360-mode" is returned to "9020-mode" whenever it is placed in either state Two or Three. This occurs when it is configured to state Two or Three by another CE executing SET CONFIGURATION (section 8.4.5.4), or when recalled to state Three by a CE-ELC (section 9.5.3.2). Similarly, system IPL and system PSW restart force a CE into state Three and thus always take it out of "360-mode".

INSTRUCTION REPERTOIRE

The following operation codes are not assigned for execution in "360-mode". When any of these instructions are encountered an operation code exception is recognized, and a program interruption is taken. The operation is suppressed.

JUNE 1, 1971

Operation Code	Name	Mnemonic	Section
01	SET CONFIGURATION	SCON	8.4.5.4
02	CONVERT and SORT SYMBOLS	CSS	8.6.1.2
03	CONVERT WEATHER LINES	CVWL	8.6.1.3
0B	DELAY	DLY	8.4.5.1
0C	LOAD IDENTITY	LI	8.4.5.2
0D	SET ADDRESS TRANSLATOR	SATR	8.4.5.7
0E	INSERT ADDRESS TRANSLATOR	IATR	8.4.5.8
0F	REPACK SYMBOLS	RPSB	8.6.1.4
52	LOAD CHAIN	LC	8.6.1.1
9A	START I/O PROCESSOR	SIOP	8.4.5.10
9B	SET PCI	SPCI	10.2.7.5
A0	STORE PS BASE ADDRESS	SPSB	8.4.5.5
A1	LOAD PS BASE ADDRESS	LPSB	8.4.5.3
D8	MOVE WORD	MVW	8.4.5.9

DIRECT-ADDRESS RELOCATION

When a CE is placed in "360-mode" the "logical" preferential-storage base address register (PSBAR) is reset to all zeros. The value in "physical" PSBAR is determined by the identifier appearing in position 1 of the address translator (ATR). Since LOAD PREFERENTIAL-STORAGE BASE ADDRESS cannot be executed, "logical" PSBAR will remain reset unless modified by a subsystem IPL, or subsystem PSW restart.

PSBAR "stepping" and the issuing of a logout-stop signal to a storage element is inhibited in "360-mode" by forcing on the inhibit logout-stop (ILOS) condition in both the CE and IOCE-1.

PREFERENTIAL-STORAGE AREA ASSIGNMENT

The preferential storage area assignment for both "9020-mode" and "360-mode" is shown in Section 9.1.2 and Table 12-II. The 9020 System is compatible with System/360 in the use of the first 32 words (bytes 000-127 inclusive) of the preferential-storage area. The diagnostic area extends from byte location 128 through byte location 511 in both modes of operation.

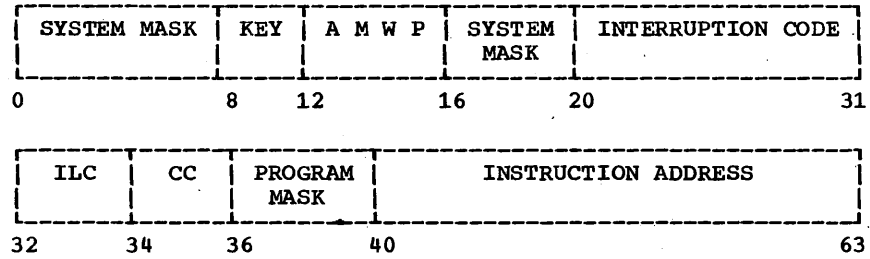
PROGRAM STATUS WORD

In "9020-mode" operation bits 0-7 and 16-19 of the PSW are interpreted as a 12-bit system mask. In "360-mode" operation the IBM System/360 PSW format applies. Bits 0-7 become the system mask and bits 16-19 are interpreted as the high-order positions of a 16-bit interrupt-

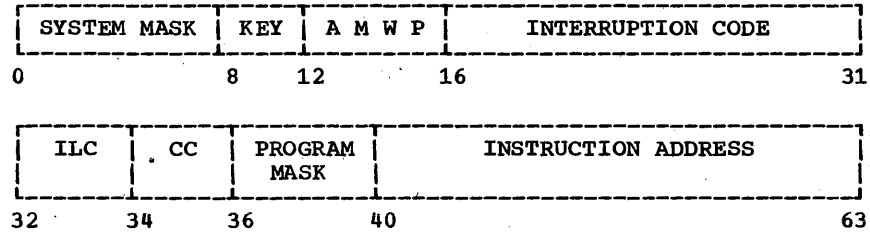
JUNE 1, 1971

tion code field.

IBM 9020 System PSW Format



IBM System/360 PSW Format



JUNE 1, 1971

"9020-mode"		"360-mode"	
0 - 7	System Mask		
0	Multiplexor Channel A mask		Multiplexor channel mask
1	Selector channel 1A mask		Selector channel 1 mask
2	Selector channel 2A mask		Selector channel 2 mask
3	Selector channel 3A mask		Selector channel 3 mask
4	Multiplexor channel B mask		
5	Selector channel 1B mask		Ignored
6	Selector channel 2B mask		
7	External mask		External mask
	8-11	Protection key	
	12	USASCII-8 Mode (A)	
	13	Machine Check mask (M)	
	14	Wait State (W)	
	15	Problem State (P)	
16-19	System Mask	16-19	Interruption Code
16	Selector Channel 3B mask		
17	Multiplexor channel C		
18	Selector channel 1C mask		
19	Selector channel 2C mask		
	20-31	Interruption Code	
	32-33	Instruction length code (ILC)	
	34-35	Condition code (CC)	
	36-39	Program mask	
	36	Fixed-point overflow mask	
	37	Decimal overflow mask	
	38	Exponent underflow mask	
	39	Significance mask	
	40-63	Instruction address	

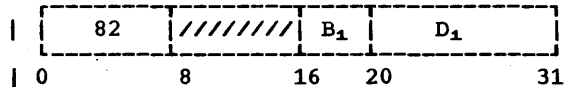
JUNE 1, 1971

STATUS-SWITCHING INSTRUCTIONS

The instructions LOAD PSW (Section 8.5.1.1), SET SYSTEM MASK (Section 8.5.1.3), and SUPERVISOR CALL (Section 8.5.1.4) are executed as defined for IBM System/360 by a CE operating in "360-mode".

LOAD PSW

| LPSW D₁(B₁) [SI]



The double word at the location designated by the operand address replaces the PSW.

The operand address must have its three low-order bits zero to designate a double word; otherwise a specification exception results in a program interruption.

The double word which is loaded becomes the PSW for the next sequence of instructions. Bits 40-63 of the double word become the new instruction address. The new instruction address is not checked for available storage or for an even byte address during a load PSW operation. These checks occur as part of the execution of the next instruction.

Bits 8-11 of the double word become the new protection key. The interruption code in bit positions 16-31 of the new PSW is not retained as the PSW is loaded. When the PSW is subsequently stored because of an interruption, these bit positions contain a new code. Similarly bits 32 and 33 of the PSW are not retained upon loading. They will contain the instruction-length code for the last-interpreted instruction when the PSW is stored during a branch-and-link operation, or during a program or supervisor-call interruption.

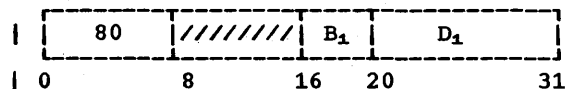
Condition Code: The code is set according to bits 34 and 35 of the new PSW loaded.

Program Interruptions:

Privileged operation
Addressing
Specification
Protection

SET SYSTEM MASK

| SSM D₁(B₁) [SI]



The byte at the location designated by the operand address replaces the system mask, bits 0-7, of the current PSW.

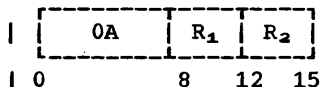
Condition Code: The code remains unchanged.

JUNE 1, 1971

Program Interruptions:
Privileged operation
Addressing
Protection

SUPERVISOR CALL

| SVC R₁,R₂ [RR]



The instruction causes a supervisor-call interruption, with the R₁ and R₂ field of the instruction providing the interruption code.

The contents of bit positions 8-15 of the instruction are placed in bit positions 24-31 of the old PSW which is stored in the course of the interruption. Bit positions 16-23 of the old PSW are made zero. The old PSW is stored at location 32 of the preferential-storage area, and a new PSW is obtained from location 96. The instruction is valid in both problem and supervisor state.

Condition Code: The code remains unchanged in the old PSW.

Program Interruptions: None.

INTERRUPTION HANDLING

An interruption consists of storing the current PSW as an "old PSW" and fetching a "new" PSW. Processing resumes in the state indicated by the "new" PSW. To permit proper programming action following an interruption, the cause of the interruption is identified by the interruption code stored in the "old" PSW. In "360-mode" bits 16-31 of the PSW are stored as the interruption code. In "9020-mode" bits 20-31 are stored (Chapter 9). Both interruption codes are shown in the following tabulations.

JUNE 1, 1971

PROGRAM INTERRUPTIONS			
"9020-mode" Interruption Source Identification	Interruption Code PSW Bits		"360-mode" Interruption Source Identification
	<-----"360-mode"----->		
	<--"9020-mode"-->		
Operation	0000	0000 00000001	Operation
Privileged operation	0000	0000 00000010	Privileged operation
Execute	0000	0000 00000011	Execute
Protection	0000	0000 00000100	Protection
Addressing	0000	0000 00000101	Addressing
Specification	0000	0000 00000110	Specification
Data	0000	0000 00000111	Data
Fixed-point overflow	0000	0000 00001000	Fixed-point overflow
Fixed-point divide	0000	0000 00001001	Fixed-point divide
Decimal overflow	0000	0000 00001010	Decimal overflow
Decimal divide	0000	0000 00001011	Decimal divide
Exponent overflow	0000	0000 00001100	Exponent overflow
Exponent underflow	0000	0000 00001101	Exponent underflow
Significance	0000	0000 00001110	Significance
Floating-point divide	0000	0000 00001111	Floating-point divide
IOCE-3 PSA lockout		0000 00010000	
IOCE-2 PSA lockout		0000 00100000	
IOCE-1 PSA lockout		0000 01000000	
SE stopped		0000 10000000	
<u>Legend</u>			
PSA -- Preferential-storage area			

PROGRAMMING NOTE

IOCE PSA lockout and SE stopped interruption requests are ignored by a CE in "360-mode". The detection of either condition causes the CE to check stop.

SUPERVISOR CALL INTERRUPTION		
"9020-mode" Interruption Source Identification	Interruption Code PSW Bits	"360-mode" Interruption Source Identification
	<-----"360-mode"-----> <-"9020-mode"->	
Instruction bits	0000 0000 rrrrrrrr	Instruction bits
<u>Legend</u> r -- Bits of R ₁ and R ₂ field of SUPERVISOR CALL		

EXTERNAL INTERRUPTIONS		
"9020-mode" Interruption Source Identification	Interruption Code PSW Bits	"360-mode" Interruption Source Identification
	<-----"360-mode"-----> <-"9020-mode"->	
DAR	0000 xxxx xxxxxxxx1	DAR
PIR	0000 xxxx xxxxxx1x	PIR
CE4 Write Direct	0000 xxxx xxxxx1xx	CE4 Write Direct
CE4 Read Direct	0000 xxxx xxxx1xxx	CE4 Read Direct
CE3 Write Direct	0000 xxxx xxx1xxxx	CE3 Write Direct
CE3 Read Direct	0000 xxxx xx1xxxxx	CE3 Read Direct
Interrupt Switch	0000 xxxx x1xxxxxx	Interrupt Switch
Timer	0000 xxxx 1xxxxxxx	Timer
CE2 Write Direct	0000 xxx1 xxxxxxxx	CE2 Write Direct
CE2 Read Direct	0000 xx1x xxxxxxxx	CE2 Read Direct
CE1 Write Direct	0000 x1xx xxxxxxxx	CE1 Write Direct
CE1 Read Direct	0000 1xxx xxxxxxxx	CE1 Read Direct
<u>Legend</u> x -- Unpredictable DAR -- Diagnose accessible register PIR -- Processor interrupt register		

JUNE 1, 1971

PROGRAMMING NOTE

The direct control instructions READ DIRECT and WRITE DIRECT are defined for 9020 System operation (Chapter 8). They are executed in "360-mode" subject to all the conditions placed on them in "9020-mode". In particular, specification exceptions and condition code settings can occur which are not part of System/360 architecture.

INPUT/OUTPUT INTERRUPTIONS		
"9020-mode" Interruption Source Identification	Interruption Code PSW Bits	"360-mode" Interruption Source Identification
	<-----"360-mode"----->	
	<--"9020-mode"-->	
Multiplexor Channel A	0000 0000 aaaaaaaaa	Multiplexor Channel
Selector Channel 1A	0000 0001 aaaaaaaaa	Selector Channel 1
Selector Channel 2A	0000 0010 aaaaaaaaa	Selector Channel 2
Selector Channel 3A	0000 0011 aaaaaaaaa	* Selector Channel 3
Multiplexor Channel B	0100 aaaaaaaaa	
Selector Channel 1B	0101 aaaaaaaaa	
Selector Channel 2B	0110 aaaaaaaaa	
Selector Channel 3B	0111 aaaaaaaaa	
Multiplexor Channel C	1000 aaaaaaaaa	
Selector Channel 1C	1001 aaaaaaaaa	
Selector Channel 2C	1010 aaaaaaaaa	

Legend

a -- Device Address Bit

* -- Optional Channel in IOCE-1

JUNE 1, 1971

MACHINE-CHECK INTERRUPTIONS		
"9020-mode Interruption Source Identification	Interruption Code PSW Bits	"360-mode" Interruption Source Identification
	<p><-----"360-mode"-----></p> <p><-"9020-mode"-></p>	
CE Malfunction	0000 0000 00000000	CE Malfunction
IOCE-1 Malfunction	0000 0000 00000001	IOCE-1 Malfunction
IOCE-2 Malfunction	0000 00000010	
IOCE-3 Malfunction	0000 00000011	
Read Direct Timeout	0000 0000 00000100	Read Direct Timeout

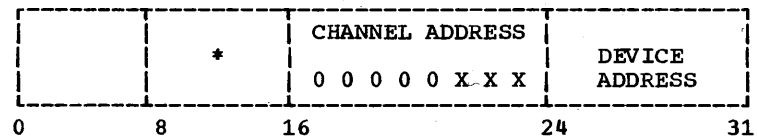
JUNE 1, 1971

INPUT/OUTPUT

Input/Output Addressing

Channel and device addresses are developed as the sum obtained by the addition of the content of register B_1 and the content of the D_1 field of the I/O instruction being executed.

This sum has the format:



Bit positions 0-7 are not part of the address. Bit positions 8-15, which constitute the high-order portion of the address are ignored. Bit positions 16-23 of the sum contain the channel address, while bit positions 24-31 identify the device on the channel.

Bit positions 16-20 of the sum must contain zeros. A condition code 3 is set for an I/O instruction that does not contain the proper number of zeros. Instruction execution is terminated.

Channel Addressing		
Channel Address	"9020-mode" Interpretation	"360-mode" Interpretation
0000 0000	Multiplexor Channel A	Multiplexor Channel
0000 0001	Selector Channel 1A	Selector Channel 1
0000 0010	Selector Channel 2A	Selector Channel 2
0000 0011	Selector Channel 3A	Selector Channel 3
0000 0100	Multiplexor Channel B	Not operational
0000 0101	Selector Channel 1B	Not operational
0000 0110	Selector Channel 2B	Not operational
0000 0111	Selector Channel 3B	Invalid
0000 1000	Multiplexor Channel C	Invalid
0000 1001	Selector Channel 1C	Invalid
0000 1010	Selector Channel 2C	Invalid
0000 1011		
thru	Invalid	Invalid
1111 1111		

PROGRAMMING NOTE

A subsystem configured for "360-mode" operation uses IOCE-1. Consequently, I/O instructions which specify other channels are terminated with condition code 3 stored to indicate that the addressed channel is not operational, or that the channel address is invalid.

INITIAL PROGRAM LOADING

System IPL cannot be carried out in "360-mode" as the CE is forced into state Three which causes it to return to "9020-mode". A 9020

JUNE 1, 1971

system IPL takes place (Chapter 11). Similar action occurs when a system PSW restart is attempted in "360-mode".

A subsystem IPL, or subsystem PSW restart may be carried out in "360-mode". Both "logical" and "physical" PSBARS are set to reference the SE designated in the main storage select switch. The first PSW is fetched from location 0 in this SE.

STORAGE PROTECTION

For protection purposes main storage is divided into blocks of 2048 bytes, each block having an address which is a multiple of 2048. A five-bit storage key is associated with each block. When data are stored in a storage block, the four high-order bits of the storage key are compared with the protection key. When data are fetched the fetch-protection bit (fifth, i.e., low-order bit) is inspected. When the fetch-protection bit is one the four high-order bits of the storage key are compared with the protection key. The protection key of the current PSW is used as the comparand when a storage access is specified by an instruction. When a storage access is specified by a channel operation, the protection key supplied to the channel by the CAW is used as the comparand.

The keys are said to match in either "9020-mode" or "360-mode" when they are equal or when either one is zero. This operation is not compatible with IBM System/360 architecture, as the keys are said to match in System/360 when they are equal or when the protection key is zero.

Protection Key	Storage Key	9020 System Interpretation	System/360 Interpretation
X	X	Match	Match
X	Y	Mismatch	Mismatch
X	Zero	Match	Mismatch
Zero	X	Match	Match
Zero	Zero	Match	Match

Where X ≠ zero, Y ≠ zero.

IOCE-PROCESSOR OPERATION

The IOCE-processor capability is disabled when an IOCE (applicable to IOCE-1 only) is placed in "360-Mode". The IOCE-processor is forced into the stopped state by hardware. "360-Mode" operation is unchanged.

JUNE 1, 1971

APPENDIX H: IOCE-PROCESSOR

1.0 GENERAL CONCEPTS

This appendix describes the architectural design of RPQ F21241 which exploits the latent computing power in each IBM 7231-02 Input/Output Control Element. This is done by providing the IOCE with 126,976 bytes of 2 microsecond internal (MACH) storage and an IOCE-processor mode in which the IOCE executes a subset of the IBM 9020 System instruction set. An IOCE-processor is utilized under control of the CE.

From the user's viewpoint the IOCE consists of two distinct and independent functional units sharing a common logic and storage unit, as shown in Figure H-1. They are:

IOCE-channel controller
IOCE-processor

Areas where strict independence and simultaneousness are not possible (e.g., malfunction detection) are virtually transparent to the problem programmer and need only concern supervisory and diagnostic programmers.

1.1 IOCE-CHANNEL CONTROLLER

The IOCE-channel controller continues to perform those functions presently implemented in the IOCE. Its distinguishing attributes are:

Input/output operations are initiated by a controlling CE.

Automatic initiation and control of device operation through control units.

Input/output interruptions are serviced by a CE through the Common Logic Unit (CLU).

Access is provided to all main storage (SE) for data transfer, and control functions (CAW, CCW, CSW).

Channel Controller is always running (under CE control) even though the processor may be stopped.

In addition, the IOCE-channel controller is modified by this RPQ to:

Access MACH storage for I/O data transfers and CCW fetching, with the selection of main or MACH storage under program control.

1.2 IOCE-PROCESSOR

The IOCE-processor has the following characteristics:

Processing (i.e., instruction execution) is initiated by a controlling CE.

Program, external, and supervisor - call interruptions are serviced by the IOCE-processor through the Common Logic Unit (CLU).

JUNE 1, 1971

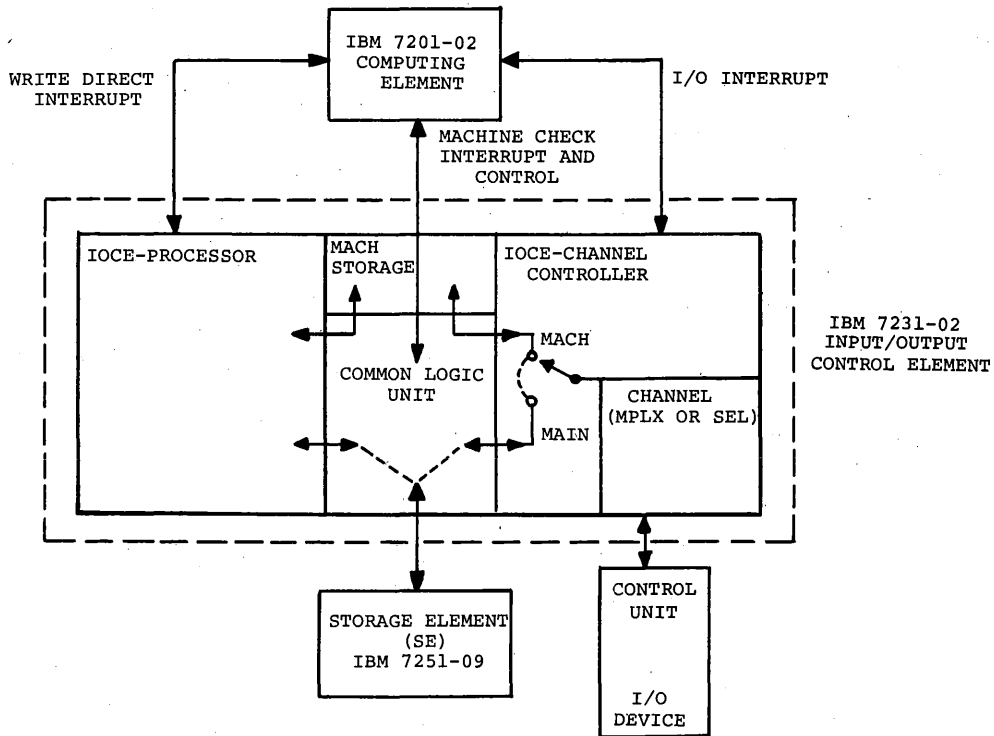


Figure H-1 IOCE Block Diagram

JUNE 1, 1971

The IOCE-processor can execute a subset of the IBM 9020 System instruction set with a few special instructions added under this RPQ (see Table H-2). The IOCE-processor cannot initiate I/O operations.

No floating-point or decimal arithmetic capability is provided.

Operation is in four program state alternatives:

Stopped/Operating
Wait/Running
Masked/Interruptable
Problem/Supervisor

The IOCE-processor has access to all main and to its internal MACH storage for instruction execution and data.

1.3 COMMON LOGIC UNIT (CLU)

That portion of the IBM 7231-02 which contains the control hardware and read-only-storage used to interface with the controlling CE, IOCE-channel controller, and IOCE-processor is called the common logic unit (CLU). Its chief functions are:

Allocation of the common resources in the 7231 to the controlling CE, IOCE-channel controller, and IOCE-processor in accordance with their assigned priority.

Creation and handling of status information (PSW, CSW, etc.)

Detection and signalling of error conditions.

1.3.1 PRIORITY CONTROL

Two levels of priority control exist in the 9020 System. The first, described in Section 13.12, is concerned with IOCE/CE-SE communications. Each SE has built-in priority circuitry to grant service to one element in cases of access request ties, and to distribute its availability where multiple-like elements require continuous service. This priority scheme remains unchanged. The SE always gives priority to an IOCE over a CE and does not distinguish between the IOCE-channel controller and the IOCE-processor.

A second level of priority control is provided within the IOCE for the allocation of common resources, and storage access requests (to both MACH and main storage). First priority is always given to the IOCE-channel controller to handle input/output data transfer. Second priority is given to the controlling CE to enable it to initiate new input/output operations in the IOCE channel controller. Third priority is assigned to the IOCE-processor. The IOCE-channel controller is allowed to break-in on other operations in the IOCE since some of the I/O devices which it controls may overrun if they are not serviced in time. The controlling CE is always permitted to break in on the IOCE-processor to execute an I/O instruction (Start I/O, Test I/O, Halt I/O, Test Channel, or Set PCI), during a move-type operation in the IOCE processor or during the Delay instruction. On other instructions, controlling-CE break in occurs when the IOCE-processor has completed one instruction, and before starting another, i.e., during I-fetch.

JUNE 1, 1971

1.3.2 STATUS CONTROL AND ERROR HANDLING

When a program, external (pushbutton, timer, or controlling CE Write Direct), or supervisor-call interruption is generated by the IOCE-processor, the common logic unit creates and stores the appropriate "old" PSW in MACH storage. On a machine-check, CLU handles the check registers, issues the element check (ELC) signal, and performs the logout. During I/O interruptions, it handles the PSW and CSW as now implemented. Detailed explanations of these functions are given later.

1.4 OTHER FUNCTIONS

Configuration control and the functions of the IOCE's configuration control register (CCR) remain unchanged. The inter-element relations established by the contents of the CCR apply to both the IOCE-channel controller and IOCE-processor sections of the IOCE. Similarly, the function of the storage address translator (ATR) in the IOCE is unchanged. Address assignments for main storage are common to both channel-controller and processor. For details, see Set Configuration (Section 8.4.5.4) and Set Address Translator (Section 8.4.5.7).

1.5 MODE CONTROL

The IBM 7231-02 may be in one of two modes: Operational or diagnostic. In operational mode the IOCE-processor is either in operating or stopped state, and the channel controller is either busy or idle. All instructions added by this RPQ are available in the diagnostic mode. Existing features of the IOCE diagnostic mode are unchanged.

