XDS

Xerox Data Systems

Reference Manual

XDS 925 Computer

XDS 925 BASIC INSTRUCTIONS

Mnemonic		Octal Code	Name	Page Ref.
LOAD / STORE				
LDA	A, T	76	Load A	9
STA	A, T	35	Store A	9
LDB	A, T	75	Load B	9
STB	A, T	36	Store B	9
LDX	A, T	71	Load Index	9
STX	A, T	37	Store Index	9
EAX	A, T	77	Copy Effective Address into Index Register	10
ARITHMETIC				
ADD	A, T	55	Add Memory to A	10
MIN	A, T	61	Memory Increment	10
MDE	A, T	60	Memory Decrement	10
SUB	A, T	54	Subract Memory from A	10
MUS	A, T	64	Multiply Step	10
DIS	A, T	65	Divide Step	11
LOGICAL				
ETR	A, T	14	Extract	11
MRG	A, T	16	Merge	11
EOR	A, T	17	Exclusive OR	11
REGISTER CHANGE				
RCH		46	Register Change	12
CLR		0 46 30000	Clear AB	12
XAB		0 46 00000	Exchange A and B	12
BAC		0 46 10000	Copy B into A, Clear B	12
ABC		0 46 20000	Copy A into B, Clear A	12
BRANCH				
BRU	A, T	01	Branch Unconditionally	12
BRX	A, T	41	Increment Index and Branch	12
BRM	A, T	43	Mark Place and Branch	13
BRR	A, T	51	Return Branch	13

Mnemonic		Octal Code	Name	Page Ref.
TEST / SKIP				
SKG	A, T	73	Skip if A Greater than Memory	13
SKM	A, T	70	Skip if A = Memory on B Mask	13
SKA	A, T	72	Skip if A and Memory Do Not Compare Ones	14
SKN	A, T	53	Skip if Memory Negative	14
SKS	A	40	Skip if Signal Not Set	24, 35, 36
SHIFT				
RSH	N, T	0 66 000XX	Right Shift AB	15
RCY	N, T	0 66 200XX	Right Cycle AB	15
LSH	N, T	0 67 000XX	Left Shift AB	15
LCY	N, T	0 67 200XX	Left Cycle AB	16
NOD	N, T	0 67 100XX	Normalize and Decrement X	16
CONTROL				
HLT		00	Halt	16
NOP		20	No Operation	16
EXU	A, T	23	Execute	16
BREAKPOINT TESTS				
BPT 1		0 40 20400	Breakpoint No. 1 Test	17
BPT 2		0 40 20200	Breakpoint No. 2 Test	17
BPT 3		0 40 20100	Breakpoint No. 3 Test	17
BPT 4		0 40 20040	Breakpoint No. 4 Test	17
OVERFLOW				
OVT		0 40 20001	Overflow Indicator Test and Reset	17
ROV		0 02 20001	Reset Overflow	17
INTERRUPT				
EIR		0 02 20002	Enable Interrupt System	21
DIR		0 02 20004	Disable Interrupt System	21
IET		0 40 20004	Interrupt Enabled Test	21
IDT		0 40 20002	Interrupt Disabled Test	21
AIR		0 02 20020	Arm Interrupts	21

Legend: A = address; T = tag field; N = number of shifts

Price: $3.75

XDS 925 COMPUTER REFERENCE MANUAL

90 00 99B

September 1966

XDS

Xerox Data Systems/701 South Aviation Boulevard/El Segundo, California 90245

Printed in U.S.A.

REVISIONS

This publication, 90 00 99B, augments and corrects the previous edition of the XDS 925 Computer Reference Manual, 90 09 99A. The additions are Appendix C, Computer Operating Procedures, and Appendix D, Detailed Machine Functions, including Instruction Execution, Typical Interrupt Cycle, and Buffered Input/Output.

Changes to the previous edition are indicated by a vertical line in the margin of the affected page.

RELATED PUBLICATIONS

Title	Publication No.
XDS 910/925 Programmed Operators Technical Manual	90 00 18
XDS 925 Computer EXAMINER Diagnostic System Technical Manual	90 06 49
XDS SYMBOL and META-SYMBOL Reference Manual	90 05 06
XDS MONARCH Reference Manual	90 05 66
XDS FORTRAN II Reference Manual	90 00 03
XDS 900 Series FORTRAN II Operations Manual	90 05 87
XDS ALGOL 60 Reference Manual	90 06 99
XDS Project Management System Reference Manual	90 08 18
XDS MANAGE Reference Manual	90 10 46
XDS Business Language Reference Manual	90 10 22
XDS Sort/Merge Reference Manual	90 09 97
XDS 900 Series Utility and Debug Package (AID)	01 20 13

CONTENTS

I. GENERAL DESCRIPTION 1

Introduction 1
XDS 925 Registers 3
XDS 925 Memory 4
Memory Word Formats 4
Special Characteristics 5

II. MACHINE INSTRUCTIONS 9

Introduction 9
Load/Store Instructions 9
Arithmetic Instructions 10
Logical Instructions 11
Register Change Instructions 12
Branch Instructions 12
Test and Skip Instructions 13
Shift Instructions 14
Control Instructions 16
Breakpoint Tests 17
Overflow Instructions 17
Floating-Point Operations 17

III. INTERRUPT SYSTEM 19

Priority Interrupt System 19
Priority Interrupt Operations 19
Interrupt Control 20
Non-Interruptable Instructions 20
Interrupt Location Assignments 20
Enable/Disable Interrupt Instructions 20
Arming Feature (Optional) 21
Channel Interrupt Designations 22
End-of-Word/Tranmission Interrupt Operations; Compatible Mode 22
Count Equals Zero/End-of-Record; Extended Mode 22

IV. INPUT/OUTPUT INSTRUCTIONS 23

Introduction 23
Direct Memory Access System 24
Primary Input/Output Instructions 24
Communications Channel Input/Output 25
Control Console 42
Peripheral Equipment Description 47
Paper Tape Input/Output 47
Card Input/Output 50
Magnetic Tape Input/Output 54
Line Printer 60

INDEX Index-1

APPENDIXES

A. CONVERSION TABLES

XDS Character Codes A-1
Table of Powers of Two A-2
Octal – Decimal Integer Conversion A-3
Octal – Decimal Fraction Conversion A-7

APPENDIXES (cont.)

B. TWO'S COMPLEMENT ARITHMETIC A-10

C. COMPUTER OPERATING PROCEDURES A-11

D. DETAILED MACHINE FUNCTIONS

Instruction Execution A-13
Typical Interrupt Cycle A-15
Buffered Input/Output A-17

E. PROGRAMMED OPERATORS A-19

F. CHANNEL MEMORY ACCESS PRIORITY A-21

G. OPTIONAL EQUIPMENT

Data Multiplexing System A-22
Memory Interface Connection A-26
Automatic Power Fail-Safe System A-26
Memory Parity Interrupts A-26
Real-Time Clock A-27

H. XDS 925 INSTRUCTION LISTS

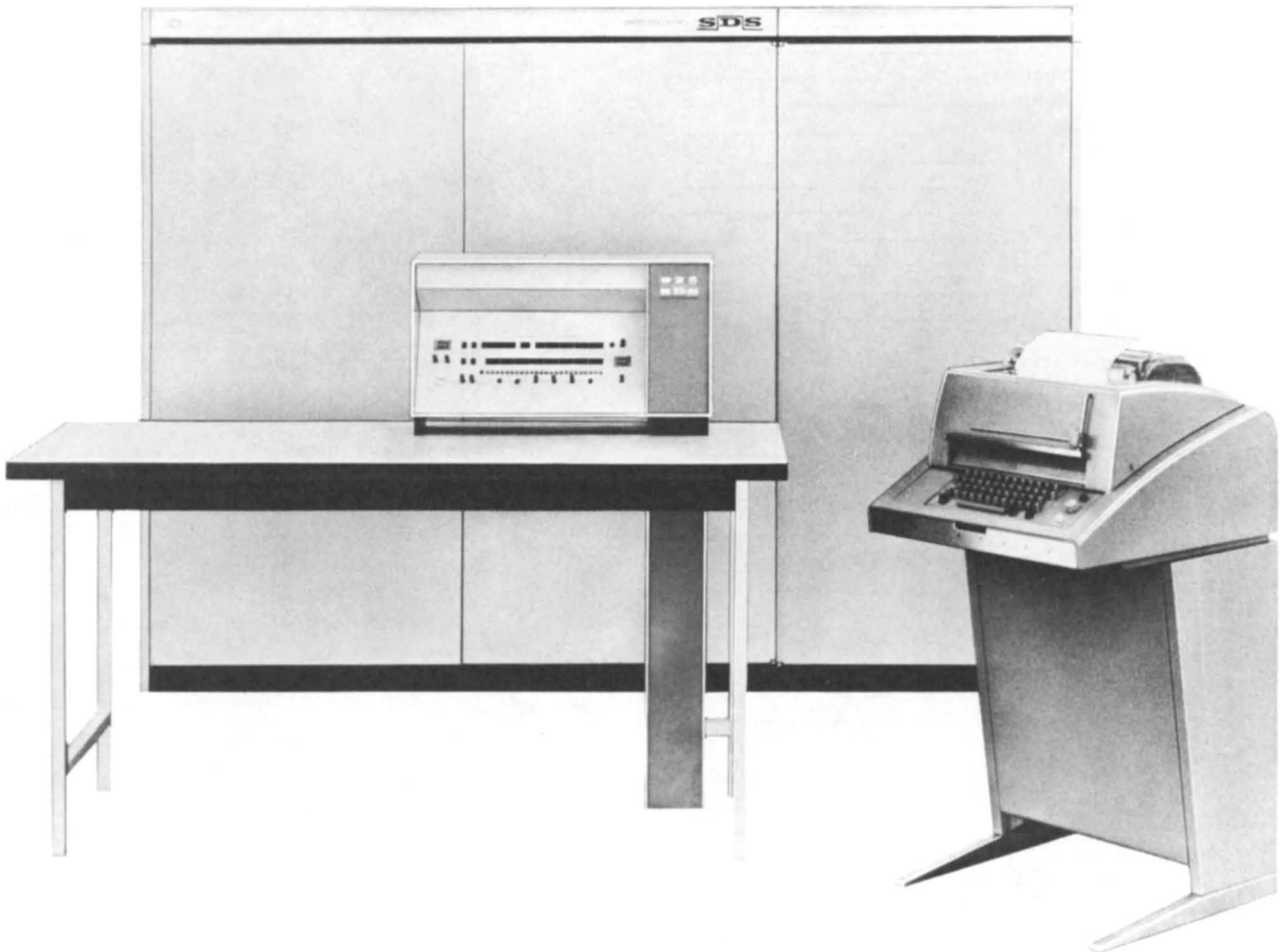
Functional Categories A-28
Numerical Order A-33
Alphabetical Order A-36

FIGURES

XDS 925 Computer (Frontispiece) iv
1-1. XDS 925 Computer Configuration 2
1-2. Basic Register Flow Diagram 3
3-1. Interrupt Arm – Enable Response 22
4-1. XDS 925 Time-Multiplexed Communication Channel, Block Diagram 27
4-2. XDS 925 Direct Access Communication Channel, Block Diagram 28
4-3. XDS 925 Computer Control Panel 43
4-4. Card Read into Memory in Hollerith 50
4-5. Printer Control Indicator Lights and Switches 60
D-1. Instruction Execution Diagram A-12
D-2. Priority Interrupt System Diagram A-14
D-3. Buffer Operation, Single-Word Transmission A-16
D-4. Buffer Operation, Interlaced Transmission A-18
G-1. XDS 925 Overall Computer Configuration A-24

TABLES

3-1. Interrupt Locations 20
4-1. Unit Address Codes 30
4-2. Format Control Characters 63



XDS 925 Computer

I. GENERAL DESCRIPTION

INTRODUCTION

The XDS 925 is a high-speed, low-cost, general-purpose, digital computer with the following characteristics:

- 24-bit word plus parity bit
- Binary arithmetic
- Single-address instructions with

  Index Register
  Indirect Addressing
  Programmed Operators

- Basic Core memory of 4096 words, expandable to 16,384 words, all directly addressable with 1.75-microsecond cycle time
- 4096, 8192, and 16,384-word memory banks available
- Typical execution times (including memory access and indexing):

Fixed-Point Operations (in microseconds)

Add	3.5
Multiply	54.25

Floating-Point Operations (in microseconds)

	24-bit Fraction (plus 9-bit Exponent)	39-bit Fraction (plus 9-bit Exponent)
Add	94.5	196
Multiply	101.5	371

- Program interchangeability with other XDS 900 Series Computers
- Parity checking of memory and I/O operations
- Priority Interrupt System

  System Interrupts, up to 896 optional
  XDS Options Interrupts, two standard, up to 38 more, optional

- Memory non-volatile with power failure; power fail-safe feature (optional) permits saving contents of programmable registers
- Up to four I/O communication channels (with optional interlacing capability), time-multiplexed with computer operation, providing input/output rates of up to one word per 3.5 microseconds
- An optional Direct Memory Access System providing I/O rates of up to one word per 1.75 microseconds

  One to four Direct Access Communication Channels that incorporate the Direct Memory Access System

  Data Multiplex Channel that uses direct memory access connection and accepts/transmits information from external devices, or sub-channels, which may operate simultaneously; thus, externally controlled and sequenced equipment may perform input/output buffering and control operations rather than the computer

- Time-Multiplexed input/output channels operate upon either words or characters. 6 bits is the standard character size; 6 and 12-bit characters, or 6, 12, and 24-bit characters can be specified as desired. Direct Access Channels operate upon words and characters. These channels accept 6, 8, 12, and 24-bit characters. The number of characters per word is specified by the external device
- Input/output with Scatter-Read and Gather-Write facility
- An optional high-speed Word Parallel Input/Output System that is independent of the communication channels and has a transfer rate of one word per memory cycle
- Standard input/output

  Time-Multiplexed Communication Channel (without interlace)

  Control Console

  Dual-channel Priority Interrupt

- Optional input/output devices

  Automatic Typewriter

  Photoelectric Paper Tape Reader and Paper Tape Punch, and Spooler mounted on cart

  MAGPAK Magnetic Tape System

  Magnetic Tape Units (IBM-compatible; binary and BCD)

  Punched card equipment

  Line Printers, Graph Plotters

  Keyboard printers with electromechanical Paper Tape Reader and Punch, auxiliary Rapid Access Data Files

  Communications equipment, Teletype Consoles, Display Oscilloscopes

  A/D Converters, digital multiplexer equipment, and other special system equipment

- FORTRAN II and Symbolic Assembler as part of complete software package
- All silicon semiconductors
- Operating temperature range: 10° to 40°C
- Dimensions: 124 inches x 25-1/2 inches x 65-1/2 inches
- Power: 3 KVA

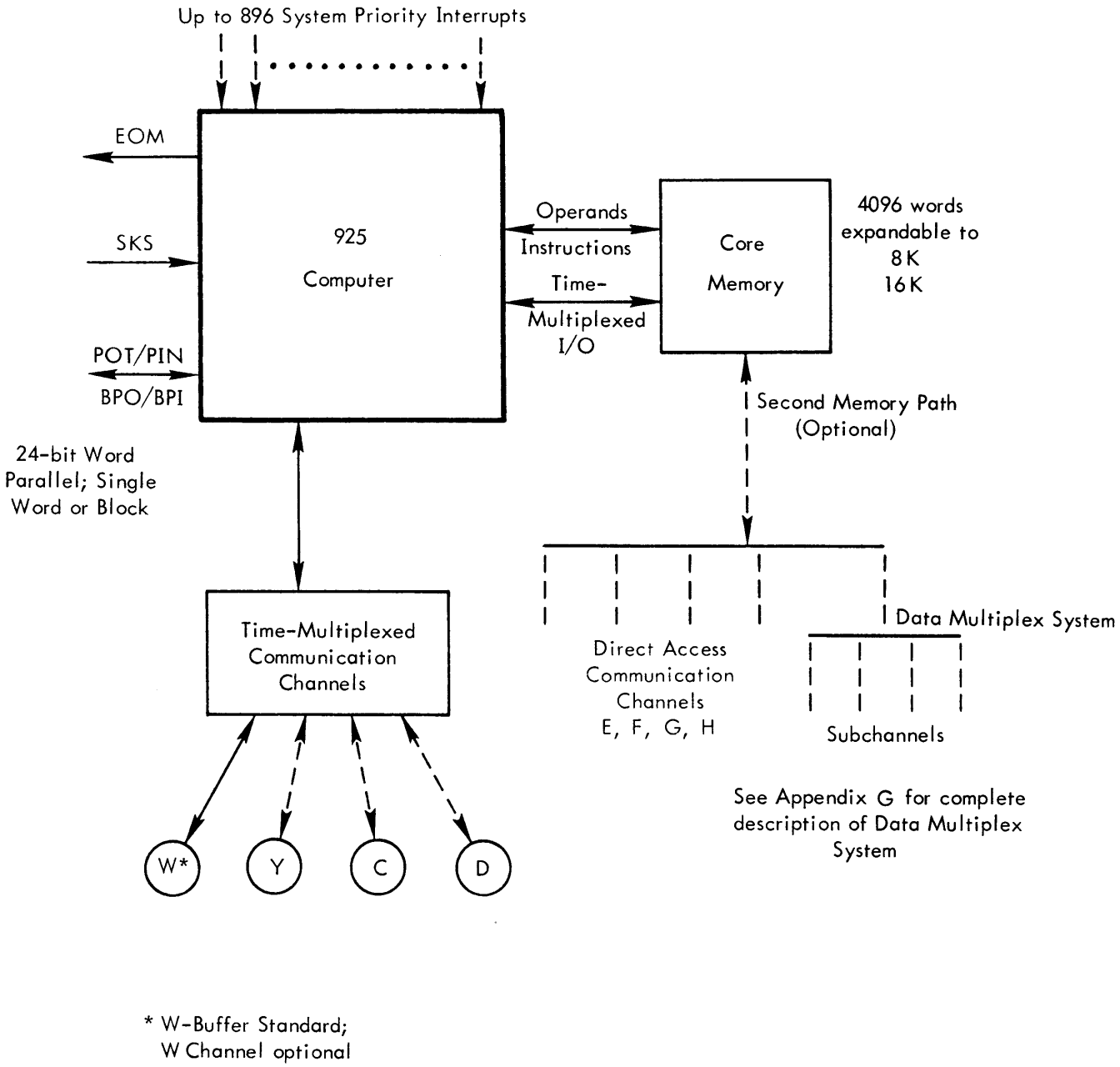


Figure 1-1. XDS 925 Computer Configuration

XDS 925 REGISTERS

The 925 Central Processor contains the following arithmetic and control registers. They are full-word, 24-bit registers except as noted.

AVAILABLE TO THE PROGRAMMER (see Fig. 1-2, dark lines)

The A Register is the main accumulator of the computer. The B Register is an extension of the A Register. The B Register contains the less significant portion of double-length numbers.

The Index Register, X, used in address modification, is a full-word register. Indexing operations occur only with the least significant 14 bits.

The P Register is a 14-bit register that contains the memory address of the current instruction. Unless modified by the program, the contents of P increase by one at the completion of each instruction.

NOT AVAILABLE TO THE PROGRAMMER (see Fig. 1-2, light lines)

The S Register is a 14-bit register that contains the address of the memory location to be accessed for instructions or data.

The C Register is an arithmetic and control register. All instructions come from memory to the C Register before decoding. Address modification and parity generation/detection take place in the C Register.

The O Register is a 6-bit register that contains the instruction code of the instruction being executed.

The M Register is a 25-bit register that holds each word as it comes from memory. Recopying of a word into memory takes place from the M Register.

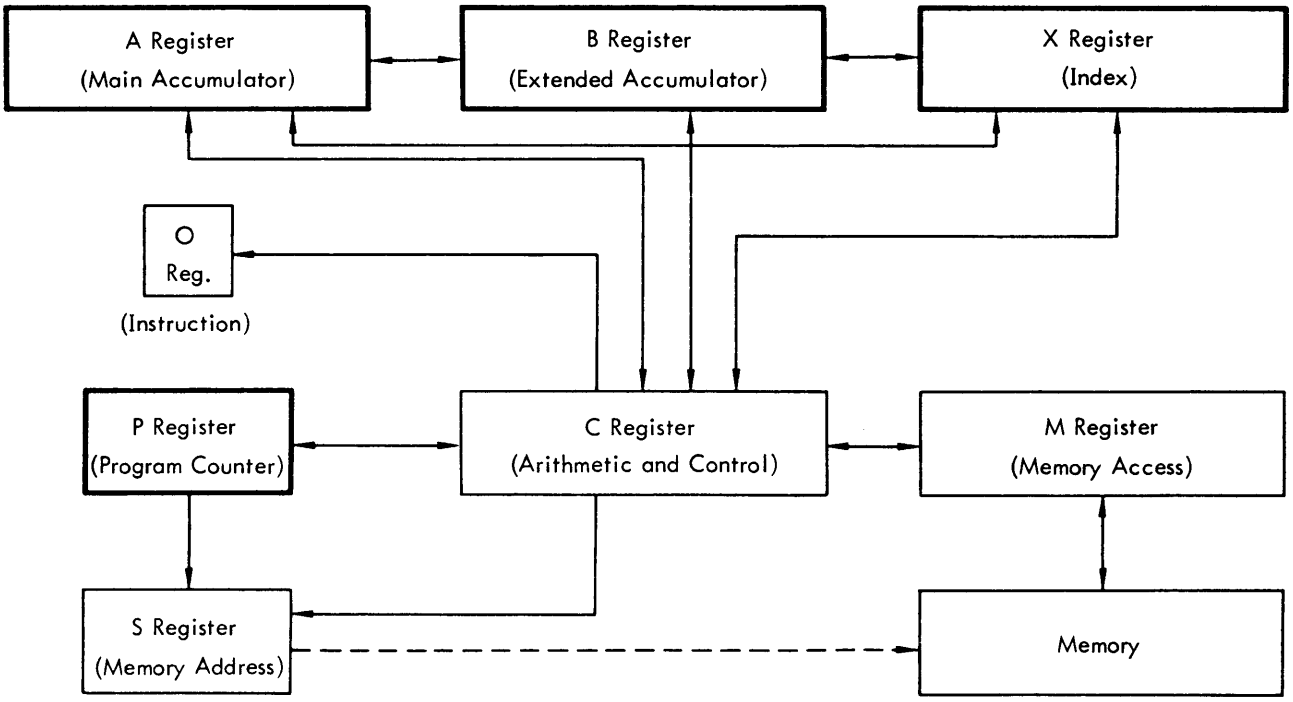


Figure 1-2. Basic Register Flow Diagram

XDS 925 MEMORY

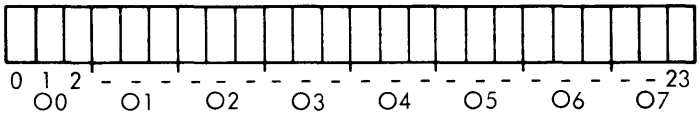
The basic XDS 925 Memory consists of one random access, 4096-word magnetic core bank with a word size of 24 bits plus parity. Optional 8192 or 16,384-word memory banks are also available. The Central Processor and the input/output channels can directly address all memory. Addresses for memory words extend from octal locations 00000 through 07777 (4K Memory), 00000 through 17777 (8K Memory), or 00000 through 37777 (16K Memory). The memory bank in a 16K system is a "wrap-around" or circular memory where the next location after 37777 is 00000. An attempt to read from a location whose address is not available causes zeros to be read. An attempt to store into such a location essentially results in a "no-op" operation, with the next instruction in sequence being executed. Thus, a program can use this property to determine the memory size of the machine on which it is operating.

Before accessing each memory word, the computer checks the power to ensure that the entire read-write cycle can be successfully completed. If it detects a power loss, the computer halts. Special logic (optional) may be included that prevents loss of information due to transient power failure or manual power shut-off.

The computer automatically generates even parity or checks for it during each read/write cycle. Setting a control panel parity switch causes the computer to halt automatically in case of parity error detection.

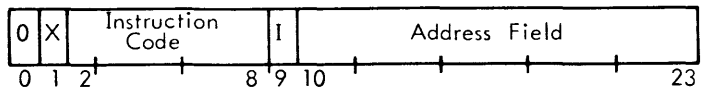
MEMORY WORD FORMATS

A computer word is 24 binary digits (bits) long.



The format above numbers the bits from the left, or most significant end of the word, to the right, or least significant end of the word. This numbering format is the basis of references to bit positions or bit numbers. Octal notation most easily describes the contents of the 24 bits of a word. Thus, one octal digit, 0 through 7, represents three binary digits. For example, the octal number, 01234567, represents its binary equivalent, 000 001 010 011 100 101 110 111.

The computer instruction word format is:



Bit position 0 is not used by the central processor decoding logic.

Bit position 1 contains the Index Register Bit (X).

Bit positions 2 through 8 contain the Instruction Code Field which determines the operation to be performed. The Programmed Operator facility uses bit position 2; it is part of the "Tag" Field (bit positions 0 - 2).

Bit position 9 contains the Indirect Address Bit (I).

Bit positions 10 through 23 contain the Address Field which usually represents the location of the operand called for by the instruction code.

The following examples use standard assembler format in expressing instructions. This format is:

LDA 1000, T

where:

LDA is a representative mnemonic instruction code, 1000 is a representative address that is written decimally, and T is a 1-digit octal integer that represents the Tag Field.

To express indirect addressing (that is, a "one" in the Indirect Address position), the programmer prefixes an asterisk to the Address Field:

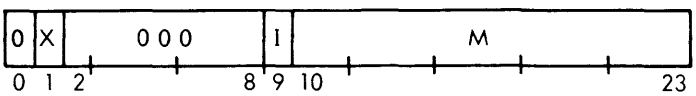
LDA *1000, T

The interpretation of the Tag Field (bit positions 0 - 2) integer, T, is:

Tag Field Integer T	Interpretation
0 (or blank)	No Relative Address, No Index, No Programmed Operator
1	Programmed Operator
2	Index
3	Programmed Operator and Index
4	Relative Address
5	Programmed Operator and Relative Address
6	Both Relative Address and Index
7	Programmed Operator, Index, and Relative Address

Note: Three-letter Programmed Operator mnemonics, not the Tag Field, are generally used to denote Programmed Operators (see Programmed Operator Instructions Appendix for example).

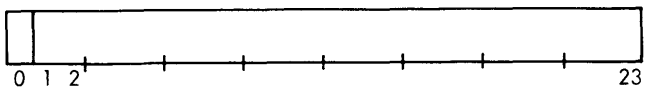
In the explanation of specific instructions, the format used for the instruction word is:



where X and I are as defined previously, 000 represents the instruction, and M represents a generalized memory address.

FIXED-POINT FORMAT

Fixed-point data words have the format:



Numbers held in this format are 8-digit, octal numbers, with the sign incorporated as the "leading bit" in the most significant octal digit. Bit position 0 is the sign bit, with negative numbers having a "1" in bit position 0 and positive numbers having a "0" in bit position 0.

The memory holds fixed-point numbers as 23-bit fractions with an assumed binary point to the left of bit position one. Numbers held in one word have the equivalent precision of over six decimal digits. The range of values of the fixed-point format is from minus one to less than plus one. Scaling is used in handling numbers during computation.

Programmers sometime consider fixed-point numbers to be integers, with the binary point to the right of bit position 23. The range of integer values is from -2^{23} to $+2^{23}-1$.

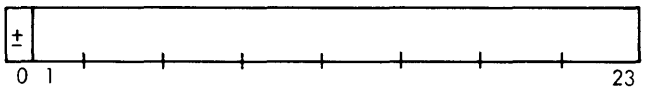
The memory holds negative, fixed-point numbers in two's complement form; the computer operates on these numbers arithmetically in a two's complement number system. See Appendix B for a discussion of two's complement arithmetic.

FLOATING-POINT FORMAT

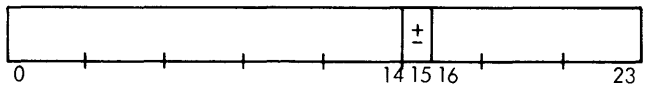
XDS offers standard Programmed Operator subroutines for performing double and single-precision floating-point arithmetic. The following paragraphs explain the standard floating-point number formats.

Double-Precision Floating-Point Format

Most Significant Word



Least Significant Word



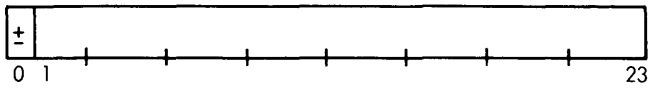
The fractional portion of a double-precision, floating-point number is a 39-bit, proper fraction, with the leading bit being the sign bit and the assumed binary point just to the left of the most significant magnitude bit (bit 1 of the upper word). The floating-point exponent is a 9-bit integer, with the leading bit being the sign. The standard routines operate on both fraction and exponent in two's complement form. If F represents the contents of the fractional field and E represents the contents of the exponent field, the number has the form $F \times 2^{\pm E}$

Double-precision, floating-point numbers have over 11 decimal digits of precision and a decimally equivalent exponent range of 10^{-77} to 10^{+77}.

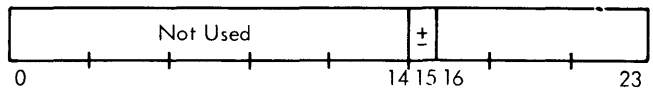
Standard Programmed Operators assume that the more significant word is in the A Register, or stored in memory location M + 1, and that the less significant word is in the B Register, or stored in memory location M.

Single-Precision Floating-Point Format

Fractional Word



Exponent Word



The fractional portion of a single-precision, floating-point number is a 24-bit proper fraction, with the leading bit being the sign and the assumed binary point just to the left of the most significant magnitude bit. The floating-point exponent is a 9-bit integer with a leading sign bit. The standard routines operate on both fraction and exponent in two's complement form.

Single-precision, floating-point numbers have over six decimal digits of precision and a decimally equivalent exponent range of 10^{-77} to 10^{+77}.

Standard Programmed Operators assume that the fractional word is in A, or stored in memory location M + 1, and that the exponent word is in B, or stored in memory location M. When entering a standard Programmed Operator routine, bits 0-14 of the exponent word are ignored.

SPECIAL CHARACTERISTICS

Certain computer features simplify programming and provide significant economies in memory and in program running time.

ADDRESS MODIFICATION

Indexing and indirect addressing, used singly or in combination, perform address modification. In both indexing and indirect addressing, the computer performs address modification after bringing the instruction from memory but before executing it. The instruction remains in memory in its original form. The results of indexing and/or indirect addressing form the "effective address".

INDEXING

The computer contains an Index Register for address modification. The use of this register to modify the address in an instruction does not increase instruction execution time.

If the content of the Index Bit in an instruction is a "one", the computer adds the contents of bits 10 through 23 of the X Register to the contents of the Address Field of the instruction prior to execution. This addition does not keep any overflow or carry beyond the fourteenth address bit.

The instruction set provides instructions for modifying and testing the X Register and for transferring information between the X Register and memory.

INDIRECT ADDRESSING

The Indirect Address Bit is in bit position 9 of the instruction. This bit determines whether the computer uses indirect addressing with the instruction being executed.

A zero in the Indirect Address Bit causes the computer to use the contents of the Address Field (bit positions 10-23 in the instruction) as the 14-bit address requested by the instruction. A one in the Index Bit causes the computer to add the contents of the Index Register to this address to form the effective address.

A one in the Indirect Address Bit causes the computer to decode the contents of the effective address, accessed as described above, as if it were an instruction without an instruction code; that is, the address logic reinitiates address decoding, using the word in the effective location (the memory cell whose address is the effective address). This is an iterative process and provides multi-level indirect addressing. Indirect addressing adds one cycle time to instruction execution time for each level of addressing. The programmer can use indexing to modify indirect addressing at every level.

EXAMPLES: INDEXING AND INDIRECT ADDRESSING

The octal instruction code for LOAD A REGISTER (LDA), used in the examples, is 76.

Location	Contents	Effect
X	00000001	
1000	00001001	
1001	00041002	
1002	00001003	
1003	00000002	
2000	0 76 01000	(1000)=00001001→A
2001	2 76 01000	(1000 + 1)=(1001)=00041002→A
2002	0 76 41000	((1000))=(1001)=00041002→A
2003	2 76 41000	((1000 + 1))=((1001))=(41002) = ((1002))=(1003)=00000002→A

Nomenclature

When discussing properties of the various instructions, including the indirect addressing facility, several terms describe specific locations or addresses.

The term "effective memory location" describes the location in memory from which the final operand is taken at the conclusion of all indirect addressing and indexing. This term is sometimes shortened to "effective location." It is the location whose address is the effective address.

The term "effective operand" means the contents of the effective memory location.

PROGRAMMED OPERATORS

Programmed Operators permit the calling of subroutines with a single instruction of the same form as built-in, machine instructions. The computer decodes the codes 100 - 177 as special instructions and transfers to a subroutine uniquely determined by the code. The computer records the return address at location 00000 so that program continuity is maintained. Through indirect addressing, the subroutine can gain access to the effective address of the calling instruction.

Programmed Operator subroutines are assigned three-letter, mnemonic designations in the same manner as built-in, machine instructions described in Section II. A program can use up to 64 Programmed Operators at any one time; however, since Programmed Operators are programmer-specified, the programmer can select alternate sets or sub-sets of the 64 Programmed Operators from program to program or from section to section of the same program. The total number of Programmed Operators is without limit; but it is inconvenient to use more than 64 in one program. Other computers in the SDS 900 Series maintain symbolic homogeneity through use of Programmed Operators. Mnemonic designations are identical in all computers. For example, while the designation "FLA" (for Floating ADD) may refer to a built-in, machine instruction in one computer, it refers to a Programmed Operator subroutine in another. This technique preserves the one-to-one instruction relationship; programs written for one 900 Series Computer can run on any other computer in the series.

A more detailed discussion and a list of standard SDS Programmed Operator routines are in Appendix E.

OVERFLOW

The Overflow Indicator in the computer permits the detection of erroneous arithmetic operations that occur during the execution of a program. The Overflow Indicator turns on if any of the following occur:

A sum or difference resulting from an addition or subtraction that cannot be contained within the A Register.

A left-shift operation that shifts a bit of absolute magnitude equal to one beyond position 1 of the A Register.

The Multiply Step instruction (page 10) can set overflow.

The instruction set (see Section II) contains instructions to reset, or test and reset the state of the Overflow Indicator.

The only instruction whose execution is altered by the state of the Overflow Indicator is OVERFLOW TEST (OVT), which skips if overflow is reset. Thus, the state of the Overflow Indicator can be ignored if desired. This is unlike some machines in which overflow causes a trap or halt.

To determine whether a particular instruction causes overflow, turn off the Overflow Indicator before executing the instruction. An instruction that may be used to turn on overflow is BRR. The instruction A BRR A, 4, where A is the location of the BRR, "branches" to the next location and turns on the Overflow Indicator.

If the Overflow Indicator is on, it remains on until the appropriate instruction turns it off.

The execution of Programmed Operator, closed, and interrupt subroutines automatically preserves the status of the Overflow Indicator. In the execution of a Programmed Operator instruction, the computer automatically places the status of the Overflow Indicator in bit position 0 of location 00000 and resets the Overflow Indicator. The instruction, MARK PLACE AND BRANCH (BRM), places the status of the Overflow Indicator in bit position 0 of the effective memory location and does not disturb the Overflow Indicator.