NOTE: Operational mode is to be assumed throughout this appendix, except where diagnostic mode is explicitly indicated.

2.0 CE OPERATION

Principle modifications to the CE are:

- a. Extension of the Write Direct instruction.
- b. Addition of two new instructions: Start I/O Processor and Move Word.
- c. Extension to the external interruption facility.
- d. Addition of new diagnostic kernel.

2.1 NEW OR EXTENDED CE INSTRUCTIONS

2.1.1 WRITE DIRECT

The Write Direct instruction is extended to provide "IOCE-processor start", "IOCE-processor stop", "IOCE-processor interrupt", and "CE external stop" commands. (See section 8.5.1.7)

2.1.2 MOVE WORD

The Move Word instruction provides a means for specifying storage to storage transfers in terms of number of words to be moved, rather than in number of bytes. (See Section 8.4.5.9)

JUNE 1, 1971

2.1.3 START I/O PROCESSOR

The Start I/O Processor instruction is used by the CE to make available to the designated IOCE-processor a protection key and the operand address from which the IOCE-processor may obtain a program status word (PSW). This PSW, when loaded by the IOCE-processor, determines its subsequent action. (Section 3.4 of this Appendix.)

Execution of Start I/O Processor terminates in the CE, returning it to instruction fetching, when either the IOCE-processor returns condition code 0 or 1, or the CE stores condition code 3 on a read-only-storage timeout. Start I/O Processor condition codes are described in Section 8.4.5.10.

2.2 EXTERNAL INTERRUPTION ACTION

An extension to the external interruption facility provides a means by which the CE responds to direct control signals from the IOCE-processors. (See Section 9.5.3.3) A processor interruption register (PIR) is provided in the CE to store the IOCE-processor requests for external interruption until removed by a diagnostic operation.

2.3 DIAGNOSTIC KERNEL

The addition of the processor interruption register (PIR) to the CE requires a change to the Logout Registers kernel, and the addition of a new CE kernel to store PIR. (See Section 8.5.1.9). PIR is logged out in logword 39, bits 27-29.

3.0 IOCE - PROCESSOR OPERATION

3.1 ADDRESS GENERATION

Instructions referring to storage have been given the capability to explicitly address operands in either main or MACH storage. Address generation in the IOCE-processor is similar to that in the CE, that is, by combining a base address, index factor, and displacement to produce a 24-bit effective address. The magnitude of the effective address determines the type of storage accessed, where:

$$0 \leq \text{Main Storage Address} \leq 12,582,911$$
$$12,582,912 \leq \text{MACH Storage Address} \leq 16,777,215$$

The upper bound on main storage addressing depends upon the available main storage in a particular installation. The maximum main storage address possible on the IBM 9020D System is 5,242,879 (i.e., when 10 SES are installed), and on the 9020E System is 2,621,439 (i.e., when 5 SES are installed). An address in the interval 5,242,880 through 12,582,911 is always invalid on either system, and causes a program interruption on an addressing exception. MACH storage addresses are always invalid in the 7201-02 Compute Element.

An address specifying a location in MACH storage may be considered to consist of two parts: a base address and a displacement. The base address is 12,582,912. The displacement provides for relative addressing of up to 126,975 bytes beyond the base address, and, therefore, spans MACH storage. (The upper 4096 bytes are reserved for the multiplexor channel UCWs and are not directly addressable by the programmer.) A valid MACH address, therefore, lies in the interval:

JUNE 1, 1971

12,582,912 ≤ effective address ≤ 12,709,887

which corresponds to

0 ≤ MACH location ≤ 126,975

The low limit is significant where address arithmetic creates an address below MACH location zero. MACH storage addresses do not wrap around to location 0 when stepped beyond the upper limit.

IMPLEMENTATION NOTE

The selection of storage is under program control, with bits 8 and 9 of the effective address designating the storage to be used. When bits 8 and 9 are both ones, the address refers to a MACH storage location. When bits 8 and 9 are other than both ones, the address refers to a main storage location.

Instructions having two operand addresses (SS format) may reference two locations in main or MACH storage, or reference one location in main, and another in MACH storage.

A change in the sequential operation of the IOCE-processor caused by branching, status switching, or interruption handling may cause instruction fetching to be switched from main to MACH storage, and vice versa, depending upon the value of bits 8 and 9 introduced in the new instruction address.

The branch address of Execute may reference main or MACH storage, independently of the storage from which the Execute instruction was fetched. Similarly, the branch and link addresses of a Branch And Link instruction may reference the same type of storage, or both main and MACH storage.

The control function associated with bits 8 and 9 is always active except when the IOCE is operating in Diagnostic Mode, or in "360 Mode" (restricted to IOCE-1). References are made to MACH or main storage when in diagnostic mode under control of the DIAGNOSE instruction. When in "360 Mode" the IOCE-processor is disabled. (See section 4.0 for IOCE channel operation.)

3.2 PROGRAM STATES IN THE IOCE-PROCESSOR

The four types of program-state alternatives which determine overall IOCE-processor status are Stopped/Operating, Wait/Running, Masked/Interruptable, and Problem/Supervisor.

All program states are independent of each other in their function, indication, and status-switching. Status-switching does not affect the contents of the arithmetic registers or the execution of I/O operations. It may affect the timer operation.

3.2.1 PROBLEM/SUPERVISOR STATE

The choice between supervisor and problem state determines whether or not the full set of instructions is valid. If bit 15 of the IOCE's PSW is set to 1 only the non-privileged instructions can be used; all privileged instructions encountered constitute a privileged instruction exception.

If the PSW bit 15 is zero all instructions provided in the IOCE-processor are valid with the exception of I/O operations (Start

JUNE 1, 1971

| I/O, Halt I/O, Test I/O, Test Channel, and Set PCI) and the Diagnose instruction.

| 3.2.2 STOPPED/OPERATING STATE

| In the Stopped state no IOCE-processor instructions are processed, and neither main nor MACH storage is accessed for this purpose. In the operating state the interval timer is updated, and the IOCE is capable of executing instructions and being interrupted.

| An IOCE-processor mode latch is provided to place the processor in either stopped or operating state. The processor mode latch can be changed from operating to stopped state by: (a) switching from 9020-mode to 360-mode in the case of IOCE-1; (b) by a system or subsystem reset associated with an IPL or PSW restart; (c) by an IOCE-processor stop via direct control. Conversely, this bit can be changed from stopped to operating state by a direct control IOCE-processor start or the Start I/O Processor instruction.

| Timer updating is affected by the choice between operating and stopped state. When the IOCE-processor is placed in the stopped state timer updating is suspended. Going into operating state enables timer updating and enables its ability to present an external interruption request in the IOCE-processor (subject to the usual PSW bit 7 masking).

| 3.2.3 WAIT/RUNNING STATE

| When bit 14 of the IOCE's PSW is one, the IOCE-processor is waiting. When bit 14 is zero, the IOCE-processor is running. Switching from running to wait state may be achieved by a Load PSW in an IOCE-processor program, or a Start I/O Processor operation which introduces an entire new PSW with bit 14 set to one, or an interrupt. Conversely, switching from wait to running state is achieved when bit 14 of the PSW introduced is zero. In the wait state, no instructions are processed, whereas, in the running state, instruction fetching and execution may proceed in the normal manner. IOCE-processor timer updating is not affected by the choice between wait state and operating state.

PROGRAMMING NOTE

| The choice of problem/supervisor, stopped/operating, wait/running, or masked/interruptable does not disturb any I/O operation, or the state of any channel, subchannel, or device. Conversely, the state of the IOCE-processor is not altered by conditions detected in any I/O operation with the exception of an unmasked machine-check interruption.

| 3.2.4 MASKED/INTERRUPTABLE STATE

| The IOCE-processor may be masked or interruptable for external, machine-check, and fixed-point-overflow interruptions depending upon the prevailing system, machine-check, and program masks respectively. External interrupts are maskable by bit 7 in the IOCE-processor's PSW. Machine-check interruptions are masked by bit 13 in the controlling CE's PSW. Fixed-point-overflow interruptions are masked by bit 36 in the IOCE-processor's PSW.

| When the IOCE-processor is masked off for fixed-point-overflow exceptions, they are ignored and processing continues. External interruptions remain pending when masked off by the IOCE-processor. When the IOCE-processor is interruptable (PSW bit 7 is one), the interruption is accepted by the IOCE-processor. When the IOCE-processor is interruptable for a machine check, this interruption is accepted by raising the

JUNE 1, 1971

machine check interruption request line to the controlling CE, and by check-stopping. Masked machine-check interruptions always remain pending in the controlling CE, and the IOCE remains check-stopped until permitted to log out.

Those program interruptions not maskable, as well as the supervisor - call interruption, are always taken by the IOCE-processor.

With the exception of machine-check interruptions, the interruptable state of the IOCE-processor is switched by changing the mask bits in its PSW. The program mask (bit 36) may also be changed by use of Set Program Mask in an IOCE-processor program. The system mask (bit 7) may also be changed by the instruction Set System Mask. Machine-check interruption masking is changed by introducing a new PSW bit 13 in the controlling CE's PSW.

IOCE-processor timer updating is not affected by the choice between masked or interruptable states.

3.3 STORAGE PROTECTION

The storage protection feature applies to main storage (SE) references. Storage protection is not provided for MACH storage. The operation of main storage protection is unchanged, and is described in Sections 2.6 and 8.2.

3.4 PROGRAM STATUS WORD IN THE IOCE-PROCESSOR

IOCE-processor control is determined by the contents of its program status word (PSW). In general, the PSW is used to control instruction sequencing, and to hold and indicate the status of the IOCE-processor relative to the program being executed. Its functions, although more restricted, are similar to the PSW in a computing element.

The IOCE-processor PSW format is shown in Figure H-2. The purpose of the PSW fields are:

System Mask: A one-bit system mask is provided for the IOCE-processor's external interruptions. When bit 7 is one, the timer overflow, interrupt switch, or controlling CE can interrupt the IOCE-processor.

Protection Key: Bits 8-11 of the PSW form the IOCE-processor's protection key on a main storage reference. The key is matched with the four high-order bits of the storage key when data are to be stored, and when data are to be fetched and the fetch-protection bit in the storage key is one. The protection key is ignored on a MACH storage reference. The use of the key F (hexadecimal) is not permitted by the IOCE-processor.

USASCII-8 (A): When bit 12 of the PSW is one, the codes preferred for the extended USASCII-8 code are generated for decimal results for the Unpack and Convert to Decimal instructions. When PSW bit 12 is zero, the codes preferred for the extended binary-coded-decimal interchange code are generated.

Wait State (W): When bit 14 of the PSW is one, the IOCE-processor is in the wait state. When PSW bit 14 is zero, the IOCE-processor is running.

Problem State (P): When bit 15 of the PSW is 1, the IOCE-processor is in the problem state. Where PSW bit 15 is zero, the IOCE-processor is in the supervisor state.

JUNE 1, 1971

Interruption Code: Bit 16-31 of the PSW identify the cause of a program, supervisor call, machine check, or timer interruption. Use of the code for all four interruption types is shown in Table H-1, Interruption Action.

Instruction Length Code (ILC): The code in PSW bits 32 and 33 indicates the length, in halfwords, of the last-interpreted instruction in the IOCE-processor program, when an interruption occurs. The code is unpredictable for a machine-check or external interruption. (The encoding of these bits is summarized in the table, Instruction Length Recording, (Section 9.1.3).)

Condition Code (CC): Bits 34 and 35 of the PSW are the two bits of the condition code. (The condition codes for all instructions are summarized in the table, Condition Code, at the end of Chapter 7.)

Program Mask: Bit 36 of the PSW is associated with a program exception specifying a fixed-point overflow. When the mask bit is one, the exception results in an interruption in the IOCE-processor. When the mask bit is zero, no interruption occurs.

Instruction Address: Bits 40-63 of the PSW are the instruction address. This address specifies the leftmost eight-bit byte position of the next instruction in main or MACH storage.

PROGRAMMING NOTE 1

Bits 0-6, 13 and 37-39 are ignored on a PSW-load operation associated with a Start I/O Processor operation, an interrupt, or LPSW instruction.

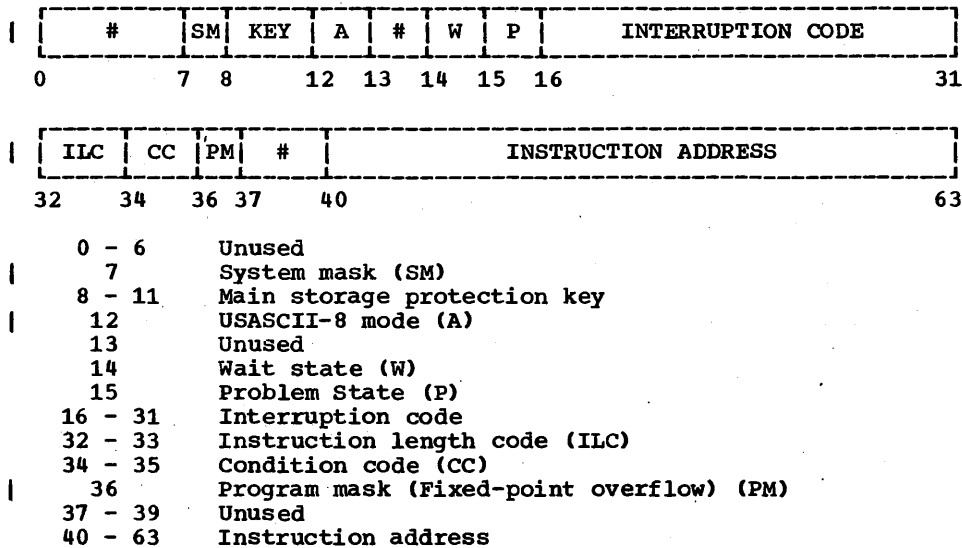


FIGURE H-2 -- IOCE-Processor PSW Format (Operational Mode)

PROGRAMMING NOTE 2

Privileged instructions which are handled in Diagnostic Mode (with the exception of Write Direct, Load PSW, Set System Mask), cause program interruptions on an invalid operation exception when encountered in the IOCE-processor programs.

JUNE 1, 1971

PROGRAMMING NOTE 3

Since IOCE-processor and IOCE-I/O operations are separate and independent functions under control of a computing element, the content of the IOCE-processor PSW can have no effect upon I/O channel operations. Conversely, the state of the system mask bits 0-6, 16-19 in the CE's PSW affect the channel operation only, and are not related to corresponding bit positions in the IOCE-processor's PSW.

3.5 INTERRUPTIONS IN THE IOCE-PROCESSOR

3.5.1 INTERRUPTION ACTION

A program, supervisor call, or external interruption in the IOCE-processor consists of storing its current PSW as an old PSW and fetching a new PSW. The old PSW is stored in MACH with interruption code (PSW bits 16-31) indicating the cause of the interruption. The new PSW is fetched from MACH and processing resumes under its control.

Machine-check masking is provided in the controlling CE's PSW, and is independent of the IOCE-processor PSW.

PROGRAMMING NOTE 1

An interruption always takes place after one IOCE-processor instruction interpretation is finished and before a new instruction interpretation is started. Similarly, an IOCE machine-check interruption is permitted by the CE after the CE instruction is completed, and before a new instruction is fetched. When the IOCE-processor receives a direct control "IOCE-processor stop" command, the current instruction is finished and all interruptions which are pending or become pending before the end of the instruction, are preserved. (It should be noted that this action is quite different from a CE-external stop which causes the CE to reset and enter the stopped state.)

JUNE 1, 1971

TABLE H-1 -- INTERRUPTION ACTION IOCE-PROCESSOR						
INTERRUPTION SOURCE IDENTIFICATION	INTERRUPTION CODE PSW BITS 16-31		MASK BITS	ILC SET	INSTRUCTION EXECUTION	
<u>Program</u> (old PSW MACH 40, new PSW MACH 104, priority 2)						
Operation	0000	0000	00000001	1,2,3	suppressed	
Privileged Operation	0000	0000	00000010	1,2	suppressed	
Execute	0000	0000	00000011	2	suppressed	
Protection	0000	0000	00000100	0,2,3	suppressed/ terminated	
Addressing	0000	0000	00000101	0,1,2,3	suppressed/ terminated	
Specification	0000	0000	00000110	1,2,3	suppressed	
Data	0000	0000	00000111	2	terminated	
Fixed-pt. overflow	0000	0000	00001000	36	completed	
Fixed-pt. divide	0000	0000	00001001	1,2	suppressed/ completed	
SE stopped	0000	0000	10000000	1,2,3	suppressed/ terminated	
<u>Supervisor Call</u> (old PSW MACH 32, new PSW MACH 96, priority 2)						
Instruction bits	0000	0000	rrrrrrrr	1	completed	
<u>External</u> (old PSW MACH 24, new PSW MACH 88, priority 3)						
Unused	0000	0000	nnnnnlnl	7	X	completed
Controlling CE	0000	0000	nnnnnlnl	7	X	completed
Unused	0000	0000	nnnnlnnn	7	X	completed
Unused	0000	0000	nnnlnnnn	7	X	completed
Unused	0000	0000	nnlnnnnn	7	X	completed
Interrupt switch	0000	0000	nlnnnnnn	7	X	completed
Timer	0000	0000	lnnnnnnn	7	X	completed
<u>Machine Check</u> (old PSW MACH 48, new PSW MACH 112, priority 1)						
IOCE malfunction	0000	0000	00000000	13*	X	terminated
<u>Legend:</u>						
n Other external interruption conditions						
x Unpredictable						
r Bits of R ₁ and R ₂ field of Supervisor Call						
* in CE PSW when operational mode, in IOCE-processor PSW when diagnostic mode.						

3.5.2 PROGRAM INTERRUPTIONS

The exceptions which cause program interruptions in the IOCE-processor are shown in Table H-1. The SE-stopped exception has been added. Its purpose and operation is similar to that in the CE (see section 9.3.17).

3.5.3 MACHINE-CHECK INTERRUPTION

Machine checks detected during an IOCE-processor operation are handled by the common logic unit (CLU). Other than the addition of storing the old machine-check PSW in MACH location 48, they are handled in the same way as those machine-check interruptions associated with IOCE-channel controller operations. (See Section 12.5.1)

3.5.4 EXTERNAL INTERRUPTIONS

3.5.4.1 Timer

The IOCE-processor timer occupies a 32-bit word at MACH storage location 80. The timer value may be changed at any time by storing a new value in MACH storage location 80. A timer value changing from positive to negative causes an external interruption request in the IOCE-processor. When the IOCE-processor is placed in the stopped state by an IOCE-processor stop via direct control, timer updating is suspended. Going into wait state does not affect timer updating, nor its ability to present an external interruption request in the IOCE-processor (subject to the usual PSW bit 7 masking).

Timer rate is determined by the line frequency with the same resolution as the CE timer. (See section 9.5.1.)

3.5.4.2 Controlling CE

Receipt of an IOCE-processor interrupt (via Write Direct) from the controlling CE requests an external interruption. When allowed, bit 30 is set in the interruption code stored.

3.6 ADDITIONAL IOCE INSTRUCTIONS

The following new instructions are added to the IOCE instruction repertoire. They are executable by the IOCE-processor in the operational and diagnostic modes.

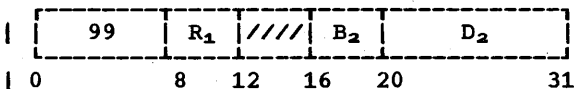
Load PSW and Set System Mask functions are restricted in operational mode. They cannot be used to alter I/O or machine-check masking.

3.6.1 MOVE WORD

This instruction is identical to the Move Word instruction in the CE (see Section 8.4.5.9). Unlike the CE, the IOCE-processor may use the instruction for MACH-to-MACH, main-to-main, MACH-to-main, or main-to-MACH moves (see Section 3.1 of this appendix.)

3.6.2 LOAD DATA ADDRESS

LDA R₁, D₂(B₂) [RS]



The data address in the unit control word for the multiplexor subchannel specified by the second operand is loaded into the low-order

JANUARY 31, 1973

TNL: GN31-3005

24 bits of the register specified by the first operand. The remaining bits of the word are set to zero.

The second operand address is not used to address data; its low-order eight bits specify a subchannel. The remainder of the address is ignored.

Condition Code: The code remains unchanged.

Program Interruptions: None.

PROGRAMMING NOTE

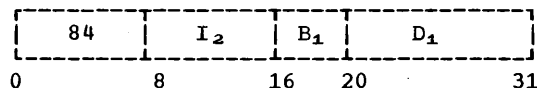
The data address obtained by Load Data Address, after an IOCE reset and prior to a SIO being issued, will be zeros. After a SIO has been issued, prior to a data transfer, the address will equal the CCW address. After data transfer, the address will refer to the next byte location to be used for data transfer to or from the multiplexor subchannel.

3.6.3 LOAD IDENTITY

The instruction is identical to the Load Identity instruction defined for the CE (see Section 8.4.5.2). IOCE-processors IOCE-1, IOCE-2, AND IOCE-3 take the identifying values 8, 9, and A (hexadecimal), respectively.

3.6.4 WRITE DIRECT

WRD D₁(B₁), I₂ [SI]



Bits 8-11 of the I₂ field specify an operation to be performed by the IOCE-processor. Bits 12-15 of the I₂ field are ignored.

Bit 8	Bit 9	Bit 10	Bit 11	Operation
0	0	0	0	Invalid
0	1	1	1	
1	0	0	0	CE external interrupt
1	0	0	1	Invalid
1	1	1	1	

If bits 8-11 specify a CE external interrupt then an external interrupt request is presented in the CE controlling the IOCE-processor.

If bits 8-11 specify an invalid operation a specification exception causes a program interrupt and the instruction is suppressed.

The content of the B₁, and D₁ fields is ignored. The location designated by the first operand address is, therefore, not accessed. A bit, which identifies the IOCE-processor executing this instruction, is

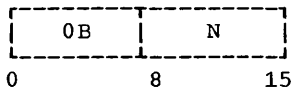
set in the processor interruption register of CE whose communication bit is set in the IOCE-processor's configuration control register. An external interruption is either taken or becomes pending in the controlling CE.

Condition Code: The code remains unchanged.

Program Interruptions:
Privileged Operation.
Specification

3.6.5 DELAY

DLY N [RR]



The operation of the IOCE-processor is delayed for approximately 256N microseconds (usec) where N is the binary value of bits 8-15 of the instruction.

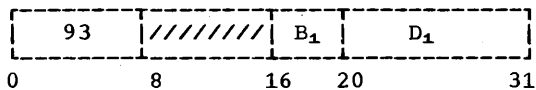
The R_1 and R_2 fields are treated as a single count field (N). A value of N equal to zero causes a zero delay.

The instruction is terminated either by the count (N) reaching zero or by the occurrence of an unmasked external or machine-check interruption condition. The timing of the delay is approximate, and may be increased by requests for service from the interval timer and I/O instructions.

Condition Code: The code remains unchanged.

3.6.6 TEST AND SET

TS $D_1(B_1)$ [SI]



The leftmost bit of the byte (bit 0) at the operand address sets the condition code and the entire byte is set to ones.

The byte in storage is fetched for the bit test and then set to all ones. No other access to this location is permitted between the moment of fetching and the moment of storing all ones.

Condition Code:
0 Leftmost bit of byte specified was zero.
1 Leftmost bit of byte specified was one.

Program Interruptions:
Addressing
Protection

JANUARY 31,1973

TNL: GN31-3005

PROGRAMMING NOTE

This byte is interpreted as a "lock" as follows:

0xxx	xxxx	Unlocked	No CE or IOCE has previously gained access to the "lock byte." The accessing CE or IOCE "locks out" all other CE's and IOCE's. The condition code is set to zero.
1xxx	xxxx	Locked	Another CE or IOCE has gained access to the "lock" byte. The accessing CE or IOCE is "locked out". The condition code is set to one

x denotes a bit position which may be either zero or one.

"Unlocking" of the byte consists of changing it from the 1xxx xxxx state to the 0xxx xxxx state.

4.0 IOCE-CHANNEL CONTROLLER OPERATION**4.1 CHANNEL ARCHITECTURE CHANGES**

The IOCE-channel controller function is modified by this RPQ to permit I/O data transfers between an I/O device and either main or MACH storage. The selection of storage is under program control, with bits 8 and 9 of the data address in the channel command word (CCW) designating the storage to be used. When bits 8 and 9 are not both ones, the data address refers to a main storage location; when bits 8 and 9 are both ones, the data address refers to a MACH storage location.

The information in the following sections supplements rather than replaces corresponding sections of Chapter 10.

When an IOCE is operating in the diagnostic mode, or 360-mode, bits 8 and 9 are always interpreted as the high order bits of the data address.

4.2 CHANNEL ADDRESS WORD

The channel address word (CAW) is unchanged as to format and general function, i.e., it specifies the main storage protection key and the address of the first CCW, associated with START I/O (see Section 10.3.2). Although the CAW is always fetched from the controlling CE's PSA (location 72) in main storage, the CCW address may reference main or MACH storage.

The protection key is matched with a key in storage whenever reference is made to main storage for a CCW or data. The key is ignored whenever reference is made to MACH storage.

4.3 CHANNEL COMMAND WORD

The channel command word (CCW) is unchanged as to format and general functions, i.e., it contains the command to be executed and the data address of the storage area, if any, to be used (see Section 10.3.3).

Bits 8-31 of the data address specify the location of an eight-bit byte in either MACH or main storage. It is the first location referenced in the area designated by the CCW. Bits 8 and 9 function as control bits. When bits 8 and 9 are not both ones, bits 8-31 refer to a location in main storage; when bits 8 and 9 both are ones, bits 10-31 refer to a location in MACH storage.

JUNE 1, 1971

During data chaining the new CCW fetched defines a new storage area for the original I/O operation. Data chaining of areas of main storage with areas of MACH storage is permitted.

Bits 48-63 of the CCW specify the number of eight-bit byte locations in either the MACH or main storage area designated by the CCW.

IMPLEMENTATION NOTE 1

The control function associated with bits 8 and 9 of the data address in the CCW is always active except when the IOCE is operating in "360 mode" (restricted to IOCE-1), or in Diagnostic Mode. In either case, bits 8 and 9 function as the high-order positions in the data address. References are made only to main storage when in 360-mode.

PROGRAMMING NOTE

The command address in the CAW designates the location of the first CCW in main or MACH storage. Since chaining takes place between CCW's located in successive double-word storage locations, if the first CCW is fetched from main storage the chained CCW's are also fetched from main storage. Conversely, when the first CCW is fetched from MACH storage, subsequent chained CCW's are fetched from MACH storage. Two chains of CCW's located in noncontiguous storage areas, including the case of areas in both main and MACH storage, can be coupled by a transfer in channel (TIC) command.

4.4 CHANNEL STATUS WORD

The channel status word (CSW) is unchanged as to format and general functions, i.e., it provides to the program status of an I/O device or conditions under which an I/O operation has been terminated. (See section 10.4.3) The CSW is always placed in the interrupted CE's preferential main storage at PSA location 64. No part of the CSW is stored in MACH location 64 unless the IOCE is running in Diagnostic Mode.

Conditions under which the CSW is stored, and those resulting in the PSA lockout exception are the same as for I/O operation using main storage. (See Chapter 10.)

Bits 0-3 of the CSW form the protection key used in the chain of operations at the subchannel or references to main storage. It has no significance when the command address and the data address in the last CCW used both reference MACH storage.

Bits 8-31 of the CSW form the command address for a main or MACH storage location that is eight higher than the address of the last CCW used.

5.0 SYSTEM MONITORING CONSIDERATIONS FOR IOCE-PROCESSOR

5.1 RESETS

The addition of the IOCE-processor RPQ results in the following additional actions to occur in an IOCE during system and subsystem reset:

- a. Parity is corrected on general purpose registers and the IOCE-processor PSW.
- b. IAR is reset to zeros.

JUNE 1, 1971

c. Pending interruption requests are cleared.

d. The IOCE-processor mode latch is set off, forcing the IOCE-processor into Stopped State.

(For actions that occur on system and subsystem reset, see Figures 26 and 27 in Chapter 12.)

5.2 RECORDING AND HANDLING CHECK CONDITIONS

5.2.1 RECORDING CHECK CONDITIONS

The recording of check conditions in the check registers (CR₁, CR₂) is retained. (See Section 12.2.1.) The fetch Data Check (CR₂, bit 4) is modified to reflect detection of incorrect parity on a data fetch from either main or MACH storage. Examination of the storage address register (SAR) in Word 81 of the logout enables the programmer to localize the check to the storage accessed.

5.2.2 SIGNALING CHECK CONDITIONS

The signaling of check conditions (OTC, OBS, and ELC) is unchanged. (See Section 12.2.2.)

5.2.3 IOCE MACHINE-CHECK HANDLING

The addition of the IOCE-processor capability does not appreciably change the way in which an IOCE handles a machine-check. Machine-checks detected during an IOCE-processor operation are handled by the common logic unit (CLU). Other than the addition of storing the old machine-check PSW in MACH location 48 at the completion of a successful logout, they are handled in the same way as those machine-check interruptions associated with IOCE-channel controller operations. (See Section 12.5.1.)

When the machine-check is such that a CLU logout occurs, IOCE-processor operation ceases and it goes into the stopped state at the completion of the logout. As in the present design, the effect of a CLU logout upon channel operation is not always predictable. When the malfunction is entirely associated with the selector channel hardware, a Selector Channel Logout is initiated. The affected channel terminates its operation, logs out, and resets. A concurrent IOCE-processor operation is suspended during the logout, and then resumed upon completion of the channel logout.

5.3 PRESERVATION OF CHECK DATA

5.3.1 CE LOGWORD FORMATS

The processor interruption register (PIR) is logged out in logword 39, bits 27-29.

5.3.2 IOCE LOGWORD FORMATS

The following additions were made to IOCE logword 88:

Bit 12 IOCE-PSW System Mask

Bits 13-16 IOCE-PSW bits A, M, W, P (when M is significant in Diagnostic Mode only.)

JUNE 1, 1971

Bit 17 IOCE-processor mode bit.

The following additions were made to IOCE logword 87:

Bits 18-19 CE Condition Code Register.

Bit 21 CE External Interrupt Request.

PROGRAMMING NOTE

The IOCE-processor mode bit (logword 88, bit 17), in conjunction with the I/O Routine Mode logout bit (word 109, bit 3), allows the program to determine whether a CLU logout was initiated by an IOCE-processor or an IOCE-input/output operation. (See Section 12.9.1, Programming Note 2.)

6.0 SYSTEM/360 MODE OF OPERATION

6.1 CE INSTRUCTION REPERTOIRE

The following operation codes are added to those which generate operation code exception conditions resulting in a program interruption when encountered in "360-mode":

START I/O PROCESSOR
MOVE WORD

6.2 IOCE-OPERATION

The IOCE-processor capability is disabled when an IOCE (applicable to IOCE-1 only) is placed in "360-Mode". The IOCE-processor is forced into the stopped state by hardware. "360-mode" operation is unchanged. (See Appendix G.)

JUNE 1, 1971

TABLE H-2 -- ALPHABETIC LIST OF IOCE-PROCESSOR INSTRUCTIONS						
NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	NOTES	
ADD	AR	RR	C	IF	1A	
ADD	A	RX	C	P A S	5A	MAIN
ADD HALFWORD	AH	RX	C	P A S	4A	MAIN
ADD LOGICAL	ALR	RR	C		1E	
ADD LOGICAL	AL	RX	C	P A S	5E	MAIN
AND	NR	RR	C		14	
AND	N	RX	C	P A S	54	MAIN
AND	NI	SI	C	P A	94	MAIN
AND	NC	SS	C	P A	D4	MAIN
BRANCH AND LINK	BALR	RR			05	
BRANCH AND LINK	BAL	RX			45	
BRANCH ON CONDITION	BCR	RR			07	
BRANCH ON CONDITION	BC	RX			47	
BRANCH ON COUNT	BCTR	RR			06	
BRANCH ON COUNT	BCT	RX			46	
BRANCH ON INDEX HIGH	BXH	RS			86	
BRANCH ON INDEX LOW OR EQUAL	BXLE	RS			87	
COMPARE	CR	RR	C		19	
COMPARE	C	RX	C	P A S	59	MAIN
COMPARE HALFWORD	CH	RX	C	P A S	49	MAIN
COMPARE LOGICAL	CLR	RR	C		15	
COMPARE LOGICAL	CL	RX	C	P A S	55	MAIN
COMPARE LOGICAL	CLI	SI	C	P A	95	MAIN
COMPARE LOGICAL	CLC	SS	C	P A	D5	MAIN
CONVERT TO BINARY	CVB	RX		P A S D IK	4F	MAIN
CONVERT TO DECIMAL	CVD	RX		P A S	4E	MAIN
DELAY	DLY	RR			0B	
DIAGNOSE	DI	SI		M P	83	SPR
DIVIDE	DR	RR		S IK	1D	
DIVIDE	D	RX		P A S IK	5D	MAIN
EXCLUSIVE OR	XR	RR	C		17	
EXCLUSIVE OR	X	RX	C	P A S	57	MAIN
EXCLUSIVE OR	XI	SI	C	P A	97	MAIN
EXCLUSIVE OR	XC	SS	C	P A	D7	MAIN
EXECUTE	EX	RX		P A S EX	44	MAIN
HALT I/O	HIO	SI		M	9E	SPR
INSERT CHARACTER	IC	RX		P A	43	MAIN

TABLE H-2 (continued)					
NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	NOTES
LOAD	LR	RR		18	
LOAD	L	RX	P A S	58	MAIN
LOAD ADDRESS	LA	RX	P	41	MAIN
LOAD AND TEST	LTR	RR	C	12	
LOAD COMPLEMENT	LCR	RR	C IF	13	
LOAD DATA ADDRESS	LDA	RS		99	
LOAD HALFWORD	LH	RX	P A S	48	MAIN
LOAD IDENTITY	LI	RR		0C	
LOAD MULTIPLE	LM	RS	P A S	98	MAIN
LOAD NEGATIVE	LNR	RR	C	11	
LOAD POSITIVE	LPR	RR	C IF	10	
LOAD PSW	LPSW	SI	L M P	82	MAIN LTD
MOVE WORD	MVW	SS	P A S	D8	MAIN
MOVE	MVI	SI	P A	92	MAIN
MOVE	MVC	SS	P A	D2	MAIN
MOVE NUMERICS	MVN	SS	P A	D1	MAIN
MOVE WITH OFFSET	MVO	SS	P A	F1	MAIN
MOVE ZONES	MVZ	SS	P A	D3	MAIN
MULTIPLY	MR	RR	S	1C	
MULTIPLY	M	RX	P A S	5C	MAIN
MULTIPLY HALFWORD	MH	RX	P A S	4C	MAIN
OR	OR	RR	C	16	
OR	O	RX	C P A S	56	MAIN
OR	OI	SI	C P A	96	MAIN
OR	OC	SS	C P A	D6	MAIN
PACK	PACK	SS	P A	F2	MAIN
SET PCI	SPCI	SI	M	9B	SPR
SET PROGRAM MASK	SPM	RR	L	04	
SET SYSTEM MASK	SSM	SI	M P	80	MAIN LTD
SHIFT LEFT DOUBLE	SLDA	RS	C S IF	8F	
SHIFT LEFT DOUBLE LOGICAL	SLDL	RS	S	8D	
SHIFT LEFT SINGLE	SLA	RS	C IF	8B	
SHIFT LEFT SINGLE LOGICAL	SLL	RS		89	
SHIFT RIGHT DOUBLE	SRDA	RS	C S	8E	
SHIFT RIGHT DOUBLE LOGICAL	SRDL	RS	S	8C	
SHIFT RIGHT SINGLE	SRA	RS	C	8A	
SHIFT RIGHT SINGLE LOGICAL	SRL	RS		88	
START I/O	SIO	SI	M	9C	SPR
STORE	ST	RX	P A S	50	MAIN
STORE CHARACTER	STC	RX	P A	42	MAIN
STORE HALFWORD	STH	RX	P A S	40	MAIN
STORE MULTIPLE	STM	RS	P A S	90	MAIN
SUBTRACT	SR	RR	C IF	1B	
SUBTRACT	S	RX	C P A S IF	5B	MAIN
SUBTRACT HALFWORD	SH	RX	C P A S IF	4B	MAIN
SUBTRACT LOGICAL	SLR	RR	C	1F	
SUBTRACT LOGICAL	SL	RX	C P A S	5F	MAIN
SUPERVISOR CALL	SVC	RR		0A	

JUNE 1, 1971

TABLE H-2 (continued)					
NAME	MNEMONIC	FORMAT	EXCEPTIONS	CODE	NOTES
TEST AND SET	TS	SI	C P A	93	MAIN
TEST CHANNEL	TCH	SI	M	9F	SPR
TEST I/O	TIO	SI	M	9D	SPR
TEST UNDER MASK	TM	SI	C P A	91	MAIN
TRANSLATE	TR	SS	P A	DC	MAIN
TRANSLATE AND TEST	TRT	SS	C P A	DD	MAIN
UNPACK	UNPK	SS	P A	F3	MAIN
WRITE DIRECT	WRD	SI	M S	84	

LEGEND:

A Addressing exception
 C Condition code is set
 D Data exception
 EX Execute exception
 IF Fixed-point overflow exception
 IK Fixed-point divide exception
 L New condition code loaded
 LS Significance exception
 M Privileged-operation exception
 P Protection exception
 SPR Operation suppressed; instruction not executable in operational mode.
 Operation exception interruption.
 MAIN Protection on main storage access only
 LTD Function limited
 S Specification exception

JUNE 1, 1971

TABLE H-3 -- PROGRAMMING EXCEPTIONS IN THE IOCE-PROCESSOR

1. Operation Exception

When an operation code is not assigned, an operation exception is recognized. The operation is suppressed.

The instruction-length code in the IOCE is 1, 2, or 3.

The following instructions are not available in the IOCE. When encountered in a program an operation exception is recognized.

<u>Name</u>	<u>Op. Code</u>
a. Decimal Arithmetic Instructions	
ADD DECIMAL	FA
SUBTRACT DECIMAL	FB
ZERO AND ADD	F8
COMPARE DECIMAL	F9
MULTIPLY DECIMAL	FC
DIVIDE DECIMAL	FD
b. Logical Instructions	
EDIT	DE
EDIT AND MARK	DF
c. Status-Switching Instructions	
SET STORAGE KEY	08
INSERT STORAGE KEY	09
DIAGNOSE	83
READ DIRECT	85
d. Multiple Computing Element Instructions	
LOAD PS BASE ADDRESS	A1
SET CONFIGURATION	01
STORE PS BASE ADDRESS	A0
SET ADDRESS TRANSLATOR	0D
INSERT ADDRESS TRANSLATOR	0E
START I/O PROCESSOR	9A
e. Floating-Point Instructions	
ALL	
f. I/O Instructions	
ALL	
g. Display Instructions	
ALL	

2. Privileged-Operation Exception (M)

When a privileged instruction is encountered in an IOCE program, a privileged-operation exception is recognized. The operation is suppressed.

The instruction-length code in the IOCE is 1 or 2.

The following instructions are in this category:

LOAD PSW
SET SYSTEM MASK
WRITE DIRECT

JUNE 1, 1971

3. Execute Exception (EX)

| When the subject instruction of EXECUTE is another EXECUTE, an
| execute exception is recognized. The operation is suppressed.

The instruction-length code in the IOCE is 2.

4. Protection Exception (P)

When the storage key of an accessed location in main storage does not match the protection key in the current PSW in the IOCE, a protection exception is recognized. A protection exception is also recognized if the IOCE-processor attempts to use the key F (hexadecimal).

| The instruction-length code in the IOCE is 0, 2 or 3.

This exception can occur on: The instructions designated in Table H-2.

The protection exception resulting from the use of key F will occur immediately upon setting the processor key to F via Load PSW instruction or interruption.

5. Addressing Exception (A)

When an address specifies any part of data or an instruction outside of MACH, or outside the available main storage, or outside the configured main storage, or main storage assigned by the storage address translator for a particular IOCE, an addressing exception is recognized.

The operation is terminated for an invalid address. Data in storage remain unchanged, except when designated by valid addresses. The operation is suppressed for an invalid instruction address. The instruction-length code in the IOCE normally 1, 2, or 3, but may be 0

JUNE 1, 1971

inthe case of a data address.

The following instructions are in this category:

<u>Name</u>	<u>Op. Code</u>	<u>Name</u>	<u>Op. Code</u>
ADD	5A	MOVE	92
ADD HALFWORD	4A	MOVE	D2
ADD LOGICAL	5E	MOVE NUMERICS	D1
		MOVE WITH OFFSET	F1
AND	54	MOVE ZONES	D3
AND	94	MOVE WORD	D8
AND	D4		
COMPARE	59	MULTIPLY	5C
COMPARE HALFWORD	49	MULTIPLY HALFWORD	4C
COMPARE LOGICAL	55		
COMPARE LOGICAL	95	OR	56
COMPARE LOGICAL	D5	OR	96
		OR	D6
CONVERT TO BINARY	4F		
CONVERT TO DECIMAL	4E	PACK	F2
DIVIDE	5D	STORE	50
		STORE CHARACTER	42
EXCLUSIVE OR	57	STORE HALFWORD	40
EXCLUSIVE OR	97	STORE MULTIPLE	90
EXCLUSIVE OR	D7		
		SUBTRACT	5B
EXECUTE	44	SUBTRACT HALFWORD	4B
		SUBTRACT LOGICAL	5F
INSERT CHARACTER	43		
		TEST AND SET	93
LOAD	58	TEST UNDER MASK	91
LOAD HALFWORD	48		
LOAD MULTIPLE	98	TRANSLATE	DC
		TRANSLATE AND TEST	DD
		UNPACK	F3

6. Specification(S)

A data, instruction or control-word address does not specify an integral boundary for the unit of information.

Or the R_1 field of an instruction specifies an odd register address for a pair of general registers which contain a 64-bit operand.

Or bits 8-11 of the I_2 field of the Write Direct specify an invalid operation.

Instruction-length code in the IOCE is 1, 2, or 3.

JUNE 1, 1971

The following instructions are in this category:

<u>Name</u>	<u>Op. Code</u>	<u>Name</u>	<u>Op. Code</u>
ADD	5A	MOVE WORD	D8
ADD HALFWORD	4A		
ADD LOGICAL	5E		
AND	54	MULTIPLY	1C
		MULTIPLY	5C
		MULTIPLY HALFWORD	4C
COMPARE	59		
COMPARE HALFWORD	49	OR	56
COMPARE LOGICAL	55		
CONVERT TO BINARY	4F	SHIFT LEFT DOUBLE	8F
CONVERT TO DECIMAL	4E	SHIFT LEFT DOUBLE LOGICAL	8D
		SHIFT RIGHT DOUBLE	8E
		SHIFT RIGHT DOUBLE LOGICAL	8C
DIVIDE	1D		
DIVIDE	5D	STORE	50
		STORE HALFWORD	40
EXCLUSIVE OR	57	STORE MULTIPLE	90
EXECUTE	44	SUBTRACT	5B
		SUBTRACT HALFWORD	4B
LOAD	58	SUBTRACT LOGICAL	5F
LOAD HALFWORD	48		
LOAD MULTIPLY	98	WRITE DIRECT	84

7. Data Exception (D)

A data exception is recognized when the sign or digit codes or operands in CONVERT TO BINARY are incorrect.

The operation is terminated.

The instruction-length code is 2.

8. Fixed-point Overflow Exception (F)

When a high-order carry occurs or high-order significant bits are lost in fixed-point add, subtract, shift, or sign-control operations, a fixed-point overflow exception is recognized.

The operation is completed by ignoring the information placed outside the register. The interruption may be masked by PSW bit 36 in the IOCE.

The instruction-length code in the IOCE is 1 or 2.

The following instructions can cause this exception to occur:

<u>Name</u>	<u>Op. Code</u>	<u>Name</u>	<u>Op. Code</u>
ADD	1A	SHIFT LEFT DOUBLE	8F
ADD	5A	SHIFT LEFT SINGLE	8B
ADD HALFWORD	4A		
LOAD COMPLEMENT	13	SUBTRACT	1B
LOAD POSITIVE	10	SUBTRACT	5B
		SUBTRACT HALFWORD	4B

9. Fixed-Point-Divide Exception (IK)

A fixed-point-divide exception is recognized when quotient exceeds the register size in fixed-point division, including division by zero.

JUNE 1, 1971

Or the result of CONVERT TO BINARY exceeds 31 bits.

Division is suppressed. Conversion is completed by ignoring the information placed outside the register.

The instruction-length code in the IOCE is 1 or 2.

<u>Name</u>	<u>Op. Code</u>
CONVERT TO BINARY	4F
DIVIDE	1D
DIVIDE	5D

10. SE-Stopped Exception

When an IOCE-processor attempts to gain access to a main storage element which is in logout-stop state, the SE-stopped exception is recognized.

- | The operation is suppressed on I-Fetch, terminated on Data Address.
- | The instruction-length code is 1, 2 or 3.

JUNE 1, 1971

TABLE H-4 IOCE PROCESSOR INSTRUCTION TIMINGS

1. Introduction

The actual time for execution of a particular instruction depends upon a great many factors, most of which are not of direct concern to the programmer. Hence, the times stated are average values, based on the assumptions stated below. In general, the effects of data format and indexing are stated explicitly, while variations due to operand values and instruction location are included in the averages. Unless stated otherwise, the following assumptions have been made in computing average instruction times.

- 1) Instructions start on even and odd halfwords with equal probability.
- 2) In data fields (except decimal) each bit position has equal probability for values of one or zero, and is independent of other bit positions.
- 3) Positive and negative operands are equally probable.
- 4) No instruction refetch occurs. An instruction refetch occurs if bits 10-12 and 18-20 of the address of an instruction that starts in an odd halfword match the corresponding bits of data address used by the previous instruction. If the instruction has the RR format, a refetch requires 1.0 usec additional. For other instruction formats, a refetch requires 2.5 usec.
- 5) No delays in accessing shared storage are caused by references to the same storage unit by other CE's or IOCE's.
- 6) No interval timer updating occurs. (Each updating requires an average of 7 usec.)
- 7) Timings assume instructions and operands are in MACH. If otherwise, add 0.5 usec for each storage cycle in main storage.

No additional time is required to add the contents of a base register B in computing the effective address of any instruction. In the RX format the addition of the contents of an index register X in computing the effective address requires 0.5 usec.

JUNE 1, 1971

2. Fixed-Point Arithmetic Instructions

INSTRUCTION	FORMAT	MNEMONIC	TIME (usec)
LOAD DATA ADDRESS	RS	LDA	5.00
LOAD	RR	LR	2.50
LOAD	RX	L	4.00
LOAD HALFWORD	RX	LH	4.75
LOAD AND TEST	RR	LTR	2.50
LOAD COMPLEMENT	RR	LCR	2.75
LOAD POSITIVE	RR	LPR	3.00
LOAD NEGATIVE	RR	LNR	2.88
LOAD MULTIPLE	RS	LM	3.00+2.00R
ADD	RR	AR	3.25+E
ADD	RX	A	4.00+E
ADD HALFWORD	RX	AH	5.50+E
ADD LOGICAL	RR	ALR	3.25
ADD LOGICAL	RX	AL	4.00
SUBTRACT	RR	SR	3.25+E
SUBTRACT	RX	S	4.00+E
SUBTRACT HALFWORD	RX	SH	5.50+E
SUBTRACT LOGICAL	RR	SIR	3.25
SUBTRACT LOGICAL	RX	SL	4.00
COMPARE	RR	CR	3.25
COMPARE	RX	C	4.00
COMPARE HALFWORD	RX	CH	6.00
MULTIPLY	RR	MR	26.13-1.50K
MULTIPLY	RX	M	27.38-1.50K
MPY HALFWORD	RX	MH	22.50-1.50K
DIVIDE	RR	DR	28.13
DIVIDE	RX	D	28.88
STORE	RX	ST	4.00
STORE HALFWORD	RX	STH	5.00
STORE MULTIPLE	RS	STM	3.00+2.00R
CONVERT TO DECIMAL	RX	CVD	
Case 1			13.00+2.00J
Case 2			23.25+2.00J
Case 3			28.75+2.00J
CONVERT TO BINARY	RX	CVB	
Case 1			10.75+2.00J
Case 2			17.50+2.00J
Case 3			23.75+2.00J

Where:

E = 1 if masked-off overflow occurs; 0 otherwise.

J = Number of significant hexadecimal digits in binary quantity.

K = Number of zero hexadecimal digits in absolute value of multiplier. Factor with the smallest absolute value is used as multiplier.

R = Number of registers transferred.

Case 1 occurs if the CVD result or the CVB operand contains eight decimal digits or less.

Case 2 occurs during CVD if the number converted from seven or less hexadecimal digits contains more than eight decimal digits, or during

JUNE 1, 1971

CVB if the number converted from more than eight decimal digits contains seven or less hexadecimal digits.

Case 3 occurs if the CVD operands or the CVB result contains more than seven hexadecimal digits.

3. Shift Instructions

Shifts are accomplished by a number of one-bit shifts followed by a number of four-bit shifts. The shift amount is stated in the instruction as a six-bit quantity. The high-order four bits (P) of this quantity determine the number of four-bit shifts. The low-order two bits (Q) determine the number of one-bit shifts.