The instruction, RETURN BRANCH (BRR), automatically merges the contents of the Overflow Indicator with the contents of bit position 0 of the effective memory location and places the result in the Overflow Indicator. Section II contains a description of the above branch instructions.

SUBROUTINE EXECUTION

The XDS 925 Computer provides three distinct methods of subroutine execution:

Normal closed subroutine where the input parameters are specified in appropriate registers such as the A Register

Interrupt subroutine that is entered as the result of an interrupt

Programmed Operator subroutine

A program enters a normal closed subroutine via a MARK PLACE AND BRANCH (BRM) instruction; BRM automatically stores the contents of the Program Counter (P) Register and the status of the Overflow Indicator in the branch-to location. The P Register value is the address of the BRM instruction. A RETURN BRANCH (BRR) instruction accomplishes the return to the main program; the BRR adds one to the stored P Register value and transfers control to that location. See Section II, Branch Group, for a description of the branch instructions.

Interrupt subroutines are closed subroutines that are initiated by the detection of program-controlling interrupts that automatically cause the specific interrupt subroutine to be entered. A BRM instruction enters an interrupt subroutine; the BRM automatically stores the contents of the P Register and the Overflow Indicator in the branch-to location. The value stored from the P Register is the address of the instruction to which program control should return after the interrupt is serviced by the interrupt subroutine. A BRANCH UNCONDITIONALLY (BRU) instruction with indirect addressing returns control to the main program at the completion of the subroutine. BRU indirect also clears the interrupt from the active state. Note that this differs from the normal closed subroutine return that uses the BRR (stored P value + 1→P). The point in an execution cycle at which the interrupt routine-entering BRM executes has already caused the proper incrementing to occur.

II. MACHINE INSTRUCTIONS

INTRODUCTION

This section describes XDS 925 instructions; the instructions are in functional groups. Lists of instructions in functional, numerical, and alphabetical order are in Appendix H.

The following statements apply to the instruction descriptions:

All instruction times are in memory cycles, where each cycle is 1.75 microseconds, and include accessing the instruction and all required operands.

Parentheses denote "contents of." For example, "(A)" denotes "contents of the A Register."

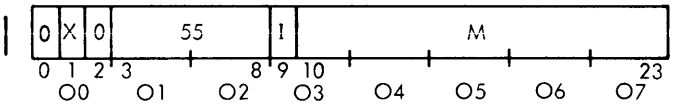
Indexing and Indirect Addressing apply to all instructions except as noted. Indexing does not change the instruction execution time. Each level of indirect addressing requires one additional memory cycle.

The interrupt system can interrupt the program sequence at the end of any instruction except as noted.

Each instruction description specifies the registers affected.

With the description of each instruction is a diagram representing the format of the instruction. Preceding this diagram is the name of the instruction and mnemonic code that identify the instruction.

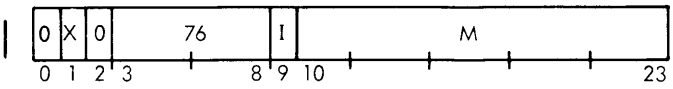
EXAMPLE:



The letter M represents the address part of the instruction. Some instructions have octal numbers in the address field. These instructions do not refer to memory.

LOAD/STORE INSTRUCTIONS

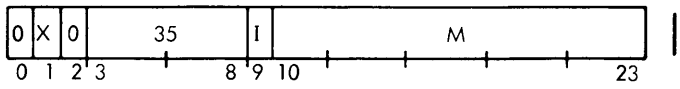
LDA LOAD A



LDA loads the contents of the effective memory location into the A Register.

Registers Affected: A Timing: 2

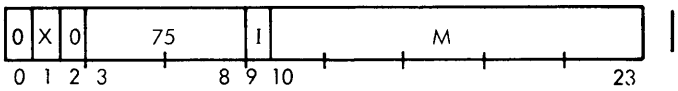
STA STORE A



STA stores the contents of the A Register in the effective memory location.

Registers Affected: M Timing: 3

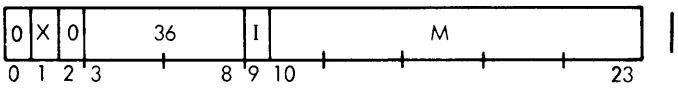
LDB LOAD B



LDB loads the contents of the effective memory location into the B Register.

Registers Affected: B Timing: 2

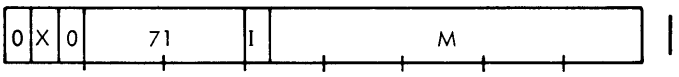
STB STORE B



STB stores the contents of the B Register in the effective memory location.

Registers Affected: M Timing: 3

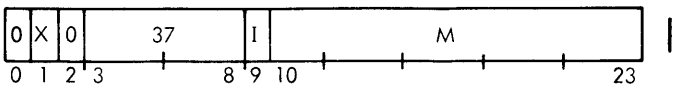
LDX LOAD INDEX



LDX loads the entire 24-bit contents of the effective memory location into the Index Register.

Registers Affected: X Timing: 2

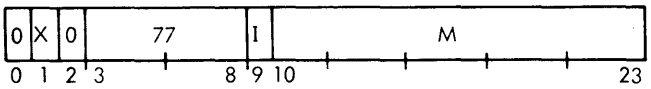
STX STORE INDEX



STX stores the entire 24-bit contents of the Index Register in the effective memory location.

Registers Affected: M Timing: 3

EAX COPY EFFECTIVE ADDRESS INTO INDEX REGISTER



EAX copies the address of the effective memory location into the Index Register.

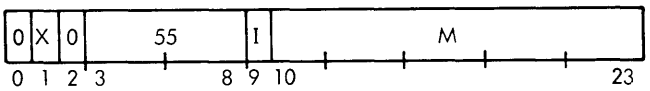
The addressing process for this instruction operates as in a Load A instruction, except that instead of obtaining the contents of the effective memory location, the effective memory address acts as the operand. This addressing process is sometimes called "immediate addressing". For example, if execution of this instruction occurs with a zero indirect address bit and a zero in the index field, then the actual bit configuration in the address field of EAX copies into the Index Register.

This instruction does not affect the ten most significant bits of the Index Register.

Registers Affected: X_{10-23} Timing: 2

ARITHMETIC INSTRUCTIONS

ADD ADD MEMORY TO A

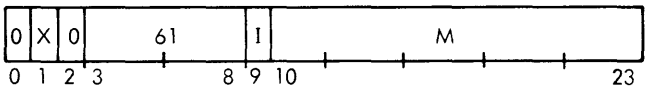


This instruction adds the contents of the effective memory location to the A Register and places the result in A.

If both numbers are of the same sign but the sign of the result is opposite, overflow has occurred and the computer has set the Overflow Indicator.

Registers Affected: A, Overflow Indicator Timing: 2

MIN MEMORY INCREMENT

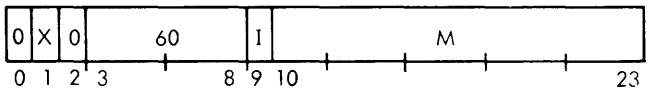


MIN increases the contents of the effective memory location by one, and places the resulting sum in the same location. The contents of the A Register do not change.

Overflow occurs only when the contents of M are 37777777 before execution. In this case, 40000000 is the result in M.

Registers Affected: M, Overflow Indicator Timing: 3

MDE MEMORY DECREMENT

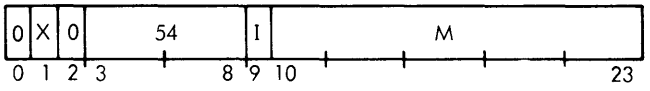


MDE decreases the contents of the effective memory location by one and places the resulting difference in the same location. The contents of the A Register do not change.

An overflow occurs if the initial contents of memory are 40000000. The result in memory in this case is 37777777.

Registers Affected: M, Overflow Indicator Timing: 3

SUB SUBTRACT MEMORY FROM A

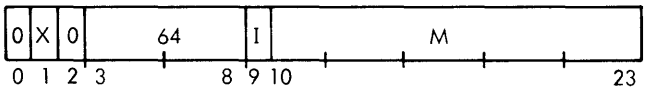


SUB subtracts the contents of the effective memory location from the A Register and places the result in the A Register.

If both numbers are of the same sign after the subtrahend has been complemented for addition but the sign of the result is opposite, an overflow has occurred and the computer sets the Overflow Indicator.

Registers Affected: A, Overflow Indicator Timing: 2

MUS MULTIPLY STEP



The sign of A temporarily extends two bit positions to the left if the Overflow Indicator is reset. If the Overflow Indicator is set, the two bits extended are zeros. Then the contents of the memory location determined by the effective address add to or subtract from the A Register, based on the contents of the three lowest order bits of the B Register. The arithmetic takes place according to the following table:

B_{21}	B_{22}	B_{23}	Arithmetic
0	0	0	None
0	0	1	(A) + 2(M) → A
0	1	0	(A) + 2(M) → A
0	1	1	(A) + 4(M) → A
1	0	0	(A) - 4(M) → A
1	0	1	(A) - 2(M) → A
1	1	0	(A) - 2(M) → A
1	1	1	None

The computer then shifts the result in the A and B Registers two bit positions to the right.

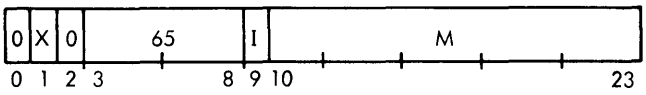
The Overflow Indicator is set if (M) is -1, the contents of B_{21} -B_{23} were 100, and (A)/2 was originally zero. Otherwise, the overflow is reset.

The MULTIPLY subroutine uses this instruction. Twelve of these instructions can be repeated to provide a complete multiplication of the form (M) x (B)→A, B. Prior to execution of the first step, the multiplier must be in the B Register, the A Register cleared, the A and B Registers shifted left one, and the Overflow Indicator turned off.

NOTE: The MULTIPLY subroutine requires 54.25 μsec for a full multiplication.

Registers Affected: A, B, Overflow Indicator Timing: 2

DIS DIVIDE STEP



DIS shifts the contents of the A and B Registers left one bit position and copies the complement of A_0 into B_{23}. If $(A_0) = (M_0)$, the contents of the memory location determined by the effective address subtract from the A Register. If $(A_0) \neq (M_0)$, the contents of the memory location determined by the effective address add to the A Register.

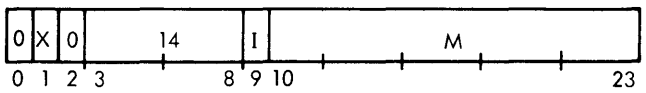
The DIVIDE subroutine uses this instruction.

NOTE: The DIVIDE subroutine requires 194.25 μsec for a full division. The subroutine provides a corrected remainder of the same sign as the original A Register.

Registers Affected: A, B Timing: 2

LOGICAL INSTRUCTIONS

ETR EXTRACT



ETR performs a logical "AND" between corresponding bits of the A Register and the effective memory location and places the result in A. This instruction performs the operation bit by corresponding bit according to the following:

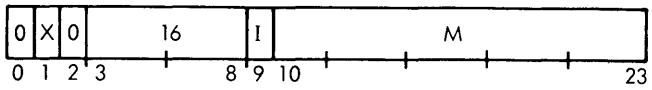
A	M	Result in A
0	0	0
0	1	0
1	0	0
1	1	1

Registers Affected: A Timing: 2

EXAMPLE:

	A	M
Before Execution	64231567	00777600
After Execution	00231400	00777600

MRG MERGE



MRG performs a logical "Inclusive OR" between corresponding bits of the A Register and the effective memory location and places the result in A. This instruction performs the operation bit by corresponding bit, as follows:

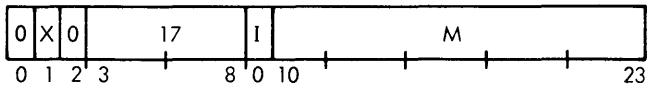
A	M	Result in A
0	0	0
0	1	1
1	0	1
1	1	1

Registers Affected: A Timing: 2

EXAMPLE:

	A	M
Before Execution	06446254	02340712
After Execution	06746756	02340712

EOR EXCLUSIVE OR



EOR performs a logical "Exclusive OR" between corresponding bits of the A Register and the effective memory location and places the result in A. This instruction performs the operation bit by corresponding bit, as follows:

A	M	Result in A
0	0	0
0	1	1
1	0	1
1	1	0

Registers Affected: A Timing: 2

EXAMPLE:

	A	M
Before Execution	34165031	70077021
After Execution	44112010	70077021

The proper memory word configuration logically inverts selected bit positions of the A Register. If all "ones" appear in the memory word, a one's complement of A results.

EXAMPLE:

	A	M
Before Execution	10357211	77777777
After Execution	67420566	77777777

REGISTER CHANGE INSTRUCTIONS

These instructions operate on and exchange data between the A and B Registers. All instructions in the group use the same operation code, 46.

EXAMPLE:

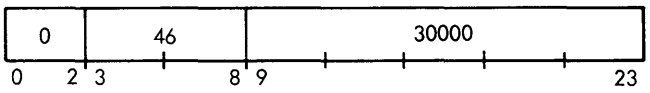
The following instruction copies (A) into B and clears the A Register:

0 46 00005

Both functions occur simultaneously, that is, within the one cycle time of the instruction.

Indirect addressing and indexing do not apply to these instructions.

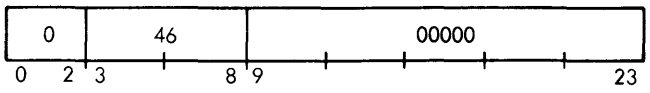
CLR CLEAR AB



CLR clears the contents of both the A and B Registers to zero.

Registers Affected: A, B Timing: 1

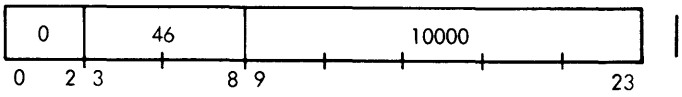
XAB EXCHANGE A AND B



XAB copies the contents of the A Register into the B Register and, simultaneously, copies the contents of the B Register into the A Register.

Registers Affected: A, B Timing: 1

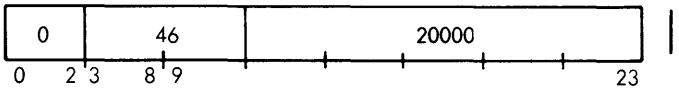
BAC COPY B INTO A, CLEAR B



BAC copies the contents of the B Register into the A Register and simultaneously, clears the B Register to zero.

Registers Affected: A, B Timing: 1

ABC COPY A INTO B, CLEAR A



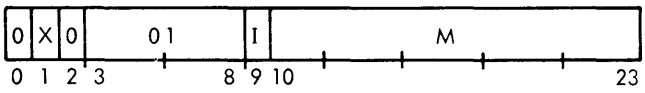
ABC copies the contents of the A Register into the B Register and, simultaneously, clears the A Register to zero.

Registers Affected: A, B Timing: 1

BRANCH INSTRUCTIONS

Branch instructions conditionally or unconditionally change the course of the program by altering the contents of the program counter. The programmer should note that these instructions branch to locations determined by the effective address; this means that the branch can operate with all levels of indirect and indexed addressing.

BRU BRANCH UNCONDITIONALLY

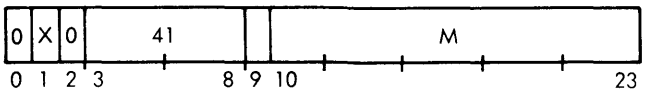


BRU takes the next instruction from the location determined by the effective address.

A BRU instruction with an Indirect Address bit equal to "one" clears the highest priority interrupt level then active in addition to branching to the effective location.

Registers Affected: P Timing: 1

BRX INCREMENT INDEX AND BRANCH



BRX increments the contents of the entire Index Register by one. If the resultant Index Register value contains a "1" in bit position 9 of the index, the computer transfers control to the effective location. If not, it takes the next instruction in sequence.

If a BRX instruction is indexed, any transfer of control is to the effective address determined by the value of the index immediately prior to the execution of BRX. The test for transfer is on the incremented value of the Index Register, just as if the BRX instruction were not indexed.

The most significant bits of the Index Register (bits 0-8) have no effect on the execution of the instruction, but may be affected by it.

If a branch occurs, an interrupt cannot occur following the execution of this instruction.

EXAMPLE:

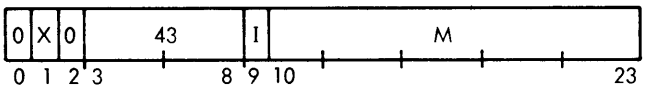
Location	Instruction	X Register
00777	STA 01500	77777776
01000	BRX 01006	77777777
01001	LDA 02000	
01006	BRX 01001	00000000
01007	LDA 02100	00000000

The execution of these instructions is in the following order, as given by their locations:

00777
01000
01006
01007

Registers Affected: X, P Timing: 1, if branch
2, if no branch

BRM MARK PLACE AND BRANCH



BRM stores the contents of the P Register (the address of the BRM instruction itself) in the effective memory location and transfers control to the effective memory location plus one. BRM also stores the status of the Overflow Indicator in bit 0 of the effective location. The contents of bits 1-9 of the effective location are zeros.

EXAMPLE:

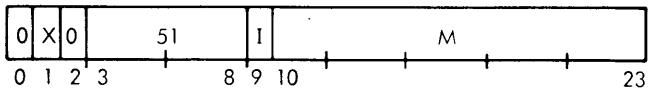
Location	Instruction
01517	BRM 0522

	Overflow Indicator	Location 0522	P Register
Before Execution	1 (on)	---	01517
After Execution	1 (on)	40001517	00523

Note: Use this instruction to enter subroutines where a return to the main program is desired after the subroutine has been completed. Use RETURN BRANCH (51) to return to the main program.

Registers Affected: M, P Timing: 2

BRR RETURN BRANCH



BRR copies the contents of the effective memory location into an internal register and increments the contents by one. The instruction then stores the least significant 14 bits in the P Register. It also performs a logical OR between bit 0 and the Overflow Indicator; and places the result in the Overflow Indicator. There is no change in the contents of the effective memory location.

EXAMPLE:

Location	Contents
02100	BRR 02000
02000	00003220

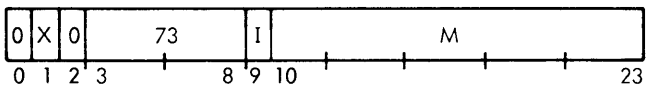
If the computer executes the instruction in location 02100, it takes the next instruction from location 03221. Location 02000 still contains 00003220.

Note: Use BRR to return to the main program after completion of a subroutine in conjunction with MARK PLACE AND BRANCH (43) except in interrupt subroutines (see Section III).

Registers Affected: P, Overflow Indicator Timing: 2

TEST AND SKIP INSTRUCTIONS

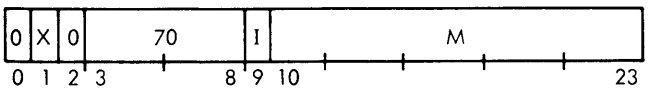
SKG SKIP IF A GREATER THAN MEMORY



SKG algebraically compares the contents of the A Register with the contents of the effective memory location. If the contents of A are greater than the contents of the effective location, the computer skips the next instruction in sequence and executes the following instruction. If the contents of A are less than or equal to the contents of the effective location, the computer executes the next instruction in sequence. SKG alters neither A nor memory.

Registers Affected: P Timing: 2, if no skip
3, if skip

SKM SKIP IF A EQUALS MEMORY ON B MASK



SKM compares selected bits of the contents of the A Register with the corresponding bits in the contents of the effective memory location. If the selected bits in A are identical to the selected bits in the contents of the effective memory location, the computer skips the next instruction in sequence and executes the following instruction. If the selected bits in the contents

of the A Register are not identical to the contents of the effective location, the computer executes the next instruction in sequence.

The programmer selects the bits in A to be compared by placing ones in the corresponding bit positions of the B Register and zeros in the remaining bit positions of B.

SKM considers the contents of A, B, and the effective location to be unsigned, 24-bit, non-numeric quantities, and does not alter them.

EXAMPLE:

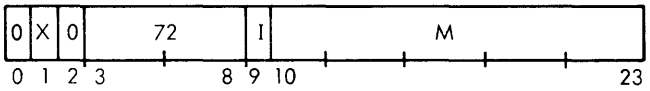
A	B	Memory
00043007	00177000	57643240

Since SKM compares bit positions 8-14 only (as determined by B), and (A) = (M) in these positions, a skip occurs. Note that if (B) = 0, a skip occurs regardless of (A) and (M).

Registers Affected: P

Timing: 2, if no skip
3, if skip

SKA SKIP IF A AND MEMORY DO NOT COMPARE ONES



SKA compares the contents of the A Register, bit by bit, with the contents of the effective memory location. If the contents of the A Register and the contents of the effective location do not have any ones in corresponding bit positions, the computer skips the next instruction in sequence and executes the following instruction. If the contents of the A Register and the contents of the effective location do have at least one pair of 1-bits in corresponding bit positions, the computer executes the next instruction in sequence.

The instruction logically ANDS corresponding bits in A and Memory, based on the following table:

A	Memory	Result
0	0	0
0	1	0
1	0	0
1	1	1

If the result produces a "1" in any bit position, a skip does not occur.

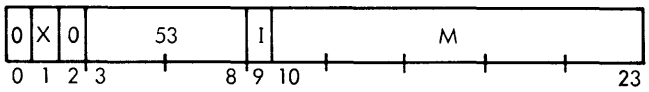
Note: Different configurations of the memory word result in a wide variety of conditional instructions for use by the programmer. Some representative configurations are:

Memory Configuration	Instruction
40000000	Skip if A is Positive
77777777	Skip if A = 0
00000001	Skip if A is Even
Contents of A Register	
40000000	Skip if Memory is Positive
77777777	Skip if Memory = 0
00000001	Skip if Memory is Even

Registers Affected: None

Timing: 2, if no skip
3, if skip

SKN SKIP IF MEMORY NEGATIVE



If the contents of the effective memory location are negative, i.e., if $(M_0) = 1$, the computer skips the next instruction in sequence and executes the following instruction. If the contents of the effective location are positive or zero, the computer executes the next instruction in sequence.

Registers Affected: None

Timing: 2, if no skip
3, if skip

SHIFT INSTRUCTIONS

The shift instructions operate on the contents of the A and B Registers and offer a complete facility for right and left shifting, cycling, and normalizing the contents of these two registers. The A and B Registers, in combination, form a double-length register whose double-length contents can be shifted, cycled, or normalized. This double-length register is named "AB."

When the contents of the AB Register shift right, bits from bit position 23 of the A Register shift into bit position 0 of the B Register. When the AB Register shifts left, bits from bit position 0 of the B Register shift into bit position 23 of the A Register.

The 48-bit contents of the AB Register may be cycled using the shift instructions. When the contents of the AB Register cycle, the bits that shift from one end of the one register copy into the other end of the other register.

These instructions use the instruction code to determine the direction of shift (66 = right; 67 = left); bits 10-11 of the instruction address determine the method of shifting as follows:

Octal Position	Bits 10, 11	Octal Value	Function
	00	0	A, B Shift
O3	10	2	A, B Cycle
	01	1	Normalize (Left only)

Indirect addressing is permissible with these instructions, bits 10 and 11 of the effective address determining the method of shifting.

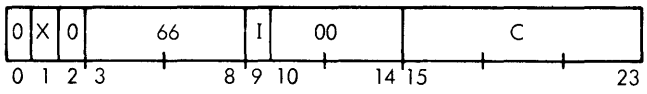
Indexing during direct address occurs only on the least significant nine bits of the address. It is thus possible to index the number of shifts without affecting the method of shifting. During indirect addressing, the full 14 bits are used.

When the computer interprets a shift instruction, bit positions 15-23 of the effective address of the instruction determine the amount of the shift. The computer treats these nine bits as an unsigned count. If the initial count is equal to zero, no shifting occurs. If the initial count is greater than 48, it is set equal to 48 before shifting begins. Once the shift begins, the count reduces by one for each position shifted until it reaches zero. The count C in the following instructions indicates the number of places to be shifted.

Shift timing is calculated as follows where N is the number of places shifted.

Timing in Cycles	Number of Places Shifted
$2 + \frac{N}{3}$	N = 0, 3, 6, 9, 12, ...
$3 + \frac{N - 1}{3}$	N = 1, 4, 7, 10, 13, ...
$4 + \frac{N - 2}{3}$	N = 2, 5, 8, 11, 14, ...

RSH RIGHT SHIFT AB



RSH shifts the contents of the AB Register (that is, A and B Registers) right the number of places specified in bits 15 through 23 of the effective address. The bit in the sign position of A does not shift, but its value copies into the vacated bit positions of the shifted number. The bit in the sign position of B shifts. Bits shifted out of A_{23} shift into B_0. Bits shifting past position B_{23} are lost.

Registers Affected: A, B Timing: 2-19

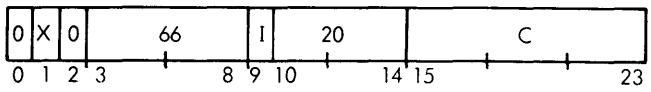
EXAMPLE:

The instruction is: RSH 00022

	A	B
Before execution	45261237	27651260
After execution	77777745	26123727

Note: This instruction may perform scaling of floating-point numbers by use of indexing, where the difference of exponents is in the Index Register as a positive quantity.

RCY RIGHT CYCLE AB



RCY shifts the contents of the AB Register right the number of places specified in bits 15 through 23 of the effective address. The bit in the sign position of B shifts like any other bit in B. Bits shifting out of A_{23} shift into B_0. Bits from bit position 23 of B go into bit position 0 of A. The computer treats the double-length register as if it were circular and cycles it onto itself; it loses no bits.

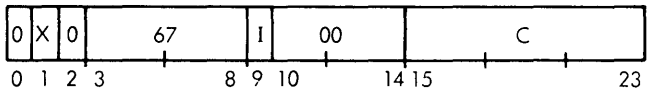
Registers Affected: A, B Timing: 2-19

EXAMPLE:

The instruction is: RCY 00017

	A	B
Before Execution	61245703	41637701
After Execution	37701612	45703416

LSH LEFT SHIFT AB



LSH shifts the contents of the AB Register left the number of places specified in bits 15 through 23 of the effective address. Bits shift left through the sign position of A, but when a bit, different in value from the original sign, shifts into the sign position, the computer sets the Overflow Indicator. Bits shifting out of B_0 go into A_{23}. Bits shifting past position 0 in A are lost. Zeros fill the vacated bit positions on the right end of the B Register.

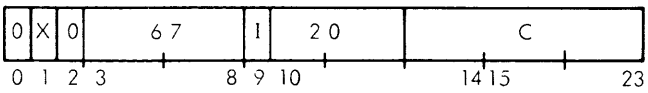
Registers Affected: A, B, Overflow Indicator Timing: 2-19

EXAMPLE:

The instruction is: LSH 00022

	A	B
Before Execution	46712370	64132711
After Execution	70641327	11000000

LCY LEFT CYCLE AB



LCY shifts the contents of the AB Register left the number of places specified in bits 15 through 23 of the effective address. The bits in the sign positions of A and B shift like any other bits in the number. Bits shifting out of B_0 shift into A_{23}.

The instruction copies bits that shift from bit position 0 of A into bit position 23 of B. The computer treats the double-length register as if it were circular and cycles it onto itself. It loses no bits.

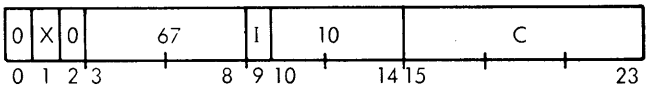
Registers Affected: A, B Timing: 2-19

EXAMPLE:

The instruction is: LCY 00011

	A	B
Before Execution	71432560	34156723
After Execution	32560341	56723714

NOD NORMALIZE AND DECREMENT X



NOD shifts the contents of the AB Register left until 1) a bit appears in position 1 of A that is not equal to the bit in the sign position of A, or 2) until C shifts occur. The computer keeps count of the number of places shifted by decrementing the contents of the X register each time a shift occurs. If, in the attempt to normalize, shifting exceeds 48 places, the contents of the AB Register were initially zero. In this case, the computer continues shifting until the shift count C reduces to zero. Zeros fill the vacated positions of AB.

The number, C, placed in address bit positions 18 through 23, is an upper limit for the number of left shifts that will occur. The programmer must ensure that C is sufficiently large to permit a complete normalization.

EXAMPLE:

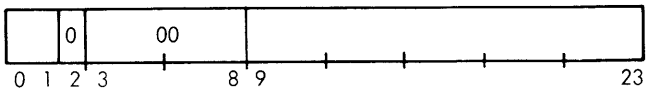
NOD

	A	B	X
Before Execution	00004632	76124035	00000000
After Execution	23153705	20164000	77777765

Registers Affected: A, B, X Timing: 2+R where R is the resultant number of shifts

CONTROL INSTRUCTIONS

HLT HALT



When the computer executes this instruction, it halts computation and lights the HALT indicator in the console. Before halting, the computer increments the P Register and brings the next instruction to the C Register to be displayed. To resume computation, the operator must first set the RUN-IDLE-STEP switch to IDLE, then back to RUN.

The computer then executes the next instruction, according to the P Register.

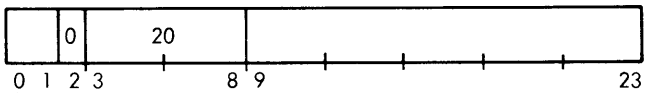
Indirect addressing and indexing do not apply to this instruction.

When the computer executes HLT, all internal computation ceases at the end of the present instruction being executed. If an input/output operation is in progress, it continues until completed. Computation automatically resumes with the occurrence of a program interrupt, if the RUN-IDLE-STEP switch is still in the RUN position and the interrupt system is enabled.

The HALT light turns off when the RUN-IDLE-STEP switch is set to IDLE, or when an interrupt occurs.

Registers Affected: None Timing: 1

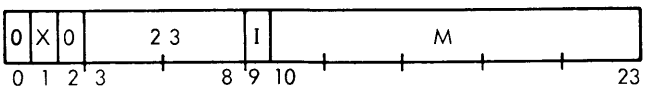
NOP NO OPERATION



Executing NOP does not affect the A Register, B Register, X Register, or memory. Indirect addressing and indexing do not apply to this instruction.

Registers Affected: None Timing: 1

EXU EXECUTE



EXU causes the contents of the effective memory location to be executed as an instruction without altering the contents of the Program Counter. If the effective location is not a Branch, Skip, or another Execute instruction, the computer executes the next instruction in sequence following the Execute instruction, after it executes the contents of the effective location.

If the contents of the effective memory location are a Branch instruction, program control goes to the effective address of the branch and not to the next instruction in sequence following the Execute instruction.

If the contents of the effective memory location are a skip instruction, then, depending on the skip decision, program control returns to the next instruction, or the next instruction plus one, following the Execute instruction.

If the contents of the effective memory location are another Execute instruction, the above process continues identically, with the normal return being the initial Execution location plus one. This process can cascade indefinitely.

Registers Affected: None Timing: 1

BREAKPOINT TESTS

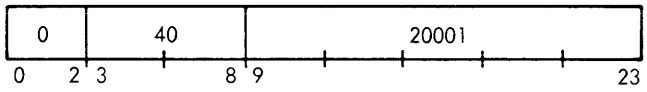
This instruction tests the status of the Breakpoint switches singly or in any combination. If any one of the Breakpoint switches tested is reset, the computer skips the next location in sequence and executes the following instruction. If none of the Breakpoint switches tested is reset, the computer executes the next instruction in sequence.

Mnemonic	Name of Instruction	Octal Configuration
BPT 1	Breakpoint No. 1 Test	0 40 20400
BPT 2	Breakpoint No. 2 Test	0 40 20200
BPT 3	Breakpoint No. 3 Test	0 40 20100
BPT 4	Breakpoint No. 4 Test	0 40 20040

Registers Affected: None Timing: 1, if no skip
2, if skip

OVERFLOW INSTRUCTIONS

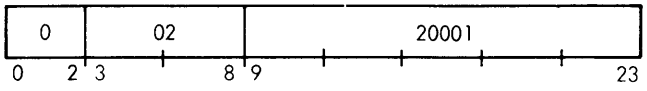
OVT OVERFLOW INDICATOR TEST AND RESET



This instruction tests the status of the Overflow Indicator, skips or not accordingly, and turns the indicator off. If the indicator is off, the computer skips the next instruction in sequence and executes the following instruction. If the indicator is on, the computer executes the next instruction in sequence.

Registers Affected: P, Overflow Indicator Timing: 1, if no skip
2, if skip

ROV RESET OVERFLOW



ROV unconditionally resets the Overflow Indicator (clears to zero).

Registers Affected: Overflow Indicator Timing: 1

FLOATING-POINT OPERATIONS

XDS Programmed Operator subroutines perform in either single or double-precision. Double-precision operation permits accuracy of approximately 11 decimal digits. Single-precision operation permits faster execution times with approximately seven decimal digits of accuracy.

The standard XDS Programmed Operators assume that the most significant word is in A, or stored in location M + 1, while the less significant word is in B, or memory location M. (See Section I, General Description, Floating-Point Format.)

DOUBLE-PRECISION FLOATING-POINT OPERATIONS

The Programmed Operators perform double-precision, floating-point operations using a fractional number of 39 bits (38 bits plus sign) and an exponent of nine bits (eight bits plus sign). Numbers have the fraction equal to 11 decimal digits plus sign and the multiplier as high as $10^{\pm 77}$.

The Programmed Operator subroutines that perform double-precision, floating-point operations are:

Mnemonic	Name	Function	Approx. Execution Time
FLA	Floating Add	Floating (A, B) + (M+1, M) → A, B	196 μsec
FLS	Floating Subtract	Floating (A, B) - (M+1, M) → A, B	222.25 μsec
FLM	Floating Multiply	Floating (A, B) x (M+1, M) → A, B	371 μsec
FLD	Floating Divide	Floating (A, B) ÷ (M+1, M) → A, B	374.5 μsec

SINGLE-PRECISION, FLOATING-POINT OPERATIONS

The Programmed Operators perform single-precision, floating-point operations using a fractional number of 24 bits (23 bits plus sign) and an exponent of nine bits (eight bits plus sign). Numbers have the fraction equal to six decimal digits plus sign and the exponent as high as $10^{\pm 77}$.

The Programmed Operator subroutines that perform single-precision floating-point operations are:

Mnemonic	Name	Function	Approx. Execution Time
FSA	Floating Add, Single-Precision	Floating (A) + (M+1) → A Exponent in B, M	94.5 μsec
FSS	Floating Subtract, Single-Precision	Floating (A) - (M+1) → A Exponent in B, M	104 μsec
FSM	Floating Multiply, Single-Precision	Floating (A) x (M+1) → A Exponent in B, M	101.5 μsec
FSD	Floating Divide, Single-Precision	Floating (A) ÷ (M+1) → A Exponent in B, M	174 μsec

III. INTERRUPT SYSTEM

PRIORITY INTERRUPT SYSTEM

XDS 900 Series Computers contain a priority interrupt system that provides added program control of input/output operations, aids in programming simultaneous input/output and compute operations, and allows immediate recognition of special external conditions.

Interrupts, as specified by the program, can signal when a single word or a block of words has been transmitted. When received, the internal logic examines the interrupt signal and causes the computer to interrupt the program sequence at the end of the execution cycle of the current instruction. Without disturbing the Program Counter Register, the computer transfers program control to one of a selected set of memory locations. A MARK PLACE AND BRANCH (BRM) instruction in this location saves the contents of the program counter, EM3, EM2, and overflow indicator and transfers to the particular interrupt servicing routine required. Entrance to the proper service routine occurs since each interrupt has a unique interrupt location. To exit from the routine, a BRANCH UNCONDITIONALLY (BRU) instruction using indirect addressing returns control to the next instruction in proper sequence in the main program; it also clears the interrupt. Note that when an interrupt occurs causing the execution of the BRM in the interrupt level, the address stored in the mark location is the location plus one of the instruction that was interrupted. In other words, the computer increments the program counter prior to inspecting its registers for an interrupt condition.

The priority interrupt system has up to 896 System interrupt levels. The levels are numbered upward from 200 and have priority according to number; the higher priority levels have a smaller number. See Table 3-1, Interrupt Locations, for the specific assignment.

The two standard as well as the additional interrupts obtained with XDS optional hardware are located at interrupt levels numbered from 30. In general, these have priority according to number like the System interrupts. Note that interrupts 30-77 have priority over any System interrupt (200 or more). The Power Fail-Safe option interrupts (in locations 36 and 37) are "out-of-order" interrupts; they have the highest priority of all.

When an interrupt has occurred and its service subroutine has been entered, an interrupt of higher priority can interrupt the subroutine and gain program control for the servicing of its more important operation. But an interrupt of lower priority cannot interrupt an interrupt-processing subroutine of a higher level. Thus, the priority interrupt system allows interrupts to be arranged according to their importance and/or according to their need for speedy servicing.

The above type of interrupt is called a normal priority interrupt to differentiate from another interrupt feature, the single-instruction interrupt. This different kind of interrupt causes the execution of only one instruction before automatically clearing itself and returning to the program that it interrupted. For example, if an external clock source is connected to the computer so that it pulses an interrupt line at set intervals, the program can maintain a programmed real-time clock. Each time the external pulse causes an interrupt, the program executes the single instruction, MEMORY INCREMENT (MIN), to add one to the memory word selected for use as the programmed real-time clock. The main program can examine this memory location whenever necessary to determine how many time increments have elapsed since the clock was started.

If the single instruction that is executed is a branch instruction, and the branch occurs, the interrupt is cleared but there is no return to the program that was interrupted. This type of interrupt needs no branch instruction to clear it.

Since the single-instruction interrupt performs just one instruction and clears itself, it can be sandwiched into a priority system without disturbance. Any of the optional System interrupts (200-1777) can be single- or normal-instruction interrupts in any combination desired.

PRIORITY INTERRUPT OPERATIONS

A normal priority interrupt level has three operational states: Inactive, Waiting, and Active.

In the inactive state, no interrupt signal has been received into the level and none is currently being processed by its interrupt servicing subroutine.

In the waiting state, an interrupt has been received into the level, but is not being processed. This situation may be due to an interrupt of higher priority being processed at this time. When all higher waiting interrupts have been processed, this level goes to the active state.

In the active state, the interrupt has caused the main program to recognize its presence and has transferred to its assigned interrupt location where it is being processed. When the interrupt processing is completed, a BRANCH UNCONDITIONALLY (BRU) instruction with indirect addressing exits from the service subroutine by transferring control to the proper return location. This branch instruction also sets the interrupt level to the inactive state.

A single-instruction interrupt operates in the same way as the normal priority interrupt in the inactive and waiting states. However, when acknowledged, this interrupt enters the active state, and remains there during the execution of one instruction. At the completion of the one instruction, the single-instruction interrupt returns to the inactive state without the aid of a branch instruction. The single instruction must have a two-cycle or greater execution time.

INTERRUPT CONTROL

Two program control features are available in the interrupt system. These features are Arm/Disarm and Enable/Disable. Arm/Disarm controls whether an interrupt can proceed from the inactive state to the waiting state. When armed, an interrupt signal sets the interrupt to the waiting state. The disarmed condition causes that level to retain no record of an interrupt signal entering the level.

Enable/Disable operates on the entire interrupt system. When the interrupt system is enabled, the System interrupts (200-1777) are enabled; when the interrupt system is disabled, the system interrupts are disabled. Enable/Disable operates differently for the interrupts obtained with XDS options (30-77). Enable/Disable has no effect on the Power Fail-Safe interrupts; they are always enabled and armed. See the last two subsections of this section for a description of how the channel interrupts are affected.

The control of the optional Arm/Disarm feature operates on individual System interrupt levels, that is, any chosen interrupt level may be selectively armed or disarmed. But the instruction structure for Arm/Disarm allows operation on these interrupts in groups of sixteen.

NON-INTERRUPTABLE INSTRUCTIONS

Three instructions prohibit interrupts following their execution. If a branch occurs, an interrupt cannot occur between the execution of INCREMENT INDEX AND BRANCH (BRX) and the instruction to which BRX branches. An interrupt cannot occur between the execution of ENERGIZE OUTPUT M (EOM) and the instruction following it or between the execution of ENERGIZE OUTPUT TO DIRECT ACCESS CHANNEL (EOD) and the instruction following it.

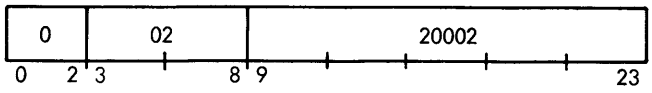
ENABLE/DISABLE INTERRUPT INSTRUCTIONS

Three instructions are available for setting, resetting, and testing the state of the INTERRUPT ENABLED indicator.

Table 3-1. Interrupt Location Assignments

30	Channel Y	Count Equals Zero (End-of-Word)
31	Channel W	Count Equals Zero (End-of-Word)
32	Channel Y	End-of-Record (End-of-Transmission)
33	Channel W	End-of-Record (End-of-Transmission)
36	Power ON	Power Fail-safe interrupt: Power Return
37	Power OFF	Power Fail-safe interrupt: Power below safe limit
60	Channel C	Count Equals Zero (End-of-Word)
61	Channel C	End-of-Record (End-of-Transmission)
62	Channel D	Count Equals Zero (End-of-Word)
63	Channel D	End-of-Record (End-of-Transmission)
64	Channel E	Count Equals Zero
65	Channel E	End-of-Record
66	Channel F	Count Equals Zero
67	Channel F	End-of-Record
70	Channel G	Count Equals Zero
71	Channel G	End-of-Record
72	Channel H	Count Equals Zero
73	Channel H	End-of-Record
74	Clock Sync.	Locations 74, 75 are for the real-time clock
75	Clock Pulse	
200 - 217	Group 0 Optional General-Purpose Interrupts	
220 - 237	Group 1 Optional General-Purpose Interrupts	
⋮	etc.	

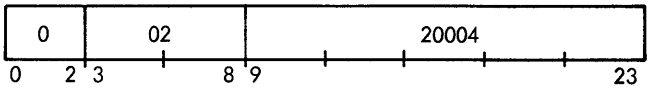
EIR ENABLE INTERRUPT SYSTEM



EIR unconditionally sets the INTERRUPT ENABLED indicator and enables the interrupt system. If any interrupt levels are waiting, the one with the highest priority becomes active.

Registers Affected: None Timing: 1

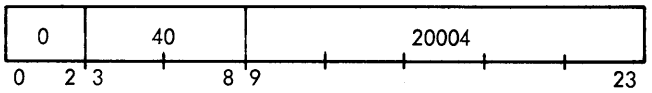
DIR DISABLE INTERRUPT SYSTEM



DIR unconditionally resets the INTERRUPT ENABLED indicator and disables the interrupt system. This instruction does not change the current state of any interrupt level.

Registers Affected: None Timing: 1

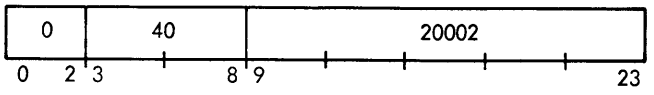
IET INTERRUPT ENABLED TEST;
SKIP IF INTERRUPT SYSTEM ENABLED



If the priority interrupt system is enabled, the computer skips the next instruction in sequence and executes the following instruction. If the priority interrupt system is disabled, the computer executes the next instruction in sequence.

Registers Affected: None Timing: 1, if no skip
2, if skip

IDT INTERRUPT DISABLED TEST;
SKIP IF INTERRUPT SYSTEM DISABLED



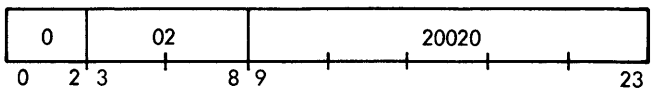
If the priority system is disabled, the computer skips the next instruction in sequence and executes the following instruction. If the priority interrupt is enabled, the computer executes the next instruction in sequence.

Registers Affected: None Timing: 1, if no skip
2, if skip

ARMING FEATURE (Optional)

The arming feature is controlled for a group of 16 interrupts by a word sent to the group with the ARM INTERRUPTS (AIR) instruction followed by the PARALLEL OUTPUT (POT) instruction. AIR operates only on System interrupts (200-1777).

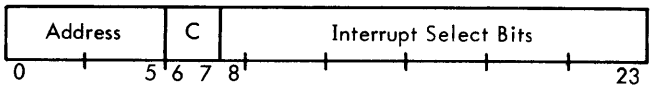
AIR ARM INTERRUPTS



AIR prepares the arm interrupt control unit to receive a control word for a group of 16 interrupt levels. A PARALLEL OUTPUT (POT) must always follow AIR, or an unpredictable operation results.

Registers Affected: None Timing: 1

Section IV, Input/Output System, contains a discussion of PARALLEL OUTPUT (POT). The word that the POT instruction addresses has the following format:



The address field in bit positions 0 through 5 identifies which group of 16 interrupts in the system is being addressed. Address 00 refers to the group of locations 200-217. The C field controls what is done to the particular interrupt levels selected in bit positions 8 through 23. Bit position 8 refers to the lowest-numbered level of the group, therefore the one with highest priority. Bit position 23 refers to the last or highest-numbered level, the one with lowest priority. For example, a word of 00240000 arms level number 201.

The control operations are:

Bit Position	Octal Position	Octal Value	Function
6-7	O2		
00		0	Not used
01		2	Arm all interrupt levels that are selected by a 1 in bit positions 8 - 23
10		4	Disarm all interrupt levels that are selected by a 0 in bit positions 8 - 23
11		6	Arm all interrupts selected by a 1 and disarm all interrupts selected by a 0 in bit positions 8 - 23.

CHANNEL INTERRUPT DESIGNATIONS

As shown in the Interrupt Location Table, each I/O channel has two interrupt levels. These reflect the two distinct uses of interrupts during channel input and output. Also, each W, Y, C, and D channel level has two names that reflect their use in the Extended or Compatible I/O Modes (see Section IV, Compatible/Extended Input/Output Modes).

END-OF-WORD/END-OF-TRANSMISSION INTERRUPT OPERATIONS; COMPATIBLE MODE

A program can use Channels W and Y as single-word, direct, program-controlled, input/output buffers. Special I/O instructions applicable to Channels W and Y control this type of operation (see Section IV). In this mode, the program can specify that interrupts occur as each word is transferred from the buffer to the peripheral device on output, or as soon as the buffer is filled from the peripheral device on input. This is the End-of-Word interrupt. The program can specify that an End-of-Transmission interrupt occurs when the buffer detects a signal such as End-of-Record from magnetic tape. During both input and output operations, this interrupt occurs when the peripheral device used in the transmission disconnects and the buffer becomes ready for another input/output operation.

These two interrupts also can control input/output termination for any communication channel when the program is operating the buffers in the block transmission or "interlaced" compatible mode (optional system). The End-of-Transmission interrupt operates in the fashion described above. In this mode, the End-of-Word interrupt only occurs on input. The End-of-Word interrupt occurs after the channel has read the number of words specified and then another word fills the buffer. If the program encounters the last word before an End-of-Transmission interrupt, the End-of-Word interrupt occurs after the next word is read. If an End-of-Record condition occurs first, the End-of-Transmission interrupt occurs. No End-of-Word interrupt occurs during output.

The Enable or Disable instructions "enable and arm" or "disable and disarm" the End-of-Word and End-of-Transmission interrupts when the channel is not operating in the extended interlace mode. When the EIR instruction is executed, the interrupt system is enabled and these interrupts are also armed; when DIR is executed, the system is disabled and these interrupts are also disarmed.

COUNT EQUALS ZERO/END-OF-RECORD; EXTENDED MODE

When the computer's input/output system uses channels within its full capabilities, input/output functions control interlaced block transmission operations (see Input/Output Functions, Section IV). The interrupts used with the extended input/output function control are Count Equals Zero and End-of-Record. The Count Equals Zero interrupt occurs when the last of the number of words specified is placed into or brought from memory. The End-of-Record interrupt occurs when the channel receives an End-of-Record signal (gap). Input/output functions can alter this latter occurrence for use with magnetic tapes.

EFFECTS OF THE ENABLE/DISABLE FEATURE ON ARMABLE INTERRUPTS

When operating an Input/Output Channel in the extended mode, the interrupt Enable feature controls the Armable interrupts (Count Equals Zero and End-of-Record). If a channel generates an extended mode I/O interrupt while the system is disabled, the designated interrupt level goes to the Waiting state. When the program again enables the interrupt system, the interrupt goes to the Active state when its priority allows.

This feature allows the programmer great ease in handling multiple channel operations. The interrupt processing subroutine for one channel can disable the interrupt system while it processes the interrupt. During this time, the system receives all other interrupts in their respective levels and goes to the Waiting state until the system is again enabled.

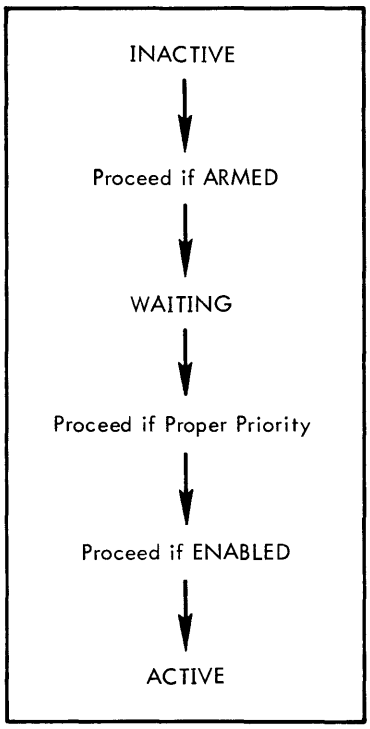


Figure 3-1. Interrupt Arm-Enable Response

IV. INPUT/OUTPUT INSTRUCTIONS

INTRODUCTION

The XDS 925 has a flexible, input/output system to complement its high, internal processing speed and versatile instructions. This system can transmit data in word, character, or single-bit form to and from the computer at the speed of internal computation. The input/output system assumes control of conditions imposed by different characteristics of a wide variety of devices, but leaves a high degree of input/output control to the programmer.

This system includes the following types of input/output:

Buffered input/output of data words, each under direct program control

Communication channel input/output of characters or words, time-shared with memory and multiplexed with computation.

Communication channel input/output of characters or words, fully buffered and simultaneous with computation.

Direct parallel input/output of up to 24 bits of information to and from external equipment, completely controlled and sequenced externally.

Direct parallel input/output of up to 24-bit words to and from external static registers under program control.

Single-bit input/output, such as equipment on/off status, sense switches, and pulsing and sensing of special devices.

DATA FLOW PATHS

The XDS 925 includes as standard equipment one Time-Multiplexed Communication Channel (TMCC), without interlacing capability, as well as provision for three additional channels. The interlace unit is available as an option. The W and Y channels are available with or without interlace; the C and D channels are available only with interlace. These channels are capable of automatically controlling the flow of data to and from memory at rates up to one word every 3.5 microseconds. These channels run independently of the central processor and only communicate with it to transfer data to or from memory.

In addition to the Time-Multiplexed Channels, a Direct Memory Access System is available. This system uses a path to memory separate from those used by the central processor. Up to four Direct Access Communication Channels (with direct access memory connections) can be attached to the Direct Access System. These channel operate like the time-multiplexed Channels except that they are faster and provide for a true overlap of input/output with processing.

A Data Multiplex System, which uses the direct access memory connection, is also available as an option. This system consists of a Data Multiplex Channel that accepts/transmits data words and memory addresses from many external devices or subchannels, all of which may be in operation at the same time. The system is capable of transmitting up to 572,000 words per second simultaneous with computation (see Appendix G).

COMMUNICATION CHANNELS

Using Channels W and Y, characters and words can be transmitted between memory and peripheral devices under the direct control of single instructions. Each channel has associated with it two instructions to facilitate direct control operations. For Channel W, W INTO MEMORY (WIM) causes a word from a peripheral transmission to be taken from the Channel W buffer register and placed directly in the specified memory location without disturbing any internal registers. MEMORY INTO W (MIW) causes a word to be taken from a specified memory location and placed in the Channel W buffer register to be read out to the currently operating peripheral device connected to the channel. WIM and MIW are preceded by instructions from the EOM group that set up the input/output operation. YIM and MIY instructions function in an analogous manner for channel Y. The general test instruction, SKIP IF SIGNAL NOT SET (SKS) provides the facility for testing error indications and/or for testing various peripheral device indicators.

Additionally, using any channel including Channels W and Y with interlace, data can be transmitted to and from core storage under channel control. Operation of a channel is initiated by the execution of a sequence of instructions in the central processor. Once started, the channel operates independently of the central processor, automatically transferring each word at the correct time.

Four instructions control the process of transmitting and receiving data between channel peripheral equipment and the central processor. These instructions are:

EOM	ENERGIZE OUTPUT M
EOD	ENERGIZE OUTPUT TO DIRECT ACCESS CHANNELS
POT	PARALLEL OUTPUT
SKS	SKIP IF SIGNAL NOT SET

EOM instructions activate one of Channels W, Y, C, or D, to select the peripheral device to be used, and to set up the initial conditions of the data transmission, including the peripheral operation to be performed. EOD instructions activate one of Channels E, F, G, or H. The other functions of EOD are similar to EOM. An EOM (EOD) instruction also specifies terminal conditions for an operation.

PARALLEL OUTPUT (POT) sends out to the channel the number of words in the transmission and the address at which the output begins.

SKIP IF SIGNAL NOT SET (SKS) can test the Error indicators, End-of-Transmission indicators, and other input/output control indicators, such as printer end-of-form or card hopper empty.

The general order of use of these instructions for interlaced operation is:

Instruction	Function
EOM	to address the channel, connect the peripheral device, specify various input/output conditions, and alert the optional channel interlace (see Communication Channel Input/Output)

Instruction	Function
EOM	to specify the terminal conditions and interrupts desired during the transmission
POT	to transmit to the channel a word containing the transmission starting address and block length

Bits 0 through 9 of this latter word contain the ten lower order bits of the word count; bits 10 through 23 contain the 14 bits of the starting address. The second EOM contains the high-order bits of the word count and starting address when needed.

DIRECT PARALLEL INPUT/OUTPUT

The direct parallel input/output (POT/PIN) facility allows any word in core memory to be presented, in parallel, at any special system connector or applicable standard peripheral connector; or, conversely, allows signals sent to a connector to be stored in any core memory location. EOM and SKS instructions control parallel input/output operations in the same way as in channel operations. POT/PIN instructions also generate or check for correct parity with each word transmitted. The system provides block transfer form of POT and PIN with the instructions, BLOCK PARALLEL INPUT (BPI) and BLOCK PARALLEL OUTPUT (BPO). By placing the word count N minus one in the A Register, BPI and BPO provide the identical function of PIN and POT, respectively, on N consecutive words.

See Direct Parallel Instructions, this section, for a detailed description of parallel input/output.

SINGLE-BIT INPUT/OUTPUT

EOM and SKS instructions also perform single-bit input/output and testing for special or standard devices. The execution of an EOM transmits a single signal of approximately 1.75 microseconds duration to an external connector and also provides the connector with a 15-bit address for the destination of this signal. SKS tests whether a similar signal is present on an external connector and skips accordingly. See Single-Bit Transmission, this section, for further description of single-bit input/output.

DIRECT MEMORY ACCESS SYSTEM

This optional system provides direct transmission between peripheral devices and core memory. Two access paths to the memory module are available. The standard path connects to the central processor; the other path connects to Direct Access Communication Channels on the Direct Memory Access Connection. Direct memory access allows data to be transmitted at the rate of one 24-bit word every 1.75 microseconds, thus sustaining an input/output rate of 572,000 words (equivalent to 2,284,000 characters) per second.

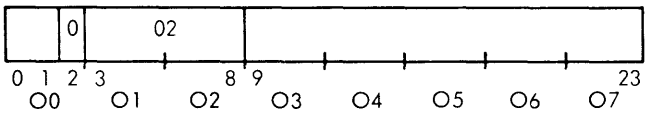
Communication Channels, E, F, G, and H, if present in a system, require the Direct Memory Access System, and therefore, are called Direct Access Communication Channels. Operation of these channels is discussed in Communication Channel Input/Output, this section.

Note that the Direct Memory Access System may be obtained separately and used to incorporate special-purpose input/output equipment instead of the standard Direct Access Channels. Externally controlled and sequenced devices may present data and addresses to the direct access connectors, thus allowing input/output operations or other memory accesses to be performed independently of the computer.

These special input/output systems present an address and various timing and control signals to the connector. External data may be stored in any specified location, or read from any location specified by the external unit. For example, the external equipment may provide an interface register, thereby allowing an entire block of data to be entered into or read from memory. Telemetry data may be automatically decommutated, thus obviating sorting and sequencing within the computer.

PRIMARY INPUT/OUTPUT INSTRUCTIONS

EOM ENERGIZE OUTPUT M



The major instruction for preparing Channel W (or Y, C, D) and an attached peripheral device to perform a data transmission or other peripheral activity is the multi-purpose instruction, ENERGIZE OUTPUT M (EOM). It operates in four distinct modes with many functional configurations. These modes are Buffer Control, Input/Output Control, Internal Control, and System Control. In the third and fourth modes, EOM controls and initiates non-communication channel operations such as special systems transmissions. Each of the frequently used EOM instruction configurations has a mnemonic tag used with standard XDS assemblers. These mnemonics appear in this manual with the description of the specified configurations. The different modes of operation are program-selectable by the setting of two bits (10, 11 of octal position 3) within the EOM instruction format:

Octal Value	Bit Position 10	Bit Position 11	Area
0	0	0	Buffer Control
1	0	1	Input/Output Control
2	1	0	Internal Control
3	1	1	System Control

A Buffer Control mode EOM operates essentially as a set-up or preparation facility for data transmissions or other peripheral activities using the channel. The channel to be used, the peripheral unit on that channel, the operation to be performed, and the type of character format to be used are all detailed within this EOM. It also details the use of BCD or binary data transmission, the allowance or not of a leader (as in paper tape),

and the direction of operation (as in forward direction for magnetic tape). Execution of such an EOM "connects" the specified peripheral unit to the channel. An EOM in this mode can also alert the interlace, which is the optional, automatic buffer control for input/output.

An EOM in the Input/Output mode directs peripheral devices to perform non-transmitting operations such as rewind magnetic tape and upspace the printer. This EOM selects certain channel operations such as interrupt response and input/output terminal function desired. It alerts peripheral devices that a PARALLEL INPUT (PIN) or PARALLEL OUTPUT (POT) instruction follows. It also can give an extension of the word count to 14 bits for the number of words to be transmitted. Without disturbing the associated channel, this EOM can also set up the interlace unit. It is with the Input/Output mode EOM that the user selects his I/O operation as Compatible or Extended I/O Modes (described later in this section).

This coding sequence initiates such an interlaced channel operation (compatible mode):

Instruction	Function
EOM (Input/ Output Control Mode)	Alert the interlace
POT	transmit starting address and block length to interlace
EOM (Buffer Control Mode)	address channel, connect peripheral device, specify various input/output conditions, start transmission

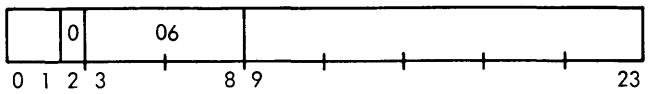
Initiating an interlaced input/output operation via this sequence of instructions facilitates checkout by allowing the programmer to single-step through this portion of the program. The first two instructions, EOM (Ioc) and POT, set up the interlace with data address and block length. Therefore, single-stepping through the sequence allows the interlaced channel to complete the input/output operation. When a single EOM (Buffer Control mode) sets up the channel and interlace with a POT instruction following, the programmer cannot step through the sequence since the input/output operation proceeds before the next stepped instruction (POT) places the address and block length in the interlace.

An EOM in the Internal Control mode enables and disables the interrupt system. EOM in this mode also can prepare the system for the selective arming and disarming of the system interrupt levels. This mode does not directly concern the input/output programmer.

An EOM in the System Control mode is specifically coded for a given installation and system. Address capability is 15 bits or 32,768 combinations for these special system designations.

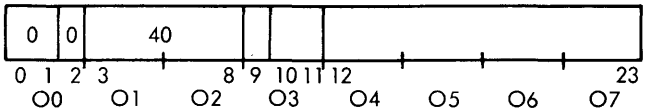
Note: If an interrupt occurs during the execution of an EOM in any mode, no acknowledgement occurs until the completion of the execution of the instruction following the EOM.

EOD ENERGIZE OUTPUT TO DIRECT ACCESS CHANNEL



The EOD instruction operates in the Buffer Control and Input/Output Control modes. It refers to Channels E, F, G, and H, when present, and performs the same functions and operations as an EOM on these channels.

SKS SKIP IF SIGNAL NOT SET



The principal instruction for testing the states and responses of data channels and their attached peripheral devices, as well as testing internal and external indicators, is the multi-purpose instruction, SKIP IF SIGNAL NOT SET (SKS). SKS is a "skip class" instruction yielding a decision and transfer capability to all channels, devices, indicators, and systems that require it. It operates in four distinct modes: Special Internal Test, Channel and Device Test, Internal Test, and Special System Test. In the second mode, SKS tests channel-oriented, input/output functions. Each of the frequently used SKS instruction configurations has a mnemonic tag, used with XDS assemblers. These mnemonics appear in this manual with the description of the specific configuration.

These different modes of operation are program-selectable by the setting of two bits (10, 11 of octal position 3) within the SKS instruction format:

Bit Positions 10	Bit Positions 11	Octal Value	Area
0	0	0	Special Internal Test
0	1	1	Channel and Device Test
1	0	2	Internal Test
1	1	3	Special System Test

In the Channel and Device Test mode, SKS tests a channel for channel Ready (not active), interlace Word Count Equal to Zero, and Error. This mode also tests peripheral devices directly. These include testing indicators in a magnetic tape unit such as Beginning-of-Tape, End-of-Tape, File-Protect Ring present, and End-of-File. For example, an SKS instruction might address an indicator within the printer to determine whether the paper is at the End-of-Form.

In the Internal Test mode, SKS tests whether the interrupt system is enabled or disabled, whether a breakpoint switch is set, and whether Overflow is set.

In the Special Internal and Special System Test modes, SKS tests signals of special configuration as the specific system requires.

COMMUNICATION CHANNEL INPUT/OUTPUT

GENERAL INFORMATION

XDS Communication Channels provide fully buffered, input/output control and transmission, multiplexed or simultaneous with computation. Up to eight data channels can connect to the central processor, all operating independently of each other.

Each channel can control as many as 30 input/output devices and automatically handles character, word assembly and disassembly, input/output parity detection and generation, data transmission to and from memory, and End-of-Transmission detection.

All channels are bi-directional and can communicate with 6-bit character devices or word devices of up to 24 bits. In the case of character-oriented devices, the program specifies the number of characters to be contained in each word during the transmission.

A channel buffer assembles and disassembles data words as they are transmitted between core memory and the peripheral equipment. The buffer maintains control of operations such as characters per word transmitted and direction of peripheral operation (as in magnetic tape forward/reverse).

A Buffer Control mode EOM or EOD sets up the channel buffer for operation. The execution of this EOM sets the operation controls, places the unit address in the buffer, and initiates data assembly/disassembly. The presence of the unit address activates the buffer, causing it to look for data coming from the peripheral device or from memory, as determined by the unit address.

When in use, a channel interlace controls the transfer of the data words going through the associated channel buffer. This interlace supplies the memory address of data coming from or going to memory and maintains the word count determining the number of words transferred. The terminal interrupts, End-of-Record and Zero Word Count, come from the interlace and are under its control. The interlace controls input/output termination functions during interlaced operation.

Two EOM instructions and a POT instruction alert and set up a channel interlace. The first EOM alerts the interlace, that is, activates the interlace and instructs it to expect a word count and starting address to be sent to it by the POT instruction. The second EOM is an Input/Output mode EOM that specifies the interrupt and the terminal function to be used. This EOM also can specify a 15th address bit and five more high-order word count bits expanding the word count from 10 bits to 15. This sequence is written: EOM (Alert), EOM (I/O), and POT. When the channel buffer is being set up at the same time, the buffer control EOM can alert the interlace. When the buffer is already set up, during a continuing I/O operation, the programmer may use the I/O EOM, ALERT CHANNEL (00250000), to alert the interlace.

When the programmer does not desire to program the Extended Mode with the input/output terminal functions, interrupts, and additional count or address, only the EOM (Alert) and the POT are necessary to set up the channel interlace (Compatible mode).

In the Extended Mode, the eight channels are programmed in the same way, though there is a distinction between Channels W through D and Channels E through H. The former group are Time-Multiplexed Channels; the latter are Direct Access Channels.

The Time-Multiplexed Channels use the memory logic of the central processor to facilitate input and output of data words. The transfer of each word between a time-multiplexed channel buffer and memory requires two memory cycles. During this time, computation stops in the central processor. Priority for the use of the word input/output logic is in the order: Channel D, C, Y, W. Any Time-Multiplexed Channel operating with interlace has priority over the central processor for memory access.

Each Direct Access Channel has its own independent memory logic. When memory access is needed to read or store a data word, computation stops for one cycle. When two or more Direct Access Channels require memory access simultaneously, determination of priority is as described in Appendix F.

Transmission to and from Direct Access Channels and core memory are under the control of the channel. At the onset of each memory cycle, the control unit interrogates all Direct Access Channels to determine whether any channel requires a transfer to or from computer memory; each channel gets priority on the basis of need. If, during a channel transmission, a transfer to or from computer memory is to take place, the computer connects the memory bank to the selected Direct Access Channel. If, simultaneously, the computer requires access to the same memory bank, the channel takes precedence and there is a delay of one memory cycle. If the computer is not accessing the same memory bank as the direct access channel, the transfer takes place without affecting computation speed. Thus, internal computation and direct access channel transmissions occur simultaneously and independently when the computer and channel are accessing separate memory banks. Channel control logic permits the transfer of only one word per memory cycle to and from the computer memory independent of the number of operating channels connected to the computer. Thus, the maximum transfer rate for the channel system is equal to one word every memory cycle, or approximately 572,000 words per second, or in excess of two and one-quarter million characters per second for direct access channels.

COMMUNICATION CHANNEL DESCRIPTION

Figures 4-1 and 4-2 contain block diagrams of the channels, the functional control of information between the channels, the Data Multiplexing System, the memory bank, and the external devices.

Up to 30 peripheral devices may be connected to one channel. Each of these devices has a unique, two-digit, octal address by which it is selected for an input/output operation. To select the peripheral device, the program loads the proper unit address into the 6-bit Unit Address Register (UAR) in the channel buffer. This address selects both the device and, if appropriate, the function to be performed. Placing a non-zero unit address in the Unit Address Register "connects" the peripheral unit addressed to the channel and it becomes "active". When the UAR contains a zero address, or any time that a terminal or initial condition clears the contents of UAR, the channel is "inactive." The zero in UAR also means that it is not connected to a peripheral unit.

When the channel and the peripheral unit to be used have been connected, the channel must have information pertaining to the location in memory of the data to be transmitted or received and pertaining to the number of data words in the transfer.

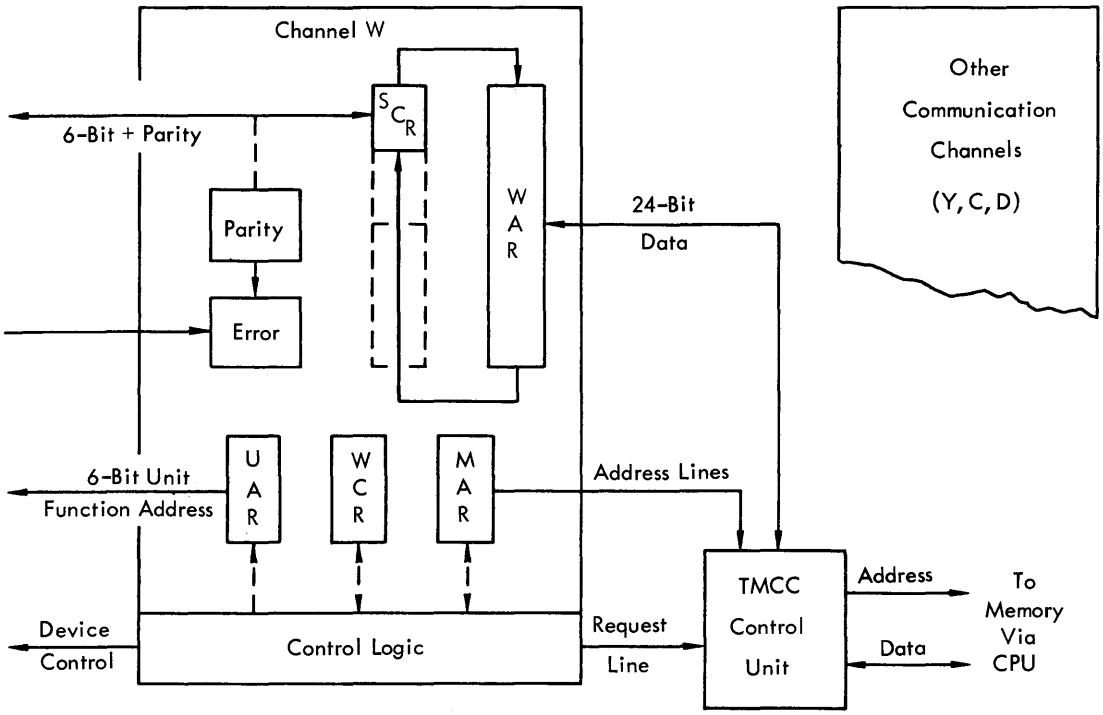


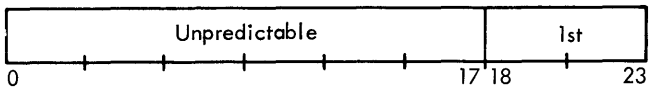
Figure 4-1. XDS 925 Time-Multiplexed Communication Channel, Block Diagram

TIME-MULTIPLEXED CHANNEL REGISTERS

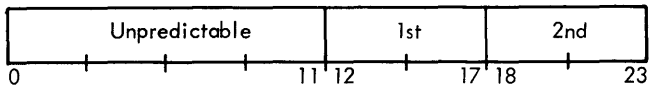
In the Time-Multiplexed Channels W through D, there are two registers important to the programmer, the Word Assembly Register (WAR) and the Single-Character Register (SCR). The WAR, a 24-bit, word-sized buffer, contains the word of data actively being received or transmitted during an input or output operation. During input, 6-bit characters (plus parity) enter the Single-Character Register where the channel buffer assembles them, one at a time, into the WAR. Then the completed word is placed in memory. Depending on the number of characters per word specified, the word assembled and placed in memory during input has the form:

Word in Memory

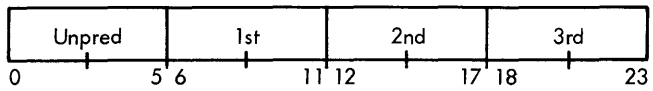
One character per word mode



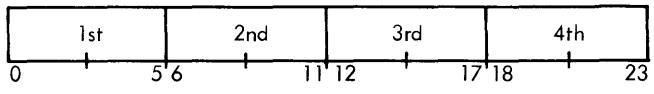
Two characters per word mode



Three characters per word mode



Four characters per word mode



The unfilled character positions contain unpredictable data. When assembled during a single-word operation, a WIM instruction places the word into memory. Under interlace control, the interlaced channel automatically places the word in memory when assembled.