INSTRUCTION	MNEMONIC	TIME (usec.)
SHIFT LEFT SINGLE (logical)	SLL	$4.00+0.50P+0.50Q+0.50S(Q)$
SHIFT LEFT DOUBLE (logical)	SLDL	$4.00+1.00P+1.00Q+0.50S(Q)$
SHIFT RIGHT SINGLE (logical)	SRL	$4.00+0.50P+0.50Q+0.50S(Q)$
SHIFT RIGHT DOUBLE (logical)	SRDL	$4.00+1.00P+1.00Q+0.50S(Q)$
SHIFT LEFT SINGLE (algebraic)	SLA	$4.00+0.50P+0.50Q+0.50S(Q)+0.50A$
SHIFT LEFT DOUBLE (algebraic)	SLDA	$5.00+1.00P+1.00Q+2.00C$ $+0.50(S(Q)(1-C))+0.50Z$
SHIFT RIGHT SINGLE (algebraic)	SRA	$4.00+0.50P+0.50Q+0.50S(Q)$
SHIFT RIGHT DOUBLE (algebraic)	SRDA	$4.00+1.00P+1.00Q+0.50S(P)$ $+0.50S(Q)+0.50Z$

Where:

P = Number of four-bit shifts.
Q = Number of one-bit shifts.
Z = 1 if high order part of result is zero; 0 otherwise.
C = 1 if operand is negative; 0 otherwise.
A = 1 if Q ≠ 0 and operand is negative; 0 otherwise.
S(X) is a function defined as S(X) = 1 if X = 0;
S(X) = 0 if X ≠ 0.

4. Logical Instructions

The instruction times stated here for the SS format instructions are approximations adequate for normal use. Although reasonably accurate in the average, individual cases may differ from these approximations by as much as 20 percent.

JUNE 1, 1971

INSTRUCTION	FORMAT	MNEMONIC	TIME (usec)
INSERT CHARACTER	RX	IC	5.00
LOAD ADDRESS	RX	LA	2.75
MOVE	SS	MVC	10.69+1.00N
MOVE	SI	MVI	4.50
MOVE NUMERIC	SS	MVN	12.19+1.50N
MOVE ZONE	SS	MVZ	12.19+1.50N
COMPARE	SS	CLC	10.00+1.00B
COMPARE	SI	CLI	4.50
COMPARE LOGICAL	RR	CLR	3.00
COMPARE LOGICAL	RX	CL	4.00
AND	SS	NC	12.19+1.50N
AND	SI	NI	6.50
AND	RR	NR	5.00
AND	RX	N	5.75
OR	SS	OC	12.19+1.50N
OR	SI	OI	6.50
OR	RR	OR	5.00
OR	RX	O	5.75
EXCLUSIVE OR	SS	XC	12.19+1.50N
EXCLUSIVE OR	SI	XI	6.50
EXCLUSIVE OR	RR	XR	5.00
EXCLUSIVE OR	RX	X	5.75
STORE CHARACTER	RX	STC	4.50
TRANSLATE	SS	TR	6.00+4.50N
TRANSLATE AND TEST	SS	TRT	7.75+3.00B
TEST UNDER MASK	SI	TM	5.50-0.50Z

Where:

B = Number of bytes processed
 N = Number of bytes in field
 Z = 1 if all selected bits are zero; 0 otherwise.

5. Decimal Arithmetic Instructions

The instruction times stated here are approximations adequate for normal use. Although reasonably accurate in the average, individual cases may differ from these approximations by as much as 20 percent.

All times are in microseconds.

Overlapped data fields are not considered.

JUNE 1, 1971

INSTRUCTION	FORMAT	MNEMONIC	TIME (usec)
MOVE WITH OFFSET	SS	MVO	11.88+1.00N1 +0.88N2
PACK	SS	PACK	10.38+1.31N1 +0.75N2
UNPACK	SS	UNPK	9.88+1.00N1 +0.63N2

Where:

N1 = Number of destination field bytes processed.
N2 = Number of source field bytes processed.

6. Branching and Status-Switching Instructions

INSTRUCTION	FORMAT	MNEMONIC	TIME (usec)
BRANCH ON CONDITION	RR	BCR	2.75+1.00B
BRANCH ON CONDITION	RX	BC	3.00+1.00B
BRANCH ON COUNT	RR	BCTR	3.25+1.00B -0.50G
BRANCH ON COUNT	RX	BCT	3.50+1.00B
BRANCH AND LINK	RR	BALR	4.25-1.25G
BRANCH AND LINK	RX	BAL	4.50
BRANCH ON INDEX HIGH	RS	BXH	4.50+1.00B
BRANCH ON INDEX LOW EQUAL	RS	BXLE	4.50+1.00B
EXECUTE OF RR	RX	EX	4.75+E
OF RX, RS, OR SI			5.00+E
OF CHARACTER SS			6.50+E
SET SYSTEM MASK	SI	SSM	5.50
SET PROGRAM MASK	RR	SPM	2.75
SUPERVISOR CALL (NOTE 1)	RR	SVC	12.75
LOAD PSW	SI	LPSW	7.50
WRITE DIRECT	SI	WRD	6.00

Note 1 -- Including time for program interruption.

Where:

B = 1 if successful branch; 0 otherwise.
E = Time for the subject instruction.
G = 1 if branch suppressed ($R_2 = 0$); 0 otherwise.

JUNE 1, 1971

7. Multiple Computing Element Instruction

INSTRUCTION	FORMAT	MNEMONIC	TIME (usec)
DELAY	RR	DLY	4.0+Length of delay
TEST AND SET	SI	TS	5.0
LOAD IDENTITY	RR	LI	3.0
MOVE WORD	SS	MVW	4.50+4.00W

Where:
W = Number of words transferred.

JANUARY 31,1973

TNL: GN31-3005

APPENDIX I. IBM 7289-04 DISPLAY ELEMENT1.1 INTRODUCTION

This appendix is provided to describe the basic operational concept of a display element to the programmer. The hardware design of this element is discussed only where there are programming restrictions or programming requirements or, where a discussion is necessary for the purpose of conceptual clarity.

Descriptions of DE addressing, configuration control, and interface priority may be found in Chapters 2, 8, and 13, respectively.

2.1 SYSTEM CHARACTERISTICS

The IBM 7289-04 Display Element is used within the 9020E System to provide storage for image data, and the controls to allow each assigned Plan View Display (PVD) to synchronously regenerate its display images at a predetermined maximum cyclical rate. Each display element communicates with both CEs and Display Generators (DG). Housed within each DG are six Character Vector Generators (CVG). The CVGs contain control logic for the PVDs. One PVD is assigned to each CVG, six CVGs are housed within each DG. Since up to four DGs may be configured to a DE, a maximum of twenty-four PVDs may be assigned to each DE at any one time. (See Figure I-1).

NOTE: The designation DCVG (Display Control Vector Generator) used in some documents is generally synonymous with CVG as used herein.

To a CE, an attached DE appears as an extension of System Storage. To a DG, the DE appears as an I/O control device feeding image and control data in response to requests from the CVGs and controlling the regeneration of image data. As an extension of System Storage, the DE, for all intents and purposes, appears as a Storage Element. As a control device the DE once started needs no further programmer initiated action to continuously regenerate images on the PVD via the CVGs contained in the DG. The programmer does however control the image seen on a PVD by properly loading control and image data into the DE as a normal storage operation.

The programmer initializes, starts, generates and manages up to 24 display images contained in a DE. He does this by first loading the initialization area of a DE, Quadwords 0-49, as described in section 4.1.1 (Image Initialization of this Appendix). He then starts regeneration by configuring the desired PVD to the DE via the Configuration Commands described in Chapter 8. Once configured the DE will commence accepting CVG requests for data from the CVG configured to the PVD via the DG configured to the DE. The DE will respond to these requests by sending programmer formatted doublewords to the DG in a rigid priority sequence. The programmer generates and manages the image appearing on a PVD by loading into the DE image data formatted as described in section 3.1.3, "DATA FORMATS", of this Appendix, and control data as described in section 3.1.4, "COMMANDS", of this Appendix.

As a System Element, the priority with which each CVG request is serviced by the DE is of interest to the programmer. No CVG has a preferred priority. The assigned CVGs asynchronously request data. All CVGs are serviced by the DE in Quadword increments with each CVG allocated a predetermined fixed Quadword slot in a continuously revolving-

ing priority scan. If a CVG is not requesting data at the time its Quadword slot appears in the priority scan it is not given to another requesting CVG. The unused Quadword slot will be made available to a

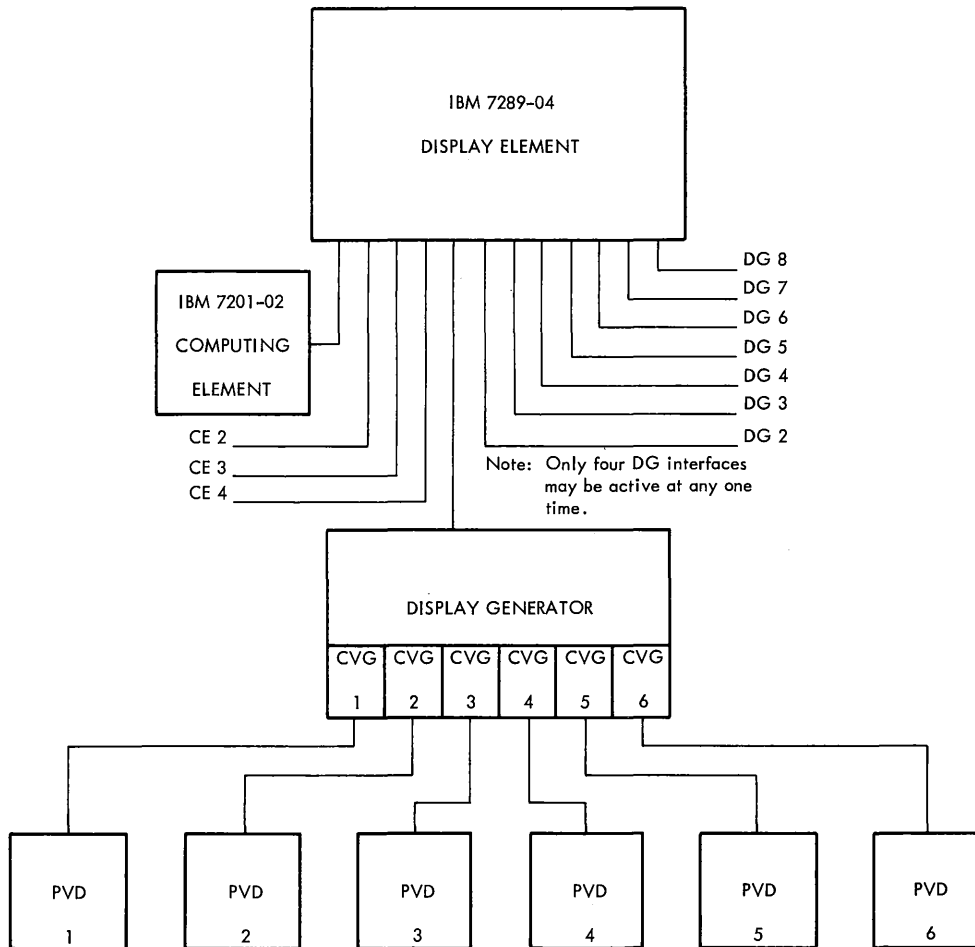


Figure I-1. Display Element Configuration

requesting CE. The DE by responding this way to asynchronous CVG requests, forces the display sub-system into a synchronous mode of operation. The minimum regeneration period is fixed by a clock in the DE for those cases where the PVD displays its data in less time than the minimum regeneration period. For those cases where the PVD takes more time than the minimum regeneration period, the regeneration period ends when an End of Display Command (EOD, see section 3.1.4.2 of this Appendix) is detected by the DE.

3.1 DATA ORGANIZATION

3.1.1 DATA MANAGEMENT

The data organization within a DE is basically a revolving storage buffer approach, each variable sized buffer containing the current image data for a particular PVD (See Figure I-2). For ease of data management and for efficient regeneration, each image is regenerated into a new

JANUARY 31,1973

TNL: GN31-3005

storage buffer during each programmer initiated update cycle. As the new image is being constructed by the programmer, the DE is providing

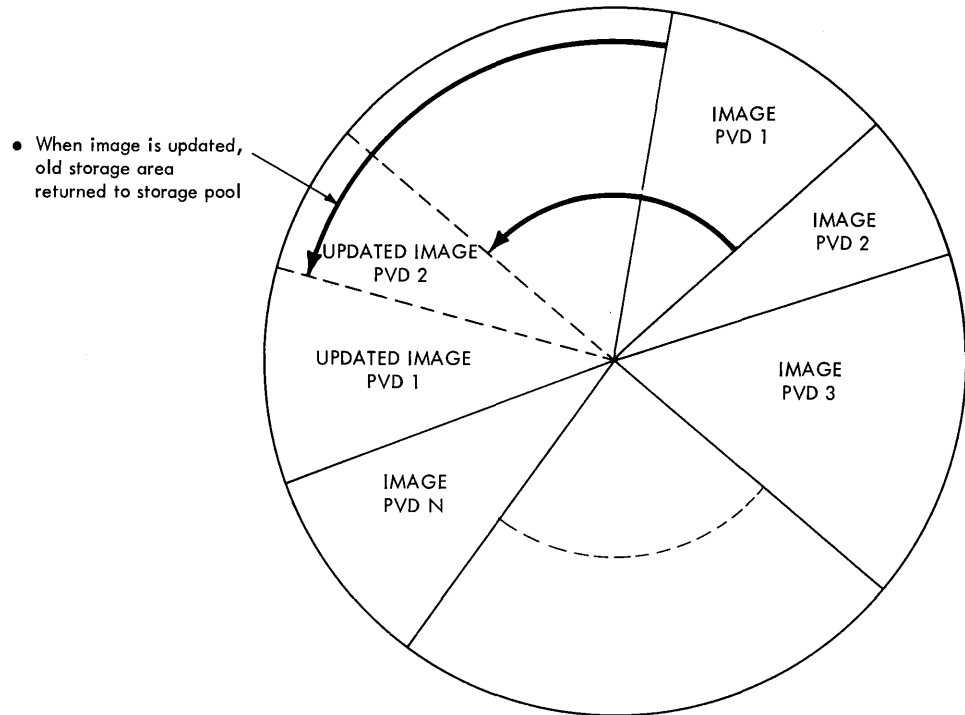


Figure I-2 Display Element Organization

regeneration data access cycles from the current data image. Upon completion of the new data image for a particular PVD, the programmer will direct the DE to use the new image for subsequent PVD regeneration cycles and will then return the storage buffer (which contained the previous image) to the available storage buffer pool.

3.1.2 DISPLAY IMAGE

The data image for a particular PVD contains the data to be displayed plus that control information (in the form of data) required by the DE and/or CVG. An image consists of a Start of Display (SOD) command, data to be displayed (variable length), and an End of Display (EOD) Command. Within a display image, TRANSFER (TFR) commands may also be inserted to allow for non-contiguous storage images. (see Figure I-3)

The placement of the SOD, EOD, and TFR (if applicable) commands, and the image data is performed by the programmer via storage operations.

NOTE: The term display image as used in this text refers loosely to the SOD command as part of the image, whereas it is actually not part of the image data area per se, but part of the hardware addressed control area for the DG. The DE does not detect an incorrectly formatted display image and the effect of such an image on the particular PVD is unpredictable.

3.1.3 DATA FORMATS

The display data contained within a DE must be of a format which is acceptable to the display subsystem. This format is based on a display word which is a doubleword in length (eight bytes).

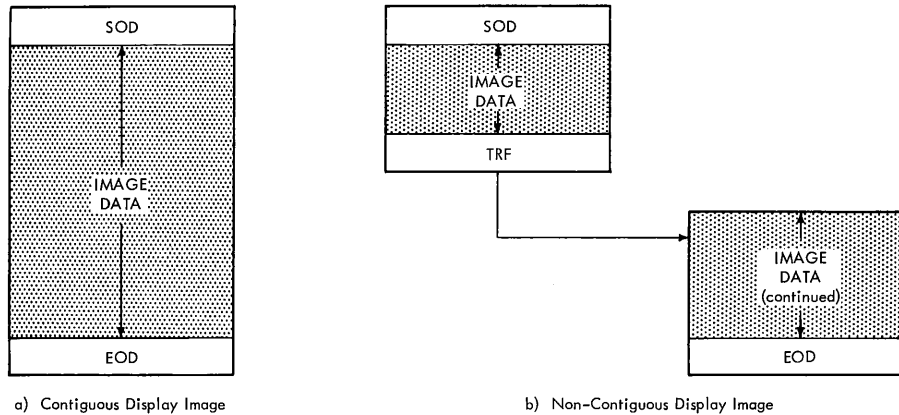


Figure I-3 Display Image

The high order bit of the display doubleword (bit 0) contains a parity bit for bits 1-63. This parity is ignored by the DE and may contain either a one or zero and provide either odd or even parity. Upon fetching display data for a CVG and before transferring it to a DG, the DE will assign correct parity to the doubleword for presentation to the display subsystem.

Data fetched in response to a CVG request is always performed on a quadword basis from storage. This quadword consists of contiguous even-odd addressed doublewords. When transferred to a DG, the data is always transferred in this sequence, even-odd.

NOTE#: The programmer must load data into the DE on quadword boundaries. The effect of violating this rule is unpredictable if the image is currently being displayed.

A DE is not sensitive to data format with the exception of the EOD and TFR command, and in these commands, only to those bits specified in the following section.

3.1.4 COMMANDS

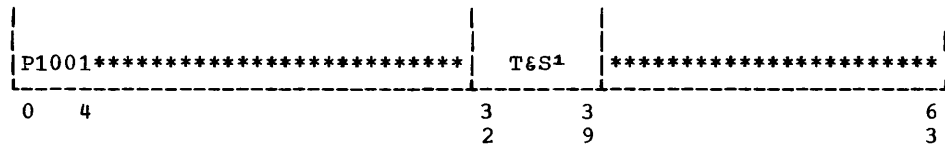
The three display control commands used with each data image are Start of Display (SOD), End of Display (EOD), and TRANSFER (TFR)

3.1.4.1 Start of Display Command

The SOD command is used by the display subsystem and must be the first data field associated with the image. The DE does not detect this command either as the first data field or elsewhere within the image. The format of the SOD command is as follows:

JANUARY 31, 1973

TNL: GN31-3005



SOD Command Format

* = don't care bit with respect to the DE.

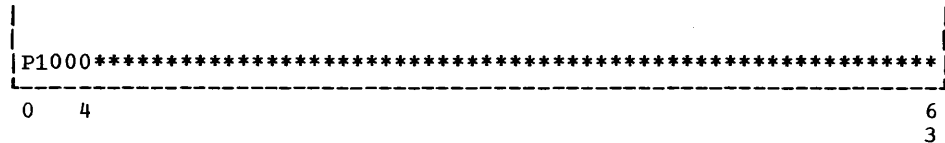
NOTE: Bits 5-31 and 40-63 may be significant to the Display Sub-system
 Bit 0 is assigned odd parity for the entire doubleword.

¹ = See section 4.1.2

The entire SOD command (bits 0-63) is transferred to a DG by the DE with no modification (except the parity bit which is assigned enroute to the DG).

3.1.4.2 End of Display Command

The EOD command is used by both the display subsystem and the DE. This command is detected by the DE and indicates the end of the display image. Subsequent data transfers from this image will reference the beginning of the image data. The format of the EOD command is as follows:



EOD Command Format

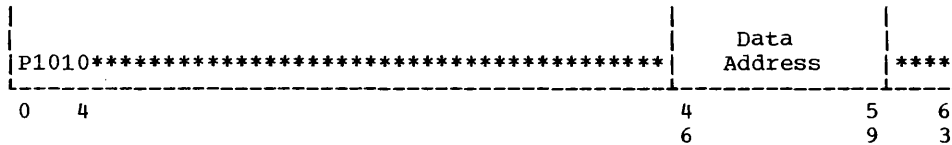
* = don't care bit with respect to the DE.

Note: Bits 5-63 may be significant to the Display Sub-system.:
 Bit 0 is assigned odd parity for the entire doubleword.

Since the storage cycles for a CVG request are performed by quadwords, the DE will transfer two doublewords even upon detection of an EOD command. If the EOD command is contained in the first doubleword (even doubleword address), the DE will transfer it twice.

3.1.4.3 Transfer Command

The TRF Command is recognized by both the DE and the DG. This command is always detected by the DE which internally executes a transfer operation to the indicated address within the DE. The format of the TFR command is as follows:



TFR Command Format

* = don't care bit with respect to the DE.

Note: Bits 5-45 and 60-63 may be significant to the Display sub-system.

Bit 0 is assigned odd parity for the entire doubleword.

PROGRAMMING NOTE:

The Transfer command has a format which is recognizable to the DG as a Header Word. Therefore, when Header Words are used in the DG image field (within the DE) they will be treated as Transfer Commands.

The placement of the TFR command(s) within a display image is critical. The DE interrogates both the odd and the even doubleword of each quadword for a TFR command. If the even doubleword (first doubleword) contains a transfer, the DE will transfer this doubleword to the DG twice - both as the even and odd doubleword. The odd doubleword fetched from storage is ignored in this case.

All TFR commands should contain valid quadword addresses within the address field (bits 46-59). The high order bits (bits 5-45) and the low order bits (bits 60-63) are, however ignored and will not cause any invalid address if set. It should be noted that a TFR to a specified doubleword address will result in a transfer to the truncated quadword address (even doubleword which precedes the specified odd doubleword).

3.1.5 ADDRESS WRAP-AROUND

The DE upon encountering the last address contained within its storage, will wrap-around the addressing to 0. This address location, as will be discussed under OPERATIONAL CHARACTERISTICS, IS A DE control area which is not available for image data. Consequently, a TFR command specifying the address of the continued image data must be stored in the last doubleword of the DE if it becomes necessary to store an image across this boundary.

3.1.6 PVD TEST WORD

It is recommended that a PVD Test Word follow the SOD in the scheme of data management presented in this section. This word is used for test purposes by the PVD. Its format is determined by the requirements of the Display Sub-system.

4.1 OPERATIONAL CHARACTERISTICS

All of the storage contained within a display element, with the exception of the first fifty quadwords, is available to the programmer for the storing of image data. These fifty quadwords are used for programmed communications between the operational program and the control logic within the DE. A storage map of this control area is presented in Figure I-4.

JUNE 1, 1971

QUADWORD	CVG	0	23	63	0	63
0		IMAGE SWITCH	NOT USED	NOT USED		
1			NOT USED			NOT USED
2			SOD			TEST
3	1A {		TRACK BALL			TRF
4						
5	2A {					
6						
7	3A {					
8						
9	4A {					
10						
11	5A {					
12						
13	6A {					
14						
15	1B {					
16						
17	2B {					
18						
19	3B {					
20						
21	4B {					
22						
23	5B {					
24						
25	6B {					
26						
27	1C {					
28						
29	2C {					
30						
31	3C {					
32						
33	4C {					
34						
35	5C {					
36						
37	6C {					
38						
39	1D {					
40						
41	2D {					
42						
43	3D {					
44						
45	4D {					
46						
47	5D {					
48						
49	6D {		SOD			TEST
			TRACK BALL			TRF

Figure I-4 Display Element Control Area

4.1.1 IMAGE INITIALIZATION

Quadwords 2-49 within each DE are used as display initialization areas. These quadwords are assigned to particular CVGs in the following manner;

<u>QUADWORD</u>	<u>CVG</u>
2-13	1-6 of DG configured to Data Register A.
14-25	1-6 of DG configured to Data Register B.
26-37	1-6 of DG configured to Data Register C.
38-49	1-6 of DG configured to Data Register D.

It will be noted that each set of twelve quadwords (e.g., 14-25) is assigned to a particular DG Data Register and not to a particular display generator. The assignment, in turn, of a particular DG to a DG Data Register is performed via the Configuration Control Register (See Chapter 8).

Each pair of quadwords should be initialized by the programmer to contain an SOD command, a PVD test word, a trackball word, and a TFR command if the related CVG/PVD is to be operational. The TFR command must contain the storage address within that DE at which the display image for the particular PVD resides. The DE, upon starting a regeneration cycle (55 regeneration cycles per second maximum), will fetch the first quadword, of the quadword pair assigned to that CVG, from its particular location within the control area. This characteristic of the DE is a hardware function and cannot be modified by the programmer i.e., there is a constant start position for each assigned CVG. (See Figure I-5)

It is through this initialization area that a programmer redirects the DE to a new, updated image for each individual CVG (PVD). It should be noted however, that the DE references this initialization control quadword for a CVG only at the start of a regeneration cycle (55 cps maximum). The DE does not reference these initialization double quadwords while in the regeneration cycle.

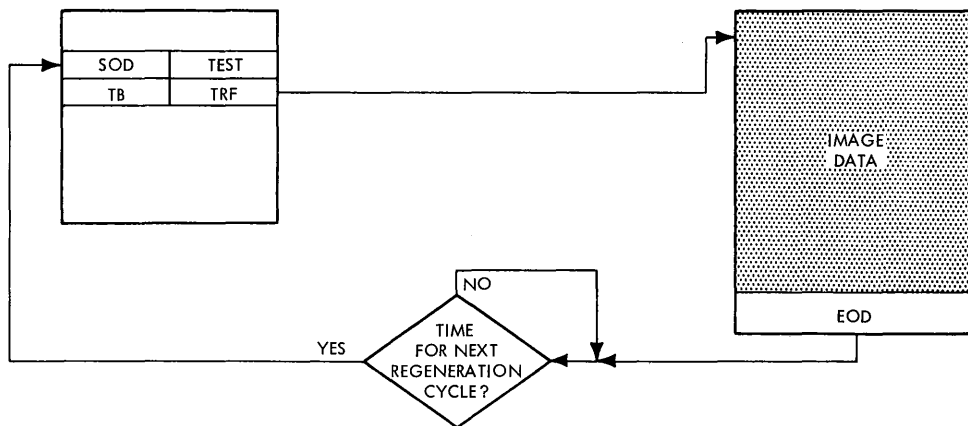


Figure I-5 Use of Image Initialization Area

JANUARY 31, 1973

TNL: GN31-3005

4.1.2 TEST AND SET

A TEST AND SET like function is provided on a CVG basis within the initialization quadwords. Byte 4 (bits 32-39) of the SOD command (doubleword one of the double quadword set) is used for this purpose. See Figure I-6.