When the end of an information record is detected by a buffer, the buffer automatically disengages from the device and is then "ready" for another operation. The buffer logic is reset, except that the state of the error indicator is maintained and the last word of the input is still in the word register. If the number of characters in the input record was not a multiple of the number of characters assembled into each computer word, then zeros are automatically forced into the least significant positions of the last word. This last word can then be stored in memory by a BUFFER INTO M WHEN READY WIM or YIM instruction after the buffer has disengaged. If the number of characters in the input record was a multiple of the number of characters assembled into each computer word, then the word remaining in the W buffer is either the last group of characters from the input device, if they were not previously transferred to memory by a BUFFER INTO M WHEN READY WIM or YIM, or zeros if the last group of characters had been transferred to memory. In either case, it is safe to issue one such instruction after the buffer has disengaged without "hanging up" the computer.

During output, words come from memory into the WAR where the channel buffer disassembles them into the SCR one 6-bit character at a time. Depending on the characters per word mode specified, the 6-bit characters within the word are output as follows:

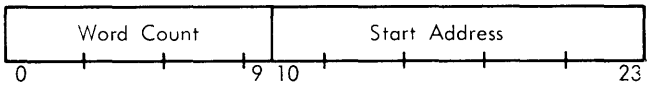
Function	Mode
Output one character from bits 0 through 5	One character per word
Output two characters from bits 0 through 5, 6 through 11	Two characters per word
Output three characters from bits 0-5, 6-11, 12-17	Three characters per word
Output four characters from bits 0-5, 6-11, 12-17, 18-23	Four characters per word

As required, the characters are transferred into the Single-Character Register and output with generated parity. After each character transfer, the word in the WAR is shifted left six bits to be ready for the next transfer. Only those characters needed from each word are used; when required, a new word is brought to the WAR for the next character. For special applications, a Time-Multiplexed Channel may be equipped with a 12- or 24-bit Single-Character Register. The external device which has a character size greater than 6 bits specifies to the channel what its size is, 12 or 24 bits. Standard 6-bit devices are unaffected by the installation of a wider SCR.

Interlace Registers

A channel interlace contains two working registers, the Word Count Register (WCR) and the Memory Address Register (MAR). In the set-up sequence -- EOM, EOM, POT -- for an interlaced input/output operation, the POT instruction transmits to the interlace a data word made up of the word count (that is, length) and the starting address of the data block. The 15-bit Word Count Register (WCR) contains the data word count during a data transfer. The number of data words is decremented by one and the new count replaces the old one in the WCR for each word transmitted.

The count is assembled into the WCR from two places: the least significant 10 bits is from the "POTted" word and the most significant 5 bits is from the "HI COUNT" field of the second EOM. The form of the "POTted" word is:



When the word count is equal to zero, the transmission is complete. During output, this causes a termination; during input, the interlace allows any further data to fill the channel buffer and generates the End-of-Word interrupt, if enabled.

The Memory Address Register (MAR) contains the starting destination or source address in memory of the transmitted data. The memory locations to or from which data words are to be transmitted enter the MAR at the same time the word count does. During transmission of data, the interlace increments the contents of the MAR after each word as it decrements the contents of the WCR. These two registers provide the interlace control of block transmissions.

DIRECT ACCESS CHANNEL REGISTERS

In the Direct Access Channels E through H, two other important registers are the Word Assembly Register (WAR) and the Input/Output Register (IOR). The Word Assembly Register is a 24-bit word-sized buffer which, during a transmission, contains the information actively being transmitted to, or received from, the external device. Information is assembled into, or disassembled from, the WAR in one of four character sizes, 6, 8, 12 or 24 bits. The 6-bit mode is the normal mode of operation. A device with a larger character size will send the channel a signal to indicate its character size. It is the programmer's responsibility to select a character/word count suitable for the character size.

(Time-Multiplexed channels can handle only 6-bit characters, standard. There are, however, two options which will increase the acceptable character size to 12 bits and to 24 bits. As with the DACCs the external device signals the TMCC with its character size.)

When receiving 6-bit characters from a peripheral device (operation is similar for other character sizes), the first character of a word enters the WAR into bit positions 18 through 23. When the WAR receives the next character, the first

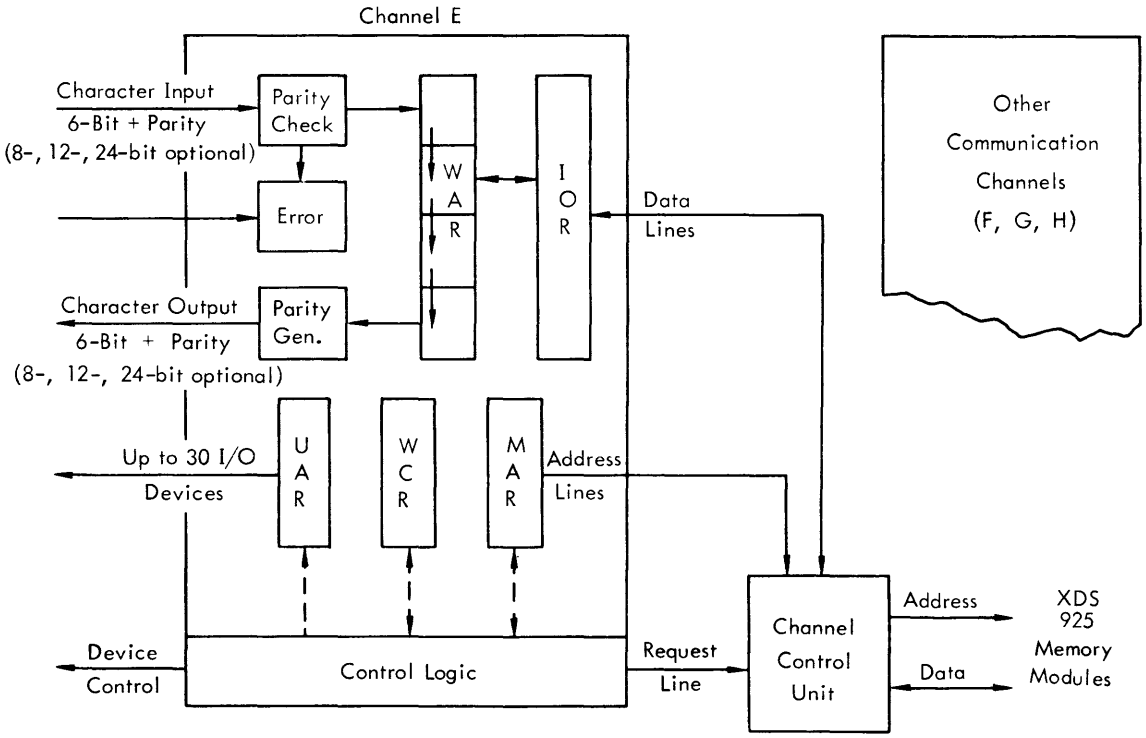


Figure 4-2. XDS 925 Direct Access Communication Channel, Block Diagram

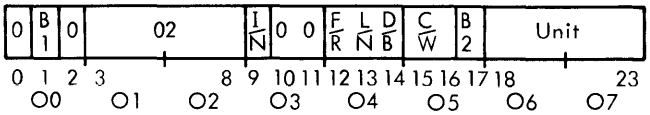
six bits in positions 18 through 23 shift into bit positions 12 through 17 and the incoming character is placed into bit positions 18 through 23. The next incoming character causes the two 6-bit characters in bit positions 12 through 23 to be shifted to bit positions 6 through 17 and the incoming character is placed into bit positions 18 through 23. The next character causes another 6-bit left shift and then the character is placed in the vacated bit positions 18 through 23. At this point, there are 24 bits completely filling the WAR. This information is now copied into the IOR to be placed into the proper memory location.

The above procedure occurs when the programmer specifies four characters per data word for the data transmission. If the specification is three characters, the data word contains three 6-bit characters in bit positions 6 through 23 and unpredictable information in bit positions 0 through 5 are transmitted to the IOR. The next incoming character is accepted as the first of another set of three characters. If the programmer specifies two characters, the data word contains two 6-bit characters in bit positions 12 through 23 and random data in bit positions 0 through 11 are transmitted to the IOR. If the specification is one character, the data word transmitted to the IOR contains only one character in bit positions 18 through 23. When transmitting data using the character format mode, characters are taken from the WAR from the most significant end. If the programmer specifies one character per word, the 6-bit character in bit positions 0 through 5 is transmitted to the external device and then another full word of information is received from the IOR. If the programmer specifies two characters per word, the 6-bit character in bit positions 0 through 5 is transmitted. Then the contents of bit positions 6 through 23 shift left into bit positions 0 through 17, the new 6-bit character in bit positions 0 through 5 is transmitted and another word is accepted from the IOR to be processed. If the programmer specifies three characters, the 6-bit character in bit positions 0 through 5 is transmitted. The contents shift left six bits and the new contents of bit positions 0 through 5 are transmitted. The contents shift left six bits again and the third character from bit positions 0 through 5 is transmitted. Then another word is received from the IOR to be processed. If the programmer specifies four characters, the above process continues to one more 6-bit left shift and the final six bits of the word are transmitted before the next word is accepted from the IOR.

The Input/Output Register (IOR) is a 24-bit register which is a full-word buffer between the WAR and memory. The Direct Access Channel control unit places words into the IOR, awaiting their transfer to the WAR to be output. During input, the IOR receives full words from the WAR and places them into memory under control of the word count and memory address being used in the transmission. During multiple data word transfers, the WAR and the IOR simultaneously contain data information.

COMMUNICATION CHANNEL EOM

The ENERGIZE OUTPUT M (EOM) used in the Buffer Control mode addresses and connects the specified Channel W, Y, C, or D, and selects the desired unit address. The detailed instruction format is:



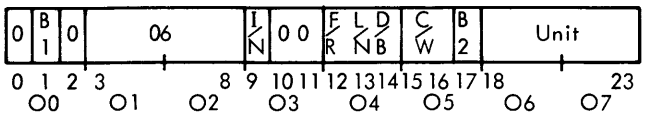
Bit Designation	Octal Position	Octal Value	Function
B1	O0	2	Bit positions 1 and 17 specify the channel to be activated.
B2	O5	1	Channel W is numbered 00, Channel Y is 01, Channel C is 10, and Channel D is 11.
02	O1-2	02	Bit positions 3 through 8 contain 02, the instruction code for EOM.
I/N	O3	4	A 1-bit in position 9 alerts the buffer interlace.
00	O3	0	Bit positions 10 and 11 contain the EOM mode indicator for the Buffer Control mode.
F/R	O4	4	Bit position 12 specifies the direction in which the peripheral device will operate. A "0" specifies the forward direction. A "1" specifies the reverse direction.
L/N	O4	2	Bit position 13 specifies whether the device should be started with a leader as in paper tape. A "0" specifies a start with leader. A "1" specifies a start without leader.
D/B	O4	1	Bit position 14 specifies the mode of character format. A "0" specifies BCD format. A "1" specifies Binary format.
C/W	O5		Bit positions 15 and 16 specify the number of characters to be assembled into, or disassembled from, each transmitted word. One character per word is specified by 00 (octal 0),
		0	
		2	two by 01 (octal 2), three
		4	by 10 (octal 4) and four
		6	by 11 (octal 6).
UNIT	O6-7		Bit positions 18 through 23 specify the unit and the function to be performed with that unit.

Table 4-1. Unit Address Codes

00	Disconnect	40	-
01	Type Input No. 1	41	Type Output No. 1
02	Type Input No. 2	42	Type Output No. 2
03	Type Input No. 3	43	Type Output No. 3
04	Paper Tape Input No. 1	44	Paper Tape Punch Output No. 1
05	Paper Tape Input No. 2	45	Paper Tape Punch Output No. 2
06	Card Reader Input No. 1	46	Card Punch Output No. 1
07	Card Reader Input No. 2	47	Card Punch Output No. 2
10	Magnetic Tape Input No. 0	50	Magnetic Tape Output No. 0
11	Magnetic Tape Input No. 1	51	Magnetic Tape Output No. 1
12	Magnetic Tape Input No. 2	52	Magnetic Tape Output No. 2
13	Magnetic Tape Input No. 3	53	Magnetic Tape Output No. 3
14	Magnetic Tape Input No. 4	54	Magnetic Tape Output No. 4
15	Magnetic Tape Input No. 5	55	Magnetic Tape Output No. 5
16	Magnetic Tape Input No. 6	56	Magnetic Tape Output No. 6
17	Magnetic Tape Input No. 7	57	Magnetic Tape Output No. 7
20	-	60	High-Speed Printer Output No. 1
21	-	61	High-Speed Printer Output No. 2
22	-	62	-
23	-	63	-
24	-	64	Incremental Plotter Output No. 1
25	-	65	Incremental Plotter Output No. 2
26	Rapid Access Device Input No. 1	66	Rapid Access Device Output No. 1
27	Rapid Access Device Input No. 2	67	Rapid Access Device Output No. 2
30	Scan Magnetic Tape No. 0	70	Magnetic Tape Erase No. 0
31	Scan Magnetic Tape No. 1	71	Magnetic Tape Erase No. 1
32	Scan Magnetic Tape No. 2	72	Magnetic Tape Erase No. 2
33	Scan Magnetic Tape No. 3	73	Magnetic Tape Erase No. 3
34	Scan Magnetic Tape No. 4	74	Magnetic Tape Erase No. 4
35	Scan Magnetic Tape No. 5	75	Magnetic Tape Erase No. 5
36	Scan Magnetic Tape No. 6	76	Magnetic Tape Erase No. 6
37	Scan Magnetic Tape No. 7	77	Magnetic Tape Erase No. 7

EOD ENERGIZE OUTPUT ON DIRECT ACCESS CHANNEL

The EOD instruction used in the Buffer Control mode alerts and connects the specified Direct Access Channel (E, F, G, H) and the desired unit address. The instruction format is:



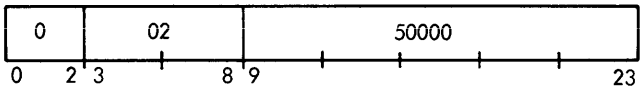
Bit Designation	Octal Position	Octal Value	Function
B1	O0	2	Bit positions 1 and 17 specify the channel to be activated. Channel E is numbered 00, Channel F is 01, Channel G is 10, and Channel H is 11.
B2	O5	1	

All other indicators in the EOD are identical with EOM and function in the same way.

STANDARD EOM AND EOD CHANNEL INSTRUCTIONS

Several EOM and EOD function configurations have standard uses. These have standard, assembler-type mnemonics and are separate instructions.

ALC ALERT CHANNEL



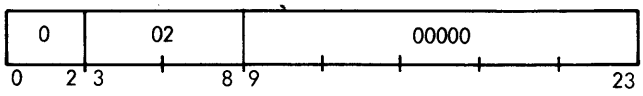
ALC alerts the channel interlace. This instruction does not disturb the channel buffer in any way. ALC has no effect on W or Y Buffers without interlace.

The channel Alerts are:

Mnemonic	Alert Channel	Instruction
ALC 0	W	0 02 50000
ALC 1	Y	0 02 50100
ALC 2	C	2 02 50000
ALC 3	D	2 02 50100
ALC 4	E	0 06 50000
ALC 5	F	0 06 50100
ALC 6	G	2 06 50000
ALC 7	H	2 06 50100

Registers Affected: None Timing: 1

DSC DISCONNECT CHANNEL

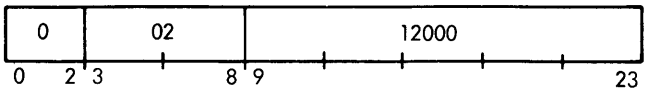


DSC disconnects the channel. It unconditionally sets the Unit Address Register to 00 regardles of whether the channel is currently addressing a device. This instruction disconnects any device which may be connected to the channel. It also unconditionally makes the channel Ready (Inactive) and clears the Error indicator.

Mnemonic	Disconnect Channel	Instruction
DSC 0	W	0 02 00000
DSC 1	Y	0 02 00100
DSC 2	C	2 02 00000
DSC 3	D	2 02 00100
DSC 4	E	0 06 00000
DSC 5	F	0 06 00100
DSC 6	G	2 06 00000
DSC 7	H	2 06 00100

Registers Affected: None Timing: 1

ASC ALERT TO STORE ADDRESS FROM CHANNEL



ASC alerts an interlaced channel so the PIN instruction that follows can store the contents of the Memory Address Register. This instruction affects the operation of the channel in no other way. See Direct Parallel Instructions, this section, for a detailed discussion of PIN.

ASC is always used in conjunction with PIN to determine the current status of a peripheral operation being performed by the selected channel. The two instructions are written together:

```
ASC    n
PIN    m, x
```

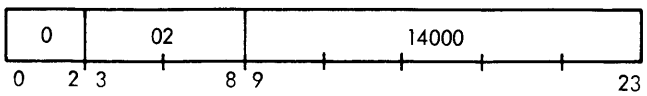
When the program executes these two instructions, the contents of the effective memory location designated by the PIN instruction are:

Bit Positions	Contents
0 through 9	Zero
10 through 23	Contents of channel's Memory Address Register

Mnemonic	Channel	Instruction
ASC 0	W	0 02 12000
ASC 1	Y	0 02 12100
ASC 2	C	2 02 12000
ASC 3	D	2 02 12100
ASC 4	E	0 06 12000
ASC 5	F	0 06 12100
ASC 6	G	2 06 12000
ASC 7	H	2 06 12100

Registers Affected: None Timing: 1

TOP TERMINATE OUTPUT ON CHANNEL



When the last word of a block enters the channel, TOP terminates channel output. After the execution of this instruction, the following occurs. When the channel buffer delivers the last character to the peripheral device, the buffer disconnects.

TOP always terminates a non-interlaced channel output operation. It may be used with all communication channels if the particular function selected is terminal function 11 but no further data output is required (see Terminal Functions, this section).

Mnemonic	Terminate Output on Channel	Instruction
TOP 0	W	0 02 14000
TOP 1	Y	0 02 14100
TOP 2	C	2 02 14000
TOP 3	D	2 02 14100
TOP 4	E	0 06 14000
TOP 5	F	0 06 14100
TOP 6	G	2 06 14000
TOP 7	H	2 06 14100

Registers Affected: None Timing: 1

COMPATIBLE/EXTENDED INPUT/OUTPUT MODES

The termination of an I/O operation and the interrupts that may be associated with that termination fall into two classes: Compatible and Extended. The choice of one of these two "modes" of input/output operation determines how the system behaves when the termination of an I/O operation occurs.

As mentioned in Section III, Interrupt System, interrupts occurring at the same level (e.g., location 30, 31, etc.) can have different names (e.g., Count Equal Zero and End-of-Word). These names reflect the different I/O mode in operation when the interrupt occurs. The differences include the timing of interrupt occurrence relative to the I/O operation and type of interrupt requested.

The Compatible mode of operation for channels W, Y, C, D is directly compatible with the XDS 910 Computer mode of I/O operation. The types of interrupts that can be requested are the End-of-Word and End-of-Transmission interrupts.

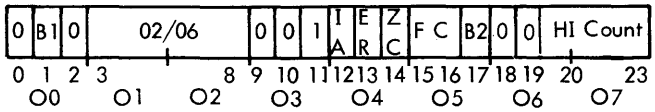
The Extended mode for all channels expands the I/O capabilities to include the "terminal functions" discussed below. The types of interrupts that can be requested are the Count Equal Zero and End-of-Record interrupts.

The I/O mode is selected in the Input/Output EOM (EOD) via bit 12, the Interrupt Arm bit. A 0-bit makes the system operate in the Compatible mode; a 1-bit sets the system in the Extended mode.

In particular, the Interrupt Arm (IA) bit determines whether any of the Extended functions operate; that is, a "0" in IA means that the other Extended mode controls, bits 13, 14, 15 and 16, have no effect.

INPUT/OUTPUT CLASS EOM/EOD

The Input/Output EOM (EOD) selects the I/O operation mode. When the Extended mode is selected, this EOM also selects (arms) which interrupts are to be operational and selects the desired terminal function. This EOM applies to Channels W, Y, C, and D. EOD applies to Channels E, F, G, and H.



Bit Designation	Octal Position	Octal Value	Function
0			Bit positions 0 and 2 are not used with this EOM.
B1	O0	2	Bit positions 1 and 17 specify the channel.
B2	O5	1	
02/06	O1-O2	02/06	Bit positions 3 through 8 contain 02/06, the instruction code for EOM/EOD.
01	O3	1	Bit positions 10 and 11 contain the EOM/EOD indicator for the Input/Output control mode.
IA	O4	4	Bit position 12 selects the mode of I/O operation. A "0" specifies the Compatible mode. The operation of bits 13, 14, 15, and 16 are disallowed. Channels W, Y, C and D operate in this mode which is completely XDS 920-compatible. If interrupts are required, the user enables the Interrupt System, thus enabling and arming the End-of-Word and End-of-Transmission interrupts. A "1" specifies the Extended mode. All channels can operate in this mode. This allows the use of bits 13, 14, 15, and 16. If interrupts are required, the user arms the associated ones by placing 1-bits in bit 13 and/or 14. The "terminal function" to be used is selected via bits 15 and 16.

Note: A 1-bit in 13 and/or 14 does the following:

1. Arms that interrupt during this complete I/O operation; disconnecting this channel disarms the interrupt.

2. Once armed by bits 13 and/or 14, the interrupt can be enabled or disabled by the Enable/Disable feature of the Interrupt System. If a channel generates an extended mode I/O interrupt while the system is disabled, the designated interrupt level goes to the Waiting state. When the program again enables the interrupt system, the interrupt goes to the Active state when its priority allows.

Direct Access Communication Channels operate only in the Extended mode; therefore, a DACC does not examine bit 12, but assumes it to be a 1. Note that with the complete omission of this second EOM, a DACC operates in the IORD or terminal function 00 mode.

Bit Designation	Octal Position	Octal Value	Function
ER		2	Bit position 13 controls the arming of the End-of-Record interrupt. A 1-bit arms the interrupt. A 0-bit disarms the interrupt.
ZC		1	Bit position 14 controls the arming of the Zero Word Count interrupt. A 1-bit arms the interrupt. A 0-bit disarms the interrupt.
FC	O5	0 2 4 6	Bit positions 15 and 16 specify the terminal condition function to be performed with the transmission. These are defined in the next topic in this section.
HI Count			Bit positions 20 through 23 contain the most significant four bits of the 15-bit word count. These positions specify a word count greater than 1023.

TERMINAL FUNCTIONS; EXTENDED MODE

A 2-bit function code in the Input/Output EOM (EOD) controls the termination of input/output operation in the extended mode. These functions are described below with the letter C representing the specified word count of the transmission.

		Bit Configuration	Octal Value
IORD	INPUT/OUTPUT OF A RECORD AND DISCONNECT	00	0

Input Read C words. If C equals zero before the End-of-Record is detected, the rest of the record is ignored. At the End-of-Record, the peripheral device is disconnected and the channel becomes inactive.

Output Write C words. When C equals zero, output is terminated (i.e.,the device is signaled that the last characters have been transmitted). When the peripheral device has generated the end of record and, if necessary, checked the validity of the record, it sends an End-of-Record response to the Channel buffer. When received by the buffer, the End-of-Record signal generates an End-of-Record interrupt (if armed) and disconnects the channel.

The line printer generates the End-of-Record response when it completes the printing of a line. If the printer encounters any print errors or faults, it sends a signal to the channel that sets the channel error indicator; this can occur since the printer has not disconnected from the channel. The IORD is useful when the program is to print several lines and the program is not otherwise to use the channel between lines. When the printer completes each line, it causes an End-of-Record interrupt (assumed to be armed), notifying the program that it can immediately transmit the next paper control instruction and the next line image.

The unbuffered card punch operates similarly. It generates the End-of-Record response after punching each row. If any faults occur during the punching of the entire card, the card punch sends a signal to the channel that sets the channel error indicator; this occurs after punching the last row (row 9).

NOTE: A program should not use IORD with devices that do not have End-of-Record conditions on input (e.g., typewriter) or generate End-of-Record responses upon output termination, (e.g., devices such as the paper tape punch and typewriter). These devices do terminate output but give the program no indication when they receive the last characters.

		Bit Configuration	Octal Value
IOSD	INPUT/OUTPUT UNTIL SIGNAL THEN DISCONNECT	01	2

Input Read C words. When C equals zero or when the End-of-Record is encountered, the device is disconnected and the channel becomes inactive. If the channel disconnects because of a zero count, an EOR interrupt (if armed) will be generated in addition to the count equal zero interrupt. If both are armed, C=0 will occur first.

Output Write C words. When C equals zero and when the last character has been transmitted, the channel disconnects the device and becomes inactive. If an End-of-Record signal is received before the count reaches zero, the channel will disconnect immediately.

NOTE: The IOSD is designed for use on devices which are normally operated on the basis of the word count only. Typewriters and paper tape devices are of this type, as are the printer and card punch when the user does not wish to stay connected until the operation is complete.

		Bit Configuration	Octal Value
IORP	INPUT/OUTPUT OF A RECORD AND PROCEED	10	4

Input Read C words. If the channel counts C down to zero before the peripheral device encounters the End-of-Record (EOR), the channel ignores the rest of the record (to the End-of-Record). When the peripheral device sends the End-of-Record signal to the channel, the channel sets its End-of-Record Indicator; this signal sets the End-of-Record interrupt (if armed). The channel does not disconnect. The channel is now in an "Inter-record" condition.

When the peripheral device is magnetic tape, the tape continues to move when the tape handler encounters the End-of-Record. The End-of-Record occurs when the tape read-heads encounter tape gap; this also causes a Tape Gap signal to "come high". If the program executes a new read tape or scan tape EOM during the inter-gap time (approximately .75 millisecond while the Tape Gap signal is high), the tape remains in motion and proceeds to read or scan the next record. If the program executes no such EOM before the Tape Gap signal drops, the channel disconnects and the tape comes to a stop. No additional interrupt occurs. This is the only condition that causes a channel to disconnect automatically.

All other input devices remain connected until the program takes further action. The paper tape reader remains in motion; the program should issue a "disconnect channel" instruction if the program is not reading any more tape. To proceed after the End-of-Record occurs, the program first executes a Buffer Control mode EOM to re-initialize the Channel Unit Address Register and then reloads the interlace portion of the channel (the program can alert the Interlace via the Buffer Control EOM). Otherwise, the channel immediately terminates any attempt to use its interlace portion since the channel is aware that it is still active and in the End-of-Record condition. When the program continues from an Inter-record condition,the program should use an extended mode terminal function. An IORP should not be used to read devices which do not have EOR signals (e.g., the typewriter and paper tape punch).

Output — Write C words. When the channel interlace counts C down to zero, the Interlace notifies the channel buffer that it has received the last word that is to be output; when the buffer outputs this last word, it sends a signal to the connected peripheral device indicating that the device has the last word now. When the peripheral device "receives, outputs and checks the validity of" this last word, it sends an End-of-Record response to the channel buffer. When received by the buffer, the End-of-Record signal generates an End-of-Record interrupt (if armed) and sets the Inter-record indicator; the channel does not disconnect.

When the peripheral device is magnetic tape, the tape continues to move after it signals End-of-Record. As in reading tape, the signal causes the Tape Gap signal to come high. If the program executes a new write tape or erase tape EOM during the inter-gap time (approximately one millisecond), the tape remains in motion and proceeds to write or erase a new record. If the program executes no such EOM before the Tape Gap signal drops, the channel disconnects and the tape comes to a stop. No interrupt occurs at this time. This is the only condition which causes a channel to disconnect automatically.

To proceed after the End-of-Record occurs, the program first executes a Buffer Control mode EOM to re-initialize the Channel Unit Address Register and then reloads the interlace portion of the channel (the program can alert the Interlace via the Buffer Control EOM). Otherwise, the channel immediately terminates any attempt to use its interlace portion, since the channel is aware that it is still active and in the End-of-Record condition. When the program continues from an Inter-record condition, the program should use an extended mode terminal function.

A program should not use IORP with devices that do not generate End-of-Record responses upon output termination; such devices are paper tape and typewriter. These devices do terminate output but give the program no indication when they receive the last characters.

The IORP should also not be used with the printer and card punch since these devices expect the channel to disconnect after they send EOR.

IOSP — INPUT/OUTPUT UNTIL SIGNAL THEN PROCEED

Bit Configuration	Octal Value
11	6

Input — Read C words. If the channel counts C down to zero before the peripheral device encounters the End-of-Record, the channel generates a Count Equals Zero interrupt (if armed). The program should reload the interlace portion of the channel to continue reading the record. As far as the peripheral device knows, nothing happens at this time. Failure to reload the Interlace before the peripheral device sends enough characters to overfill the channel buffer causes a rate error; this sets the channel error indicator.

When the peripheral device encounters the End-of-Record, IOSP operates identically like the IORP command.

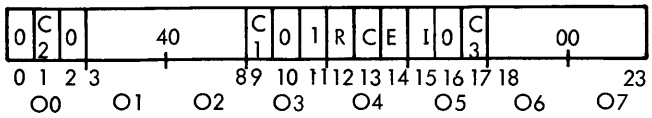
Output — Write C words. When the channel counts C down to zero, the channel generates a Count Equals Zero interrupt (if armed); the channel does not terminate output. The program should reload the interlace portion of the channel to continue writing in the same record. Failure to reload the Interlace before the buffer transmits all of the characters in its registers and before the peripheral device requests the next character from the buffer results in a rate error; this sets the channel error indicator.

If the program executes a TERMINATE OUTPUT (TOP) instruction after the channel has counted C down to zero, the channel terminates the output and operates identically like the IORP from this point on.

CHANNEL AND DEVICE SKS

The Channel and Device Test mode SKIP IF SIGNAL NOT SET (SKS) tests the indicators in a channel as well as devices attached to it. To test the channel, use unit address 00. The instruction format is:

CHANNEL TESTS



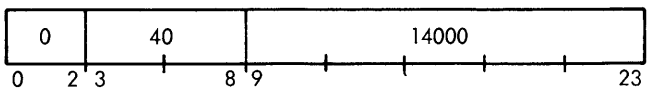
Bit Designation	Octal Position	Octal Value	Function
40	O1-O2	40	Bit positions 3 through 8 contain 40, the SKS instruction code.
01	O3	1	Bit positions 10 and 11 contain the mode selection.
C1 C2 C3	O3 O0 O5	4 2 1	Bits C1, C2, C3, used as an octal address, specify the channel to be tested. Channel W is 0, Channel Y is 1, and so on, Channel H being 7.
R			Test for ready. A 1-bit selects the test. Skip if Ready or Inactive.
C	O4	2	Test if indicator for Word Count Equal to Zero is set. A 1-bit selects the test. Skip if word count zero.
E		1	Test for error indicator reset. A 1-bit selects the test. Skip if no error.
I	O5	4	Test for Inter-record condition.
00			Bit positions 18 through 23 are zero to specify a channel test. Each of these tests causes a skip when the test condition is true.

STANDARD SKS INSTRUCTIONS

Several SKS function configurations have standard uses. These have standard, assembler-type mnemonics and are always used as shown.

CAT CHANNEL ACTIVE TEST;

SKIP IF CHANNEL NOT ACTIVE



If the channel is ready to accept a new input/output instruction, the computer skips the next instruction in sequence and executes the following instruction. If the channel is active, or in the process of disconnecting a peripheral unit, the computer executes the next instruction in sequence.

Mnemonic	Channel Active Test	Instruction
CAT 0	W	0 40 14000
CAT 1	Y	0 40 14100
CAT 2	C	2 40 14000
CAT 3	D	2 40 14100
CAT 4	E	0 40 54000
CAT 5	F	0 40 54100
CAT 6	G	2 40 54000
CAT 7	H	2 40 54100

Registers Affected: None Timing: 2, if no skip
3, if skip

The following XDS 910-compatible instructions make the identical test as the above instructions on Channels W and Y;

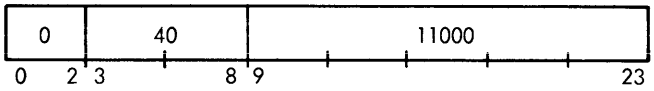
BRTW	0 40 21000	W BUFFER READY TEST
BRTY	0 40 22000	Y BUFFER READY TEST

Registers Affected: None Timing: 1, if no skip
2, if skip

The indicator that CAT tests is reset only by the next EOM that connects and alerts the same channel.

CET CHANNEL ERROR TEST;

SKIP IF NO ERROR ON CHANNEL



CET tests the error indicator in the channel for being in the set condition. If the error indicator has not been set, the computer skips the next instruction in sequence and executes the following instruction. If the error indicator has been set, the computer executes the next instruction in sequence.

Mnemonic	Channel Error Test	Instruction
CET 0	W	0 40 11000
CET 1	Y	0 40 11100
CET 2	C	2 40 11000
CET 3	D	2 40 11100
CET 4	E	0 40 51000
CET 5	F	0 40 51100
CET 6	G	2 40 51000
CET 7	H	2 40 51100

Registers Affected: None Timing: 2, if no skip
3, if skip

The following XDS 910-compatible instructions make the identical test of Channels W and Y:

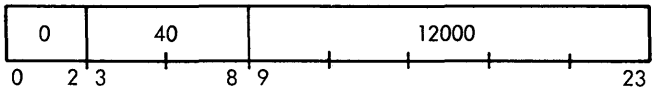
BETY	0 40 20020	Y BUFFER ERROR TEST
BETW	0 40 20010	W BUFFER ERROR TEST

Registers Affected: None Timing: 1, if no skip
2, if skip

The indicator that CET tests is reset only by the next EOM that connects and alerts the same channel.

CZT CHANNEL ZERO COUNT TEST ;

SKIP IF CHANNEL WORD COUNT IS ZERO



CZT tests whether the contents of the Word Count Register in the channel have been reduced to zero. If the contents of WCR are zero, the computer skips the next instruction in sequence and executes the following instruction. If the contents of the WCR are non-zero, the computer executes the next instruction in sequence.

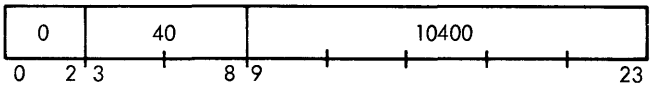
Mnemonic	Channel Zero Count Test	Instruction
CZT 0	W	0 40 12000
CZT 1	Y	0 40 12100
CZT 2	C	2 40 12000
CZT 3	D	2 40 12100
CZT 4	E	0 40 52000
CZT 5	F	0 40 52100
CZT 6	G	2 40 52000
CZT 7	H	2 40 52100

Registers Affected: None Timing: 2, if no skip
3, if skip

The indicator that CZT tests is reset only by a POT instruction to set up the word count and data address in the same channel.