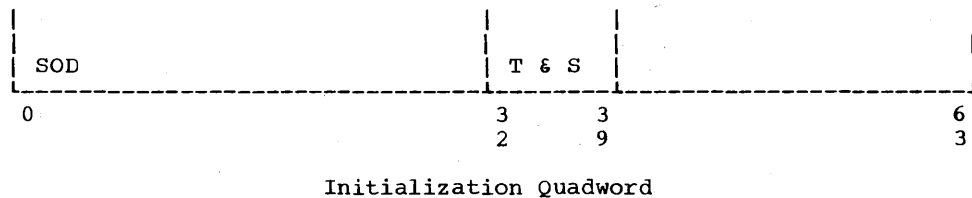


Figure I-6 Test and Set Byte

The DE upon accessing an initialization quadword for the DG interface, will set the TEST AND SET byte to all ones. The programmer may, by setting this byte to zero, determine if and when the DE initialized a regeneration cycle for a particular CVG/PVD.

4.1.3 REGENERATION CONTROL

The first quadword within each DE control area contains a bit switch for each assigned CVG (bits 0-23). (See Figure I-4). A programmer, by storing a switch mask into the first doubleword of a DE, may stop the regeneration cycle for any number of CVGs by setting the particular switch bits to one. The DE upon detecting a store into this location will interrogate this doubleword and immediately set the EOD latch for all CVGs which have their assigned bits set to one. A zero in any bit position indicates that no action is required (i.e., regeneration cycle for the CVG may proceed). This switch action is effective only until the next regeneration cycle begins. This action will cause all referenced CVGs operating in the synchronous mode to become idle for the remainder of the current refresh period. All referenced CVGs operating in the asynchronous mode will restart their images at the addresses in their respective initialization quadwords.

This facility may allow the programmer to immediately redirect a DE to a new CVG/PVD image and to assure the programmer that the old image storage area may be returned to the available storage pool. While the DE may not immediately initiate a new regeneration cycle for the CVG (regeneration cycles are initiated at fixed time intervals), it is assured that the old image data will not be accessed again. The programmer must, of course, redirect the DE to the new image by previously storing the new image address in the CVGs initialization quadword.

NOTE: The bit switches were designed to be used as a programming aid under abnormal conditions and not intended for use during normal image switchover. For example, if DE storage is full it may be necessary to preempt the old DE image in order to reallocate the storage area for the new image. The switches are activated by

the transition from 0 to 1. The programmer must insure that the bit was zero before changing it to a one.

4.1.4 REGENERATION CYCLE

The regeneration cycles for all assigned and configured CVGs are simultaneously allowed to be initiated. The DE will begin a scan of each configured CVG in a fixed sequence and, upon receiving a request, will fetch the first initialization quadword for that CVG. The data contained in this quadword (SOD and PVD Test) is transferred to the CVG (for example, CVG 1). when that CVG raises its request signal again, the second initialization quadword (Trackball and TFR) is fetched, transferred to the CVG, and the address contained within the TFR is retained by the DE in a nonaddressable Local Storage within the DE. The DE will then scan CVG 2 and, if a request is pending, initialize CVG 2 in a like manner. Upon servicing or at least interrogating requests for all twenty-four CVGs, the DE returns to CVG 1. If a request is pending, the DE will fetch the transfer address stored for this CVG in Local Storage, fetch the addressed quadword, and transfer the data to CVG 1. The DE will then update the address and store it back into Local Storage. CVG 2-24 are then similarly serviced. This servicing of CVGs continues until the DE detects an EOD within the image data of a particular CVG/PVD. At this time, the DE will stop accepting requests from the particular CVG until the next regeneration cycle. Other CVGs which have not yet encountered an EOD will continue to be serviced. (See Figure I-7).

It will be noted on Figure I-7 that if the data load for a particular CVG/PVD requires greater than a regeneration cycle (approximately 18.2 msec) to transfer, the DE will allow another regeneration cycle immediately. An example of this is shown as CVG 5A in Figure I-7.

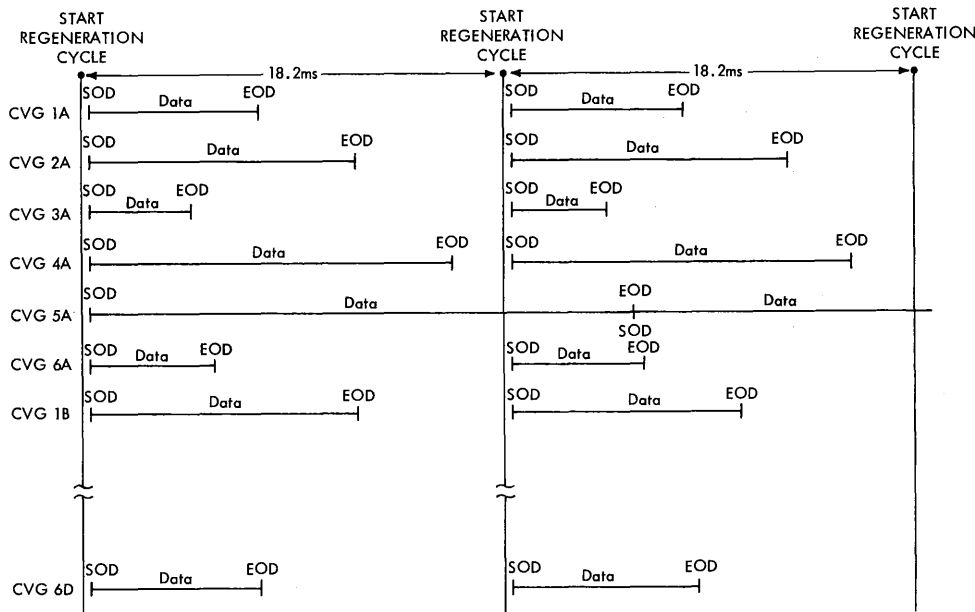


Figure I-7 Regeneration Cycle

JUNE 1, 1971

APPENDIX J: CHANNEL-TO-CHANNEL ADAPTER

1.1 INTRODUCTION

The Channel-to-Channel Adapter is an RPQ feature available on the IBM 7231-02 IOCE which provides both the path and the necessary synchronization for operations to take place between two IOCE selector channels of different systems. This feature provides the facility to move blocks of data from MACH or main storage of one system to MACH or main storage of the other system.

Connected to the two IOCE selector channels (one on the CCC and the other on the DCP) by means of the I/O interface, the adapter is installed on one of the two communicating IOCE selector channels, and one control unit position on each of those channels is used. On the channel on which it is installed, the adapter is assigned the highest control unit priority, on the other channel, it may be assigned any priority. The adapter receives its power from the IOCE in which it is installed; however, when that power is off, the other channel to which the adapter is connected continues to operate with its control units. Likewise, the Channel-to-Channel Adapter ONLINE-OFFLINE Switch and the ADAPTER DISABLED indicator is located on the panel of the IOCE in which the adapter is housed. This switch is operational only when the IOCE in which the adapter is housed is in state Zero with the Test Switch ON.

The adapter operates in burst mode and transmits data at the rate of the selector channel. To each of the channels to which it is connected, the adapter appears to be a control unit, and, in essentially the same manner as a control unit, it accepts and decodes commands from the channel. The adapter, however, differs from a control unit in that it does not use these commands to operate and control input/output devices; instead, it uses them to open a path between the two channels it connects and then synchronizes the operations performed between the two channels.

The adapter functionally consists of two control elements connected to and communicating with each other by means of a common one-byte buffer register and several signal lines. One of the two control elements serves one channel and the other control element the other channel. (See Figure J-1).

2.1 GENERAL OPERATION

When a channel sends a command to an adapter, the adapter responds by sending to the channel a status byte indicating the status of the adapter. If the operation is an 'immediate' one, that is, one not requiring a data transfer other than command byte over the interface, the adapter responds with a status byte indicating either (1) that the command will be executed and the channel is freed for the next operation, (2) that the adapter is busy or, (3) that the adapter is not ready. If the operation is one requiring data transfer, however, the adapter responds with a status byte indicating either (1) that the command has been accepted and the operation is proceeding but the channel is not freed, (2) that the adapter is busy or, (3) that the adapter is not ready. If the command has been accepted, the adapter decodes the command byte and, in certain instances, places it in the buffer register of the adapter.

JUNE 1, 1971

2.1.1 COMMANDS

The adapter decodes and makes use of seven commands: Read, Read Backward, Write, Control, Sense Command Byte, Sense Adapter, and Write End of File. (See Table J-I).

The adapter decodes only those bits of the command byte that identify the basic command and ignores the other bits. Therefore, in most cases, the programmer may use those other bits in ways he finds advantageous.

The type of command determines what adapter operation will ensue after the command is decoded.

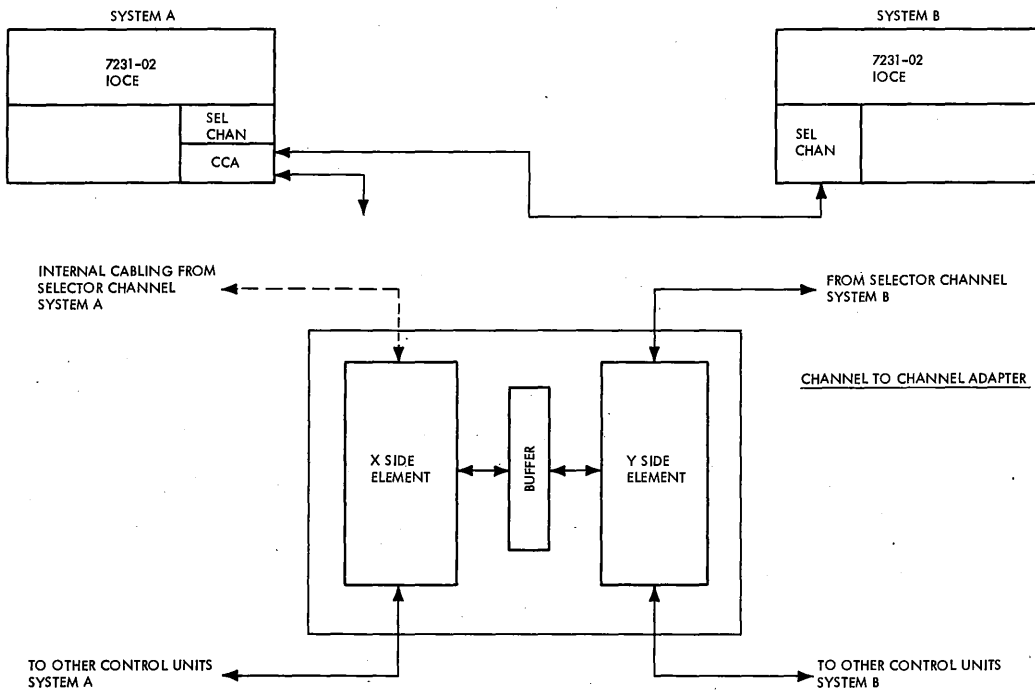


Figure J-1 Data Flow Through a Channel-to-Channel Adapter

For example, a No Operation command will cause an adapter which is ready to return a status byte containing channel end and device end bits and to disconnect from the channel immediately after accepting the command. On the other hand, certain commands issued by a channel to an adapter which is ready can be executed only if the other channel to which the adapter is connected issues an appropriate command. For example, a Read or Read Backward operation issued by one channel can be executed by an adapter only after a Write command has been issued to the

JUNE 1, 1971

TABLE J-I COMMANDS USED BY THE CHANNEL-TO-CHANNEL ADAPTER

Command	COMMAND CODES (in binary)							
	0	1	2	3	4	5	6	7
Read	X	X	X	X	X	X	1	0
Read Backward	X	X	X	X	1	1	0	0
Write	0	X	X	X	X	X	0	1
Write End of File	1	X	X	X	X	X	0	1
Sense Command Byte	X	X	X	1	0	1	0	0
Sense Adapter	X	X	X	0	0	1	0	0
Control (other than No Operation)	X	X	X	X	X	1	1	1
No Operation (Special Control)	X	X	X	X	X	0	1	1

"X" indicates the bit may be a 1 or 0. These bits may be used by the programmer in ways he finds advantageous.

adapter by the other channel. Likewise, a Write command issued as the initial part of an operation requires an accompanying Read command from the other channel.

When the adapter decodes a command from one channel requiring an appropriate command from the other channel, the adapter places the command byte in the buffer register and signals the other channel (via an attention interrupt) that an operation is waiting in the adapter. The channel program responds by issuing a Sense Command Byte which causes the adapter to transmit (to the channel issuing that command) the contents of the adapter buffer register, which contains the command byte issued by the initiating channel. The program to which the byte was sent then examines the byte to determine what command should be issued in response to the adapter. The program then issues that command. Only after both commands have been accepted by the adapter will the adapter operation continue to completion.

2.1.2 ADAPTER READY CONDITION

A not-ready condition is forced in either adapter control element upon receipt from its respective channel, of a Halt I/O instruction, a selective reset, or a system reset. A not-ready condition on one side of the adapter causes the opposite side to reject all commands (except the Sense Adapter and Sense Command Byte commands) with unit check status. A not-ready side is forced to a ready condition upon receipt of any command except Sense Adapter and Sense Command Byte. The transition to a ready condition on one side of the adapter forces device end status to the opposite side. For example, a Halt I/O sequence from channel X forces the adapter X side to a not-ready condition; a Read, Read Backward, Write, Control, or Write End of File command, or a Test I/O instruction from channel Y is rejected with unit check status. A subsequent Read, Read Backward, Write, Control, or Write End of File command from channel X returns the adapter side to a ready condition and forces device end status to the adapter Y side. Channel Y then presents the device end status to the program, indicating that the not-ready condition has been changed to ready.

2.1.3 STATUS INDICATIONS

In indicating unit status to the channels, the adapter makes use of six bits of the eight-bit unit-status byte, as follows:

<u>Bit</u>	<u>Indication</u>
P	Parity
0	Attention
1	Not Used
2	Not Used
3	Busy
4	Channel End
5	Device End
6	Unit Check
7	Unit Exception

These six bits are defined as follows:

Bit 0 - Attention

Indicates that an operation that requires an appropriate response from the channel has been initiated by the other channel.

Bit 3 - Busy

Indicates that the adapter cannot accept a command. This condition occurs if the initiating channel has a previous Control command (other than a No Operation) still in progress or, the command issued does not "match" a pending command previously issued by the other channel.

Bit 4 - Channel End

Indicates to the channel that the adapter has been disconnected from the channel.

Bit 5 - Device End

Indicates that the operation specified by the channel has been completed.

Bit 6 - Unit Check

Indicates to the issuing program (in the initial selection or termination status byte) that the adapter's opposite side is in the not-ready condition. Where the opposite side is not ready, the adapter rejects all commands from an issuing channel except Sense Adapter and Sense Command Byte and responds with unit check status.

Bit 7 - Unit Exception

Issued in response to a Read command (in the initial selection or termination status byte) if the Read command is terminated by an opposite-side Write End of File command.

The adapter may send status bytes with the following bits set to 1.

Channel End

Device End

Attention

Busy

Unit Check

Channel End and Device End

JUNE 1, 1971

Busy and Attention

Busy and Device End

Unit Exception, Channel End, and Device End

Unit Check, Channel End and Device End

The conditions under which the adapter sends these status bytes to the channels are described in the 'Operations' section of this Appendix.

During I/O interruptions and during execution of the START I/O, TEST I/O and HALT I/O instructions, the unit-status byte (under certain conditions) will be stored as bits 32-39 of the channel status word. The conditions under which the unit-status byte will be stored are described in Chapter 10, Input-Output Operations.

2.1.4 SENSE INDICATIONS

A Sense Adapter command should be initiated upon detecting a unit check condition in the status byte. In adapter operations, a unit check status occurs only when the opposite adapter element is in a not-ready condition. The sense byte received in response to the Sense Adapter command has the following format:

<u>Bit</u>	<u>Indication</u>
P	Parity
0	Not Used
1	Intervention Required
2	Not Used
3	Not Used
4	Not Used
5	Not Used
6	Not Used
7	Halt I/O-Selective Reset

Bit 1 - Intervention Required

Indicates to the program issuing the Sense Adapter command that the opposite side is in the not-ready condition. When the intervention required bit appears alone in the sense byte, the opposite side is not ready due to a system reset from its channel. When accompanied by bit 7 (Halt I/O-selective reset), the not-ready condition was caused by either a Halt I/O or selective reset.

Bit 7 - Halt I/O-Selective Reset

Indicates to the program issuing the Sense Adapter command that the adapter opposite side is in the not-ready condition. The Halt I/O - selective reset bit is always accompanied by the intervention required bit and indicates that the not-ready condition is due to either a Halt I/O or a selective reset operation.

2.1.5 UNUSUAL CONDITIONS

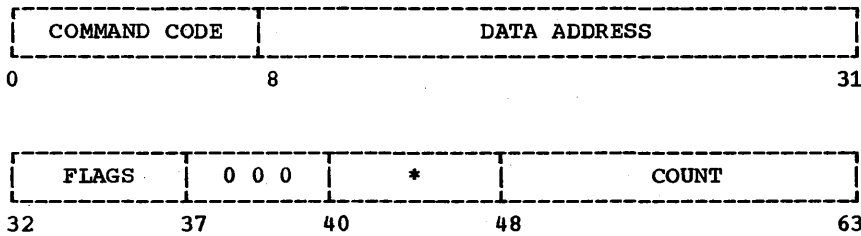
The adapter neither detects nor indicates error or other unusual conditions existing in the data it transfers. The function of the adapter is to transfer data between channels regardless of the correctness of the data or whether proper parity exists. Checking is performed only in the channels connected by the adapter.

3.1 OPERATIONS

All adapter operations are initiated by a START I/O instruction. Successful execution of the START I/O instruction causes the channel to fetch a channel address word (CAW). The CAW specifies the location in storage where the channel program begins (the address of the first byte of the first channel command word).

3.1.1 CHANNEL COMMAND WORD

The channel command word (CCW) specifies the operation to be executed. The channel refers to a CCW in storage only once; thereafter, the pertinent information is stored in the IOCE. The CCW has the following format:



The following is a brief description of the CCW. For a complete description, see Chapter 10, Input-Output Operations.

In the CCW, the command code (bits 0-7) specifies the operation to be performed. The command codes pertinent to the adapter are shown in Table J-I.

The data address (bits 8-31) specifies the location of a byte in storage. It is the first location referred to in the area designated by the CCW.

Bit 32 is the chain-data flag and bit 33 the chain-command flag. When bit 32 is set to 0 and bit 33 to 1, command chaining is specified. In adapter operations, command chaining may be performed by one of the two channels or both simultaneously. Note, however, that operations performed using the adapter occur functionally in pairs. If the channels issue commands in an illogical sequence, the out-of-sequence command will not be accepted. For example, if both channels are chaining and channel X initiates a Read Command but Channel Y answers with a Control command, the Control command will not be accepted. The adapter will disconnect from channel Y but will remain connected to channel X until channel Y sends a correct response or a HALT I/O instruction is issued. Since the Sense command does not require a reply, Sense commands cannot be chained in answer to a chain of control commands.

Bit 34 is the suppress-length-indication flag. It controls whether an incorrect length condition is to be indicated to the program.

Bit 35 is the skip flag. It specifies the suppression of transfer of information to storage during a Read, Read Backward, or Sense Operation.

Bit 36, the program-control-interruption flag, causes the channel to generate an interruption condition upon fetching the CCW. The count, bits 48-63, specifies the number of eight-bit-byte locations in the storage area designated by the CCW. In adapter operations, the count for Sense and Control commands should be 1. The counts of paired

JUNE 1, 1971

commands used during Read, Read Backward, and Write operations should be the same. If they are not, the CCW having the lowest count will control the number of bytes of data transferred. (Note that the chaining of CCW's necessarily requires that the count fields be correctly matched.) The incorrect length indication will be set in the channel-status byte as listed in "Programming Note" in the "Read and Read Backward Operations" section (3.1.2.4).

3.1.2 COMMANDS

In the following descriptions of operations performed by the adapter, the channel initiating an adapter operation is termed "Channel X", and the other channel is termed "Channel Y". The pertinent status indications sent by the adapter to the channels during adapter operation are summarized in Table J-II.

3.1.2.1 Control

The command byte for the Control command is shown in Table J-I. If bit 5 of that Control command byte is set to 0, the adapter interprets the command to be a No Operation command.

All control commands are 'immediate', that is, the adapter sends channel end status to the issuing channel during the initial selection process (if accepted).

3.1.2.1.1 NO OPERATION

A No Operation command is used for programming convenience. The contents of the adapter are not affected by this operation. If the adapter is idle on receipt of the command, channel end and device end bits are immediately returned to the issuing channel; the adapter then disconnects from the channel.

If either element of the adapter is not ready or if the adapter is busy, the adapter responds to a No Operation command as indicated in Table J-II.

A simplified flowchart of the operations performed by the adapter in response to a No Operation command is presented in Figure J-2.

3.1.2.1.2 OTHER CONTROL COMMANDS

When channel X issues a Control command other than No Operation, an idle adapter places the Control command byte in the adapter buffer register and signals attention status to channel Y. To free the adapter, channel Y must issue a Sense Command Byte command to the adapter. At the termination of this sense operation, device end status is sent to channel X, signalling completion of the Control operation. If either element of the adapter is not ready, or if the adapter is busy, the adapter responds to the Control Command as indicated in Table J-II.

A simplified flowchart of the operations performed by the adapter in response to a Control command other than No Operation is presented in Figure J-3.

3.1.2.2 Sense Command Byte

The command byte for the Sense Command Byte operation is shown in Table J-I. This command is usually issued in response to an attention and will sense the content of the one byte adapter buffer. A simplified flowchart of the operations performed by the adapter in response to a Sense Command Byte operation is presented in Figure J-4. If either

JUNE 1, 1971

element of the adapter is not ready, or the adapter is busy, the adapter
| responds to the Sense Command Byte command as indicated in Table J-II.