CIT CHANNEL INTER-RECORD TEST ;

SKIP IF INTER-RECORD INDICATOR IS SET



CIT tests the Inter-record indicator in the selected channel. If the Inter-record indicator is set, the computer skips the next instruction in sequence and executes the following instruction. If the indicator is reset, the computer executes the next instruction in sequence. (See IORP instruction description under TERMINAL FUNCTIONS for Inter-record definition).

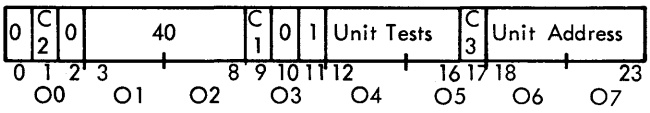
Mnemonic	Channel Inter-Record Test	Instruction
CIT 0	W	0 40 10400
CIT 1	Y	0 40 10500
CIT 2	C	2 40 10400
CIT 3	D	2 40 10500
CIT 4	E	0 40 50400
CIT 5	F	0 40 50500
CIT 6	G	2 40 50400
CIT 7	H	2 40 50500

Registers Affected: None Timing: 2, if no skip
3, if skip

The Inter-record indicator is set only during extended mode operation when using a Proceed Function; the indicator is set for an inter-record or zero count condition. The indicator is reset by the next alert and connect EOM.

DEVICE TESTS

The SKIP IF SIGNAL NOT SET (SKS) below, used in the Channel and Device Test mode, tests the condition of the peripheral devices in the system directly. The peripheral device sections contain the individual instruction descriptions.



Bit Designation	Octal Position	Octal Value	
C1	O3	4	Bit positions 9, 1, and 17 are used as an octal digit to specify the channel.
C2	O0	2	Channel W is 0, Channel Y is 1, and so on.
C3	O5	1	
40	O1-O2	40	Bit positions 3 through 8 contain the SKS instruction code 40.
01	O3	1	Bit positions 10 and 11 contain the mode selection.
Unit Tests	O4-O5		Bit positions 12 through 16 select the particular test and are system dependent.
Unit Address	O6-O7		Bit positions 18 through 23 specify the unit address.

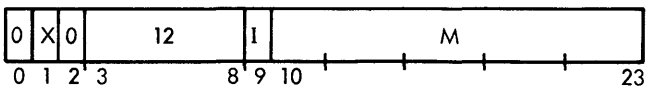
SINGLE-WORD DATA TRANSFER VIA CHANNELS W AND Y

INSTRUCTIONS

Channels W and Y can be programmed as single-word input/output buffers. Data transfer is performed under direct program control or with the aid of the interrupt system. Interlace is not used with these instructions.

The following two instructions perform data transfer using Channel W.

MIW MEMORY INTO CHANNEL W WHEN EMPTY

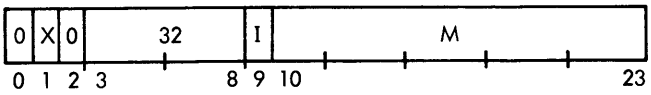


MIW transfers the contents of the effective memory location into the Channel W word buffer. If necessary, the central processor "hangs up" until the buffer is empty and ready to accept the data word.

The W buffer must be connected to the desired peripheral device by a previous "connect" EOM instruction that selects the buffer, the unit address, and all appropriate control functions.

Registers Affected: None Timing: 2 + wait

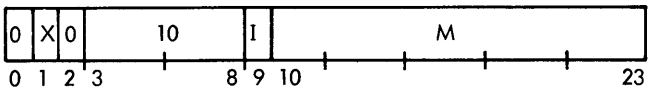
WIM CHANNEL W INTO MEMORY WHEN FULL



WIM transfers contents of the Channel W word buffer into the effective memory location. If necessary, the central processor "hangs up" until the buffer is full and ready to deliver the data word.

Registers Affected: M Timing: 3 + wait

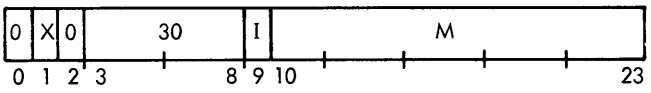
MIY MEMORY INTO CHANNEL Y WHEN EMPTY



MIY transfers the contents of the effective memory location into the Channel Y word buffer. If necessary, the central processor "hangs up" until the buffer is empty and ready to accept the data word.

Registers Affected: None Timing: 2 + wait

YIM CHANNEL Y INTO MEMORY WHEN FULL



YIM transfers the contents of the Channel Y word buffer into the effective memory location. If necessary, the central processor "hangs up" until the buffer is full and ready to deliver the data word.

Registers Affected: M Timing: 3 + wait

SINGLE-WORD OPERATIONS

The single-word buffer operations are used in two ways. Data words transfer between the channel and memory under direct program control. The "connect" EOM and the input or output channel instruction are in sequence and the computer "hangs up" until the buffer is ready to perform the transfer. This delay is usually due to buffer tie-up while the buffer is actively transmitting or receiving the previously requested data word.

Use of the priority interrupt system eliminates the tie-up of the central processor. The interrupt system allows the program to connect the device to be used in the transfer, to enable the interrupt, and then to continue processing in the main program. When the buffer is ready to receive from, or transfer to, memory, the End-of-Word interrupt to the corresponding interrupt location notifies the program that the buffer is Ready. A service routine entered via a BRANCH AND MARK PLACE (BRM) instruction in the appropriate interrupt location processes the interrupt. This routine contains the instruction (MIW or WIM, for example) that can execute immediately without computer tie-up.

During single-word operations, a parity error or incorrect timing error sets the buffer error indication in the channel. The incorrect timing error occurs when characters enter the buffer during input before the removal of the previous word; during output, buffer error indication occurs if characters are needed for output before the buffer receives the next word. The transmission does not terminate upon detection of any of these errors.

The interrupt system can detect an End-of-Record termination. During output, use of TERMINATE OUTPUT (TOP) after the final MIW (MIY) causes an interrupt to the appropriate End-of-Transmission location when that final data word has been processed by the buffer. This interrupt takes the place of the End-of-Word interrupt; the End-of-Transmission condition inhibits the End-of-Word interrupt. During input, the End-of-Transmission interrupt is sent to the End-of-Transmission location when the End-of-Record is detected. During input from devices which do not generate an End-of-Record, an EOM disconnects (DSC) the channel to terminate the transmission. This termination generates no End-of-Transmission interrupt.

EXAMPLE: WIM

This program reads a block of binary paper tape of any length, using the W buffer without interlace. There is an integral multiple of four characters in the block. This subroutine uses the End-of-Word and End-of-Transmission interrupts of the W buffer and reads data into memory beginning in a table at location 1024.

Location	Instruction	Address	Comments
H	PZE		This is an assembler instruction used to reserve the entry location by filling H with Zero.
	EIR		This instruction enables the interrupts. An End-of-Word interrupt will be received after each word is assembled in the W buffer.
	RPT	0, 1, 4	This instruction initiates the paper tape read on Channel W, four characters per word (see Paper Tape Input/ Output, this section).
	BRR	H	Return to the main program while awaiting the filling of the buffer with the first word read from tape.
TABLE	00002000		This location contains the input table starting address.

When the buffer fills with the first word, it generates the End-of-Word interrupt to location 31.

31	BRM	H1	This branch and mark instruction transfers to the read routine.
H1	PZE		Reserved entry location.
	WIM	*TABLE	This instruction transfers the contents of the W buffer into the location specified in the contents of location TABLE. The * indicates indirect addressing. If desired, indexing can be used.
	MIN	TABLE	This instruction increments the location for the next input word.
	BRU	*H1	This instruction transfers indirectly back to the main program to await the next End-of-Word interrupt and clears the currently active interrupt.

The above procedure continues until the end of the block. When gap is detected, the remaining character positions of the word being assembled in the buffer fill with zeros, and the End-of-Transmission interrupt to location 33 inhibits the End-of-Word interrupt.

33	BRM	H2	This instruction transfers and marks to location H2.
H2	PZE		This instruction reserves an entry location.
	BET	0	This instruction tests for the occurrence of an error during the input operation. If there were none, the next instruction is skipped and the following one is executed.
	BRM	ERR	This instruction transfers to an assumed error routine.
	BRU	*H2	This instruction returns to the main program. The read operation is complete.

Since in this example the input record has integral word-length, no characters are in the buffer when the End-of-Record is reached. If there are one, two, or three characters in the buffer when it detects the gap, an additional WIM has to be executed to place these characters into memory.

DIRECT PARALLEL INPUT/OUTPUT INSTRUCTIONS (PIN/BPI, POT/BPO)

Two instructions, PARALLEL OUTPUT (POT) and PARALLEL INPUT (PIN), permit any word in core memory to be presented in parallel at a connector; or, inversely, permit signals sent to a connector to be stored in any core memory location. The execution of a POT or PIN instruction causes a signal to be sent to the external device involved in the input/output operation. This signal notifies the device to send its data word as soon as it is operational. When the device becomes operational during a Read or PIN operation, it transmits a Ready signal to the central processor while at the same time presenting its data word. The computer places the received data word into a specified memory location without disturbing any arithmetic registers. The computer "hangs up" during the execution of PIN until it receives the Ready signal from the external device.

During the execution of a POT instruction, the central processor transmits a signal to the external device, alerting it to receive a data word. When the device becomes operational, it transmits a Ready signal to the central processor, which releases the data word to the external device. The computer "hangs up" during the execution of POT until it receives the Ready signal from the external device.

The block transfer forms of these instructions are BLOCK PARALLEL INPUT (BPI) and BLOCK PARALLEL OUTPUT (BPO). These instructions operate like PIN/POT except that they transfer many words instead of one word. The computer "hangs up" during a BPO/BPI execution except for preset I/O channels. The transfer rate, however, may be as high as one 24-bit word per memory cycle.

To initiate a block parallel transfer:

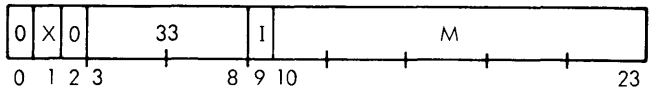
1. Load the number of words to be transferred minus one, or the maximum acceptable for input, into the A Register.

2. Alert the external device by means of an EOM instruction.

3. Start the block transfer with a BPO or BPI instruction.

For special system requirements, POT and PIN can be used effectively with other instructions to produce high-speed, synchronized, data transfers between the central processor and external devices without the use of a communication channel. Selective input/output to and from these devices is accomplished by preceding POT or PIN with an EOM to alert the desired device by specific address. By preceding the POT or PIN with an SKS, the Ready signal of the special device can be tested after the execution of the EOM but prior to execution of the parallel transfer instruction; a possible computer "hang-up" can thereby be avoided. If the Ready signal from the external device sets one of the priority interrupts, parallel input/output operation can occur as soon as the external device is able to transmit or receive. Since the Ready signal initiating the interrupt is present through the POT or PIN execution, no computer "hang-up" occurs.

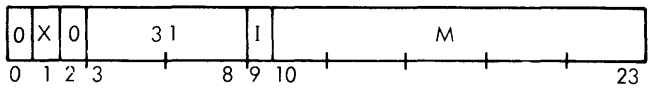
PIN PARALLEL INPUT



PIN stores the contents of 24 input lines in parallel in the effective memory location.

Registers Affected: M Timing: 4 + wait

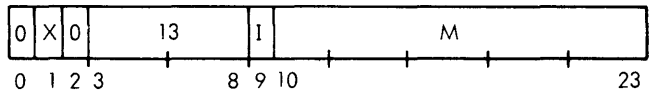
BPI BLOCK PARALLEL INPUT



BPI transfers N words from the 24 input lines in parallel into sequential locations, starting with the effective memory location. The A Register contains the word count N minus one; BPI can input up to 16,384 words per execution. The effective address goes to the S Register and the first word is transferred in. After each transfer, (S) is incremented and (A) is decremented. When (A) equal zero or an End signal is received from the external device, the execution of the instruction is completed. BPI can not be interrupted until the instruction following it has been executed.

Registers Affected: M to M + (A), A Timing: 4+ N + Wait

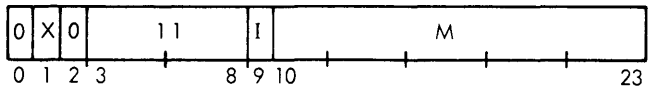
POT PARALLEL OUTPUT



POT transmits the contents of the effective memory location in parallel to 24 output lines of an external device.

Registers Affected: None Timing: 3 + wait

BPO BLOCK PARALLEL OUTPUT



BPO transfers the contents of N sequential locations in parallel to 24 output lines of an external device, starting with the effective memory location. The A Register contains the word count N minus one; BPO can output up to 16,384 words per execution. The effective address goes to the S Register and the first word is transferred out. After each transfer, (S) are incremented and (A) are decremented. When (A) equal zero or an End signal is received from the external device, execution of the instruction is completed. BPO can not be interrupted until the instruction following it has been executed.

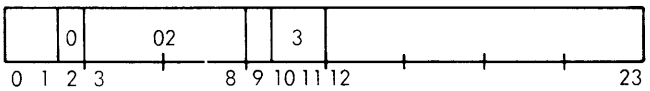
Registers Affected: A Timing 3 + N + Wait

SINGLE - BIT INPUT/OUTPUT

Operating in the System mode, the two instructions, ENERGIZE OUTPUT M (EOM) and SKIP IF SIGNAL NOT SET (SKS), provide single-bit input/output transmissions.

Execution of a System Mode EOM causes a 1.75-microsecond signal to be transmitted to one of a possible 32,768 signal destinations. The System Mode EOM format is:

EOM ENERGIZE OUTPUT M



Bit positions 3 through 8 contain the EOM instruction code, 02.

Bit positions 10 and 11 contain the System Mode Indicator.

Bit positions 12 through 23 contain the 12-bit address field that specifies the special system destinations.

Bit position 2 contains 0.

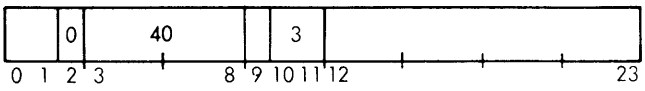
Bit positions 0, 1, and 9 are reserved for special system address bits.

Registers Affected: None Timing: 1

Execution of a System Test Mode SKS causes a 14-bit address to be presented to the collection of special system devices. If the addressed external device is supplying a set signal to the central processor, the computer executes the next instruction in sequence from the SKS. If no signal is set, the computer skips the next instruction in sequence and executes the following instruction.

The SKS System Test format, which has each corresponding bit-set identical to the System EOM format, is:

SKS SKIP IF SIGNAL NOT SET



Registers Affected: None Timing: 2, if no skip
3, if skip

COMMUNICATION CHANNEL PROGRAMMING

Extended Mode

Programming a block transmission of data using the full facility of the input/output system includes these instructions: EOM (Alert), EOM (I/O Control), and POT (PARALLEL OUTPUT).

A sample sequence of instructions from a magnetic tape read operation follows. The octal configuration of each instruction is given.

Location	Instruction	Comments
1000	0 02 42610	This EOM specifies Channel W, bits 1 and 17, (no 2 in O0, no 1 in O5) alerts the interlace, bit 9 (a 4 in O3) is in the Buffer Control mode (no 2 or 1 in O3) specifies forward direction of tape motion with no leader and BCD character format, bits 12, 13, 14, (a 4, no 2 and no 1 in O4) selects four characters per word assembly mode, bits 15, 16, (a 6 in O5) and connects the unit function address 10 to read tape number 0.
1001	0 02 15001	This EOM is in the Input/Output Control mode, selects the channel interrupt mode, bit 12,(a 4 in O4) disarms the End-of-Record interrupt, bit 13, (no 2 in O4) arms the Zero Count interrupt, bit 14, (a 1 in O4) selects terminal function 00, bits 15, 16, (no 4 or 2 in O5) and specifies high order word count of 01 (bits 20 through 23).
1002	0 31 01020	This POT transmits to the channel the contents of location 1020. The location contains the word count and the starting location for data input.
1020	0 03 13500	Bit positions 0 through 9 of this location contain the low order 10 bits of the word count. Bit positions 10 through 23 contain the 14 bits of the starting address

The channel assembles the starting address from the EOM, bit 18, and from the word transmitted by the POT. In this sample, the starting address for the read operation is 13500_8. The word count is assembled from the same EOM, bits 20-23, and from the word transmitted by the POT. In this sample, the word count is 020068. This is assembled as follows. Bits 20 through 23 of the EOM in location 1001 are 00 01; bits 0 through 9 of the transmitted word are 0 000 000 110. Assembling these bits into one 14-bit count, 00 010 000 000 110, the word count becomes 020068.

These three instructions read one magnetic tape record of 2006-word length into memory starting at location 13500. When the word count equals zero during the transmission, an interrupt is sent to Channel W interrupt level 31. Any further information is ignored and when the tape reaches the End-of-Record, it is stopped, disconnected, and the channel becomes inactive.

Compatible Mode

In the Compatible mode of channel operation, the second EOM may be omitted if the word count is less than 1023 (1777_8) words and the starting addresses are less than 16383 (37777_8). The End-of-Word and End-of-Transmission interrupts are used when interrupts are desired. They can be armed/enabled or disarmed/disabled by the Enable/Disable instructions. Since the Extended input/output functions that are specified in the second EOM cannot be used, the latter two interrupts are used along with SKS tests to determine the terminal conditions of input/output transmissions. This I/O mode operates only for Channels W, Y, C, D.

A sample line print sequence programmed in the compatible mode follows:

Location	Instruction	Comments
1000	0 02 42660	This EOM specifies Channel W, alerts the interlace, specifies four characters per word, and connects the unit function address 60 for Printer Number 1.
1001	0 31 01030	This POT transmits to the channel the contents of location 1030.
1030	0 20 42000	The location contains the word count and the starting address for output. Bits 0 through 9 contain the word count of 41_8; the starting address is 2000_8.

Since the input/output facility is less comprehensive in this mode, the user should be aware of the terminal conditions that will occur. For output, the mode is equivalent to functions 00 and 01; that is, when C words have been transmitted, the output terminates, and when the last character has been sent, the device disconnects. If the interrupt system is enabled, an End-of-Transmission interrupt to location 33 occurs when the device disconnects. No interrupt occurs on level 31.

For input, this mode is equivalent to functions 00 and 01 if the End-of-Record is encountered before the word count is reduced to zero. If the word count is reduced to zero before the End-of-Record is encountered, the interlace portion of the channel disengages all control of the channel buffer. The buffer continues to assemble characters until a word is completed. If the interrupt system is enabled, the buffer then generates an End-of-Word interrupt on level 31. The program has approximately 1.5 character times to reload the interlace if reading is to continue. On Channel W (or Y) the contents of the buffer at this time can be stored with the WIM (YIM) instruction.

If this form (EOM, POT) is used with Channels E through H, the Terminal Function mode is 00 with no interrupts armed.

This mode of channel operation should generally not be used on input unless the record length of the input records is fixed and known.

CONTROL CONSOLE

The basic XDS 925 Computer System provides a console for operator control. This console connects directly to the central processor, contains switches for operation, and displays the contents of operational registers.

DISPLAYS

The registers displayed on the console directly reflect the contents of the hardware registers. If the operator changes or clears a display, the contents of the actual register also change identically.

PROGRAM LOCATION

The program counter is a 14-bit register that contains the location of the next instruction to be executed. The programmer may change the counter by inserting a BRU into the Instruction Register and executing it. When the computer is in the IDLE state, this register displays the location of the instruction to be executed next.

INPUT-OUTPUT

The UNIT lights contain the unit address of the peripheral device currently connected to the selected channel.

The ERROR light reflects the status of the channel error indicator. Setting the I/O DISPLAY SELECT thumbwheel switch selects the channel to be displayed.

HALT

The HALT light is on whenever the computer executes an HLT instruction while in the RUN position. To clear this indicator, set the RUN-IDLE-STEP switch to IDLE.

OVERFLOW

This display shows the status of the Overflow Indicator.

REGISTER DISPLAY

This display consists of 24 binary indicators with a clear button for the entire register and a set button for each indicator. The REGISTER thumbwheel switch selects the internal register whose contents are to be displayed. The selectable registers are:

C	C Register, which contains the full instruction immediately prior to its execution
A	A Register
B	B Register
X	Index Register X

To change the contents of the selected register, press the indicator button(s) in the corresponding bit positions. The computer must be in the IDLE state and the register previously cleared. Pressing a button places a 1-bit into the selected position of the register.

MEMORY PARITY

If an operand or instruction access from memory encounters a parity error, this light turns on. If this switch is set to the HALT position, the computer will halt; if the switch is set to CONTINUE, the computer ignores the parity error, clears the indicator, turns off the light, and proceeds with the program.

INTERRUPT ENABLED

The INTERRUPT ENABLED light is on whenever the interrupt system is enabled.

SWITCHES

POWER Switch

The POWER switch turns the computer system power on or off. When power is on, the switch is lit.

FILL

The operator has the option of four input media to initially load or "fill" the computer. The pair of three-position, spring-loaded, center-return, toggle FILL switches are labeled: PAPER TAPE - MAG TAPE and CARDS-DRUM. For example, to select and initiate filling from paper tape on Channel W, set the first toggle switch to PAPER TAPE and release.

The fill procedure is:

a) Set up the selected input device with the input program. The initial portion of the program contains the "bootstrap" (the short load program).

b) Set the RUN-IDLE-STEP switch in the IDLE position.

c) Press the START switch.

d) Set the RUN-IDLE-STEP Switch in the RUN position.

e) Set one of the FILL switches. This will cause a WIM2 (03200002) instruction to be inserted into the Instruction Register and will load the Index Register with 7777777 1. Depending on which switch is pressed, activation of one of the following four devices on Channel W will occur.

Paper Tape Reader No. 1 - Unit Address 04

Card Reader No. 1 - Unit Address 06

Rapid Access Device No. 1 – Unit Address 26

Magnetic Tape Unit No. 0 - Unit Address 10

The FILL switch also prepares the channel to operate in the forward, binary, four characters per word mode.

A "bootstrap" program must be in position to be read as the first input from the device. A typical bootstrap program is:

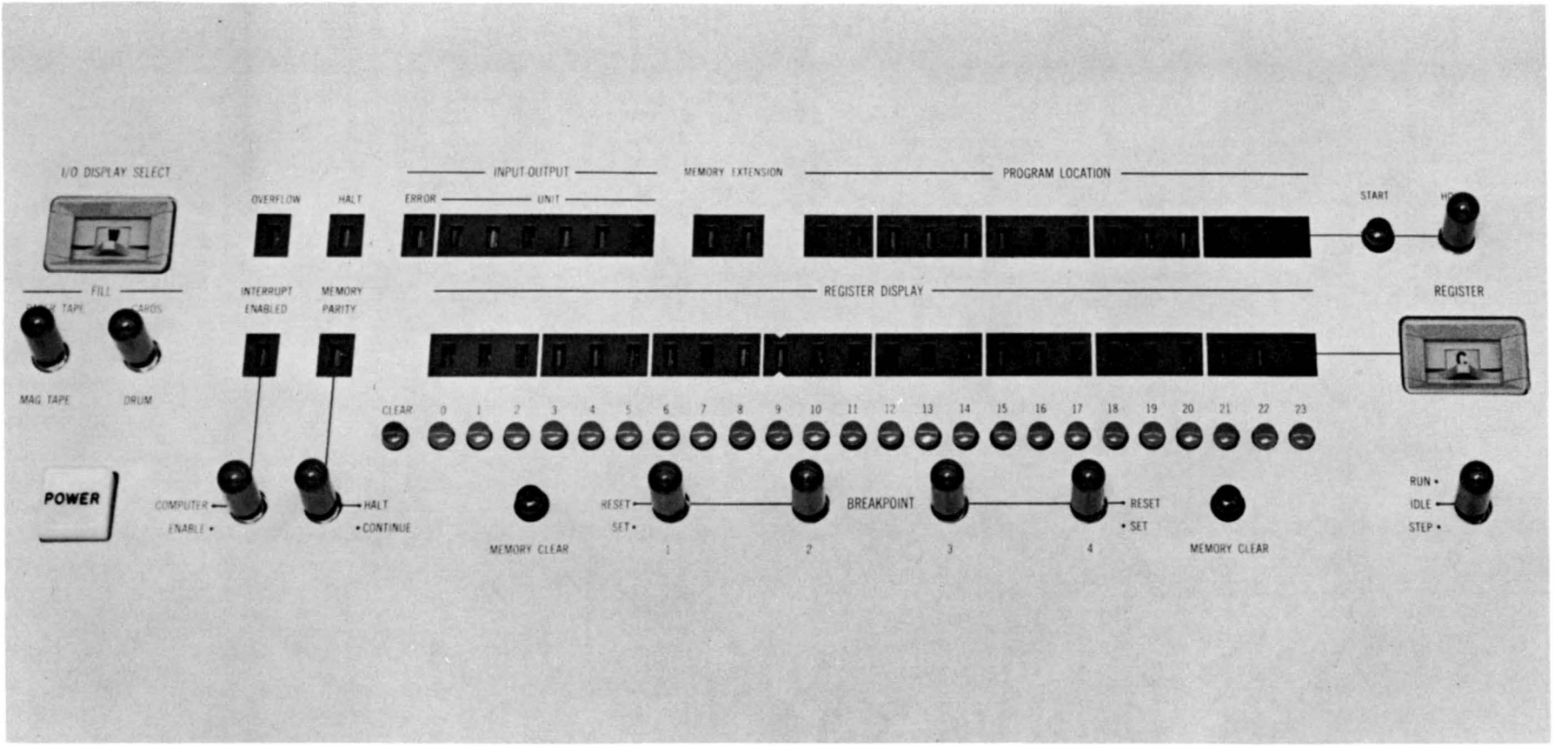


Figure 4-3. XDS 925 Control Panel

Location	Instruction	Address
00002	WIM	00012, 2
00003	BRX	00002
00004	LDX	00011
00005	WIM	00000, 2
00006	SKS	021000 Buffer Ready
00007	BRX	00005
00010	BRU	BEGIN
00011	OCT	Starting address with indirect address "tag"

The WIM 00002 instruction that is forced into the Instruction Register stores the first word of the "bootstrap" program in location 2. The computer then executes the contents of location 2. The Index Register, which contains -7, modifies the WIM in 2. The effective address of the WIM is then 00003 so that the second word is stored in 3. This word is a BRX back to the WIM.

These two instructions then load the remainder of the "bootstrap" program. The remaining six words can be those needed for the specific loading that is to be done. The one shown loads a record of any length. The Buffer Ready test in location 6 skips when the End-of-Record has been reached. In "bootstrapping" from paper tape or magnetic tape, the record may be of any length. From cards, the record is 40 words.

RUN-IDLE-STEP Switch

This is a three-position, toggle switch with two stationary positions and a spring-loaded, momentary position in STEP. In the RUN position, computation occurs at machine speed. In the IDLE position, the computer idles immediately after an instruction has been read from memory. If the REGISTER switch is in the "C" position, the REGISTER DISPLAY shows the complete instruction. In the STEP position, the computer executes the instruction and returns to the Idle state. Release the switch to the IDLE position before performing another step.

HOLD Switch

When the HOLD switch is lit, the current contents of the program counter are held. Instructions inserted into the C Register and executed do not step the program counter (i.e., it is inhibited from counting).

START Switch

This switch initializes the control section of the computer. It resets all channels and the OVERFLOW indicator, clears the P Register, the MEMORY PARITY error indicator, and all interrupt levels and disables the interrupt system; and places a zero in the displayed register. For proper operation, the RUN-IDLE-STEP switch must be in the IDLE and the REGISTER select switch at C when pressing this switch.

REGISTER Select Switch

This four-position, thumbwheel switch selects the register to be shown on the Register Display lights.

I/O DISPLAY SELECT Switch

This eight-position, thumbwheel switch selects the channel from which the unit address and error indicator are displayed in the INPUT-OUTPUT lights.

INTERRUPT ENABLED Switch

If this switch is in the COMPUTER position, the Interrupt System may be enabled or disabled under program control. Placing the switch in the ENABLE position enables the Interrupt System regardless of program operations. The switch is stationary in the COMPUTER position and momentary in the ENABLE position.

MEMORY PARITY Switch

If this switch is in the HALT position, the computer enters an Idle state whenever a memory parity error occurs. If this switch is in the CONTINUE position, the computer does not change state when memory parity occurs.

BREAKPOINT Switches

The program may detect the status of these four switches by using a breakpoint test. The switches labeled RESET and SET, control pre-determined options within the program.

MEMORY CLEAR Switches

Operating both of these switches simultaneously clears all of memory to zero.

INPUT/OUTPUT TYPEWRITER

The control console contains an electric, input/output typewriter for operator control, error or status messages, and similar functions. The Typewriter is connected to Channel W, has the input unit address 01, and the output unit address 41. Appendix A lists the typewriter codes.

The typewriter control instructions follow. These sample instructions use typewriter Number 1 on Channel W with four characters per word mode.

```
RKB  0, 1, 4  READ KEYBOARD W1,
                4 characters/word              0 20 00601
```

This instruction alerts Channel W and connects Typewriter Number 1 to it. RKB prepares the channel to read input from the keyboard. It also lights the input indicator on the typewriter.

```
TYP  0, 1, 4  WRITE TYPEWRITER W1,
                4 characters/word              0 02 00641
```

This instruction alerts Channel W and connects it to Typewriter Number 1. TYP prepares the channel to write output to the typewriter.

PROGRAMMING EXAMPLES

These examples present a straightforward sample of reading and writing with the typewriter under program control.

EXAMPLE: Typewriter Output

This routine causes the following message

ASSEMBLY DONE
ENTER NEW PROGRAM

to be typed out under program control. The computer stores the internal codes for these characters in memory beginning in location 2000. The routine inserts the carriage return code, 52, and the space code, 12, where needed and requests End-of-Record interrupt. It is written as a closed subroutine using interrupts, and uses Channel W and Typewriter Number 1.

Location	Instruction	Address	Comments
1000	PZE		This instruction is an assembler instruction, used here as a convenient way to reserve the entry location for subroutine use.
	CLR		This clears the A and B Registers.
	STA	SWICH	This clears the location called SWICH. SWICH later indicates to the main program that output is complete.
	TYP	*0, 1, 4	This instruction connects Typewriter Number 1 to Channel W for output, specifies four characters per word mode, and alerts Channel W interlace. The instruction is an EOM with octal configuration, 0 02 42641.
	EXU	WRITE	This instruction causes the Input/Output EOM in location WRITE to be executed.
	POT	WRITE + 1	This instruction sends the word count and starting address in WRITE + 1 to the channel.
	BRR	1000	This instruction branches back to the main program.
WRITE	EOM 00403720	16200	This EOM specifies terminal output function code 01 (IOSD) and the End-of-Record interrupt. The word in WRITE + 1 specifies that eight words will output from memory beginning in location 2000. According to terminal function 01, when the word count equals zero during the transmission, the output terminates, and when the last character is out, the device disconnects; at this time, the interrupt occurs.

The computer processes the main program while the channel performs the output operation. When finished with the output, an interrupt to interrupt level 33, the End-of-Record location for Channel W, occurs.

33	BRM	OKAY	This instruction, placed in location 33, branches and marks to location OKAY elsewhere in memory.
OKAY	PZE		This instruction saves the entry location.
	MIN	SWICH	This instruction increments location SWICH as an indicator for the main program.
	BRU	* OKAY	This instruction branches to the main program and clears the active interrupt.

This is the internal code for the output message:

	A	S	S	E	M	B	L	Y	Sp	D	O	N	E	C/R	E	N
2000	21	62	62	25	44	22	43	70	12	24	46	45	25	52	25	45
					N	E	W	Sp	P	R	O	G	R	A	M	Sp
2004					45	25	66	12	47	51	46	27	51	21	44	12

EXAMPLE: Typewriter Input

The operator requests control to input four control characters. The subroutine is assumed to have been entered under program control. There is no request for terminal interrupts in this example.

Location	Instruction	Address	Comments
INPUT	PZE		This instruction saves the entry location.
	RKB	*0, 1, 4	This instruction connects Channel W to Typewriter Number 1, specifies the four characters per word mode, and alerts the interlace. The input request light is lit. The octal configuration of the instruction is 0 02 42601. The asterisk prefixed to the address of read and write controlling EOM instructions indicates the setting of the interlace alert bit (9).
	EXU	CHARS	This instruction executes the instruction at location CHARS.
	POT	CHARS + 1	This instruction transmits to the channel the word count and starting address.
	CAT	0	This instruction tests for channel not active. If the channel is active when the computer executes CAT, it executes the next instruction in sequence. If the channel is inactive, the computer skips the next instruction and executes the following one.
	BRU	$-1	This instruction branches to the CAT instruction. The dollar sign and accompanying signed integer in the address field is an assembler declaration for the indicated number of locations prior to or following the current one. Plus indicates following.
	BRU	CHECK	This instruction branches to an assumed routine to determine what characters were typed in.
CHARS	EOM 00047640	14200	This EOM specifies terminal input function 01 and no interrupt at the end of transmission. The word in CHARS +1 specifies that one word can be input into location 4000. Only one word is accepted before the channel disconnects and goes inactive. The Count Equals Zero causes channel disconnect.

PERIPHERAL EQUIPMENT DESCRIPTION

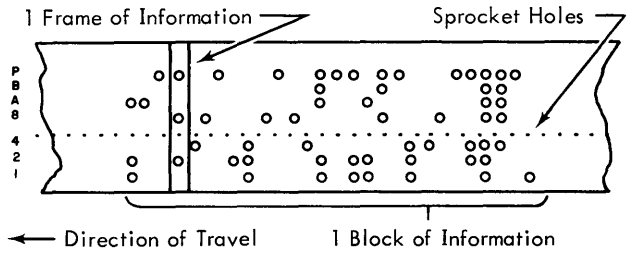
Communication channels facilitate a wide range of input/output operations. Combinations of input/output functions can perform Scatter-Read and Gather-Write operations. A channel may read many records into one contiguous area of memory, or skip portions of records and read subsequent portions.

This section describes some of the input/output devices, available in the computer system and explains their use.

PAPER TAPE INPUT/OUTPUT

Format

Paper tape used with the computer is one-inch wide, affording space for eight data holes and a sprocket hole in each frame of information. There are ten frames per inch of paper tape. Six hole positions contain information, one contains the odd parity check, and the eighth is unused.



The organization of information on the tape is in blocks. A block is a group of frames set off by a gap of at least one blank frame (in which only the sprocket hole is punched) at either end. Blocks may be of variable lengths.

For some operations a tape may consist of only one block, such as a source language tape prepared off-line. In this case, the program need not read the entire block at one time, but may stop the reader between frames, and then start again to read the remainder or another portion of the block.

Reading

All channel functions may be used in reading paper tape. An input/output function that terminates because of a zero count stops the tape between frames. A subsequent paper tape "read" starts the reader and allows the next frame to be read. An input/output function that terminates because of gap (End-of-Record) stops the tape after the first blank frame of the gap. When the tape starts, the tape reader ignores any leading blank frames. After reading information from the tape, the reader recognizes a blank frame as gap and signals the channel with an End-of-Record indication.