JANUARY 31, 1973

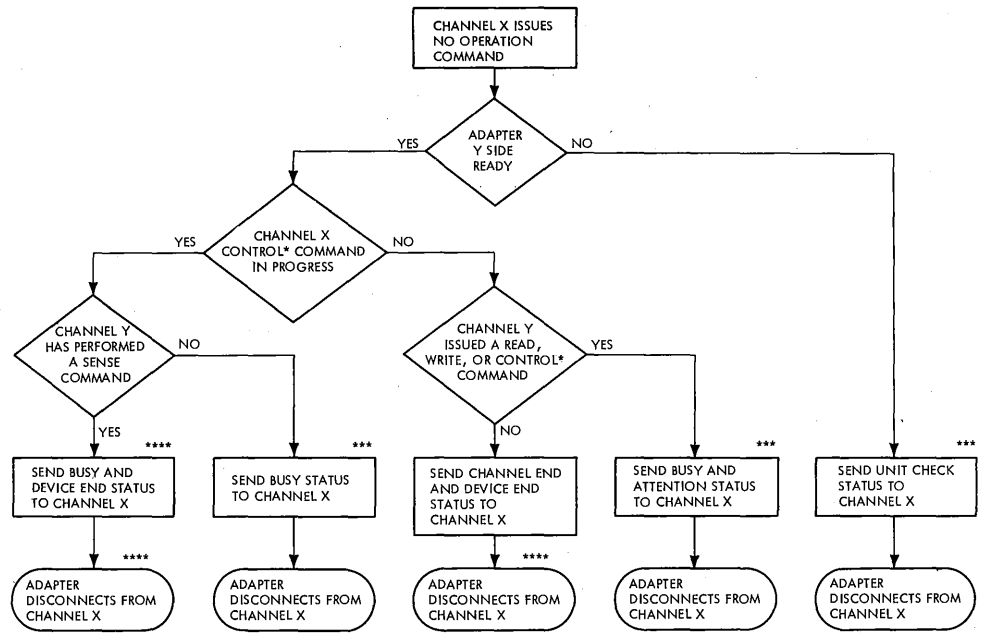
TNL: GN31-3005

TABLE J-II SUMMARY OF STATUS INDICATIONS

COMMAND ISSUED BY CHANNEL X	UNIT STATUS INDICATIONS								
	ADAPTER IS IDLE AND READY	CHANNEL X CONTROL COMMAND STILL IN PROGRESS *		CHANNEL Y HAS IN PROGRESS			ADAPTER Y SIDE NOT READY	ADAPTER X SIDE NOT READY	
		CHANNEL Y HAS SENSED **	CHANNEL Y HAS NOT SENSED	READ OR READ BACKWARD COMMAND	WRITE COMMAND	CONTROL COMMAND*			WRITE END OF FILE COMMAND
CONTROL a. NO OPERATION b. OTHERS	CHANNEL END AND DEVICE END TO CHANNEL X. CHANNEL END TO CHANNEL X AND ATTENTION TO CHANNEL Y; ON COMPLETION OF SENSE OPERATION, DEVICE END TO CHANNEL X.			BUSY AND ATTENTION TO CHANNEL X			RESET WRITE END OF FILE LATCH. IF NO OTHER COMMAND IS IN PROGRESS, UNIT STATUS INDICATION IS AS SHOWN UNDER "ADAPTER IS IDLE AND READY". IF ANOTHER COMMAND IS IN PROGRESS, STATUS WILL BE AS SHOWN UNDER OTHER COLUMNS OF THIS TABLE. NO UNIQUE STATUS INDICATION IS GIVEN FOR THE WRITE END OF FILE CONDITION.	X SIDE GOES READY. IF ADAPTER Y IS READY, CHANNEL END AND DEVICE END TO CHANNEL X. DEVICE END TO CHANNEL Y. X SIDE GOES READY. CHANNEL END TO CHANNEL X. DEVICE END TO CHANNEL Y. UPON ACCEPTANCE OF DEVICE END STATUS BY CHANNEL Y, ATTENTION TO CHANNEL Y. ON COMPLETION OF SENSE OPERATION DEVICE END TO CHANNEL X.	
SENSE COMMAND BYTE	ZERO STATUS TO CHANNEL X; ON COMPLETION OF ADAPTER BYTE TRANSFER, CHANNEL END AND DEVICE END TO CHANNEL X.			ZERO STATUS TO CHANNEL X; ON COMPLETION OF ADAPTER BYTE TRANSFER, CHANNEL END AND DEVICE END TO CHANNEL X.		ZERO STATUS TO CHANNEL X; ON COMPLETION OF ADAPTER BYTE TRANSFER, CHANNEL END AND DEVICE END TO CHANNEL Y.	ZERO STATUS TO CHANNEL X; ON COMPLETION OF ADAPTER BYTE TRANSFER, CHANNEL END AND DEVICE END TO CHANNEL X. Y SIDE STAYS NOT READY.	ZERO STATUS TO CHANNEL X; ON COMPLETION OF ADAPTER BYTE TRANSFER, CHANNEL END AND DEVICE END TO CHANNEL X. X SIDE STAYS NOT READY.	
SENSE ADAPTER	ZERO STATUS AND SENSE BYTE TO CHANNEL X; ON COMPLETION OF ADAPTER BYTE TRANSFER, CHANNEL END, AND DEVICE END TO CHANNEL X.			ZERO STATUS TO CHANNEL X; ON COMPLETION OF ADAPTER BYTE TRANSFER, CHANNEL END AND DEVICE END TO CHANNEL X.		ZERO STATUS TO CHANNEL X; ON COMPLETION OF ADAPTER BYTE TRANSFER, CHANNEL END AND DEVICE END TO CHANNEL Y.	ZERO STATUS TO CHANNEL X; IF Y NOT READY DUE TO SYSTEM RESET, INTERVENTION REQUIRED SET IN SENSE BYTE. IF Y NOT READY DUE TO HALT I/O OR SELECTIVE RESET, INTERVENTION REQUIRED AND HALT I/O-SELECTIVE RESET SET IN SENSE BYTE. ON COMPLETION OF BYTE TRANSFER, CHANNEL END AND DEVICE END TO CHANNEL X. Y SIDE STAYS NOT READY.	ZERO STATUS AND SENSE BYTE TO CHANNEL X; ON COMPLETION OF TRANSFER CHANNEL END AND DEVICE END TO CHANNEL X. X SIDE STAYS NOT READY.	
READ OR READ BACKWARD	ZERO STATUS TO CHANNEL X AND ATTENTION TO CHANNEL Y; ON COMPLETION OF DATA TRANSFER, CHANNEL END AND DEVICE END TO CHANNEL X.			BUSY AND ATTENTION TO CHANNEL X	ZERO STATUS TO CHANNEL X; DATA TRANSFER OCCURS; ON COMPLETION OF DATA TRANSFER, CHANNEL END AND DEVICE END TO CHANNEL X AND CHANNEL Y. ***	BUSY AND ATTENTION TO CHANNEL X	CHANNEL END, DEVICE END, AND UNIT EXCEPTION TO CHANNEL X.	X SIDE GOES READY. ZERO STATUS TO CHANNEL X. DEVICE END TO CHANNEL Y. UPON ACCEPTANCE OF DEVICE END STATUS BY CHANNEL Y, ATTENTION TO CHANNEL Y. ON COMPLETION OF DATA TRANSFER, CHANNEL END AND DEVICE END TO CHANNEL X.	
WRITE				ZERO STATUS TO CHANNEL X; DATA TRANSFER OCCURS; ON COMPLETION OF DATA TRANSFER, CHANNEL END AND DEVICE END TO CHANNEL X AND CHANNEL Y. ***			CHANNEL END, DEVICE END AND UNIT EXCEPTION TO CHANNEL Y.		
WRITE END OF FILE	CHANNEL END AND DEVICE END TO CHANNEL X. IF A READ COMMAND IS ISSUED BY CHANNEL Y, CHANNEL END, DEVICE END AND UNIT EXCEPTION TO CHANNEL Y; ANY OTHER COMMAND ISSUED BY CHANNEL Y RESETS WRITE END OF FILE.			CHANNEL END AND DEVICE END TO CHANNEL X. CHANNEL END, DEVICE END AND UNIT EXCEPTION TO CHANNEL Y.		BUSY AND ATTENTION TO CHANNEL X	SAME AS CONTROL AND SENSE COMMANDS	X SIDE GOES READY. CHANNEL END AND DEVICE END TO CHANNEL X. DEVICE END TO CHANNEL Y. IF A READ COMMAND IS ISSUED BY CHANNEL Y, CHANNEL END, DEVICE END AND UNIT EXCEPTION TO CHANNEL Y; ANY OTHER COMMAND ISSUED BY CHANNEL Y RESETS WRITE END OF FILE.	
TEST I/O INSTRUCTION ISSUED	ZERO STATUS TO CHANNEL X	DEVICE END TO CHANNEL X	BUSY TO CHANNEL X	ATTENTION TO CHANNEL X			ZERO STATUS TO CHANNEL X	UNIT CHECK TO CHANNEL X. Y SIDE STAYS NOT READY.	ZERO STATUS TO CHANNEL X. X SIDE STAYS NOT READY.
HALT I/O INSTRUCTION ISSUED	CHANNEL END AND DEVICE END SET ON X SIDE. X SIDE GOES NOT READY. X SIDE GOES NOT READY. SET INTERVENTION REQUIRED AND HALT I/O SELECTIVE RESET LATCHES ON Y SIDE.	CHANNEL END AND DEVICE END SET ON X SIDE. X SIDE GOES NOT READY. SET INTERVENTION REQUIRED AND HALT I/O SELECTIVE RESET LATCHES ON Y SIDE FOR SUBSEQUENT SENSING BY CHANNEL Y.		NO STATUS TO CHANNEL X; X SIDE GOES NOT READY. Y SIDE COMMAND TERMINATED WITH UNIT CHECK, DEVICE END AND CHANNEL END. DISCONNECT BOTH X AND Y SIDE. SET INTERVENTION REQUIRED AND HALT I/O SELECTIVE RESET LATCHES ON Y SIDE FOR SUBSEQUENT SENSING BY CHANNEL Y.			NO STATUS TO CHANNEL X; X SIDE GOES NOT READY. RESET WRITE END OF FILE LATCH. SET INTERVENTION REQUIRED AND HALT I/O-SELECTIVE RESET LATCHES ON Y SIDE.	CHANNEL END AND DEVICE END SET ON X SIDE. X SIDE GOES NOT READY. INTERVENTION REQUIRED AND HALT I/O-SELECTIVE RESET LATCHES SET ON Y SIDE.	NO STATUS TO CHANNEL X; X SIDE STAYS NOT READY. INTERVENTION REQUIRED AND HALT I/O-SELECTIVE RESET LATCHES SET ON Y SIDE.
SELECTIVE RESET (MALFUNCTION RESET)	RESET BUFFER AND STATUS. DISCONNECT BOTH X AND Y SIDE. X SIDE GOES NOT READY. SET INTERVENTION REQUIRED AND HALT I/O SELECTIVE RESET LATCHES FOR SUBSEQUENT SENSING BY CHANNEL X.			RESET BUFFER AND STATUS. Y SIDE COMMAND TERMINATED WITH UNIT CHECK, DEVICE END AND CHANNEL END. DISCONNECT BOTH X AND Y SIDE. SET INTERVENTION REQUIRED AND HALT I/O SELECTIVE RESET LATCHES FOR SUBSEQUENT SENSING BY CHANNEL Y. X SIDE GOES NOT READY.			RESET BUFFER, STATUS AND WRITE END OF FILE. DISCONNECT BOTH X AND Y SIDE. SET INTERVENTION REQUIRED AND HALT I/O-SELECTIVE RESET LATCHES ON Y SIDE.	RESET BUFFER AND STATUS; X SIDE GOES NOT READY. INTERVENTION REQUIRED AND HALT I/O-SELECTIVE RESET LATCHES SET ON Y SIDE. Y SIDE STAYS NOT READY.	RESET BUFFER AND STATUS; X SIDE STAYS NOT READY. INTERVENTION REQUIRED AND HALT I/O-SELECTIVE RESET LATCHES SET ON Y SIDE.
SYSTEM RESET	RESET BUFFER AND STATUS. DISCONNECT BOTH X AND Y SIDE. X SIDE GOES NOT READY. SET INTERVENTION REQUIRED LATCH FOR SUBSEQUENT SENSING BY CHANNEL Y.			RESET BUFFER AND STATUS. Y SIDE COMMAND TERMINATED WITH UNIT CHECK, DEVICE END AND CHANNEL END. DISCONNECT BOTH X AND Y SIDE. SET INTERVENTION REQUIRED LATCH FOR SUBSEQUENT SENSING BY CHANNEL Y. X SIDE GOES NOT READY.			RESET BUFFER, STATUS AND WRITE END OF FILE. DISCONNECT BOTH X AND Y SIDE. SET INTERVENTION REQUIRED LATCH ON Y SIDE.	RESET BUFFER AND STATUS; X SIDE GOES NOT READY. INTERVENTION REQUIRED LATCH SET ON Y SIDE. Y SIDE STAYS NOT READY.	RESET BUFFER AND STATUS; X SIDE STAYS NOT READY. INTERVENTION REQUIRED LATCH SET ON Y SIDE.

* Other than a No Operation command.
** Device End will be cleared and adapter left idle.
*** In this operation, Channel Y may be thought of as the initiating channel

JUNE 1, 1971



LEGEND:

- * OTHER THAN NO OPERATION COMMAND
- ** THIS CONDITION OCCURS ONLY IF Y SENSE COMMAND RESULTS IN STACKED STATUS ON X SIDE
- *** X NO OPERATION COMMAND REJECTED
- **** ADAPTER IS LEFT IDLE

| Figure J-2 No Operation Command

JUNE 1, 1971

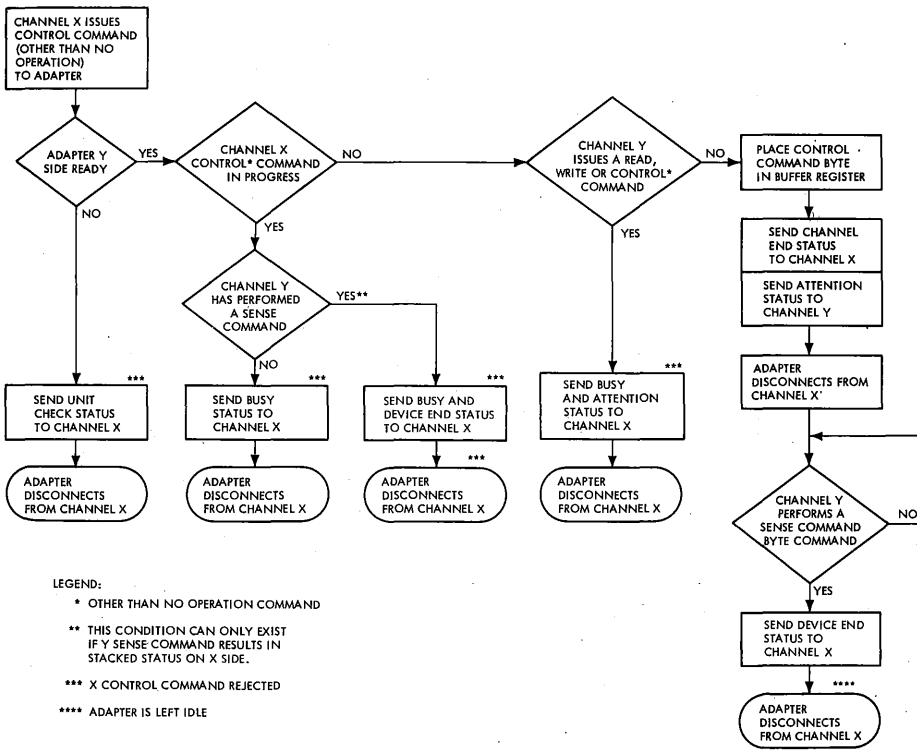


Figure J-3 Control Command

JUNE 1, 1971

3.1.2.3 Sense Adapter

The command byte for the Sense Adapter command is shown in Table J-I. When a Sense Adapter command is issued by a channel and accepted by the adapter, the adapter:

1. Sends a status byte of all zeros to that channel.
2. Transmits a sense byte with one of the following bit configurations to that channel:
 - a. Intervention required (bit 1) if the adapter opposite side is not ready due to a system reset.
 - b. Intervention required (bit 1) and Halt I/O-Selective reset (bit 7) if the adapter opposite side is not ready due to a Halt I/O or selective reset.
 - c. All zeros with correct parity if the adapter opposite side is ready.
3. On completion of the transfer of the byte, sends channel end and device end bits to that channel.

Usually, the Sense Adapter command is issued by a channel after that channel has had a previous command rejected with a unit check bit set in the initial status; thus, the sense byte is usually other than all zeros and indicates the cause of the not-ready condition. (The previous unit check bit only indicates that the not-ready condition exists, but not the cause of the condition.) Examination of the sense byte (other than all zeros) is the only means by which the channel issuing the Sense Adapter command can determine whether the not-ready condition is due to a system reset or Halt I/O-selective reset from the other channel. The Sense Adapter command does not clear the not-ready condition; this condition is cleared only by a Read, Read Backward, Write, Write End of File, or Control command from the channel originally causing the condition.

Table J-II shows (1) the conditions under which a channel issuing a Sense Adapter command finds the adapter busy, and (2) the status indications sent by a busy adapter.

A simplified flowchart of the operations performed by the adapter in response to a Sense Adapter command is presented in Figure J-5.

3.1.2.4 Read and Read Backward

The command bytes for Read and Read Backward commands are shown in Table J-I. The adapter performs the same operation whether the command is a Read or Read Backward command. For either command (Read or Read Backward) the function of the adapter is to transmit data bytes to the channel issuing the command from the other channel. Placing those data bytes in main storage in the proper order is the function of the channels. (In the following discussion, both "Read" and "Read Backward" are designated by "Read").

A Read command issued by channel X causes an idle adapter to return a status byte of all zeros to the channel, places the Read command byte in the adapter buffer register, and signals attention status to channel Y. For the operation to continue, channel Y must issue a Write command to the adapter (normally channel Y issues a Sense command preceding the Write command). When the adapter has accepted both the Read and Write commands, the data transfer takes place. When the transfer has been completed, the adapter sends channel end and device end status to each channel.

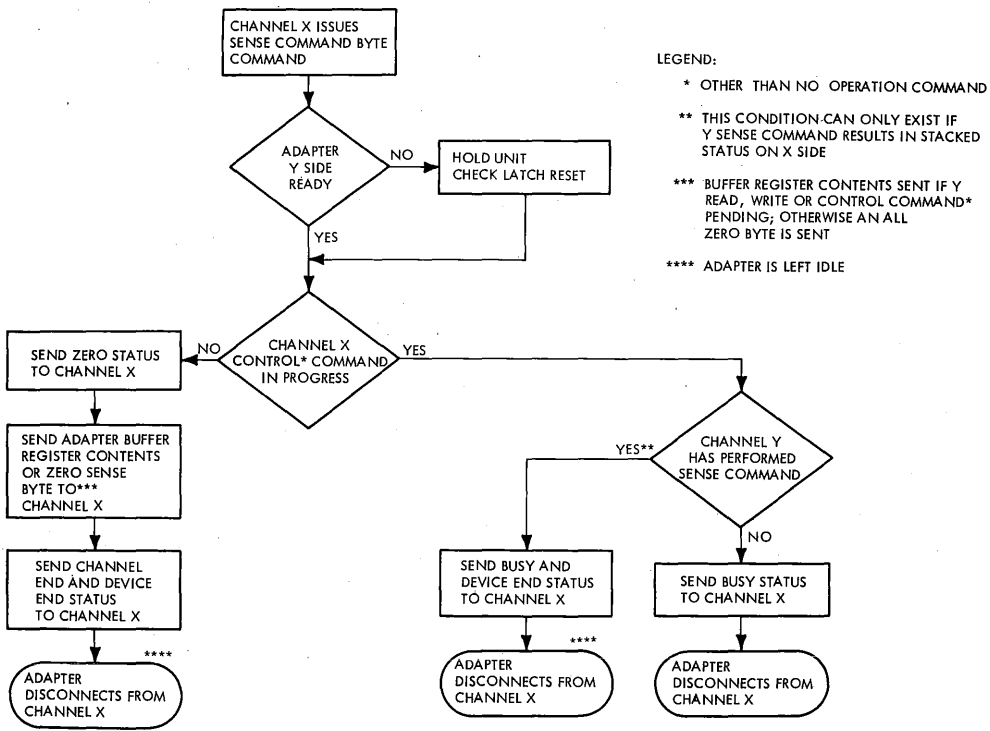


Figure J-4 Sense Command Byte Command

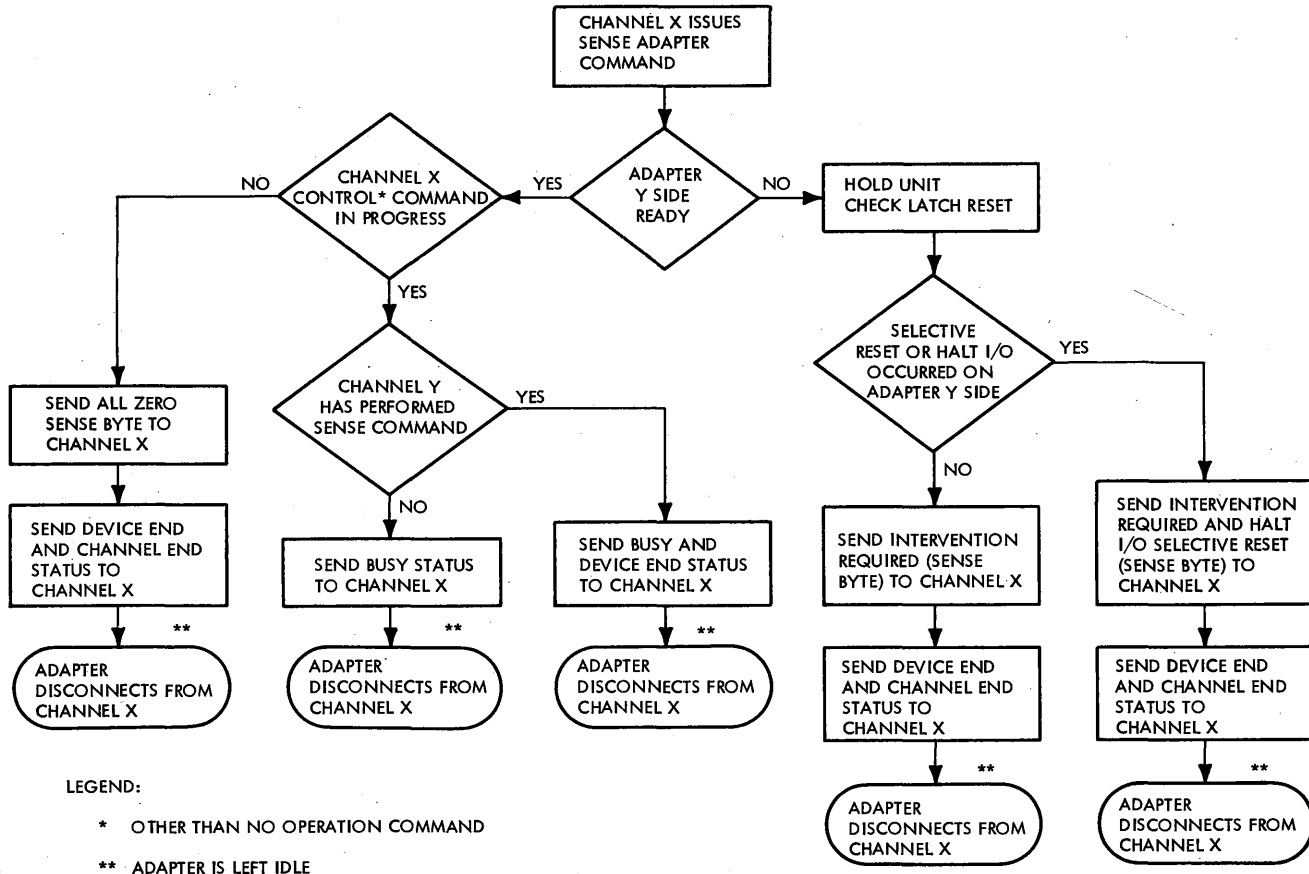


Figure J-5 Sense Adapter Command

JUNE 1, 1971

If a Write command from channel Y is waiting in the adapter when channel X issues a Read command to the adapter, the adapter recognizes a "match", and the specified data transfer occurs.

Channel Y issues a Read command when a Sense command issued by channel Y in response to attention status from the adapter has indicated that a Write operation has been initiated by channel X and is waiting in the adapter. When the adapter accepts the Read command, it sends a status byte of all zeros to channel Y. Then, the data transfer occurs. At the completion of the transfer, the adapter sends channel end and device status to each channel.

The Write End of File Operation, causes the adapter to terminate a Read command by sending channel end, device end, and unit exception status to the channel issuing the Read command. For this to occur, the Write End of File command must be issued by the other channel. For example, a Write End of File command issued to the adapter by channel Y terminates a Read command issued by channel X. If the Write End of File command is present in the adapter when the Read command is issued, the status bits are presented to channel X in the initial status byte; if the Write End of File command is issued after the Read command, the status bits are presented to channel X in the ending status byte. In either case, termination of the Read command also terminates the Write End of File command.

A Read command presented to a busy adapter or to an adapter which has either element not ready causes the adapter to respond as indicated in Table J-II.

A simplified flowchart of the operations performed by the adapter in response to a Read command is presented in Figure J-6.

PROGRAMMING NOTE 1:

During an initial program loading (IPL) operation, the IPL (a Read command with the modifier bits of the command byte set to zeros) must be issued before the Write command. If a Write command byte were placed in the adapter before the IPL was issued, the channel issuing the Write command would receive channel end, device end and unit check status at the time the other channel initiated the IPL.

PROGRAMMING NOTE 2:

At the end of reading and writing operations, the incorrect length indication, unless suppressed, will be set in the channel-status byte as indicated in the following listing.