Punching

When a channel addresses the paper tape punch, the punch motor also starts (if not already on). If the punch instruction so indicates, the punch unit punches a segment of leader (gap, or blank frames). Bit position 13 of a Channel EOM or EOD instruction, which addresses the punch, contains a "0" to punch leader; bit position 13 contains a "1" to punch without leader.

EXAMPLES:

This instruction

PPT 0, 1, 1

prepares the punch on Channel W to punch without leader. It sets the channel to operate with one character per word.

This instruction

PTL 0, 1, 4 0 02 00644

prepares the punch and produces about 12 frames of leader. It sets the channel to operate with four characters per word.

No channel terminal function produces End-of-Record gap after punching a block. The EOM instruction that addresses the punch can only generate gap.

The punch operates at 60 characters per second, asynchronously. If the channel does not supply characters to the punch fast enough for operation at 60 cps, the punch waits for each character, losing no data and making no errors.

Programming

There are no status tests for the Paper Tape Reader or Punch, that is, they are always ready for operation. When a channel addresses either device, the device starts to send or accept data within approximately one character time. The reader and punch operate only in the binary mode and the forward direction; they ignore any different mode specified, and use the forward-binary mode. Unit addresses of 04 and 05 are for Paper Tape Readers 1 and 2, respectively, and unit addresses 44 and 45 are for Paper Tape Punches 1 and 2.

Paper Tape Instructions

The following instructions use Channel W, Paper Tape Number 1, with four characters per word format.

RPT 0, 1, 4 READ PAPER TAPE 0 02 02604

RPT initiates a paper tape read operation on tape read station number 1 connected to Channel W in the four characters per word format.

PTL 0, 1, 4 PUNCH PAPER TAPE WITH LEADER 0 02 00644

PTL initiates a paper tape punch operation on tape punch station number 1 connected to Channel W in the four characters per word mode. It generates approximately twelve frames of leader preceding the first punched frame.

PPT 0, 1, 4 PUNCH PAPER TAPE WITH NO LEADER 0 02 02644

PPT initiates a paper tape punch operation on tape punch station number 1 connected to Channel W in the four characters per word format. It generates no leader preceding the first punched frame.

The desired EOM, POT combination follows each of these instructions to control the input/output of data.

EXAMPLE: Punch Paper Tape

This program punches one block of 20 words beginning in location 2000. A twelve-frame leader precedes the block. The routine is a closed subroutine that uses interrupts.

Location	Instruction	Address	Comments
1000	PZE		This instruction saves a place for the entry location.
	CLR		This instruction clears the A and B Registers.
	STA	WHERE	This instruction clears a switch location used as an indicator to the main program for completion of the punch operation.
	PTL	*0, 1, 4	This instruction connects Channel W to Paper Tape Punch Number 1, specifies four characters per word mode, and alerts the interlace. The instruction specifies leader to be punched, and if not already on, turns the punch motor on. The octal configuration of this EOM is 0 02 40644.
	EXU	PUN20	This instruction executes the I/O Control EOM that sets the interrupt and selects output function 00.
	POT	PUN20 + 1	This instruction transmits to the channel the word count and starting address of the transmission.
	BRR	1000	This instruction branches back to the main program.
PUN20	EOM 01202000	16000	The EOM specifies terminal output function 00 (IORD) and the End-of-Record interrupt. The word in PUN20 + 1 specifies 20 words of output from memory to the punch beginning at location 2000 (0120 is 024_8 shifted right one place; it is merged with 02000 to make the "POTted" control word). According to terminal output function 00, when the word count equals zero during the transmission, the output terminates. The last word has not been fully transmitted at this time. When it is and the output is complete, the channel disconnects and the interrupt occurs.

When the Count Equals Zero interrupt occurs:

Location	Instruction	Address	Comments
33	BRM	END	This instruction branches and marks to END.
END	PZE		This instruction saves a place for the entry location.
	MIN	WHERE	This instruction increments WHERE as a flag.
	BRU	* END	This instruction returns to the main program and clears the interrupt level.

EXAMPLE: Read Paper Tape

This program reads a block of 64 characters from paper tape. The routine uses the four characters per word mode, making the input 16 words. It turns the tape station on and requests a Count Equals Zero interrupt, level 31, for the operation on Channel W. The routine is a closed subroutine.

Location	Instruction	Address	Comments
1000	PZE		This instruction saves a place for the entry location.
	CLR		This instruction clears the A and B Registers.
	STA	SWICH	This instruction clears location SWICH used as an input-finished indicator.
	RPT	* 0, 1, 4	This instruction connects Paper Tape Reader Number 1 to Channel W, specifies the four characters per word mode, and alerts the interlace. The octal configuration of this EOM instruction is 0 02 42604.
	EXU	REED	This instruction executes the EOM at location REED.
	POT	REED + 1	This instruction transmits to the channel the word count and starting address.
	BRR	1000	This instruction branches back to the main program for processing while the input operation is in progress.
REED	EOM 01003720	15200	This EOM specifies terminal input function 01 (IOSD) and the Count Equals Zero interrupt. The word in REED + 1 specifies that input into memory begins in location 2000 and that 16 words will be read before the operation terminates. When the word count equals zero, the interrupt occurs. Then the channel disconnects. When the tape read operation is complete, the Count Equals Zero interrupt occurs at level 31.
31	BRM	FNISH	This instruction, in location 31 for this example, branches and marks to location FNISH.
FNISH	PZE		This instruction saves the entry location.
	MIN	SWICH	This instruction sets an input-finished switch for use by the main program.
	BRU	*FNISH	This instruction branches back to the main program and clears interrupt level 31 from the active state.

The programmer can make a test to the channel, CET, for parity error during the read operation before the BRU instruction.

Format

The computer uses 80-column cards in two formats. The card reader reads Hollerith-coded information from cards and transmits the corresponding XDS character codes to memory. In this mode, each column contains the equivalent of one 6-bit internal character. Appendix A-1 lists the character codes.

The card reader reads binary-coded information from the card with two 6-bit characters per column. In binary mode, two columns form a word. The top six rows (12-3) of column 1, for example, form the first character and the bottom six (4-9) the next character. The reader reads from column 1 to 80 in this top-bottom order. A single card holds 160 characters or 40 binary words.

Figure 4-4 shows the relation of Hollerith information on a card and in memory. Hollerith output to the punch is identically the reverse.

Reading

The card reader scans the card, column by column, starting with column one, and transmits either 80 or 160 characters to the channel depending on the mode of operation. When power is on and cards are in the hopper, the operator makes the card ready by pressing the START button. During program operation, the program must test for the ready condition before initiating a card read operation. Once an EOM instruction starts the card read, the desired channel function (EOM, POT) may control the flow of information into memory. In the Hollerith mode, any column read that is not punched in one of the 64 combinations listed in Appendix A-1 results in a Validity check. The presence of a Validity check causes an error signal to be sent to the channel and lights the VALIDITY CHECK light on the reader.

If the stacker should become full, or the hopper empty, the reader is not ready and the NOT READY indicator lights. The card reader remains in the NOT READY state until the operator corrects the situation and presses the START button. Upon reading the last card, the reader sets an End-of-File signal if the EOF ON switch is on. The central processor can test the End-of-File condition to determine if more cards are in the hopper.

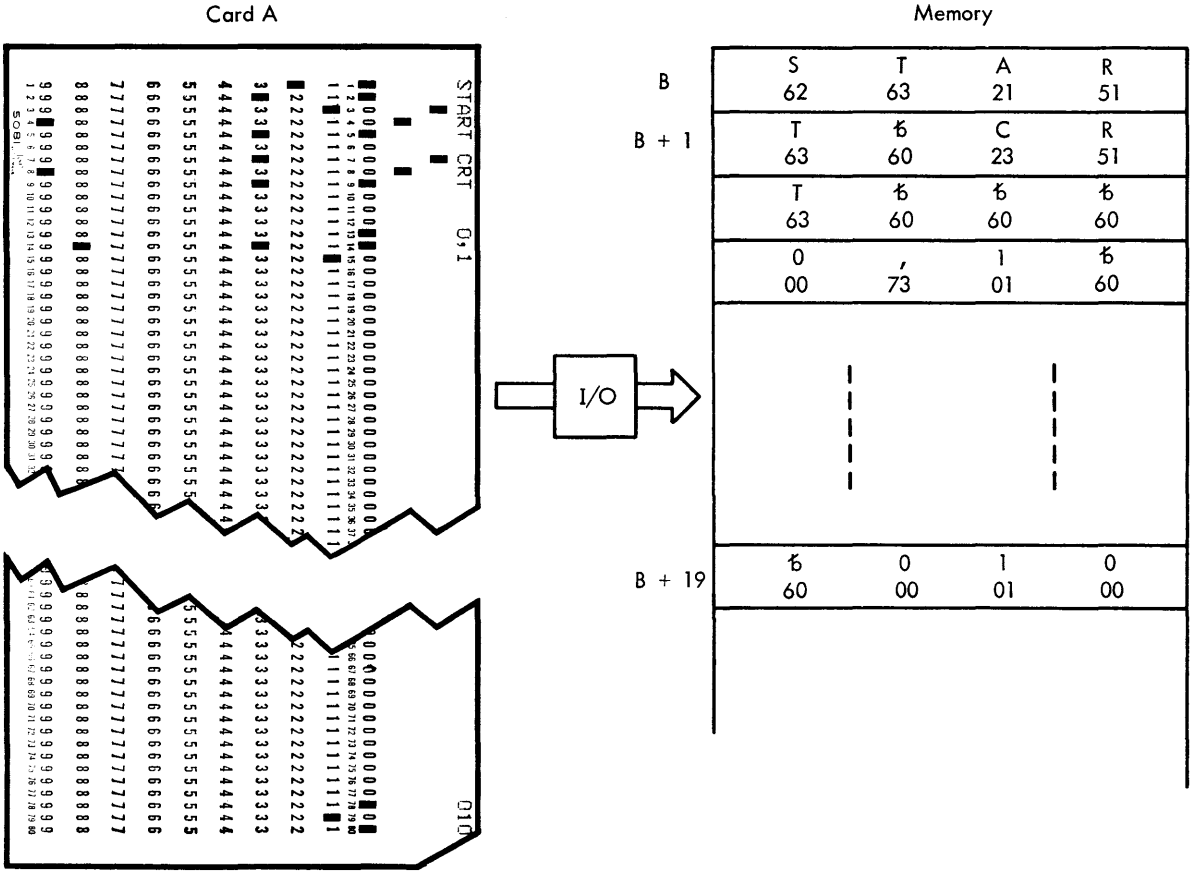


Figure 4-4. Card Read Into Memory in Hollerith

Punching

The card punch punches cards a row at a time, starting with row 12. The punch coupler, in both the Hollerith and binary modes, automatically rearranges the information to be punched. The card punch program must present the entire card image, 80 or 160 characters, to the punch 12 times for each card. This is necessary because of the way the punch operates. As each row of the card approaches the punch station, the coupler examines every character of the image to determine which column positions in that row should be punched. After the 12th output, the card punch punches row 9 and completes the card cycle.

The card punch is Ready to punch if there are cards in the magazine, the stacker is not full, and the operator has pressed the START button. The punch remains Ready as long as the above conditions are true. A Punch Card instruction given when the punch is Ready causes a card to feed past the punch station. The program must then address the punch and give the same instructions 12 times to transmit the card image to the coupler.

Programming Instructions

The Card Reader and Punch instructions follow. They use unit number 1 on Channel W with the four characters per word transmission mode.

Card Read Instructions

CRT 0, 1 CARD READER READY TEST 0 40 12006

This test determines if the selected card reader is Ready to read. If so, the computer skips the next instruction in sequence and executes the following instruction. If the reader is Not Ready, the computer executes the next instruction in sequence.

FCT 0, 1 FIRST COLUMN TEST 0 40 14006

This test determines if the first column is about to be read by the card reader. Since the time elapsing between the execution of a card reader EOM and the reading of the first column is approximately 85 milliseconds (48, 450 computer cycles), this test allows the computer to perform other operations during this time. If FCT is executed less than 1.2 milliseconds (approximately 685 computer cycles) before the first column is due to be read, the computer skips the next instruction in sequence and executes the following instruction. If FCT is executed 1.2 milliseconds (or more) before the first column is due to be read, the computer executes the next instruction in sequence (does not skip).

CFT 0, 1 CARD READER END-OF-FILE TEST 0 40 11006

This test determines if the End-of-File condition from the card reader has been detected. If not, the computer skips the next instruction in sequence and executes the following instruction. If the EOF condition has been detected, the computer executes the next instruction in sequence.

The reader remains in the End-of-File condition until the operator adds cards to the hopper or turns off the EOF ON switch.

RCD 0, 1, 4 READ CARD DECIMAL (Hollerith) 0 02 02606

RCD alerts the card reader, causes a card to feed from the hopper, and selects the Hollerith mode (as each column is read it is translated to an XDS internal code). This mode can read up to 80 characters (20 words) from a card.

RCB 0, 1, 4 READ CARD BINARY 0 02 03606

RCB alerts the card reader, causes a card to feed from the hopper and selects the binary mode (as each column is read it is transmitted as two 6-bit binary characters). This mode can read up to 160 characters (40 words) from a card.

SRC 0, 1 SKIP REMAINDER OF CARD 0 02 12006

This instruction causes the reader to stop transmission of characters to the channel. The remaining characters are not checked for validity, but a read check, feed check, or end-of-record condition will cause an End-of-Record interrupt and disconnect the card reader from the channel.

Card Punch Instructions

PBT 0, 1 PUNCH BUFFER TEST 0 40 12046

This instruction is used to test the status of the punch buffer. If the punch buffer is clear (empty) and ready for loading when PBT is executed, the computer skips the next instruction in sequence and executes the following instruction. If the punch buffer is not clear when PBT is executed, the computer executes the next instruction in sequence (does not skip). The punch buffer is always clear if the punch is ready to feed and punch.

CPT 0, 1 CARD PUNCH READY TEST 0 40 14046

This test determines if the selected card punch is Ready to punch. If so, the computer skips the next instruction in sequence and executes the following instruction. If the punch is Not Ready, the computer executes the next instruction in sequence.

Before the punch is Ready, the operator must place blank cards in the magazine and press the START button.

PCD 0, 1, 4 PUNCH CARD DECIMAL (Hollerith) 0 02 02646

PCD alerts the punch, causes a card to feed past the punch station, and selects the Hollerith mode. A transmission of 80 characters (20 words) must follow this instruction. The instruction PCD followed by the transmission instructions for 80 characters per card is repeated 12 times.

PCB 0, 1, 4 PUNCH CARD BINARY 0 02 03646

PCB alerts the punch, causes a card to feed past the punch station, and selects the binary mode. A transmission of 160 characters (40 words) must follow this instruction. The instruction PCB followed by the transmission instructions for 160 characters per card is repeated 12 times.

EXAMPLE: Card Read

This program reads one card in Hollerith mode. It is a closed subroutine that uses interrupts; assume the interrupt system is enabled.

Location	Instruction	Address	Comments
1000	PZE		This is an assembler instruction. It conveniently reserves a location for the subroutine entry.
	CRT	0, 1	This instruction is the card reader Ready test for Card Reader Number 1 on Channel W. If Not Ready, the computer executes the next instruction. If Ready, the computer skips the next one and executes the following instruction. The octal configuration is 0 40 12006.
	BRU	$-1	This instruction branches back to the test on Not Ready. The programmer can put an exit to a Not-Ready corrective routine here.
	RCD	* 0, 1, 4	This instruction connects Card Reader 1 to Channel W, alerts the interlace, starts a card moving toward the read station, and specifies Hollerith mode. The octal configuration for this instruction is 0 02 42606.
	EXU	READ	This instruction executes the I/O Control EOM at location READ.
	POT	READ + 1	This instruction transmits to the channel the word count and starting address.
	BRR	1000	This instruction branches back to the main program.
READ	EOM 01203720	15200	This EOM specifies terminal input function 01 (IOSD) and the Count Equals Zero interrupt. The word in READ + 1 specifies that a record will be read into memory beginning at location 2000 and specifies a 20-word limit.

The computer processes the main program while the channel performs the card read operation. When finished with the input, transmission of an interrupt will occur to the interrupt level 31, the Count Equals Zero location for Channel W.

31	BRM	TEST	This instruction, placed in location 31 for this example, branches and marks to location TEST.
TEST	PZE		This instruction saves a location for the routine entry.
	CET	0	This instruction tests for an error on Channel W. Its octal configuration is 0 40 11000.
	BRM	ERR	The computer executes this instruction if there is an error on Channel W. Assume that ERR is the entry to a corrective subroutine.
	BRU	* TEST	This instruction returns control to the main program and clears interrupt level 31. The computer executes this instruction if no error is detected.

EXAMPLE: Card Punch

This program punches one card in Hollerith mode. It is a closed subroutine that uses interrupts. The Index Register counts the 12 times the program presents the card image to the punch.

Location	Instruction	Address	Comments
1000	PZE		Saves the location for the subroutine entry.
	CLR		Clears the A and B Registers.
	STA	SWICH	Clears a switch for later use.
	LDA STA	1000 ENTR2	This pair of LDA and STA place the main program mark address in location ENTR2.
	MIN	ENTR2	MIN adds one to the stored contents.
MCRDS	LDX	ROWS	Initializes the Index Register with 00077765 (octal), which is -11 decimal.
	CPT	0, 1	Tests the card punch for a Ready condition. The card punch is Number 1 on Channel W.
	BRU	$-1	The computer executes this instruction if the punch is Not Ready. It branches back to the test, CPT 0, 1. The programmer can place an exit to a time loop here with the facility to tell the operator that the card punch will not become Ready.
GETRW	PCD	* 0, 1, 4	The computer executes this instruction if the punch is Ready. It alerts Channel W with interlace, connects Card Punch Number 1 to Channel W, starts a card moving toward the punch station, and specifies four characters per word and Hollerith mode.
	EXU	PNCH	Executes the EOM located in PNCH.
	POT	PNCH + 1	Transmits to the channel the word count and starting address.
	BRU	ENTR2	Branches back to the main program.
PNCH	EOM 01203720	16000	This EOM specifies terminal output function 00 (IORD) and the End-of-Record interrupt. The word in PNCH + 1 specifies that 20 words will be output from memory beginning in location 2000.
ROWS	00077765		Note that the program must send the card image to the channel twelve times to punch a card.

The computer processes the main program while the channel performs the output. When finished with the output, transmission of an interrupt will occur to the interrupt level 33, the End-of-Record location for Channel W.

33	BRM	ENTR2	
ENTR2	PZE		Saves a location for routine entry.
	BRX	GETRW	Increments the index by one. If the base has not been incremented through zero, the next instruction executed is at location GETRW. When the base increments to zero, the computer executes the next instruction in sequence. The Index counts row times on the card.
	MIN	SWICH	Sets a switch to indicate to the main program that the punch operation is complete.
	BRU	* ENTR2	Returns control to the main program and clears the interrupt.

MAGNETIC TAPE INPUT/OUTPUT

Format

Magnetic tape units used in XDS computer systems are IBM-compatible. The tape is one-half inch wide, Mylar base material, 1.5 mils thick. Tape reels (10 1/2-inch, plastic) contain up to 2400 feet of tape. A reflective marker, placed on the back of the tape approximately ten feet from the beginning of it, indicates the load point. The leading ten feet leave space for threading tape through the guides on the unit. The load point marker is on the Mylar side of the tape along the edge nearest the operator when the tape is mounted. A similar marker is along the other edge of the tape to mark the end-of-reel. About 14 feet of tape are reserved between the End-of-Reel marker and the end of the tape. This space includes at least ten feet of leader and enough tape to hold a record of 9,600 characters in 200 bpi density after sensing of the End-of-Reel marker.

Character recording on tape is in seven parallel tracks. A change in the magnetic flux in a track records a 1-bit for a given character position. No change in magnetic flux indicates a 0-bit. Six of the tracks are for information; the seventh track is a parity check. The system allows both even and odd parity, as needed. Binary recording uses odd parity. In this mode, the tape unit records the six-bit characters from the channel without change. Binary-coded decimal (BCD) recording uses even parity. In this mode, the tape control unit transforms characters from the channel to conform to standard IBM, BCD interchange code (see Appendix A-1).

Arrangement of information on tape is in blocks that may contain one or more records. Only the capacity of available core storage in the computer limits block length. An inter-record gap (section of blank tape) about 3/4-inch long separates blocks of records on tape. In writing, the tape unit automatically produces gap at the end of a record or block. Reading begins with the first character sensed after the gap and continues until encountering the next gap.

An inter-record gap, followed by a special, single-character record, marks the end of a file of information. The character is a Tape Mark (0001111). Writing a one-word record in BCD with one-character-per-word format can record such a mark. A reel of tape may contain one or more files. On reading an End-of-File record, the tape control unit stops the tape and sets its End-of-File indicator, which the program may test.

The tape control unit considers any record that contains only Tape Mark (0001111) characters an End-of-File. The tape unit reads such characters into memory like any other characters.

As the tape unit writes information it makes an odd-even count of the number of 1-bits in each track. At the end of each record it writes a bit for each track such that the total number of 1-bits in each track is even. This parity check sum is always even whether the character parity is even or odd. The character containing these check bits is the longitudinal parity character; the tape unit writes it slightly past the end of recorded information in the block.

The longitudinal check character always reflects an even parity check for each channel. In the BCD mode, the check character itself always has an even number of 1-bits. In the binary mode, however, the check character may have either an even or an odd number of 1-bits. This means that a reverse scan over a binary record may result in turning on the error indicator in the channel even though the record is correct. As a general rule, the program ignores the error indicator after a reverse operation.

It is possible to write tape in a 1-, 2-, or 3-character-per-word mode provided that the rate of characters is sufficient. On reading, however, the tape unit uses the character count to ascertain when it has read two characters and can look for gap. If a 1-character-per-word "read" is in operation, a single noise character will stop the tape. In reverse scan a 1-character-per-word operation causes the tape to stop after detecting the longitudinal check character at the end of the record. This means that the tape stops in the recorded information.

All scan operations must be in 3- or 4-character-per-word mode or the tape does not stop when it reaches gap.

As a general rule, the user should program tape units for three or four characters per word, if possible. The write-tape-mark operation is an exception to this rule.

Use of the TAPE READY TEST (TRT) between tape operations of opposite direction ensures that the tape unit stops and reverses. It is an advisable programming practice to terminate tape writing by several inches of erasure whenever subsequent resumption of recording is anticipated. This eliminates the effects of a possible extraneous character that might arise through subsequent tape repositioning.

Reading

Once a tape starts with a Read Binary or Read BCD EOM or EOD, it continues until the tape unit detects an End-of-Record gap. If the computer does not instruct the tape unit to continue, it stops in the middle of that gap. When the tape stops, the tape unit disconnects from the channel. If the tape encounters an End-of-File, the tape control unit sets its EOF indicator. The central processor can test this indicator, which remains set until the tape unit control receives a new EOM/EOD on that channel. The tape always stops after the Tape Mark.

At the end of the file the program reads the EOF character (0001111) into memory along with its check character. In a four-character-per-word "read", this appears in the first word of the input area as a 17170000 word.

When the tape unit is writing on tape, it may transmit flux disturbing surges ahead of the current writing positions; these surges affect previously written records further down the tape. This means that a record in the middle of a file cannot be updated or rewritten if the records that follow it are to be read.

Any errors detected either by the channel in the character parity check or by the control unit with longitudinal parity check sets the error indicator in the channel. When detecting such an error in reading, the routine should backspace the tape over the erroneous record and attempt to reread the record.

The tape unit backspaces over records using the Scan feature. A Scan reverse EOM or EOD starts the tape in reverse. The program then waits for the channel to become ready or waits for the End-of-Transmission (if enabled). When the buffer becomes ready or the End-of-Transmission interrupt occurs, the

tape stops in front of the backwardly traversed record. If the program hs enabled the interrupts, the End-of-Word (I1) interrupt occurs prior to the End-of-Transmission interrupt; executing a WIM to a dummy location and clearing the interrupt with a BRU indirect ignore the interrupt.

A Scan operation is similar to a Read operation except that the channel shifts the characters read through its Word Assembly Register, but does not consider a word complete until it encounters a tape gap. When the tape reaches the gap, the channel uses the last four characters in the word assembly as the only word read from the record. When scanning in reverse, the word consists of the last four characters scanned, which are the first four logical characters of the record. This operation assembles these characters in reverse. For example, if the first four characters of the record are 1234 and the tape is scanning the record in reverse, these appear as 4321 in the word stored for that record.

The same operation occurs in the forward scan with the last four characters of the record forming the word stored. The Scan is useful for reverse searching on the first word of the records in the file being searched. In this case, the routine starts the tape in a reverse scan and loads the channel interlace with a terminal function 10 with a word count of 1 and arms the Count Equals Zero interrupt. When the tape reaches the beginning of the record, the channel stores the first word and interrupts the program which checks the key word against a search key. If they agree, then the program need only wait for the channel to become inactive (ready) and the routine reads the record forward. If the record is not the desired one, the program gives another "scan reverse" without waiting for the channel to become inactive, and reloads the channel interlace to scan the next record.

If the tape encounters the End-of-Reel marker while reading, the tape logic sets the End-of-Reel indicator in the tape unit; the program can test this at any time. An End-of-File normally indicates the end of recorded information on tape. It is possible, however, to use the End-of-Reel indicator to mark the last record on the reel.

Writing

Once a tape unit is ready and the file-protect ring is on the tape reel, that is, the file-protect test is false, a Write operation can begin. The write tape EOM starts tape motion; the tape remains in motion until it receives the termination signal from the buffer. The tape control unit then writes the remaining characters of the record and writes the longitudinal check character. When the read-after-write head reads this check character, the tape signals the channel it has reached gap. If the unit receives no further write instruction within one millisecond, the tape stops and disconnects.

If the user wishes to backspace or rewind and then to return at some later time to record additional information at the end of the previous series of records, the routine should write an End-of-File character or erase a segment of tape after the series of written records. This practice provides positive identification of the end of a record and facilitates return to a specific location on the tape. If the programmer does not use this method, the tape may not subsequently stop in the same location at the end of the series of records as it did when writing the last record. This would leave a segment of tape in the gap which has not been written and may cause erroneous operation when reading the tape.

In addition to writing under program control, the program can also erase magnetic tape. When using an erase EOM with an erase unit address, the tape unit operates as though it were in a Write mode, except that it records no information. The program or interlace supplies the count of the number of words to be erased.

This type of erase is useful for the correction of a write error. When a write error occurs, an ERASE TAPE REVERSE (ERT) starts the tape in reverse. The same count, used to write the record originally, controls the erase. This procedure ensures that the tape always returns to the beginning of the erroneous record, even if a bad spot on the tape might appear as a gap. The routine may now rewrite the record. If the Write still produces an error, the routine erases the record backward and then erases it foward, using the same count, and by-passes the section of tape where the difficulty occurred. The routine may now rewrite the record on a new section of tape.

The erase procedure can produce the required 3.75 inches of blank tape between the load point and the first record. This is done by erasing 150 words at 200 bpi density, 417 words at 556 bpi density, or 600 words at 800 bpi density.

Use of a one-character-per-word, BCD, Write instruction writes an End-of-File record. Then the program loads the channel interlace with a count of 1 and loads the address of a word containing the Tape Mark character (17) in the left-most position. EOM or EOD instructions to the tape units specify start-without-leader since the tape unit automatically generates gap at the end of all records for leader. A magnetic tape program should never include a leader instruction because an attempt to generate leader may cause an erroneous operation.

Programming

The SKS and EOM instructions for normal tape operations follow. EOM instructions use four character per word format for units on Channel W.

TRT 0,n TAPE READY TEST 0 40 1041n

TRT 0 tests tape unit number n on Channel W for Not Ready. If the tape is Not Ready, it skips the next instruction in sequence and executes the following instruction. If the tape is Ready, it executes the next instruction in sequence.

A tape is Not Ready if: (1) there is no physical unit set to the logical unit number being tested, (2) the selected unit is not in the Automatic mode, or (3) the tape is in motion for any operation.

FPT 0,n FILE PROTECT TEST 0 40 1401n

Tests tape unit number n on Channel W for file protect. If the file-protect ring is present, the computer skips the next instruction in sequence and executes the following instruction. If not inserted, it executes the next instruction in sequence. The skip does not occur if there is no logical unit n on the channel.

BTT 0,n BEGINNING OF TAPE TEST 0 40 1201n

Tests tape unit number n on Channel W for the beginning of the tape. If not positioned on the load point marker, the computer skips the next instruction in sequence and executes the following instruction. If positioned on the load point marker, it executes the next instruction in sequence. The skip does not occur if there is no logical unit n on the channel.

ETT 0, n, END OF TAPE TEST 0 40 1101n

Tests whether tape unit number n on Channel W is not positioned at the end of the tape. If the tape unit has not sensed the End-of-Reel marker, the computer skips the next instruction in sequence and executes the following instruction. If the End-of-Reel marker has been sensed, it executes the next instruction in sequence. The End-of-Reel condition is reset when the tape unit moves the tape backward over the End-of-Reel marker. The skip does not occur if there is no logical unit n on the channel.

DT2 0, n DENSITY TEST, 200 BPI 0 40 1621n

Tests tape unit number n on Channel W for being set at 200 bpi density. If not, the computer skips the next instruction in sequence and executes the following instruction. If so, it executes the next instruction in sequence.

DT5 0, n DENSITY TEST, 556 BPI 0 40 1661n

Tests tape unit number n on Channel W for being set at 556 bpi density. If not, the computer skips the next instruction in sequence and executes the following instruction. If so, it executes the next instruction in sequence.

DT8 0, n DENSITY TEST, 800 BPI 0 40 1721n

Tests tape unit number n on Channel W for being set at 800 bpi density. If not, the computer skips the next instruction in sequence and executes the following instruction. If so, it executes the next instruction in sequence.

TFT 0 TAPE END-OF-FILE TEST 0 40 13610

Tests whether a tape under control of the tape control unit on Channel W encountered an End-of-File during the last Read or Scan operation. If not, the computer skips the next instruction in sequence and executes the following instruction. If so, it executes the next instruction in sequence.

The End-of-File indicator remains set until the program calls for another tape operation.

TGT 0, n TAPE GAP TEST 0 40 1261n

Tests whether tape unit n on Channel W has encountered gap since it received the last EOM/EOD instruction. If not, the computer will skip the next instruction in sequence and execute the following instruction. If so, it executes the next instruction in sequence. TGT will execute the next instruction during the approximately 0.75 millisecond that the tape-gap indicator is "true".

MAGPAK TEST 0 40 1021n

Tape unit n is tested for being a MAGPAK. If the tape unit is not a MAGPAK, the computer skips the next instruction in sequence and executes the following instruction. If the tape unit is a MAGPAK, the computer executes the next instruction in sequence.

WTB 0, n, 4 WRITE TAPE IN BINARY 0 02 0365n

Starts tape unit n on Channel W in a Binary Write mode.

WTD 0, n, 4 WRITE TAPE IN DECIMAL (BCD) 0 02 0265n

Starts tape unit n on Channel W in a BCD Write mode.

EFT 0, n, 4 ERASE TAPE FORWARD 0 02 0367n

Starts tape unit n on Channel W in an Erase mode.

ERT 0, n, 4 ERASE TAPE IN REVERSE 0 02 0767n

Starts tape unit n on Channel W in reverse in an Erase mode.

RTB 0, n, 4 READ TAPE IN BINARY 0 02 0361n

Starts tape unit n on Channel W in a Binary Read mode.

RTD 0, n, 4 READ TAPE IN DECIMAL (BCD) 0 02 0261n

Starts tape unit n on Channel W in a BCD Read mode.

SFB 0, n, 4 SCAN FORWARD IN BINARY 0 02 0363n

Starts tape unit n on Channel W forward in a Binary Scan mode.

SFD 0, n, 4 SCAN FORWARD IN DECIMAL (BCD) 0 02 0263n

Starts tape unit n on Channel W forward in a BCD Scan mode.

SRB 0, n, 4 SCAN REVERSE IN BINARY 0 02 0763n

Starts tape unit n on Channel W in reverse in a Binary Scan mode.

SRD 0, n, 4 SCAN REVERSE IN DECIMAL (BCD) 0 02 0663n

Starts tape unit n on Channel W in reverse in a BCD Scan mode.

REW 0, n REWIND 0 02 1401n

Starts tape unit n on Channel W in a Rewind. REW does not use the channel.

RTS 0 CONVERT READ TO SCAN 0 02 14000

The tape unit currently in a read mode on the channel is instructed to convert from the read mode of operation to the scan mode of operation.

SRR 0 SKIP REMAINDER OF RECORD[†] 0 02 13610

The tape unit currently on the channel is instructed to skip the remainder of the record being read.

[†]Note: This instruction applies only to 41.7-kc and 96-kc magnetic tape systems.

MAGNETIC TAPE EXAMPLE PROGRAMS

The following examples show samples of complete input/output programs for magnetic tape.

EXAMPLE: Magnetic Tape Read

This program reads one record from Magnetic Tape Number 1 on Channel W. It uses the End-of-Record interrupt. The tape is not at its beginning or end.

Location	Instruction	Address	Comments
1000	PZE		Saves a location for the subroutine entry.
	TRT	0, 1	Tests Ready Magnetic Tape 1 on Channel W. If Magnetic Tape 1 is ready to perform an input/output operation, the computer executes the next instruction in sequence. If not, it skips the next instruction and executes the following one. The octal configuration is 0 40 10411.
	BRU	$+2	Skips one instruction.
	BRU	$-2	Branches back to TRT 0, 1. The programmer can place here an exit to a routine that determines reasons for the Non-Ready condition.
	RTD	*0, 1, 4	Addresses Channel W, alerts the interlace, connects it to Magnetic Tape 1, specifies four characters per word and BCD modes, and starts tape motion.
	EXU	REDTP	Executes the EOM located in location REDTP.
	POT	REDTP+1	Transmits to the channel the word count and starting address.
	BRR	1000	Branches back to the main program.
REDTP	EOM 06203720	16000	This EOM specifies terminal input function 00 (IORD) and the End-of-Record interrupt. The word in REDTP+1 specifies that one record or 100 words, whichever is smaller, will be read into memory beginning in location 2000. Any remaining words in the record after the first 100 will be ignored. (0620 is equal to 144_8 shifted right one place; it is merged with 03720 to generate the "POTted" word.)

The main program continues while the channel performs the input operation. When finished, the End-of-Record interrupt goes to location 33.

33	BRM	COMPL	This instruction in interrupt location 33 branches and marks to COMPL to finish the read operation.
COMPL	PZE		Saves a location for the routine entry.
	CET	0	Tests for error in Channel W. If it detects an error, the computer executes the next instruction in sequence. If not, it skips the next one and executes the following instruction. The octal configuration is 0 40 22000.
	BRM	ERTST	Branches to an assumed routine to re-read the block a number of times and, if the error continues, to notify the operator.
	BRU	* COMPL	Returns control to the main program and clears interrupt level 33.

EXAMPLE: Gather-Write Magnetic Tape

The program writes one record on magnetic tape. The gathering of the data written in that record is from three non-contiguous areas of memory. This program is a closed subroutine that uses the Count Equals Zero interrupt; it uses Channel W and Magnetic Tape Number 1 on Channel W with interlace.

A similar program can perform a scatter-read operation. The difference is the exchange of the read instruction (RTD) with the write instruction (WTD) and the deletion of the file-protect testing instruction.