READ		WRITE		WRONG LENGTH RECORD	
Channel X		Channel Y		Channel X	Channel Y
CCW count	=	CCW count		No	Yes
CCW count	>	CCW count		Yes	Yes
CCW count	≤	(CCW count) - 2		Yes	Yes
CCW count	=	(CCW count) - 1		Yes	No

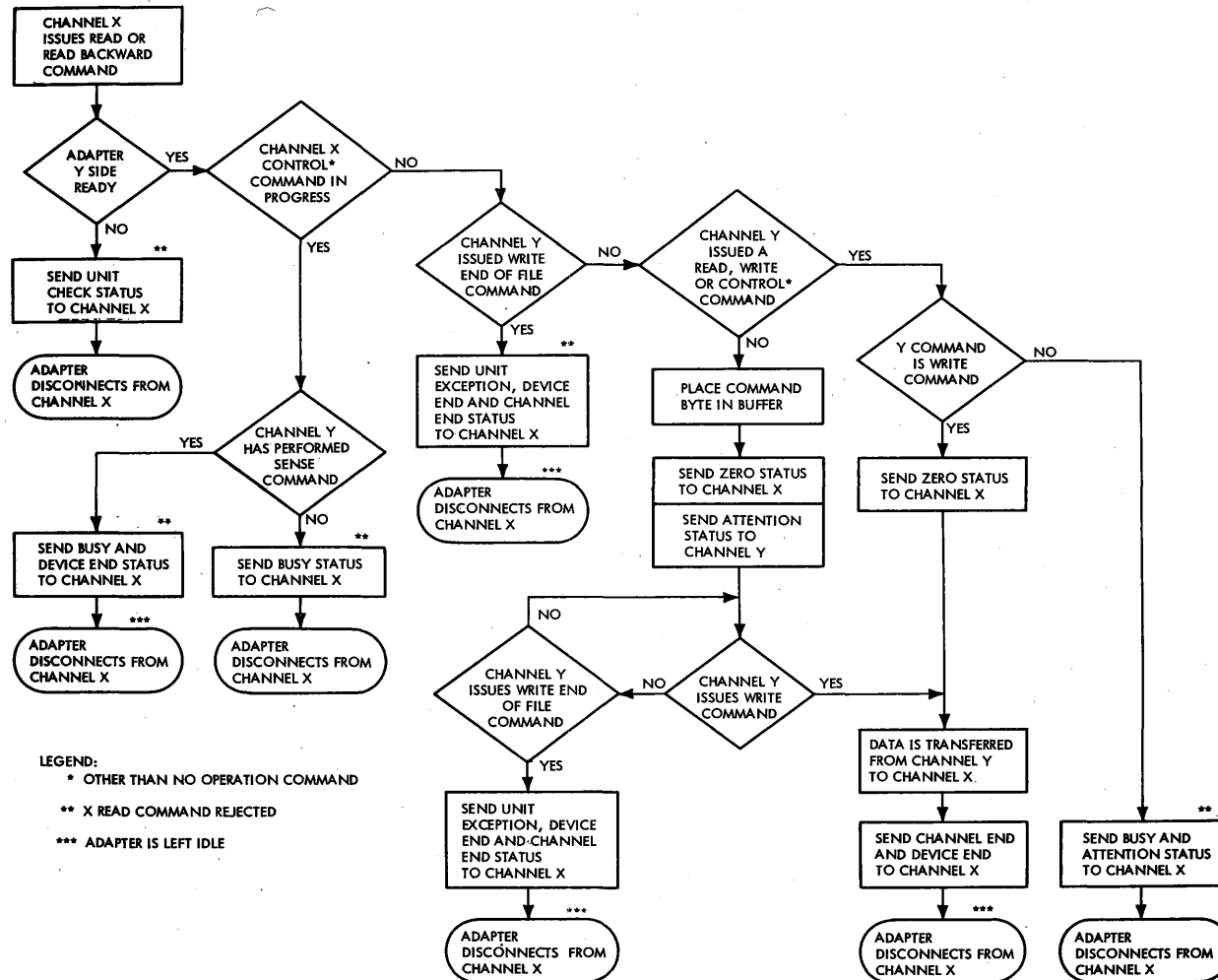


Figure J-6 Read or Read Backward Command

JUNE 1, 1971

3.1.2.5 Write

The command byte for the Write command is shown in Table J-I. In executing a Write command, the adapter transmits data bytes from the channel issuing the command to the other channel.

A Write command issued by channel X will cause an idle adapter to return a status byte of all zeros to the channel, place the Write command byte in the adapter buffer register, and signal attention status to channel Y. For the operation to continue, channel Y must issue a Read command to the adapter. (Normally channel Y issues a Sense command preceding the Read commands.) When the adapter has accepted both the Write and Read commands, the data transfer will take place. When the data transfer has been completed, the adapter will send channel end and device end status to each channel.

If a Read command from channel Y is waiting in the adapter when channel X issues a Write command to the adapter, the adapter recognizes a 'match', and the specified data transfer occurs.

Channel Y issues a Write command when a Sense command issued by channel Y in response to attention status from the adapter has indicated that a Read operation has been initiated by channel X and is waiting in the adapter. The adapter sends a status byte of all zeros to channel Y when it accepts the Write command. Then the data transfer will take place. At the completion of the transfer, the adapter sends channel end and device end status to each channel.

If either element of the adapter is not ready or if the adapter is busy the adapter responds to the Write command as indicated in Table J-II.

A simplified flowchart of the operations performed by the adapter in response to a Read command is presented in Figure J-7.

Programming Note:

During an initial program loading (IPL) operation, the Write command must be issued after the IPL (a Read command with the modifier bits of the command byte set to zero) is issued. If the write command were issued and placed in the adapter before the IPL was issued, the channel issuing the Write command would receive channel end and device end status at the time the other channel initiated the IPL.

At the end of writing and reading operations, the incorrect length indication, unless suppressed, will be set in the channel-status byte under the conditions listed in 'Programming Note 2' in the 'Read and Read Backward Operations' section.

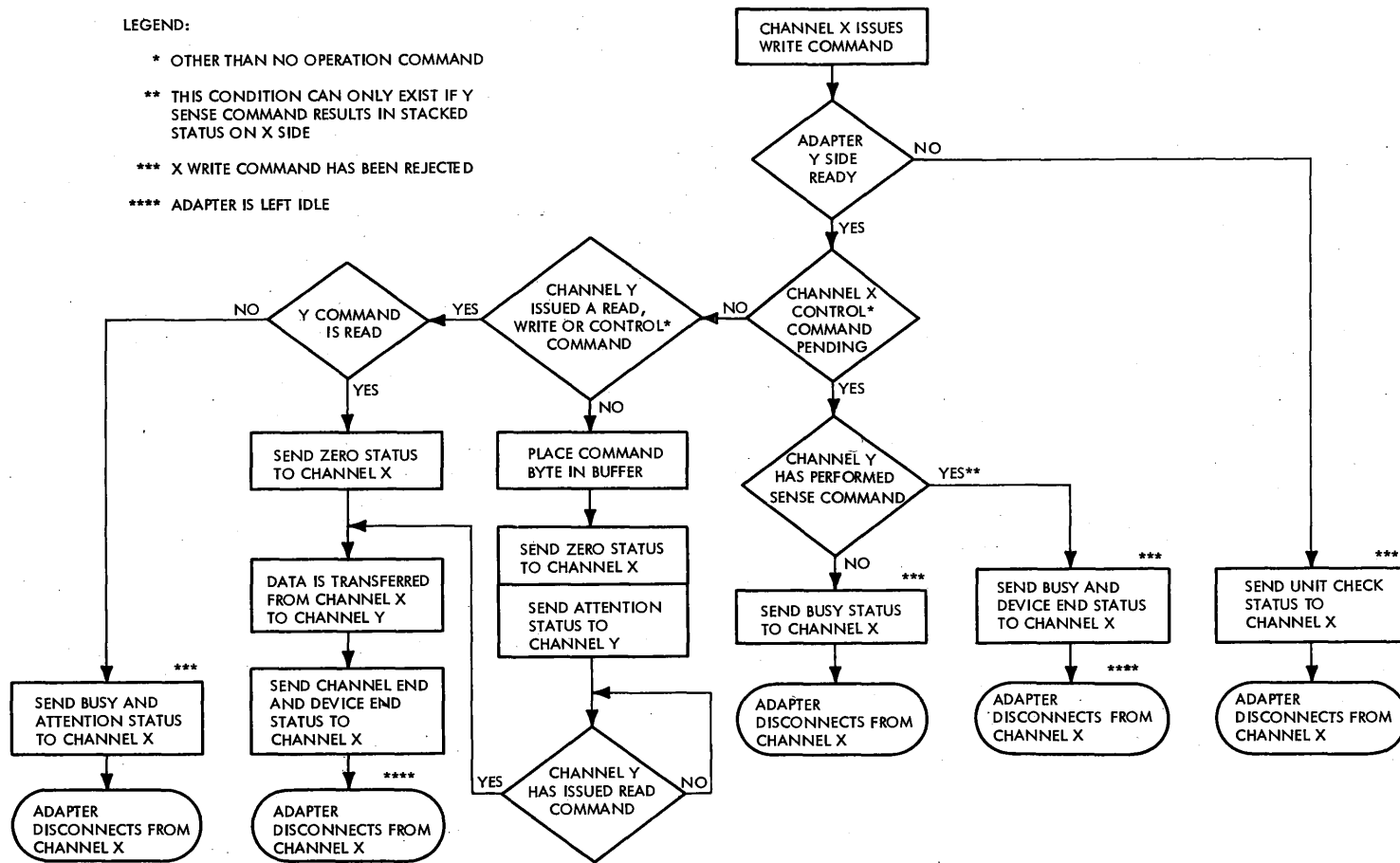


Figure J-7 Write Command

3.1.2.6 Write End of File

The command byte for the Write End of File command is shown in Table J-I. The function of a Write End of File command issued to the adapter by one channel is to terminate a Read command issued by another channel. Usually, the Write End of File command is used when individual Write commands are chained by one channel and individual Read commands are chained by the other channel. To the adapter, the Write End of File command indicates (1) that further data is not to be written to the channel issuing the command and (2) that chained Read commands issued by the other channel are to be terminated. If the adapter accepts the Write End of File command, the Write End of File command is operationally similar to a No Operation command in that channel end and device end bits are sent to the issuing channel in the initial status byte and the adapter is available to receive other commands from the channel issuing the Write End of File command; however, the Write End of File command differs from a No Operation command in that the adapter retains the Write End of File command in the form of active controls in the adapter element receiving the command.

A Write End of File command issued by a channel causes termination of a Read command already issued by the other channel; in this case, the Read command is terminated with channel end, device end and unit exception status sent to the channel issuing the Read command and the Write End of File command is terminated after initial status channel end and device end bits are sent to the channel issuing the Write End of File command. If the Read command is issued to the adapter, the Read command is terminated by the adapter and channel end, device end, and unit exception status is sent to the channel issuing the Read command; the Write End of File command is reset without sending further status bits to the channel issuing the command. (Channel end and device end bits have already been sent to the channel issuing the Write End of File command).

Since the only function of the Write End of File command is to terminate a Read command from the other channel, any command other than Read from the other channel resets a Write End of File command already received by the adapter. When the command is reset in this manner, no status indication is generated and the Write End of File command is lost to the program.

Commands issued to the adapter by a channel which has previously issued a Write End of File command to the adapter do not affect the previously issued Write End of File command. This is the only case where one adapter element may retain two commands from the same channel. When this condition exists, the command other than Write End of File responds to opposite side commands in the normal manner after the Write End of File command is cleared. One example of such a sequence is as follows:

1. Channel Y issues Write End of File command to the adapter.
2. Channel Y issues Control command (other than No Operation) to the adapter.
3. Channel X issues a Read command to the adapter, the Read command is terminated and the Write End of File command is cleared.
4. Channel X issues a Sense Byte command to the adapter and the Control command is cleared from the adapter.

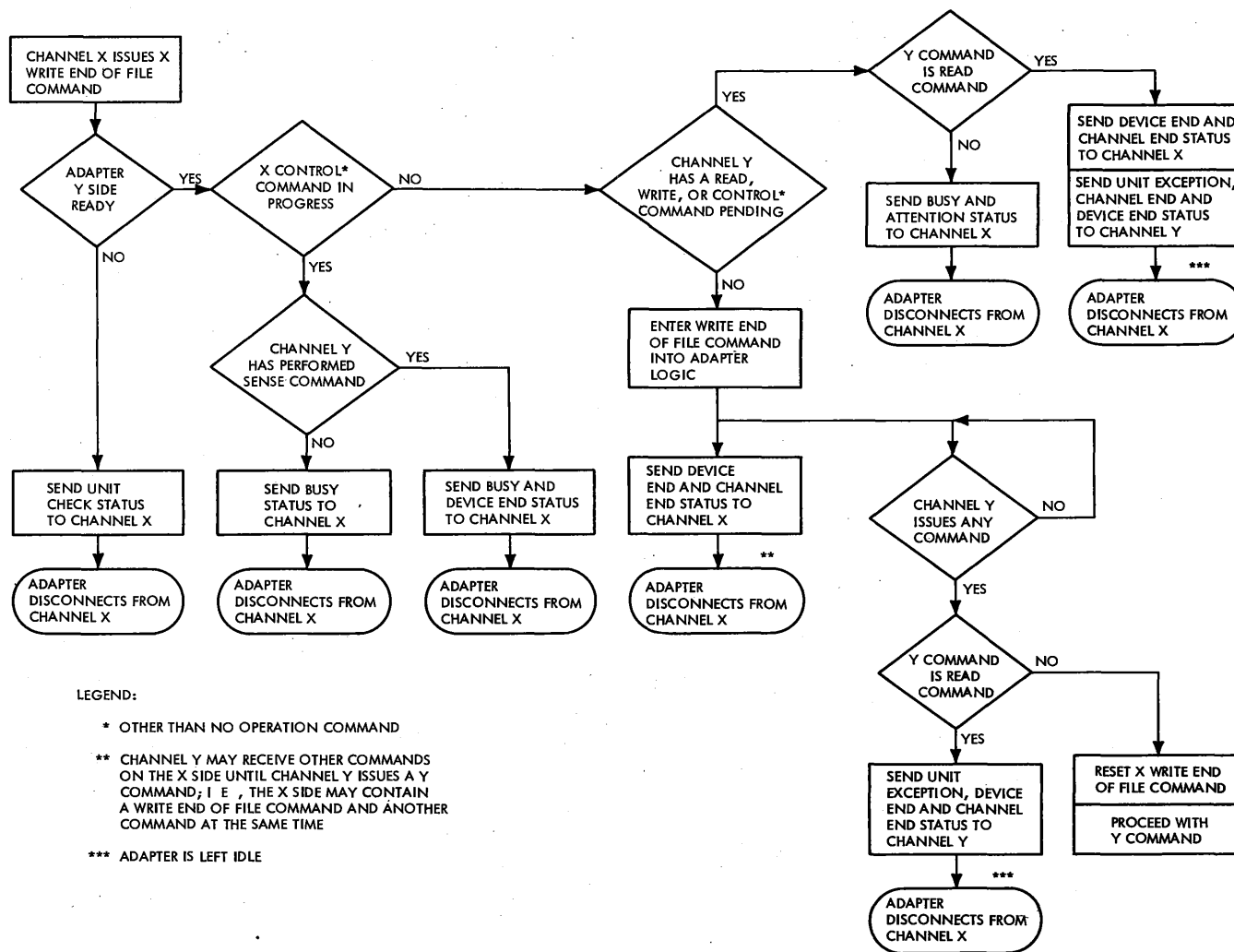


Figure J-8 Write End of File Command

JUNE 1, 1971

If either element of the adapter is not ready, or if the adapter is busy, the adapter responds to the Write End of File command as indicated in Table J-II.

A simplified flowchart of the operations performed by the adapter in response to a Write End of File command is presented in Figure J-8.

3.1.3 EFFECT OF TEST I/O AND HALT I/O

3.1.3.1 Test I/O

The Test I/O instruction is used to determine the status of the adapter. An idle adapter responds to the instruction by returning a status byte of all zeros. A Test I/O instruction presented to a not ready or busy adapter will cause the adapter to respond as indicated in Table J-II.

Issuance of a Test I/O instruction to an adapter resets only a device end indication. If the attention bit is set and the adapter is waiting to interrupt, the interruption will be cleared but the attention bit will not be reset. Any subsequent Test I/O instruction will receive attention status until the channel has responded appropriately to the attention status.

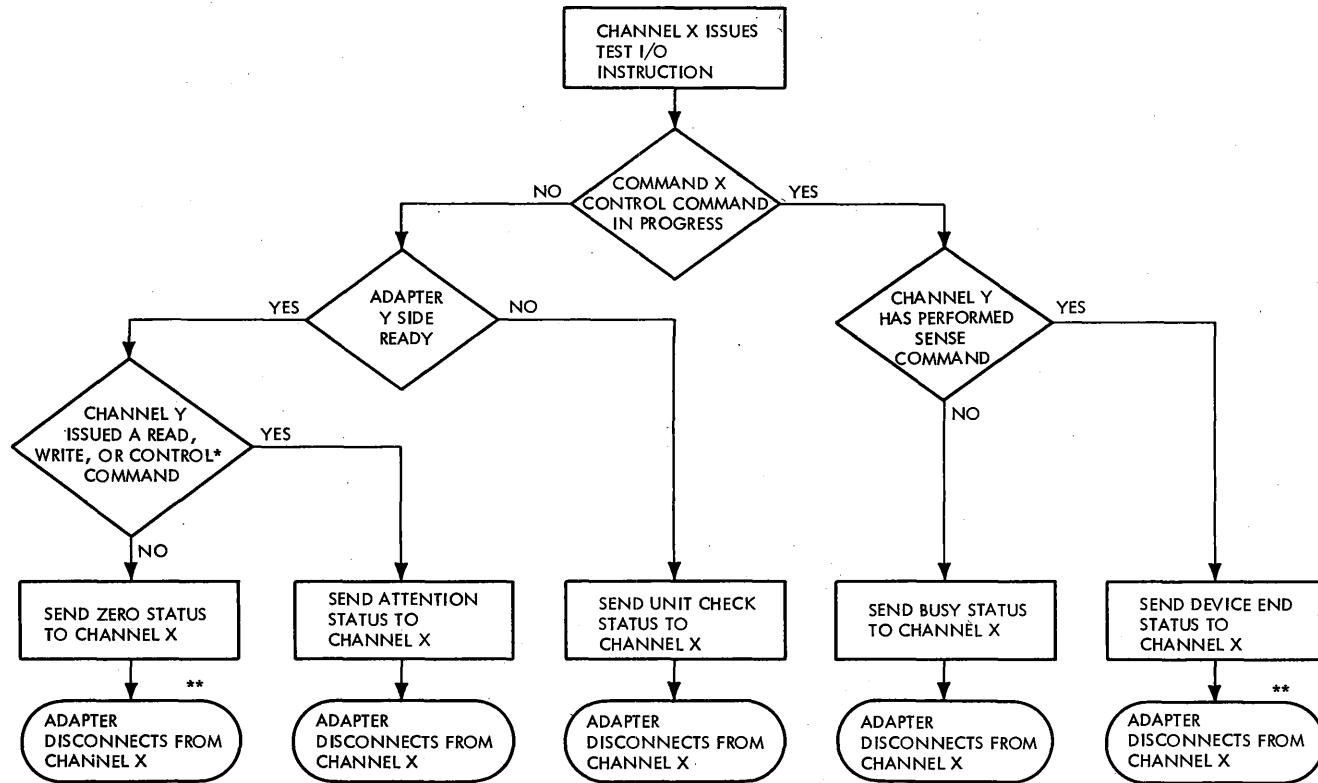
A simplified flowchart of the operations performed by the adapter in response to a Test I/O instruction is presented in Figure J-9.

3.1.3.2 Halt I/O

A Halt I/O instruction is directed to the adapter to terminate any operation currently being performed by the adapter or to place one side of the channel in a not-ready condition.

By making the adapter element receiving the instruction not ready, the Halt I/O instruction causes commands (other than the Sense commands) issued to the other adapter element to be rejected with a unit check bit in the initial status bytes. Both intervention required and Halt I/O-Selective reset bits are set for subsequent sensing by the channel which did not issue the Halt I/O instruction.

The response to an issuing channel is presented in Table J-II while a simplified flowchart is presented in Figure J-10.



LEGEND:

* OTHER THAN NO OPERATION COMMAND

** ADAPTER IS LEFT IDLE

Figure J-9 Test I/O Instruction

JUNE 1, 1971

3.1.4 SELECTIVE RESET (MALFUNCTION RESET)

The adapter responds to a selective reset from either channel by disconnecting from both channels and resetting its buffer register and status indications. In addition, the adapter element receiving the reset is made not ready and the intervention required and Halt I/O-Selective reset sense bits are activated. A subsequent Sense Adapter command to the element not receiving the reset senses the bits, indicating that the Not-ready condition is due to either a selective reset or Halt I/O instruction.

A simplified flowchart of the operations performed by the adapter in response to a selective reset is presented in Figure J-10.

3.1.5 SYSTEM RESET

The adapter responds to a system reset from either channel by disconnecting from both channels and resetting its buffer register and status indications. In addition, the adapter element receiving reset is made not-ready and the intervention required sense bit is activated. A subsequent Sense Adapter command to the element not receiving the reset senses the intervention required bit, indicating that the not-ready condition is due to a system reset.

A simplified flowchart of the operations performed by the adapter in response to a system reset is presented in Figure J-10.

JUNE 1, 1971

APPENDIX K. IBM 2701-01 DATA ADAPTER UNIT

1.1 INTRODUCTION

This appendix is provided to describe the basic operational concept of the 2701-01 Data Adapter Unit. The topics discussed provide insight into hardware operation from both the system and unit standpoint. Emphasis is placed upon providing a description which will serve as a reference guide for the programmer.

2.1 SYSTEM CHARACTERISTICS

A brief discussion of how the Data Adapter Unit attaches to and communicates with the 9020E System is presented as an aid to understanding the operation of the Adapter.

2.1.1 SYSTEM CONFIGURATION

The 2701 Data Adapter Unit (DAU) functions as a control unit and an interface converter for Radar Keyboard Multiplexers (RKMs). Each DAU can have up to five RKMs attached to it, but can only operate with one RKM at a time (simplex operation, Figure K-I).

The DAU also connects to two IOCEs via separate interfaces. These paths allow data and commands to be passed between the 9020E System and the RKMs. These interfaces can be enabled and disabled under program control via a unique interface between the Configuration Console (CC) and the DAU. Each DAU contains an interface control register which exercises the interface control.

The RKMs also have two interfaces which allow connection to two separate DAUs. These interfaces are also under program control via the CC. Figure K-I shows a 9020E System having two DAUs and five RKMs.

2.1.2 DAU COMMUNICATIONS

The RKM expects to receive commands from the IOCE and either receive data from, or send data to it as a result of its executing these commands. The DAU is an intermediate device which acts as a control unit between IOCE and RKM. The DAU receives the commands from the IOCE and translates, or reformats, them into command codes that are recognizable by the RKMs. The DAU also directs the commands and data to the proper RKM.

2.1.2.1 Commands

All commands sent from IOCE to RKM must undergo a reformatting process before they are recognizable to the RKM. This process is discussed in section 3.1.2.5 of this Appendix. Commands received by the DAU will be of five general categories; Control, Write, Read, Test, and Status.

Control commands are those which do not require exchange of data at any time other than Initial Selection Time. When the transfer of the command has been completed to the RKM, the interfaces become idle.

Write commands are those that require a variable number of data bytes to be sent from the IOCE to the RKM. Buffering is provided within the

JUNE 1, 1971

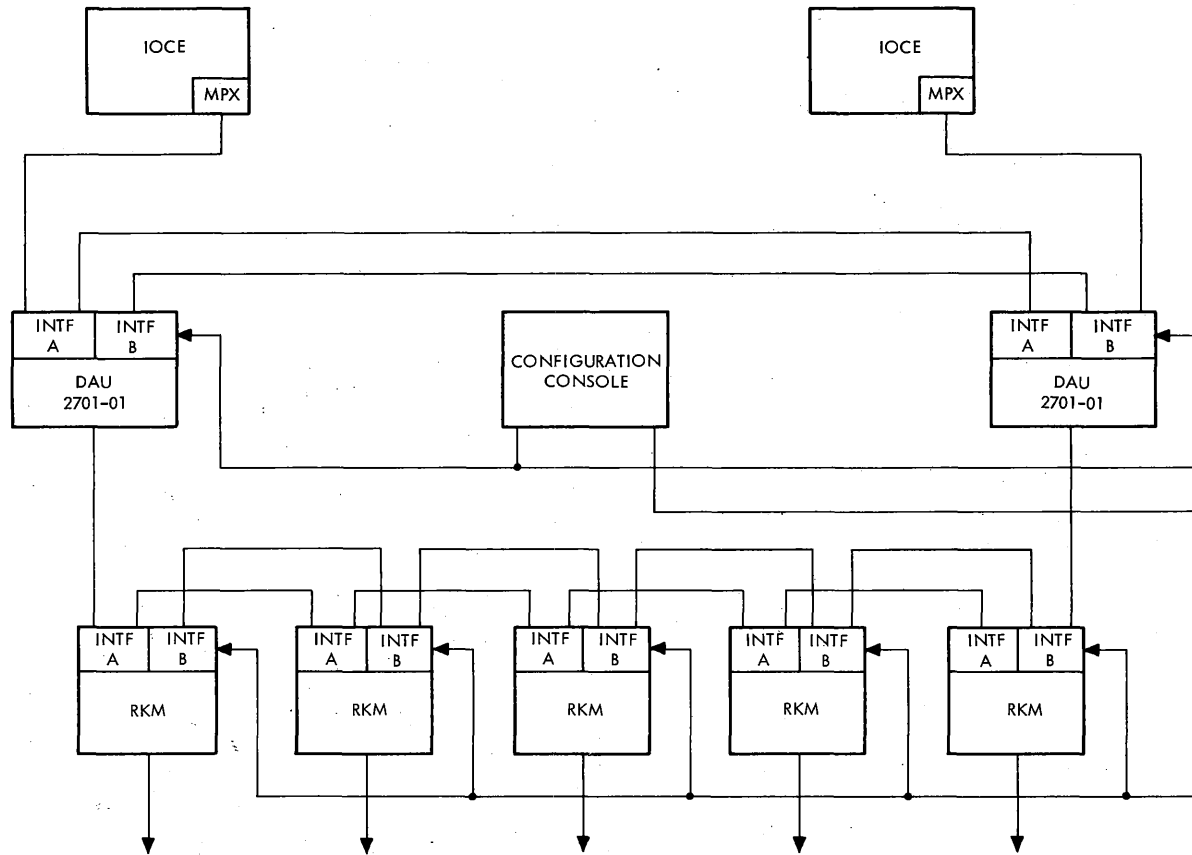


Figure K-1 DAU-RKM System Configuration

JUNE 1, 1971

DAU to allow IOCE/DAU and DAU/RKM interfaces to operate independently, to avoid holding the IOCE for prolonged periods. The number of bytes to be transferred is specified in the Channel Command Word executed by the IOCE (See Chapter 2).