Location	Instruction	Address	Comments
1000	PZE		Saves a location for the subroutine entry.
	CLR		Clears the A and B Registers.
	STA	COUNT	Clears location COUNT for use later as a switch.
	TRT	0, 1	Tests whether Magnetic Tape 1 on Channel W is Ready.
	BRU	$ + 2	Branches two locations ahead. The computer executes it if the magnetic tape unit is Ready.
	BRU	$ - 2	Branches back to the Ready test.
	FPT	0, 1	Tests whether the file-protect ring is present on the tape reel. If so, the computer skips the next instruction and executes the following one. The octal configuration is 0 40 14011.
	BRM	OPER	Branches and marks to an assumed routine to call the operator and instruct him to insert file-protect ring on Magnetic Tape 1.
	LDA STA MIN	1000 FAST FAST	These three instructions place the marked subroutine entry location plus one into location FAST.
	WTD	0, 1, 4	Connects Magnetic Tape 1 to Channel W, specifies BCD transfer mode and four characters per word, and starts the tape moving. The octal configuration is 0 02 02651.
	BRU	FAST + 1	Branches around location FAST.
FAST	PZE		Saves a location for entry to the multiple write area of the subroutine.
	LDX	COUNT	Loads the Index Register with the contents of COUNT, which picks up the proper input/output control instructions.
	LDA LDB SKM BRU BRU	OKAY MASK COUNT $ + 2 * FAST	These five instructions determine when the write operation is complete. When it is, location COUNT contains the number 6 and the active interrupt, level 31, is cleared. Location MASK contains 77777777_8.
	ALC	0	Alerts the interlace in Channel W for subsequent loading.
	EXU	A, 2	Executes the EOM located in address A modified by the Index.
	POT	A + 1, 2	Transmits to the channel the word count and starting address.
	MIN MIN	COUNT COUNT	These instructions add two to the contents of COUNT.
	BRU	* FAST	Branches back to the main program.

The main program continues while the channel performs the output. When finished, the Zero Word Count interrupt goes to interrupt location 31.

Location	Instruction	Address	Comments
31	BRM	FAST	Branches and marks at location FAST.

The routine repeats this for the output words in A + 2 and in A + 4. Then the test in location FAST + 4 causes a final Branch to clear interrupt (BRU) back to the main program.

Location	Instruction	Address	Comments
A	EOM 06203720	15600	This EOM specifies terminal output function 11 (IOSP) and the Count Equals Zero interrupt. The word in A+1 specifies that 100 words will be read out from memory beginning in location 2000.
A + 2	EOM 14404740	15600	This EOM specifies terminal output function 11 (IOSP) and the Count Equals Zero interrupt. The word in A+3 specifies 200 words from memory beginning in location 2500.
A + 4	EOM 06205670	15000	The EOM specifies terminal output function 00 (IORD) and the Count Equals Zero interrupt. The word in A+5 specifies 100 words from memory beginning in location 3000. Upon completion of the output of this sub-record, the channel disconnects.
OKAY	00000006		This is the stored number 6 used in the completion test above.

NOTE: This sample program is for clarification of magnetic tape programming. It does not include extra programming to save the contents of the A or the Index Register for the main program.

LINE PRINTER

XDS buffered line printers are capable of printing up to 1000 lines per minute at 132 characters per line, with a standard set of 56 characters. Printing is accomplished by means of a rotating character drum and a bank of 132 print hammers. The drum passes 56 different characters, in lines of 132 each, past the hammer bank. Upon command from the computer, the selected print hammers drive the paper against the ribbon and onto the appropriate character typeface as it passes the print position. The characters are transmitted sequentially for storage in the printer buffer before printing. A programmable format tape loop provides fixed (or preselected) space control. Upspacing of 1 to 7 lines, as well as page control, may be accomplished by program instructions.

An optional, off-line facility allows the program or the operator to initiate card-to-printer or magnetic tape-to-printer operations simultaneous with computation (see Off-Line Printing).

Printer Controls

The printer controls, Figure 4-5, for XDS line printers consist of eight switches and indicators.

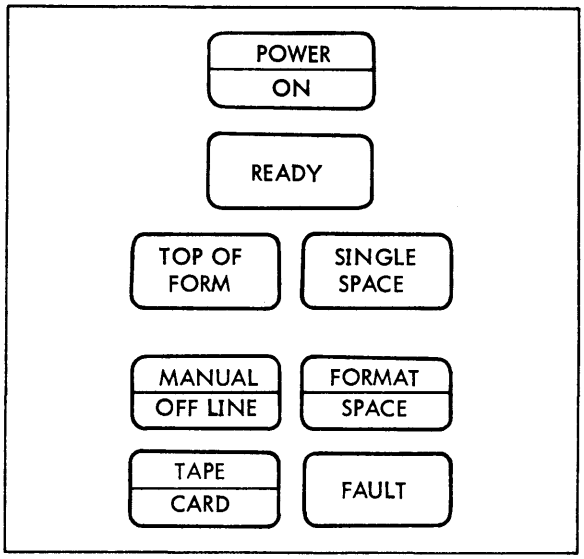


Figure 4-5. Printer Control Indicator Lights and Switches

The POWER/ON switch is an alternate action switch. The computer must be turned on for this switch to be activated. Pressing POWER/ON lights the top half of the indicator, turns on the motors and hammer driver power supply, and starts a timer that allows the motors to reach proper speed. After 20 seconds the bottom half lights, indicating that the printer is operable.

When the printer is initially turned on, the READY indicator is off. When pressed, it is turned on if:

1. paper is loaded in the line printer,
2. the lower half of the POWER/ON switch is lighted, and
3. the hammer power supply is on.

This indicator automatically goes off when the above conditions are not realized. The printer is ready for either on-line or off-line operation when READY is turned on. Ready is reset to preclude computer intervention while changing paper or ribbon, or operating the TOP OF FORM or SINGLE SPACE switches.

Pressing TOP OF FORM causes the printer to position paper according to format tape channel 1. This indicator is lighted only when the format tape is positioned at channel 1, that is, top-of-form on a standard tape loop. This switch is operative when there is paper in the printer and the READY indicator is off.

Pressing SINGLE SPACE causes the printer to upspace paper one single space, independently of the vertical format tape. This switch is operative when there is paper in the machine and READY is off.

The FAULT indicator lights when the printer detects a parity error as information transfer from the buffer to the print hammers, or when it detects a parity error in incoming data from magnetic tape or cards during an off-line operation. It remains lighted until the next EOM addresses the printer. The condition of the light corresponds to the status of a program-testable fault indicator in the printer.

MANUAL/OFF LINE[†] is a combination of a switch and two independent indicators. The program or the operator may initiate off-line operation, which is indicated by the illumination of OFF LINE (the bottom half of this switch). If the operator presses this switch to initiate off-line operation, MANUAL (the top half of the switch) is also lighted and remains lighted until the operator presses the switch again. OFF LINE is normally reset when the end-of-file is detected from the input unit. Pressing READY also resets OFF LINE, that is, by switching the printer from the "ready" to the "not ready" state.

The FORMAT/SPACE[†] switch is used in off-line operation. The operator may use either mode, spacing a single space after each line of print, or using the first character stored on tape or cards as a vertical format character.

The TAPE/CARD[†] switch selects the desired input device.

Paper Tape Format Loop

A paper tape format loop, placed in the printer, allows upspacing to proceed to prespecified vertical positions on the print page. The format loop is an eight-channel paper tape. Putting a punch in the specified channel at the desired vertical spacing selects the channel upspace. Channel 1 is the top-of-form channel, channel 7 is the bottom-of-form channel, and channel 0 is the single-upspace channel. In the off-line mode with SPACE control, channel 0 controls single spacing. When printing with no format loop inserted in the printer, single upspacing occurs regardless of the channel specified.

[†]If an off-line coupler is not attached to the printer, the MANUAL/OFF LINE, FORMAT/SPACE, and TAPE/CARD indicators neither light nor affect printer operation.

Line Printer Instructions

PLP 0, 1, 4 PRINT LINE PRINTER 0 02 02660

This instruction connects the line printer to channel W and specifies a character transmission of 4 characters per word.

This instruction is followed by the transmission of up to 132 characters. If the character count is less than 132, the characters are printed left-justified on the page. If the character count is more than 132, the printer produces an undetectable error.

The following control instructions are coded for Channel W using unit number 1:

POL 0, 1 PRINTER OFF-LINE 0 02 10260

This instruction places the printer off-line and initiates an off-line print operation. The selected input device (card reader 1 or magnetic tape unit 7) also goes off-line (See Off-Line Printing).

PSC 0, 1, n PRINTER SKIP TO FORMAT CHANNEL n 0 02 1n460

This instruction causes the printer to eject paper until the paper tape format loop detects the first punched hole in the channel specified by the number n (0 to 7). (See PSP for timing.)

PSP 0, 1, n PRINTER UPSPACE n LINES 0 02 1n660

This instruction causes the printer to upspace n (0 to 7) lines. Consecutive upspace instructions must be separated by a sufficient time delay. Otherwise, the two PSP instructions may be merged by the printer.

Approximate completion times for PSP (from initiation of instruction to paper stop) are:

Upspace 1 line: 25 milliseconds (14, 275 cycles)

Upspace more than 1 line: add 10 milliseconds (5, 690 cycles) for each additional line.

Line Printer Tests

The line printer tests to follow are coded for channel W using unit number 1:

PFT 0, 1 PRINTER FAULT TEST (Skip if no Printer Fault) 0 40 11060

This test determines if the printer has detected a parity error during a transfer of information from the printer buffer to the print hammers. If such an error occurs, a fault detector is set and the FAULT indicator is lighted. If the fault detector is set when PFT is executed, the computer executes the next instruction in sequence (does not skip). If the fault detector is not set, the computer skips the next instruction in sequence and executes the following instruction.

PRT 0, 1 PRINTER READY TEST (Skip if Printer Ready) 0 40 12060

This instruction tests the printer for a "ready" condition. The criteria for a printer "ready" condition are:

1. Paper is loaded in the machine,
2. The lower half of the POWER/ON switch is lighted, and
3. The hammer power supply is on.

If the printer is ready when PRT is executed, the computer skips the next instruction in sequence and executes the following instruction. If the printer is not ready, the computer executes the next instruction in sequence (does not skip). Since the printer tests ready while ejecting paper, the program should allow a definite time interval to pass (see PSP) after a PSC or PSP instruction before executing a new PSC or PSP. A dummy PLP instruction may be issued between two space instructions (PSC or PSP). This instruction will provide the timing required. A ready test may be used to determine when the second paper space instruction may be sent.

EPT 0, 1 END OF PAGE TEST (Skip if not End of Page) 0 40 14060

This instruction tests the printer for paper position. If the paper is positioned at the end of page (specified by format channel 7) the computer executes the next instruction in sequence (does not skip). If the paper is not positioned at the specified end of page, the computer skips the next instruction in sequence and executes the following instruction.

Terminating Line Printer Output

When the single-word mode of transmission is used for printing on the line printer, each character transmission for a line must be followed by a TERMINATE OUTPUT (TOP) instruction. TOP is automatically generated with interlaced outputs.

Error Conditions

1. Print fault - parity error during transfer of character information from print buffer to print hammers.
2. Buffer error - parity of character rate error during transfer of information through buffer.
3. Input fault - parity error in incoming data from cards or magnetic tape (during off-line operation only).

Off-Line Printing

The optional, off-line facility allows the line printer to produce printed records from card or magnetic tape sources without computer attention. The character transmission proceeds directly from the source to the printer and the channel may still be used by the computer for other input/output operations (e.g., card reading on card reader 2, card punch, paper tape read/punch, disk read/write, etc.). Once initiated, the printing operation is controlled by the source and proceeds until the source generates an end-of-file signal (see card input and magnetic tape input for appropriate end-of-file conditions).

The FAULT indicator lights when a parity error is detected during the reading of a tape record; the off-line printer rereads the record in an attempt to read good data. If this reread record contains an error, FAULT lights, the off-line operation terminates, and the printer goes back on-line if physically connected

EXAMPLE: Print Two Lines

This program positions the paper at the top of the page and prints two lines with a single upspace between them. It assumes that the printer is ready to print or is becoming ready after a print operation. This program, written as a closed subroutine, uses channel W, Line Printer 1, and the Count Equals Zero and End-of-Record interrupts.

Location	Instruction	Address	Comments
1200	PZE		Reserves a location for subroutine entry.
	CLR		Clears the A and B Registers.
	STA	SWICH	Initializes a location, SWICH, which indicates that printing is completed.
	PRT	0, 1	Tests for printer ready. The octal configuration for this instruction is 0 40 12060.
	BRU	$-1	Returns control to the ready test; if the printer is not ready, the computer executes this instruction.
	PSC	0, 1, 1	Instructs the printer to move paper to the top of the page. The octal configuration for this instruction is 0 02 11460.
	PLP	*0, 1, 4	Connects Printer 1 to Channel W, and specifies four characters per word transfer mode, and alerts the interlace. The octal configuration for this instruction is 0 02 42660.
	EXU	PRINT	Executes the EOM located in location PRINT.
	POT	PRINT+1	Transmits the word count and starting address.
	BRR	1200	Branches back to the main program while the line is being printed.
PRINT	EOM 02043720	16200	This EOM specifies output function 01 and the End-of-Record interrupt. The word in PRINT+1 specifies that 33 words will be output from memory beginning in location 2000.

The main program continues while the data transfer and printing is being completed. When completed, the End-of-Record interrupt goes to interrupt level 33. This indicates that all the data from memory has been obtained, and that the printing of the line has been completed.

33	BRM	UPSPC	Branches and marks to location UPSPC elsewhere in memory.
UPSPC	PZE		Reserves a location for an entry.
	PRT	0, 1	Tests for printer ready condition. Since the current line has been printed, the printer will be ready.
	BRU	*-1	Returns to the test.
	PSP	0, 1, 1	Causes the printer to upspace one line. The octal configuration is 0 02 11660.
HEAR	PLP	*0, 1, 4	Sets up the printer with interlace.
	EXU	PRNT	Executes the EOM in location PRNT.
	POT	PRNT+1	Transmits to the channel the word count and starting address.
	BRU	*UPSPC	Branches and clears the interrupt to the main program to await completion of the data transfer.
31	BRM	DONE	Branches and marks to location DONE elsewhere in memory.
DONE	PZE		This pseudo operation reserves a location for an entry.
	MIN	SWICH	Sets the printing complete flag.
	BRU	*DONE	Branches back to the main program and clears interrupt 31. This is the final exit.
PRNT	EOM 02043761	15000	This EOM specifies terminal output function 00(IORD) and the Count Equals Zero interrupt. The word in PRNT+1 specifies that 33 words will be read out from memory beginning in location 2033. The channel disconnects at the end of the output.

At location HEAR, note that the computer executes the instructions to print and control the printing before the printing has had time to completely upspace the paper as requested. The instructions cause an immediate transfer of data into the Print Buffer and printing begins immediately after completion of upspacing.

to the computer and the MANUAL indictor is off. When a validity check occurs during a card read, FAULT lights, the operation terminates, and the printer goes back on-line if the MANUAL indicator is off. The next EOM addressing the printer resets FAULT if the printer is on-line. If the MANUAL indicator is on, the error condition may be cleared by pressing READY off and then on again. If a fault occurs in an off-line operation initiated by the computer, the usual method for clearing the error is:

1. Press MANUAL on.
2. Press READY off.
3. Press READY on.
4. Press MANUAL off.

In a manually initiated off-line operation, steps 1 and 4 are not required.

Off-line printing can be formatted as desired through the use of a single upspace or the format control mode (see Table 4-2). Off-line printing terminates by an end-of-file indicator from either device. Upon termination of an off-line operation, a physically connected off-line printer system returns on-line, provided the MANUAL indicator is off.

Table 4-2. Format Control Characters

Code	Character	Function
00	0	Skip to format channel 0
01	1	Skip to format channel 1
02	2	Skip to format channel 2
03	3	Skip to format channel 3
04	4	Skip to format channel 4
05	5	Skip to format channel 5
06	6	Skip to format channel 6
07	7	Skip to format channel 7
40	- (hyphen)	Do not space
41	J	Upspace 1 line
42	K	Upspace 2 lines
43	L	Upspace 3 lines
44	M	Upspace 4 lines
45	N	Upspace 5 lines
46	O	Upspace 6 lines
47	P	Upspace 7 lines

Printing Off-Line Under Operator Control

The procedure for operator control of off-line printing is:

1. Switch on the desired input device. (Magnetic tape is selected by dialing it to logical tape number 7.)
2. Place paper at top of form, as desired, by means of the TOP OF FORM switch.
3. Select desired input device by means of the TAPE/CARD switch.
4. Select either the FORMAT or SPACE mode as required.
5. Press MANUAL/OFF LINEswitch.
6. Press READY switch on, which initiates actual data transfer.

Printing Off-Line Under Computer Control

The procedure for computer control of off-line printing is:

1. Turn the equipment on.
2. Prepare the desired input device for operation.
3. Select desired input device by means of the TAPE/CARD switch.
4. Select either the FORMAT or SPACE mode as required.
5. Press the READY switch on.
6. Under program control, test the tape or card unit and the line printer for "ready" condition.
7. Then, to start transfer of data, give the POL instruction to print off-line.

Off-Line Print Termination

Off-line printing terminates when an end-of-file indicator from the magnetic tape unit or card reader occurs. When printing from magnetic tape, the print operation terminates when the first character read from a record is the end-of-file code, octal 17.

When printing from cards, the print operation terminates when the end-of-file signal comes from the reader. This occurs when the card hopper becomes empty and the EOF ON switch on the reader is on (END OF FILE indicator lights). If the hopper becomes empty when EOF ON is not lighted, the printer waits for more cards to be placed in the hopper and the reader to become ready. When the reader is again ready, printing resumes.

APPENDIX A CONVERSION TABLES

XDS CHARACTER CODES

Characters		XDS Internal Code	Card Code	Magnetic Tape BCD Code on Tape
Typewriter	Printer			
ø	0	00	0	12
1	1	01	1	01
2	2	02	2	02
3	3	03	3	03
4	4	04	4	04
5	5	05	5	05
6	6	06	6	06
7	7	07	7	07
8	8	10	8	10
9	9	11	9	11
Space	Blank	12	8-2	12 (3)
# or =	=	13	8-3	13
@ or '	'	14	8-4	14
:	:	15	8-5	15
>	>	16	8-6	16
↓	↓	17	8-7	17
& or +	+	20	12	60
A	A	21	12-1	61
B	B	22	12-2	62
C	C	23	12-3	63
D	D	24	12-4	64
E	E	25	12-5	65
F	F	26	12-6	66
G	G	27	12-7	67
H	H	30	12-8	70
I	I	31	12-9	71
Backspace ? (1)	? (5)	32	12-0 (4)	72
	.	33	12-8-3	73
¤ or)	)	34	12-8-4	74
[	[	35	12-8-5	75
<	<	36	12-8-6	76
‡ Stop	‡ (5)	37 (2)	12-8-7	77

Characters		XDS Internal Code	Card Code	Magnetic Tape BCD Code on Tape
Typewriter	Printer			
-	-	40	11	40
J	J	41	11-1	41
K	K	42	11-2	42
L	L	43	11-3	43
M	M	44	11-4	44
N	N	45	11-5	45
O	O	46	11-6	46
P	P	47	11-7	47
Q	Q	50	11-8	50
R	R	51	11-9	51
Car. Ret. ! (1)	! (5)	52	11-0 (4)	52
$	$	53	11-8-3	53
*	*	54	11-8-4	54
]	]	55	11-8-5	55
;	;	56	11-8-6	56
Δ	Δ	57	11-8-7	57
↑	Blank	60	Blank	20
/	/	61	0-1	21
S	S	62	0-2	22
T	T	63	0-3	23
U	U	64	0-4	24
V	V	65	0-5	25
W	W	66	0-6	26
X	X	67	0-7	27
Y	Y	70	0-8	30
Z	Z	71	0-9	31
Tab ‡ (1)	‡ (5)	72	0-8-2	32
,	,	73	0-8-3	33
% or (	(	74	0-8-4	34
~	~ (5)	75	0-8-5	35
\	\	76	0-8-6	36
* Delete	* (5)	77 (2)	0-8-7	37

NOTES:

(1) The characters ? ! and ‡ are for input only. The functions Backspace, Carriage Return, or Tab always occur on output.

(2) On the off-line paper tape preparation unit, 37 serves as a stop code and 77 as a code delete.

(3) The internal code 12 is written on tape as a 12 in BCD. When read, this code is always converted to 00.

(4) The codes 12-0 and 11-0 are generated by the card punch; however, the card reader will also accept 12-8-2 for 32 and 11-8-2 for 52 to maintain compatibility with earlier systems.

(5) For the 64-character printers only.

TABLE OF POWERS OF TWO

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
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
4 398 046 511 104	42	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
8 796 093 022 208	43	0.000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
17 592 186 044 416	44	0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
35 184 372 088 832	45	0.000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
140 737 488 355 328	47	0.000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
281 474 976 710 656	48	0.000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625

OCTAL - DECIMAL INTEGER CONVERSION TABLE

0000 to 0777 (Octal)	0000 to 0511 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

1000 to 1777 (Octal)	0512 to 1023 (Decimal)

	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

Octal-Decimal Integer Conversion Table

	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

Octal	Decimal
2000	1024
to	to
2777	1535
(Octal)	(Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

Octal	Decimal
3000	1536
to	to
3777	2047
(Octal)	(Decimal)

Octal-Decimal Integer Conversion Table

4000 (Octal)	2048 (Decimal)
to	to
4777 (Octal)	2559 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

5000 (Octal)	2560 (Decimal)
to	to
5777 (Octal)	3071 (Decimal)

	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

Octal-Decimal Integer Conversion Table

	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

6000	3072
to	to
6777	3583
(Octal)	(Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	1022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

7000	3584
to	to
7777	4095
(Octal)	(Decimal)

OCTAL - DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000	.000000	.100	.125000	.200	.250000	.300	.375000
.001	.001953	.101	.126953	.201	.251953	.301	.376953
.002	.003906	.102	.128906	.202	.253906	.302	.378906
.003	.005859	.103	.130859	.203	.255859	.303	.380859
.004	.007812	.104	.132812	.204	.257812	.304	.382812
.005	.009765	.105	.134765	.205	.259765	.305	.384765
.006	.011718	.106	.136718	.206	.261718	.306	.386718
.007	.013671	.107	.138671	.207	.263671	.307	.388671
.010	.015625	.110	.140625	.210	.265625	.310	.390625
.011	.017578	.111	.142578	.211	.267578	.311	.392578
.012	.019531	.112	.144531	.212	.269531	.312	.394531
.013	.021484	.113	.146484	.213	.271484	.313	.396484
.014	.023437	.114	.148437	.214	.273437	.314	.398437
.015	.025390	.115	.150390	.215	.275390	.315	.400390
.016	.027343	.116	.152343	.216	.277343	.316	.402343
.017	.029296	.117	.154296	.217	.279296	.317	.404296
.020	.031250	.120	.156250	.220	.281250	.320	.406250
.021	.033203	.121	.158203	.221	.283203	.321	.408203
.022	.035156	.122	.160156	.222	.285156	.322	.410156
.023	.037109	.123	.162109	.223	.287109	.323	.412109
.024	.039062	.124	.164062	.224	.289062	.324	.414062
.025	.041015	.125	.166015	.225	.291015	.325	.416015
.026	.042968	.126	.167968	.226	.292968	.326	.417968
.027	.044921	.127	.169921	.227	.294921	.327	.419921
.030	.046875	.130	.171875	.230	.296875	.330	.421875
.031	.048828	.131	.173828	.231	.298828	.331	.423828
.032	.050781	.132	.175781	.232	.300781	.332	.425781
.033	.052734	.133	.177734	.233	.302734	.333	.427734
.034	.054687	.134	.179687	.234	.304687	.334	.429687
.035	.056640	.135	.181640	.235	.306640	.335	.431640
.036	.058593	.136	.183593	.236	.308593	.336	.433593
.037	.060546	.137	.185546	.237	.310546	.337	.435546
.040	.062500	.140	.187500	.240	.312500	.340	.437500
.041	.064453	.141	.189453	.241	.314453	.341	.439453
.042	.066406	.142	.191406	.242	.316406	.342	.441406
.043	.068359	.143	.193359	.243	.318359	.343	.443359
.044	.070312	.144	.195312	.244	.320312	.344	.445312
.045	.072265	.145	.197265	.245	.322265	.345	.447265
.046	.074218	.146	.199218	.246	.324218	.346	.449218
.047	.076171	.147	.201171	.247	.326171	.347	.451171
.050	.078125	.150	.203125	.250	.328125	.350	.453125
.051	.080078	.151	.205078	.251	.330078	.351	.455078
.052	.082031	.152	.207031	.252	.332031	.352	.457031
.053	.083984	.153	.208984	.253	.333984	.353	.458984
.054	.085937	.154	.210937	.254	.335937	.354	.460937
.055	.087890	.155	.212890	.255	.337890	.355	.462890
.056	.089843	.156	.214843	.256	.339843	.356	.464843
.057	.091796	.157	.216796	.257	.341796	.357	.466796
.060	.093750	.160	.218750	.260	.343750	.360	.468750
.061	.095703	.161	.220703	.261	.345703	.361	.470703
.062	.097656	.162	.222656	.262	.347656	.362	.472656
.063	.099609	.163	.224609	.263	.349609	.363	.474609
.064	.101562	.164	.226562	.264	.351562	.364	.476562
.065	.103515	.165	.228515	.265	.353515	.365	.478515
.066	.105468	.166	.230468	.266	.355468	.366	.480468
.067	.107421	.167	.232421	.267	.357421	.367	.482421
.070	.109375	.170	.234375	.270	.359375	.370	.484375
.071	.111328	.171	.236328	.271	.361328	.371	.486328
.072	.113281	.172	.238281	.272	.363281	.372	.488281
.073	.115234	.173	.240234	.273	.365234	.373	.490234
.074	.117187	.174	.242187	.274	.367187	.374	.492187
.075	.119140	.175	.244140	.275	.369140	.375	.494140
.076	.121093	.176	.246093	.276	.371093	.376	.496093
.077	.123046	.177	.248046	.277	.373046	.377	.498046

Octal-Decimal Fraction Conversion Table

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000000	.000000	.000100	.000244	.000200	.000488	.000300	.000732
.000001	.000003	.000101	.000247	.000201	.000492	.000301	.000736
.000002	.000007	.000102	.000251	.000202	.000495	.000302	.000740
.000003	.000011	.000103	.000255	.000203	.000499	.000303	.000743
.000004	.000015	.000104	.000259	.000204	.000503	.000304	.000747
.000005	.000019	.000105	.000263	.000205	.000507	.000305	.000751
.000006	.000022	.000106	.000267	.000206	.000511	.000306	.000755
.000007	.000026	.000107	.000270	.000207	.000514	.000307	.000759
.000010	.000030	,000110	.000274	.000210	.000518	.000310	.000762
.000011	.000034	.000111	.000278	.000211	.000522	.000311	.000766
.000012	.000038	.000112	.000282	.000212	.000526	.000312	.000770
.000013	.000041	.000113	.000286	.000213	.000530	.000313	.000774
.000014	.000045	.000114	.000289	.000214	.000534	.000314	.000778
.000015	.000049	.000115	.000293	.000215	.000537	.000315	.000782
.000016	.000053	.000116	.000297	.000216	.000541	.000316	.000785
.000017	.000057	.000117	.000301	.000217	.000545	.000317	.000789
.000020	.000061	.000120	.000305	.000220	.000549	.000320	.000793
.000021	.000064	.000121	.000308	.000221	.000553	.000321	.000797
.000022	.000068	.000122	.000312	.000222	.000556	.000322	.000801
.000023	.000072	.000123	.000316	.000223	.000560	.000323	.000805
.000024	.000076	.000124	.000320	.000224	.000564	.000324	.000808
.000025	.000080	.000125	.000324	.000225	.000568	.000325	.000812
.000026	.000083	.000126	.000328	.000226	.000572	.000326	.000816
.000027	.000087	.000127	.000331	.000227	.000576	.000327	.000820
.000030	.000091	.000130	.000335	.000230	.000579	.000330	.000823
.000031	.000095	.000131	.000339	.000231	.000583	.000331	.000827
.000032	.000099	.000132	.000343	.000232	.000587	.000332	.000831
.000033	.000102	.000133	.000347	.000233	.000591	.000333	.000835
.000034	.000106	.000134	.000350	.000234	.000595	.000334	.000839
.000035	.000110	.000135	.000354	.000235	.000598	.000335	.000843
.000036	.000114	.000136	.000358	.000236	.000602	.000336	.000846
.000037	.000118	.000137	.000362	.000237	.000606	.000337	.000850
.000040	.000122	.000140	.000366	.000240	.000610	.000340	.000854
.000041	.000125	.000141	.000370	.000241	.000614	.000341	.000858
.000042	.000129	.000142	.000373	.000242	.000617	.000342	.000862
.000043	.000133	.000143	.000377	.000243	.000621	.000343	.000865
.000044	.000137	.000144	.000381	.000244	.000625	.000344	.000869
.000045	.000141	.000145	.000385	.000245	.000629	.000345	.000873
.000046	.000144	.000146	.000389	.000246	.000633	.000346	.000877
.000047	.000148	.000147	.000392	.000247	.000637	.000347	.000881
.000050	.000152	.000150	.000396	.000250	.000640	.000350	.000885
.000051	.000156	.000151	.000400	.000251	.000644	.000351	.000888
.000052	.000160	.000152	.000404	.000252	.000648	.000352	.000892
.000053	.000164	.000153	.000408	.000253	.000652	.000353	.000896
.000054	.000167	.000154	.000411	.000254	.000656	.000354	.000900
.000055	.000171	.000155	.000415	.000255	.000659	.000355	.000904
.000056	.000175	.000156	.000419	.000256	.000663	.000356	.000907
.000057	.000179	.000157	.000423	.000257	.000667	.000357	.000911
.000060	.000183	.000160	.000427	.000260	,000671	.000360	.000915
.000061	.000186	.000161	.000431	.000261	.000675	.000361	.000919
.000062	.000190	.000162	.000434	.000262	.000679	.000362	.000923
.000063	.000194	.000163	.000438	.000263	.000682	.000363	.000926
.000064	.000198	.000164	.000442	.000264	.000686	.000364	.000930
.000065	.000202	.000165	.000446	.000265	.000690	.000365	.000934
.000066	.000205	.000166	.000450	.000266	.000694	.000366	.000938
.000067	.000209	.000167	.000453	.000267	.000698	.000367	.000942
.000070	.000213	.000170	.000457	.000270	.000701	.000370	.000946
.000071	.000217	.000171	.000461	.000271	.000705	.000371	.000949
.000072	.000221	.000172	.000465	.000272	.000709	.000372	.000953
.000073	.000225	.000173	.000469	.000273	.000713	.000373	.000957
.000074	.000228	.000174	.000473	.000274	.000717	.000374	.000961
.000075	.000232	.000175	.000476	.000275	.000720	.000375	.000965
.000076	.000236	.000176	.000480	.000276	.000724	.000376	.000968
.000077	.000240	.000177	.000484	.000277	.000728	.000377	.000972

Octal-Decimal Fraction Conversion Table

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000400	.000976	.000500	.001220	.000600	.001464	.000700	.001708
.000401	.000980	.000501	.001224	.000601	.001468	.000701	.001712
.000402	.000984	.000502	.001228	.000602	.001472	.000702	.001716
.000403	.000988	.000503	.001232	.000603	.001476	.000703	.001720
.000404	.000991	.000504	.001235	.000604	.001480	.000704	.001724
.000405	.000995	.000505	.001239	.000605	.001483	.000705	.001728
.000406	.000999	.000506	.001243	.000606	.001487	.000706	.001731
.000407	.001003	.000507	.001247	.000607	.001491	.000707	.001735
.000410	.001007	.000510	.001251	.000610	.001495	.000710	.001739
.000411	.001010	.000511	.001255	.000611	.001499	.000711	.001743
.000412	.001014	.000512	.001258	.000612	.001502	.000712	.001747
.000413	.001018	.000513	.001262	.000613	.001506	.000713	.001750
.000414	.001022	.000514	.001266	.000614	.001510	.000714	.001754
.000415	.001026	.000515	.001270	.000615	.001514	.000715	.001758
.000416	.001029	.000516	.001274	.000616	.001518	.000716	.001762
.000417	.001033	.000517	.001277	.000617	.001522	.000717	.001766
.000420	.001037	.000520	.001281	.000620	.001525	.000720	.001770
.000421	.001041	.000521	.001285	.000621	.001529	.000721	.001773
.000422	.001045	.000522	.001289	.000622	.001533	.000722	.001777
.000423	.001049	.000523	.001293	.000623	.001537	.000723	.001781
.000424	.001052	.000524	.001296	.000624	.001541	.000724	.001785
.000425	.001056	.000525	.001300	.000625	.001544	.000725	.001789
.000426	.001060	.000526	.001304	.000626	.001548	.000726	.001792
.000427	.001064	.000527	.001308	.000627	.001552	.000727	.001796
.000430	.001068	.000530	.001312	.000630	.001556	.000730	.001800
.000431	.001071	.000531	.001316	.000631	.001560	.000731	.001804
.000432	.001075	.000532	.001319	.000632	.001564	.000732	.001808
.000433	.001079	.000533	.001323	.000633	.001567	.000733	.001811
.000434	.001083	.000534	.001327	.000634	.001571	.000734	.001815
.000435	.001087	.000535	.001331	.000635	.001575	.000735	.001819
.000436	.001091	.000536	.001335	.000636	.001579	.000736	.001823
.000437	.001094	.000537	.001338	.000637	.001583	.000737	.001827
.000440	.001098	.000540	.001342	.000640	.001586	.000740	.001831
.000441	.001102	.000541	.001346	.000641	.001590	.000741	.001834
.000442	.001106	.000542	.001350	.000642	.001594	.000742	.001838
.000443	.001110	.000543	.001354	.000643	.001598	.000743	.001842
.000444	.001113	.000544	.001358	.000644	.001602	.000744	.001846
.000445	.001117	.000545	.001361	.000645	.001605	.000745	.001850
.000446	.001121	.000546	.001365	.000646	.001609	.000746	.001853
.000447	.001125	.000547	.001369	.000647	.001613	.000747	.001857
.000450	.001129	.000550	.001373	.000650	.001617	.000750	.001861
.000451	.001132	.000551	.001377	.000651	.001621	.000751	.001865
.000452	.001136	.000552	.001380	.000652	.001625	.000752	.001869
.000453	.001140	.000553	.001384	.000653	.001628	.000753	.001873
.000454	.001144	.000554	.001388	.000654	.001632	.000754	.001876
.000455	.001148	.000555	.001392	.000655	.001636	.000755	.001880
.000456	.001152	.000556	.001396	.000656	.001640	.000756	.001884
.000457	.001155	.000557	.001399	.000657	.001644	.000757	.001888
.000460	.001159	.000560	.001403	.000660	.001647	.000760	.001892
.000461	.001163	.000561	.001407	.000661	.001651	.000761	.001895
.000462	.001167	.000562	.001411	.000662	.001655	.000762	.001899
.000463	.001171	.000563	.001415	.000663	.001659	.000763	.001903
.000464	.001174	.000564	.001419	.000664	.001663	.000764	.001907
.000465	.001178	.000565	.001422	.000665	.001667	.000765	.001911
.000466	.001182	.000566	.001426	.000666	.001670	.000766	.001914
.000467	.001186	.000567	.001430	.000667	.001674	.000767	.001918
.000470	.001190	.000570	.001434	.000670	.001678	.000770	.001922
.000471	.001194	.000571	.001438	.000671	.001682	.000771	.001926
.000472	.001197	.000572	.001441	.000672	.001686	.000772	.001930
.000473	.001201	.000573	.001445	.000673	.001689	.000773	.001934
.000474	.001205	.000574	.001449	.000674	.001693	.000774	.001937
.000475	.001209	.000575	.001453	.000675	.001697	.000775	.001941
.000476	.001213	.000576	.001457	.000676	.001701	.000776	.001945
.000477	.001216	.000577	.001461	.000677	.001705	.000777	.001949

APPENDIX B TWO'S COMPLEMENT ARITHMETIC

XDS computer systems hold negative numbers in memory in binary two's complement form. The two's complement of a binary number is formed by adding one to the one's complement (logical inverse) of the number. This convention allows the sign of a number to be used as an integral part of the number in all arithmetic operations and obviates the need for keeping track of a detached sign with computer logic.