Read commands are those that require data to be transferred from the RKM to the IOCE. The DAU interfaces (IOCE and RKM) operate independently during Read execution and a variable number of data bytes can be transferred during the execution of a single Read command. The number of bytes to be transferred is specified in the Channel Command Word executed by the IOCE (See Chapter 2).

Test commands are those which are directed to the DAU and are not passed on to the RKMs. These commands direct the DAU to perform some special test function which is intended to aid in the maintenance of the DAU. These commands are not intended for use in an operational environment.

Status commands can be directed to either the DAU or the RKM. Both units maintain hardware registers which are used to accumulate both the operational and error status of the respective units. This information is set into the registers as soon as the condition occurs. Status commands allow the programmer to inspect, at his discretion, the information accumulated in the various status registers.

2.1.2.2 Data

The DAU has four data buses connecting it to the system. Two of these buses provide a path to and from the IOCE, while the other two buses provide paths to and from the RKM. The IOCE sends device addresses, commands, and data to the DAU on one bus and expects device addresses, data, and status on the other bus. Likewise commands and data are sent from DAU to RKM on another data bus, and data and status are received from the RKM on the fourth data bus.

All data buses to and from the DAU are one byte wide (8 data bits and 1 parity bit). All commands, data, status, and addresses start and end on byte boundaries. The most significant bit being left justified and least significant bit being right justified within the bytes. All parity checking is done on a per byte basis and is made as each byte is received at the DAU.

The DAU transfers data in a demand response mode of operation. That is, whenever service is required by any unit (IOCE, DAU, RKM) it will signal the appropriate unit to send or receive the data.

2.1.2.3 Chaining

The DAU responds to the IOCE in the same manner as does other control units and does not prohibit chaining. The IOCE can command chain and data chain as described in chapter 10, Input-Output Operations.

2.1.2.4 Mode of Transfer

The DAU is capable of transferring data to the IOCE in either byte or burst mode. The DAU can be manually set to operate in either mode.

3.1 OPERATIONAL CHARACTERISTICS

The DAU provides a data and control interface for the RKM's and the control logic necessary to operate this interface. The DAU operates with up to five RKMs in byte or burst mode but can only operate with one RKM at a time (simplex). The DAU is modified to recognize only five RKM

JUNE 1, 1971

addresses. The DAU can also remember up to five attention interrupts, one per RKM.

3.1.1 ADDRESSING

The DAU will recognize and store five different RKM addresses. During DAU selection time the IOCE will send an 8-bit address to the DAU on the Data Bus Out. The high order five bits of this address are used to identify the DAU while the low order three bits identify the RKM attached to that DAU. If the high order five bits do not compare with the wired DAU address (address is pluggable), or the low order three bits do not identify an attached RKM, the IOCE selection attempt is ignored by the DAU. If the selection is successful the low order three bits are sent to the RKMs on the Address Bus Out (ABO). The ABO is a four bit bus with the high order bit position always equal to zero. The address structure is shown in Table K-I. If the requested RKM is not available, unit check status will be sent to the IOCE and the timeout bit will be set in the sense byte.

Since the DAU is addressed as a single adapter, which can communicate with five attached devices, any of the five RKM addresses may be used for DAU commands such as Sense, Test I/O, or No-Op. For example, if RKM-1 caused unit check to be set, a sense issued for any RKM will retrieve the sense data for RKM-1.

TABLE K-I DAU ADDRESSING STRUCTURE

IOCE Bus Out	DAU ABO	RKM
0 1 2 3 4 5 6 7	0 1 2 3	ID
x x x x x 0 0 1	0 0 0 1	1
x x x x x 0 1 0	0 0 1 0	2
x x x x x 0 1 1	0 0 1 1	3
x x x x x 1 0 0	0 1 0 0	4
x x x x x 1 0 1	0 1 0 1	5

xxxxx = Pluggable DAU Address

3.1.2 COMMANDS

The RKM is controlled by the use of sixteen unique commands. The bit codes for these commands are generated in the DAU. The DAU will decode each command sent from the IOCE as one of the following; Read type, Write Type, Control type, or Status seeking types.

3.1.2.1 Read Type Commands

When issued, this type of command will condition the DAU to receive data from the RKM.

3.1.2.2 Write Type Commands

When issued this type of command will condition the DAU to send data to the RKM.

JUNE 1, 1971

3.1.2.3 Control Type Commands

When issued the DAU will present a normal ending to the IOCE and send the control information to the RKM.

3.1.2.4 Status Seeking Commands

This type of command conditions the DAU to accept status information from the RKM and present it to the IOCE with a normal ending.

3.1.2.5 Interface Control Commands

These two commands (Reserve and Release) control the dedication of the DAU to one of its two IOCE channel interfaces.

3.1.2.5.1 RESERVE. When the DAU accepts a Reserve command from either of its two IOCE channel interfaces, the two channel switch will be dedicated to that interface until a Release command is received (over that interface), or the interface is switched via configuration control. Once reserved, or dedicated to an interface, the DAU will be unavailable to the other interface.

3.1.2.5.2 RELEASE. After a Release command, the DAU is free to accept selections from either interface. This is contingent upon the setting of the configuration register (must be in states 1,2, or 3), or the setting of the enable switches when in state 0.

3.1.2.6 Command Formatting

The command that appears on the IOCE Bus Out will be reformatted according to the type of command and will be presented to the RKM on the Data Bus Out during the RKM selection sequence. Refer to Table K-II for command bit codes for both IOCE/DAU codes and the reformatted equivalent DAU/RKM codes. Table K-III gives a complete list of IOCE/DAU commands including an explanation of the command modifier bits.

3.1.3 STATUS

3.1.3.1 RKM Status

The RKM maintains three bytes of status that are callable by the program. The first byte contains general information pertinent to the operation of the RKM. This byte is obtainable via the execution of the Read RKM Status command.

There are two bytes of error status that are maintained in the RKM that are sent to the system when the Read Error Status 1 or 2 commands are issued. The formats of these status bytes are listed in the following text.

NOTE

Unless otherwise stated all bits in the three status bytes will be reset whenever the corresponding status byte is read via one of the three commands.

3.1.3.1.1 READ RKM STATUS COMMAND. Yields 8-bits of general operational status to the system. The DAU treats the status byte as data to be placed in storage at the address specified in the CCW. The bit assignments are as follows:

<u>Bit</u>	<u>Name</u>
0	RKM Busy
1	Operator Busy

JUNE 1, 1971

2	Interrupt
3	Display Constants Fetched
4	Console Entry
5	Polling
6	Error 2
7	Error 1

RKM Busy - This bit is set to a "1" whenever the RKM is performing an operation and can only respond to a disconnect order.

It is set by the reception of a valid order (non-status, non-disconnect) and reset by reception of a disconnect signal or disconnect order.

TABLE K-II RKM-DAU COMMANDS AND FORMATS

DAU/RKM BUS							IOCE/DAU BUS							HEX						
b ₀	b ₁	b ₂	b ₃	b ₄	b ₅	b ₆	b ₇	RKM DATA BUS		SYSTEM 360 BUS		0	1	2	3	4	5	6	7	
0	0	0	0	1	0	0	0	DISCONNECT				0	1	0	0	0	1	1	1	47
1	0	0	1	1	b	0	0	WRITE RKM				1	1	b	0	0	0	0	1	V1
1	0	0	1	0	0	0	0	CLEAR RKM MEMORY				1	0	0	0	0	0	0	0	B1
1	0	0	1	0	y	1	0	BLINK DISPLAY SET/RESET				1	0	y	1	0	0	0	1	W1
1	0	0	0	0	1	0	0	FETCH DISPLAY CONSTANT				0	0	1	0	0	0	0	1	21
1	0	0	0	0	0	0	0	LOAD QUEUE ADDRESS				0	0	0	0	0	0	0	1	01
1	0	0	0	1	1	0	0	MOVE DATA				0	1	1	0	0	0	0	1	61
1	0	0	0	1	0	0	0	SPECIAL MOVE				0	1	0	0	0	0	0	1	41
1	1	0	1	1	b	0	0	READ RKM				1	1	b	0	0	0	1	0	X2
1	1	0	0	0	0	0	0	READ QUEUE ADDRESS				0	0	0	0	0	0	1	0	02
1	0	1	0	0	0	0	0	LOAD I/O ADDRESS				0	0	0	0	0	1	0	1	05
1	0	1	1	0	0	0	0	KEYBOARD WRITE				1	0	0	0	0	1	0	1	85
1	0	1	B	1	L ₀	L ₁	L ₂	BIT MODIFY				B	1	L ₀	L ₁	L ₂	1	0	1	YZ
0	0	0	0	0	0	0	0	READ RKM STATUS				0	0	0	0	0	1	1	0	06
0	0	0	0	0	0	0	1	READ RKM ERROR STATUS 1				0	0	0	0	1	1	1	0	0E
0	0	0	0	0	0	1	0	READ RKM ERROR STATUS 2				0	0	0	1	0	1	1	0	16
ALTERNATE COMMANDS																				
1	1	0	0	0	0	0	0	READ QUEUE ADDRESS				0	0	1	0	0	0	1	0	22
1	0	1	0	0	1	0	0	LOAD I/O ADDRESS				0	0	1	0	0	1	0	1	25
1	0	0	1	0	1	0	0	CLEAR RKM MEMORY				1	0	1	0	0	0	0	1	A1
1	0	1	1	0	1	0	0	KEYBOARD WRITE				1	0	1	0	0	1	0	1	A5

U = 0 or 2 X = C or E
V = C or E Y = 4, 5, 6, 7,
W = 9 or B C, D, E or F
Z = 5 or D

B Bit to be stored (1 or 0)
L Binary Encoded position of bit B
b When b=1 Codes -- 11 1111
and -- 11 1110 are interchanged
at RKM interface during
function execution
y 1 = Set 0 = Reset
M Modifier Bit (1 or 0)

COMMANDS FOR DAU ONLY

DIAGNOSTIC WRITE	0	0	M	0	1	0	0	1	U9
DIAGNOSTIC READ	0	0	0	0	1	0	1	0	0A
TEST I/O	0	0	0	0	0	0	0	0	00
SENSE	0	0	0	0	0	1	0	0	04
NO-OP	0	0	0	0	0	0	1	1	03
RESERVE	1	1	1	1	0	1	0	0	F4
RELEASE	1	1	0	1	0	1	0	0	D4

JANUARY 31, 1973

TNL: GN31-3005

TABLE K-III IOCE/DAU EXECUTABLE COMMANDS

COMMAND	BIT CODE	HEX CODE
Disconnect	0100 0000	47
Write RKM	11b0 0001	C1, E1
Clear RKM Memory	10z0 0001	81, A1
Blink Display Set/Reset	10Y1 0001	91, B1
Fetch Display Constants	0010 0001	21
Load Queue Address	0000 0001	01
Move Data	0110 0001	61
Special Move	0100 0001	41
Read RKM	11b0 0010	C2, E2
Read Queue Address	00z0 0010	02, 22
Load I/O Address	00z0 0101	05, 25
Keyboard Write	10z0 0101	85, A5
Bit Modify	B1xx x101	XX
Read RKM Status	0000 0110	06
Read RKM Error Status 1	0000 1110	0E
Read RKM Error Status 2	0001 0110	16
Diagnostic Write	00m0 1001	09, 29
Diagnostic Read	0000 1010	0A
Test I/O	0000 0000	00
Sense	0000 0100	04
No-Op	0000 0011	03
Reserve	1111 0100	F4
Release	1101 0100	D4

b = When b = 1, codes --11 1111 and --11 1110 are interchanged at the RKM interface during function execution

m = modifier bit

z = 0 or 1

Y = 1 for Set, 0 for Reset

B = Bit to be stored at designated RKM storage location

xxx = Binary encoded bit location of 'B'

Operator Busy - This bit is set to a "1" whenever the operator unit of the RKM is busy performing a clear, blink control or move operation. Any non operator instruction may be executed.

Interrupt - This bit is set to a "1" when an interrupt signal has been sent to the DAU. It is reset to "0" by the Read RKM Status message.

Display Constants Fetched - The bit is reset to "0" each time the Fetch Display Constants message is received by the RKM. When the fetch is complete, the bit is set to "1". When the bit makes the transition to the one state the interrupt A signal is activated and the interrupt bit set to "1".

Console Entry - This bit is reset to "0" when the queue address is read and is set when, at the end of a poll of each console, data has been stored in the queue list. When the bit makes the transition to one state, the interrupt B signal is activated and the interrupt status bit set to "1".

Polling - This bit is a "1" whenever the console polling logic is actually polling the consoles.

Error 2 - This bit is set if any bit in RKM Error Status 2 is set indicating an internal RKM failure.

Error 1 - This bit is set if any bit in RKM Error Status 1 is set indicating a procedural or interface error.

3.1.3.1.2 READ RKM ERROR 1. This status command yields the following information to the system. The DAU treats this information as data to be stored at the address specified in the CCW.

<u>Bit</u>	<u>Name</u>
0	Overtemperature indication
1	I/O Power Supply Failed
2	Bus Out Parity Error
3	Display Constants Fault - Parity Error
4	No Response to Interrupt.
5	Procedural Error
6	Display Constants Fault - No Start
7	Queue Overflow

Overtemperature - This bit is set whenever an overtemperature condition occurs.

I/O Power Supply failure - This bit is set whenever the I/O power supply develops a fault condition.

Bus Out Parity Error - This bit is set whenever the RKM detects a parity error on any data or order sent from DAU to RKM.

Display Constants Parity - This bit is set if a parity error is detected for any of the eight bytes of constants.

No Response to Interrupt - This bit is set whenever a Read Queue command is not received within 36 to 54 msec following an Interrupt B signal. (See Interrupts, Section 3.1.5)

Procedural Error - This bit is set whenever the RKM detects a procedural error at the DAU/RKM interface. Conditions which set this bit are:

- a. Incorrect number of bytes for a fixed length message.
- b. Receipt of a non-status non-disconnect command while busy.

Display Constants fault - If there is no response from the addressed console, this bit is set. Whenever the request for display constants is received by the RKM, this bit is reset.

Queue Overflow - This bit is set to indicate that the queue address counter has reached the maximum address. The bit is reset by a Load Queue Address Instruction.

3.1.3.1.3 READ RKM ERROR 2. This status command yields the following information to the system. The DAU treats the status as data to be stored at the address specified in the CCW.

JANUARY 31, 1973

TNL: GN31-3005

<u>Bit</u>	<u>Name</u>
0	Invalid Command Code
1	Console Parity
2	Memory Parity
3	Data Operator Address Overflow
4	Data Operator Hangup
5	Interface Hangup
6	Data Control Hangup
7	Test Bit

Invalid Command Code - This bit is set whenever the RKM detects an invalid command or a parity error on a command.

Console Parity - This bit is set whenever a parity error is detected on data coming from the console during polling, not during display constants retrieval.

Memory Parity - This bit is set whenever a parity error is detected when reading the RKM memory.

Data Operator Address Register Overflow - This bit is set whenever an overflow is detected in the Data Operator Address Register.

Data Operator Hangup - This bit is set whenever the data operator detects an operational area associated with it.

Interface Hangup - This bit is set whenever a DAU/RKM interface sequence is violated.

Data Control Hangup - This bit is set whenever a functional or sequencing error is detected in the RKM data control circuitry.

Test Bit - This bit is set whenever the CC commands a test operation.

3.1.3.2 DAU Status

The DAU maintains one byte of status data which is transmitted to the IOCE during normal and abnormal ending sequences. This status is stored in the CSW location whenever a status cycle occurs.

<u>Bit</u>	<u>Name</u>
0	Attention
1	Status Modifier
2	Control Unit End
3	Busy
4	Channel End
5	Device End
6	Unit Check
7	Unit Exception

The following combinations of the status bit can be set simultaneously.

Channel End, Device End - Indicates a normal end of an operation.

Channel End, Device End, Attention* - Indicates a normal end of an operation and an attention interrupt signal has been received from the RKM.

Channel End, Device End, Unit Check - Indicates an unusual ending. The sense bits should be inspected for a better definition of the error.

Attention* - Indicates that an interrupt has been received from one of the RKMs.

Busy - Indicates that the DAU was operating at time an attempt was made to select it. It may accompany an A or B type interrupt.

Channel End, Device End, Unit Check, Attention* - Indicates an unusual ending and an attention signal has been received from that RKM. The sense byte should be inspected for a definition of the error condition.

Busy, Status Modifier - Indicates that the DAU is operating and an attempt has been made to select it with an address other than the operating address.

Channel End, Device End, Control Unit End - Indicates a normal end and that the DAU had previously presented Busy and Status Modifier to the channel.

Channel End, Device End, Control Unit End, Attention* - Indicates a normal end, an attention signal had been received from that RKM, and the DAU had previously presented Busy and Status Modifier to the channel.

Channel End, Device End, Control Unit End, Unit Check - Indicates an unusual ending and that the DAU had previously presented Busy and Status Modifier to the channel. The sense bits should be inspected for a definition of the error condition.

Channel End, Device End, Control Unit End, Unit Check, Attention* - Indicates an unusual ending, an attention signal has been received from that RKM, and that the DAU had previously presented Busy and Status Modifier to the channel. The sense bits should be inspected for a definition of the error condition.

Attention*, Control Unit End - Indicates that while trying to present this stand-alone attention, from an RKM, Busy and Status Modifier was presented to the channel.

Unit Check - Indicates that during an initial selection a Bus Out check or a Command Reject occurred.

* Attention will be accompanied by indication of the interrupt. Interrupt In A by Status Modifier, and Interrupt In B by Unit Exception.

3.1.4 SENSE

3.1.4.1 RKM Sense Bits

The RKM does not maintain any sense information as such. The two RKM Error Status bytes described in sections 3.1.3.1.2 and 3.1.3.1.3 contain a summary of all the RKM error data.

3.1.4.2 DAU Sense Bits

The DAU maintains one byte of sense information which can be called for by the IOCE via the execution of the SENSE command. These bits are provided as an extension of the Status byte. They further define the situation that produced abnormal status indications. The following paragraphs describe the sense byte format.

Bit	Name
0	Command Reject
1	Intervention Required
2	Bus Out Check
3	----
4	Data Check
5	RKM Error

JANUARY 31,1973

TNL: GN31-3005

6 ----
7 Timeout

Command Reject - is set whenever the DAU detects an invalid command code from the IOCE.

Intervention Required - is set when the time between subsequent data transfers (signaled by the Request In line) is 112 usec or more. Intervention required is also set if the RKM does not raise its Request In signal within 112 usec after the completion of the 2701/RKM selection sequence.

Bus Out Check - is set when a parity error is detected on the IOCE Bus Out.

Data Check - Is set when a parity check occurs on the RKM Bus In during a Read Operation. When writing, Data Check is set when a parity check occurs at the output of Transmit Buffer A.

RKM Error - is set when the RKM raises its Error In signal during a data transfer operation.

Timeout - is sent whenever the Response In Line is not activated by the RKM within 1.9 usec after the DAU raises the Order Out, or Acknowledge Out signal.

3.1.5 INTERRUPTS

The DAU retains interrupts from up to five RKMs. The interrupts can be received asynchronous to any operation and will be presented to the system when the DAU is free. The only DAU interrupts to the IOCE are those that originate in the RKM.

Each RKM has two interrupt signals that are sent to the DAU. These signals represent the availability of data in the RKM. The interrupts are designated Interrupt In A and Interrupt In B. Interrupt In A will set Status Modifier and Attention; Interrupt In B will set Unit Exception and Attention.

3.1.5.1 Interrupt In A

This signal from the RKM indicates that display constants, which were requested by the execution of a Fetch Display Constants command, are in the RKM memory ready to be used. This signal sets bits 2 and 3 of the RKM Status Register (Section 3.1.3.1.1).

3.1.5.2 Interrupt In B

This signal from the RKM indicates the completion of a poll of all attached consoles (32 max) if data has been stored in the queue as a result of the poll. The data has been stored in the RKM memory according to the queue address. This signal accompanies bits 2 and 4 of the RKM status register (section 3.1.3.1.1).

4.1 RECONFIGURATION

The DAU has provisions for program control of the enabling and disabling of the Two Processor Switch (TPS) interfaces to the IOCEs. This control is provided over a unique interface between the DAU and the CC. No commands are executed within the DAU to perform reconfiguration consequently the DAU can accept reconfiguration signals at any time.

At the start of the reconfiguration sequence the CC will send a Reconfiguration Strobe signal and the desired configuration to the DAU. If the configuration is valid (all valid except when both TPS interfaces are enabled) the DAU sends an Accept signal to the CC. If the configuration is not valid, an Accept signal will not be sent and the existing configuration will remain active.

Switches are provided on the DAU to allow manual overriding of the program set configuration. These switches are under State control and are only active when the DAU is in State 0. When the DAU is in the manual configuration mode a not ready indication is sent to the CC, and the DAU will ignore any attempts, by the program, to perform reconfiguration.

The DAU may not change configuration immediately upon accepting it from the CC. Configuration changes only occur when the DAU is idle. Therefore, if the DAU is busy the new configuration will remain pending the completion of the operation.

Two signals are sent to the CC which indicate the current configuration of the TPS; Interface A Enabled and Interface B Enabled. The program can inspect these signals by issuing a Read Status command to the CC. When the DAU accepts a configuration, the program should read back the current TPS configuration, to make sure the new configuration has been set before attempting to access the DAU.

Reset capabilities also exist between the CC and the DAU. Upon detection of the reset signal from the CC the DAU will force a DAU reset and enter the idle state.

5.1 PROGRAM MAINTENANCE AIDS

The DAU provides commands that are designed as maintenance aids. They are Diagnostic Write, and Diagnostic Read. These commands are executed via the IOCE interface and are used to exercise the data paths and control logic within the DAU.

The Diagnostic Write command (09) transfers data from the IOCE through the DAU into the Diagnostic Register (DR). No restriction exists as to the number of bytes that can be transferred by the command. However, only the last byte received will be retained in the DR. All other bytes will be accepted by the DAU but not retained in the DR.

The Diagnostic Write Modified command (29) is handled the same as the Diagnostic Write in that it transfers an unrestricted number of bytes between the IOCE and the DR. In addition the modified write sets all five of the RKM attention interrupt latches with indication of both A and B interrupts. Upon completion of the command five attention interrupts will be presented (one at a time) to the system. This function allows the interrupt handling and associated address generation circuitry to be tested.

The Diagnostic Read command (0A) is used to transfer data from the DR through the DAU to the IOCE. No restriction exists as to the byte count used for the command. The contents of the DR will be gated out until the count goes to zero.

5.1.1 COMMAND CHECKING

To provide the software with the capability of checking the output of the command formatter, the Diagnostic Register will ingate the formatter output every time a command passes through it. This action is taken independent of the state of the DAU. The Diagnostic Read command can be

JANUARY 31,1973

TNL: GN31-3005

used to retrieve the reformatted command following the execution of any RKM command.

When the DAU is in state zero and a command is issued to an RKM, a unit check will occur with the timeout bit set in the sense register. This action does, however, load the reformatted RKM command into the DR which can then be inspected by issuing a Diagnostic Read command.



Technical Newsletter

This Newsletter No. GN31-3005

Date January 31, 1973

Base Publication No. A27-2734-2

Previous Newsletters None

IBM Systems Reference Library
IBM 9020D and 9020E System Principles of Operation

This Technical Newsletter provides replacement pages for the subject publication. The following pages are to be inserted:

ii-xxii ✓	361, 362 ✓
107, 108 ✓	391, 392 ✓
161-164 ✓	423, 424 ✓
169-174 ✓	437, 438 ✓
181, 182 ✓	459, 460 ✓
191-214 ✓	533, 534 ✓
214.1, 214.2 (added) ✓	559-562 ✓
219-222 ✓	579-588 ✓
237-240 ✓	597, 598 ✓
240.1, 240.2 (added) ✓	619-626 ✓
325, 326 ✓	627 (added) ✓

A change to the text or to an illustration is indicated by a vertical line to the left of the change.

Summary of Amendments

- Display Instructions
- Channel-to-Channel Adapter
- 2701-01 Data Adapter Unit
- Miscellaneous typographical errors corrected
- Clarification of selected areas due to comments received.

NOTE: Please file this cover letter at the back of the manual to provide a record of changes.

A27-2734-2
Contract No. FA64WA-5223

Contract No. FA64WA-5223 Printed in U.S.A. A27-2734-2

IBM

**International Business Machines Corporation
Federal Systems Division, FAA Project Office
18100 Frederick Pike, Gaithersburg, Maryland 20760**