In XDS systems, the sign bit is in the first bit position to the left of the most significant magnitude bit. Thus, if an XDS computer word was only 6 bits long instead of 24, some common decimal values would be represented in binary format as follows:

Decimal Number	Octal Equivalent	Complement Plus 1	Binary Equivalent
3	03	-	000 011
2	02	-	000 010
1	01	-	000 001
0	00	-	000 000
-1	(-)01	77	111 111
-2	(-)02	76	111 110
-3	(-)03	75	111 101
31	37	-	011 111
-31	(-)37	41	100 001

This table suggests the following algorithms:

1. To find the binary, two's complement of a negative decimal number:

   a. Find the octal equivalent of the absolute of the number

   b. Form the complement and add one

   c. Express as a binary number.

   The result is the binary, two's complement equivalent.

2. To find the decimal equivalent of a binary two's complement number:

   a. Express as an octal number

   b. Subtract one and form the complement

   c. Find the decimal equivalent.

   The negative of the result is the decimal equivalent.

The following examples show how two's complement numbers automatically yield the correct result when used arithmetically in the computer:

Decimal Number	Binary Equivalent
+20	010 100
-03	111 101
+17	1 010 001 = 21_8 = 17_{10}
	↳ lost carry

Note that the carry out of the most significant (sign bit) position is lost. Nevertheless, the value remaining is the correct answer.

Decimal Number	Binary Equivalent
-32	100 000
+24	011 000
- 8	111 000 = (-)10_8 = -8_{10}

When performing additions or subtractions in the computer, carries out of the sign bit do not always signify a true overflow condition or cause the OVERFLOW indicator to be set. In an addition, it is impossible to produce an overflow if the signs of the operands are unlike. The computer sets the OVERFLOW indicator in an addition only when the signs of the two operands are the same but the sign of the result is opposite. In a subtraction, which in the computer is accomplished by forming the two's complement of the subtrahend and then adding to the minuend, the test for overflow is similar to that for addition. That is, overflow occurs when both numbers have the same sign after the subtrahend has been complemented but the sign of the result is opposite.

APPENDIX C COMPUTER OPERATING PROCEDURES

TURN COMPUTER ON

1. Set the RUN-IDLE-STEP switch to IDLE.
2. Press POWER switch.

LOAD PROGRAM WITH FILL SWITCH

1. Set up selected input device with input program. The initial portion of the program contains the "bootstrap" (short load program – see page 44).
2. Set RUN-IDLE-STEP switch to IDLE.
3. Press START switch.
4. Set RUN-IDLE-STEP switch to RUN.
5. Set one of the FILL switches (see page 42). For example, to initiate filling from paper tape on Channel W, set the first FILL toggle switch to PAPER TAPE and release.

LOAD PROGRAM WITH LOADING SYSTEM

Refer to the operating procedures furnished with the particular assembler, compiler, monitor, diagnostic, or utility system being used.

EXECUTE PROGRAM

1. Set the RUN-IDLE-STEP switch to IDLE.
2. Set the REGISTER switch to C.
3. Press CLEAR and enter a BRU to the program starting location into REGISTER DISPLAY, using the set button. Format of the instruction is

   000 000 001 0xx xxx xxx xxx xxx

   BRU Program starting location

4. Set the RUN-IDLE-STEP switch to RUN. The computer then executes the BRU and continues instruction execution at computer speed. Or, set the RUN-IDLE-STEP switch to STEP and release the switch. The computer executes the BRU and returns to the idle state with the contents of the first instruction of the program displayed in REGISTER DISPLAY, and the address of the first instruction of the program displayed in PROGRAM LOCATION. The operator may continue to cause the computer to execute instructions in this manner by repeatedly setting the RUN-IDLE-STEP switch to STEP, allowing the switch to return to IDLE each time. This process is called "stepping" instructions.

INSPECT MEMORY CONTENTS

1. Set the RUN-IDLE-STEP switch to IDLE.
2. Set the REGISTER switch to C.
3. Press CLEAR and enter a BRU to the memory location to be examined into REGISTER DISPLAY, using the set buttons. Format of the instruction is

   000 000 001 0xx xxx xxx xxx xxx

   BRU Memory location

4. Set the RUN-IDLE-STEP switch to STEP and release the switch. PROGRAM LOCATION now contains the 14-bit address of the location to be inspected and REGISTER DISPLAY contains the 24-bit contents of the location.
5. To inspect other memory locations, repeat steps 3 and 4 above.

MODIFY MEMORY CONTENTS

1. Set the RUN-IDLE-STEP switch to IDLE.
2. Set the REGISTER switch to A.
3. Press CLEAR and enter the desired configuration into the A register, using the set buttons below REGISTER DISPLAY.
4. Set the REGISTER switch to C.
5. Enter 035 XXXXX into REGISTER DISPLAY, using the set buttons. (035 is the octal instruction code for STORE A and XXXXX is the octal address of the memory location to be changed.)
6. Set the RUN-IDLE-STEP switch to STEP and release the switch. The computer executes the STORE A instruction and returns to the idle state.

INSPECT/MODIFY REGISTER CONTENTS

1. Set the RUN-IDLE-STEP switch to IDLE.
2. Set the REGISTER switch to the desired register (A, B, C, or X). The contents of the selected register are immediately displayed in REGISTER DISPLAY and may be changed by pressing CLEAR and inserting a new configuration with the set buttons.
3. Set the REGISTER switch back to C before placing the RUN-IDLE-STEP switch into RUN or STEP.

CLEAR HALT CONDITION

1. Set the RUN-IDLE-STEP switch to IDLE. The Halt flip-flop is now reset.
2. To continue with the displayed instruction, set the RUN-IDLE-STEP switch to RUN (for automatic operation) or to STEP (for single-stepping).

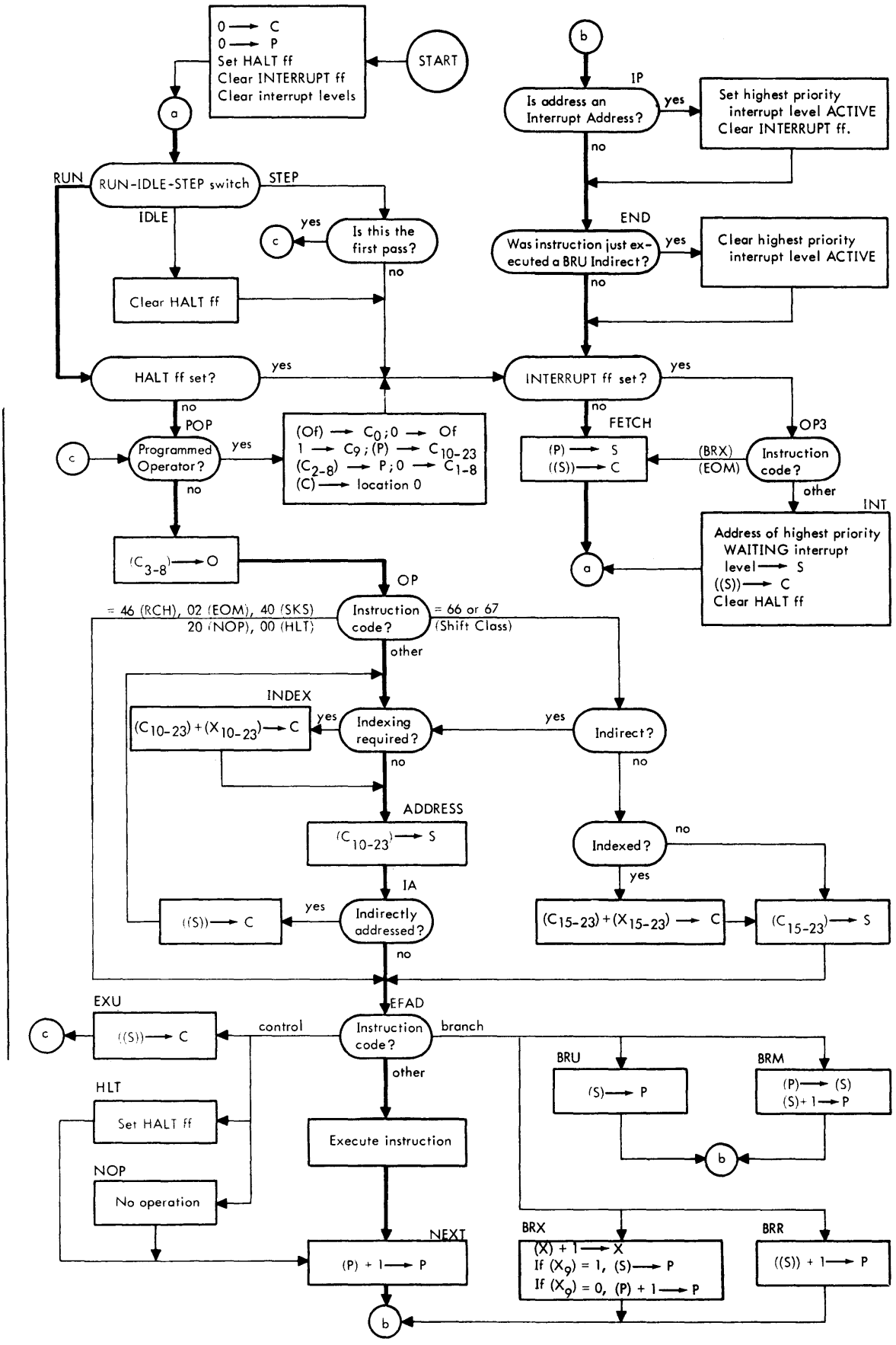


Figure D-1. Instruction Execution Diagram

APPENDIX D DETAILED MACHINE FUNCTIONS

INSTRUCTION EXECUTION

Figure D-1. shows the major relationships between certain operating and program conditions during instruction execution, but does not necessarily correspond to actual computer operations. The following are considered:

START switch
RUN-IDLE-STEP switch
HALT ff (flip-flop)
Programmed Operators
Indexing
Indirect addressing
Control and branch instructions
Subroutine interrupts

The figure assumes that the START switch has been pressed, a program is being executed, and an instruction is in the C register. The following paragraphs provide additional explanations of the functions performed at various steps in the instruction execution cycle. The labels below correspond to the labels that appear in Figure D-1

POP If bit 2 of the instruction is a 1, the instruction is a Programmed Operator. See Appendix E for a detailed discussion of this feature.

OP If bit 2 of the instruction is a zero, the O register contains the 6-bit code for the operation to be performed. Shift and cycle instructions require special address modification, and some other instructions (REGISTER CHANGE, ENERGIZE OUTPUT M, TEST/SKIP, and HALT) do not allow address modification.

INDEX If the instruction is indexed (a 1 in bit 1 of the instruction word), add the address field of the X register to the address field of the C register.

ADDRESS Copy the address field of the C register into the S register.

IA If the instruction operand is indirectly addressed (a 1 in bit 9 of the instruction word), load the C register with the contents of the S register, and go back to check for further address modification.

EFAD After all address modification, the S register contains the address of the effective operand of the operation to be performed.

NEXT For most instructions, the P register is incremented near the end of the execution cycle, in preparation for accessing the next instruction. However, since branch instructions operate directly on the P register, the effect of these instruction is shown. Also shown are control instructions. (Note that the EXU instruction loops back to OP after the effective operand is copied into the C register.)

IP If the instruction just executed was at an interrupt address (single instruction or subroutine), set the interrupt level ACTIVE and clear the INTERRUPT ff.

END If the instruction just executed was a BRU indirect (or was a single-instruction interrupt), clear the highest priority interrupt level in the active state.

FETCH If the INTERRUPT ff is not set, copy the next instruction into the C register and return to (a).

OP3 If the instruction just executed was a BRX or EOM, wait until the next instruction is executed before going to INT.

INT Copy the address of the highest priority WAITING interrupt level into the S register, copy the contents of the memory location specified by the S register into the C register, clear the HALT ff, and return to (a).

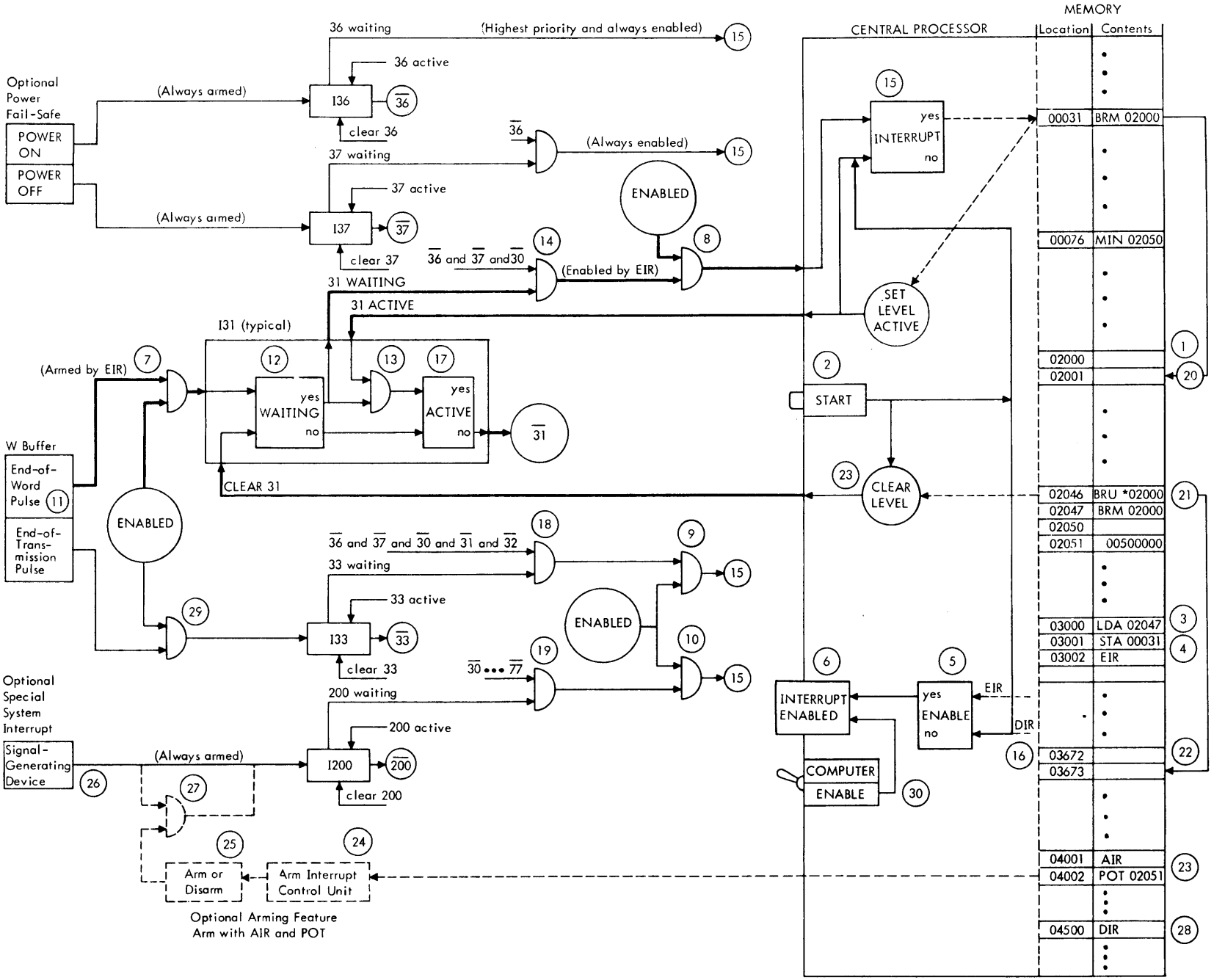


Figure D-2. Priority Interrupt System Diagram

TYPICAL INTERRUPT CYCLE

Figure D-2 shows the progress of a typical interrupt cycle, and does not necessarily reflect actual circuitry. The circled numbers in the paragraphs below refer to specific portions of this figure.

(1) Program (02000-04500) is loaded into memory.

(2) START button is pressed, clearing all interrupts, Arm Interrupt Control Unit (24), Enable ff (5), and the Interrupt (15). Program is entered by means of a BRU 03000 placed in C and RUN-IDLE-STEP switch placed in RUN.

(3) Instructions 03000 and 03001 store the entrance to servicing subroutine in address of W buffer End-of-Word interrupt level (00031)

(4) EIR in location 03002) sets Enable ff (5), turns on INTERRUPT ENABLED indicator (6), arms gate (7), and enables gates (8), (9), (10), etc.

(11) W buffer transmits end-or-word pulse through gate (7) to set the level Waiting ff (12). If the INTERRUPT ENABLED indicator had been off, then the pulse at gate (7) would have been lost.

(12) The Waiting ff presents a steady signal at gates (13) and (14); the interrupt level is now in the waiting state, and remains in the waiting state until cleared by a BRU indirect or by the START switch. The waiting state is not affected by a DISABLE INTERRUPTS (DIR) instruction.

(14) If no higher-priority interrupts (i.e., 36, 37 and 30) are in the active state, the signal passes through the priority gate (14) and through the ENABLE gate (8) to set the Interrupt ff (15). If the INTERRUPT ENABLED indicator had been off, the signal would not pass through gate (8) and the interrupt level would remain in the waiting state.

(16) Assuming that the Interrupt ff is set during the execution cycle of the instruction in 03672, and the instruction is not a BRX or EOM, the S register is set to 031, BRM 02000 in that location is brought out to the C register and executed, with the following results:

a. The contents of the Program Counter (03673) are placed in bits 10-23 of location 02000.

b. An Interrupt Active pulse is transmitted to gate (13) and the level Active ff (17) is set. Lower-priority interrupts are inhibited at priority gates (18), (19), etc.

c. The Interrupt ff is cleared, allowing interrupt levels 36, 37 and 30 to interrupt the servicing subroutine for level 31.

d. Program control is transferred to the second location within the servicing subroutine (location 02001).

(21) At the end of the servicing subroutine (location 02046), execution of BRU *02000 causes the following:

a. The contents of location 02000 (the address at which the program should resume) are placed back into the Program Counter.

b. A clear-interrupt pulse is transmitted to the interrupt level to clear the Waiting ff and the Active ff. The steady signal ($\overline{31}$) is now presented at all lower-priority interrupt levels and they may now interrupt the program.

c. Program control is transferred to the next instruction in sequence after the instruction at which the interrupt occurred.

(22) The interrupted program continues at location 03673.

(23) Instructions in locations 04001 and 04002 instruct the Arm Interrupt Control Unit (24) to set a group of arming flip-flops (52) to allow an interrupt pulse from a signal-generating device (26) to pass through an arming gate (27) to the interrupt level 0200.

(28) DIR (instruction in location 04500) resets the Enable ff, disarms gates (7), (29), etc., disables gates (8), (9), (10), etc., and turns off the INTERRUPT ENABLE INDICATOR.

(30) The INTERRUPT ENABLE switch causes an interrupt enabled condition when it is manually held in the ENABLE position - regardless of the state of the Enable ff.

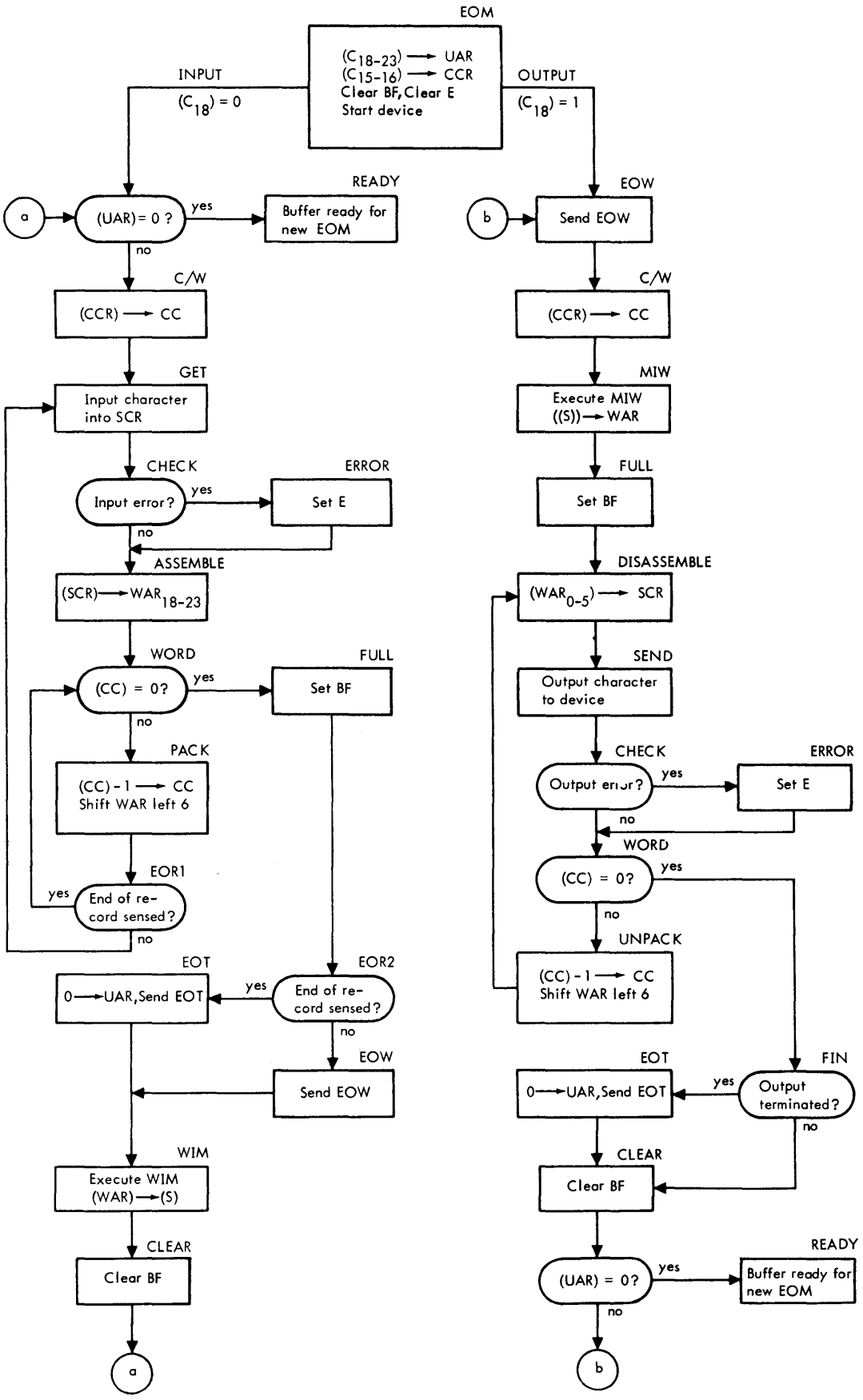


Figure D-3. Buffer Operation, Single-Word Transmission

SINGLE WORD TRANSFER

Figure D-3 shows the major relationships between certain buffer conditions in input/output operations. The following paragraphs refer to this figure and assume the W buffer is being used in the single-word mode of operation, and that buffer interrupts are enabled.

EOM Execution of a buffer control EOM:

1. Places bits 18-23 of the EOM into the buffer unit address register (UAR)
2. Places bits 15 and 16 of the EOM (characters per word) into a character count register (CCR)
3. Clears the buffer full (BF) indicator
4. Clears the buffer error (E) indicator
5. Starts the device specified by the unit address. The contents of bits 18-23 of the EOM are displayed on the control panel in UNIT ADDRESS. Bit 18 of the EOM (the first bit of the 6-bit unit address code) specifies input or output.

INPUT

READY If the UAR contains all zeros, the buffer is currently disconnected and is ready for a buffer control EOM; this will cause a skip if a W BUFFER READY TEST (BRTW) is executed.

C/W Accept a character from the peripheral device specified by the UAR into the single character register (SCR).

CHECK Was there a parity error?

ERROR Set the ERROR indicator on the control panel. This will cause a skip if W BUFFER ERROR TEST (BETW) is executed when the indicator is set.

ASSEMBLE The contents of the SCR are copied into bits 18-23 of a 24-bit word assembly register (WAR).

WORD If the number of characters/word specified by the EOM have been assembled, set BF and go to EOR2.

PACK Decrease CC by 1, and shift WAR 6 places left to make room for new input.

EOR1 If no end-of-record is sensed by the input device, go back to GET for the next character. If an end-of-record is sensed, go back to WORD until CC = zero. Thus, if the last word in an input record does not contain the specified count of characters/word, zeros fill the least-significant portion of that word.

FULL Set BF; word is ready to be stored in memory.

EOR2 If an end-of-record is sensed by the input device, go to EOT. If not, go to EOW (or COUNT).

EOT Clear UAR (disconnect buffer), and generate the End-of-Transmission (EOT) interrupt (I2W).

EOW Generate the End-of-Word (EOW) interrupt (I1W).

WIM Computer executes W BUFFER INTO MEMORY (WIM) instruction.

CLEAR Clear the BF indicator in preparation for the next input word.

OUTPUT

EOW Same as for input.

C/W Same as for input.

MIW Computer executes MEMORY INTO W BUFFER (MIW) instruction.

FULL Same as for input.

DISASSEMBLE Copy the contents of bits 0-5 of the WAR into the SCR.

SEND Transmit the contents of the SCR to the device specified by the UAR.

CHECK Same as for input.

ERROR Same as for input.

WORD Same as for input.

UNPACK Decrease CC by 1, shift WAR left 6 places for new character output.

FIN Has output been terminated with a TERMINATE OUTPUT (TOP) instruction?

EOT Same as for input.

CLEAR Same as for input.

READY Same as for input.

INTERLACE CONTROL

Figure D-4 shows the automatic operations of the buffer during interlaced transmission. The following paragraphs refer to this figure and assume that an interlaced I/O operation has been initiated by a buffer control EOM.

INTERLACE

The word count is stored in the word count register (WCR), the starting address is stored in the memory address register (MAR), and the automatic interlace control begins.

EOR

If an end-of-record signal has been received by the buffer from the peripheral device, the I/O operation is terminated.

FIN

If the specified number of words have been processed, the I/O operation is terminated.

TRANSMIT

During input, the contents of the WAR are copied into the memory location specified by the contents of the MAR. During output, the contents of the memory location specified by the contents of the MAR are copied into the WAR. Words are assembled or disassembled as described in Single-Word Transmission.

NEXT

The contents of the MAR are incremented by 1 and the contents of the WCR are decremented by 1.

EOT

If an end-of-record signal has been received, the unit address register (UAR) is cleared and an End-of-Transmission (EOT) interrupt signal is transmitted to the EOT interrupt level. If the Interrupt System is enabled, a program interrupt occurs.

OUTPUT

When the specified number of words have been transmitted to the peripheral device, the buffer automatically terminates the output operation, clears the UAR, and transmits the EOT interrupt signal.

INPUT

When the specified number of words have been stored in memory, the interlace is disengaged and the buffer returns to the single-word mode of transmission, as depicted in Figure D-3.

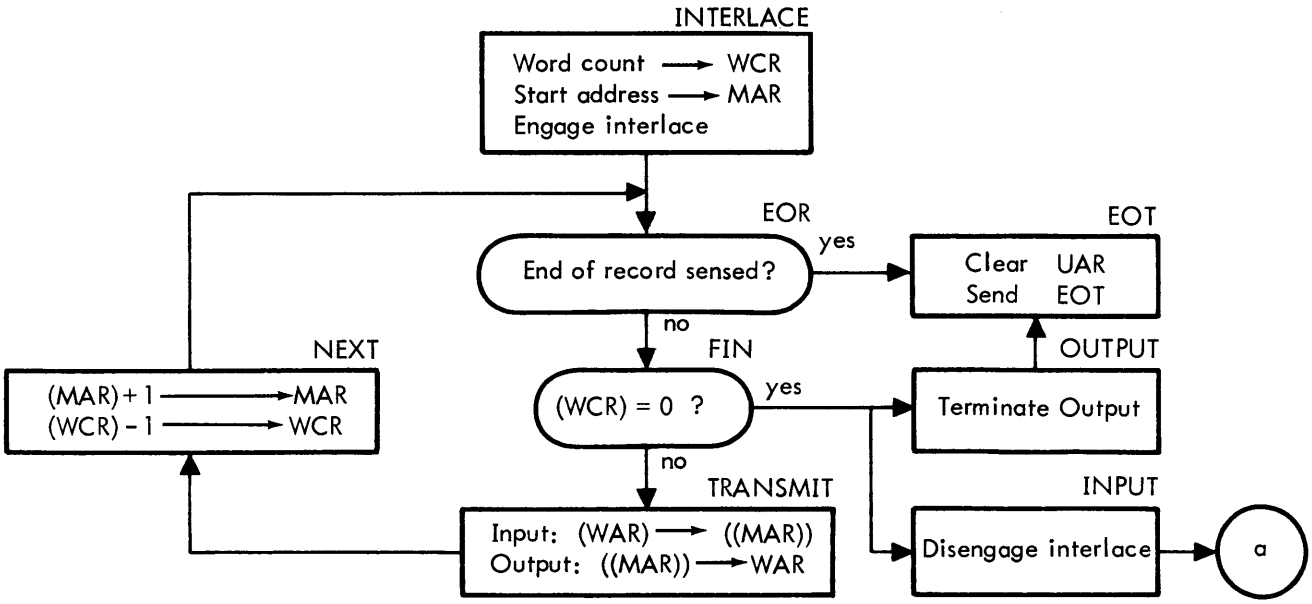


Figure D-4. Buffer Operation, Interlaced Transmission

APPENDIX E PROGRAMMED OPERATORS

The XDS Programmed Operator enables a programmer to code a subroutine call with a single instruction, just as if the subroutine were an actual machine instruction. Other computers usually perform standard subroutine calls by executing a transfer to the starting location of the subroutine and, at the same time, preserving a return address. This procedure requires an operation code (indicating a transfer) and an operand address (indicating the starting address of the subroutine). If the subroutine should require an additional operand, as in a floating point add subroutine, for example, the calling sequence must be longer to accommodate the specification of the operand.

The XDS Programmed Operator (abbreviated POP) uses the operation code to indicate the transfer address. When the computer detects a "one" in bit position 2 of an instruction, bit positions 2 through 8 are not interpreted as a normal instruction, but instead, are treated as an address to which the computer transfers control. Thus the operand address field is free to designate an address for use by the subroutine. There are 64 (decimal) locations $[(100)_8$ through $(177)_8]$ to which a transfer may occur. These 64 locations constitute a linkage table; they normally contain appropriate unconditional transfer instructions (BRU) to maintain the communication link between the POP code and the subroutine being called by it.

The location from which the transfer is made, at the time the computer detects the POP code (that is, the contents of the P Register), is preserved in location 0. Thus the normal BRR instruction may be used to leave the POP subroutine and return to the main program. Also, the state of the overflow toggle is preserved in the sign bit position of location 0 and the overflow toggle is immediately reset.

To allow access to the operand in the main program by the POP subroutine, bit position 9 (the indirect address bit) is unconditionally set to "one". In this manner, when the subroutine refers indirectly to location 0, the indirect addressing is perpetuated one more level, thereby enabling the subroutine to gain access to the operand in the main program. (See the Programmed Operator Example for further explanation.)

By judicious use of the programmed operator principle, a one-to-one correspondence may be maintained between XDS 930 instructions and XDS 925 instructions. For example, XMA is a 930 machine instruction; its function may be simulated on the XDS 925 by a subroutine, and this subroutine may be called by means of a programmed operator. Thus, the main program requires the same number of instructions for either the XDS 925 or 930.

Another advantage of the programmed operator is the ability to change the arithmetic mode of a program without recoding the arithmetic portions of the program. For example, if the programmer codes all arithmetic instructions as programmed operators, he could simply change the arithmetic subroutine package and, hence, the arithmetic mode of the main program.

The following operations take place when the computer detects a programmed operator:

1. $(P) \rightarrow (0)_{10-23}$; save P Register for return address
2. $1 \rightarrow (0)_9$; insert indirect address bit
3. $(O_f) \rightarrow (0)_s$; preserve status of overflow toggle
4. $(C)_{2-8} \rightarrow (P)$; branch to location indicated in POP code

A library of programmed operator subroutines is available to greatly extend the XDS 925 instruction list. A list of these subroutines is given in this appendix. Each subroutine is specified by a unique mnemonic code and represents an available instruction which may be used directly in preparing 925 programs. Up to 64 of these programmed operator instructions may be used to prepare any one program.

The program loading system automatically organizes the interconnection between the programmed operator instructions and the corresponding subroutines. Each programmed operator instruction mnemonic code is converted on input to an instruction code of 100_8 to 177_8. A memory location from 100_8 through 177_8 corresponding to each assigned instruction code is loaded with an unconditional branch to the corresponding subroutine.

XDS 925 PROGRAMMED OPERATOR EXAMPLE

Location	Instruction	Effective Address	Contents of Effective Address	Location 0	O_f	A Register	B Register
01342	1 XMA 02163	02163	77777777		Set	25252525	01234567
00162	0 BRU 00300			40041342	Reset		
00300	0 STB 00306						
00301	0 LBD *00000	02163	77777777			25252525	77777777
00302	0 STA *00000	02163					
00303	0 XAB 00014						
00304	0 LDB 00306	00306	01234567			77777777	01234567
00305	0 BRR 00000				Set		
01343	Continue in main program						

Explanation: XMA is a programmed operator that exchanges the contents of the A Register and the effective address. The contents of the B and X Registers are unaffected by this "instruction". In this example, XMA is POP code 162.

PROGRAMMED OPERATOR INSTRUCTIONS

Mnemonic	Name
ADM	Add A to M
ATD	Arctangent of A – Double Precision
ATF	Arctangent of A – Floating Point
ATN	Arctangent of A
BDD	Binary to Decimal Conversion – Double Precision, Fixed Point
BDF	Binary to Decimal Conversion – Double Precision, Floating Point
BFS	Binary to Decimal Conversion – Single Precision, Floating Point
BID	Binary to Decimal Conversion – Single Precision, Fixed Point
CAB	Copy A into B
CAX	Copy A into Index
CBA	Copy B into A
CBX	Copy B into Index
CLA	Clear A
CLB	Clear B
CNA	Copy Negative into A
COS	Cosine of A
CSD	Cosine of A – Double Precision
CSF	Cosine of A – Floating Point
CXA	Copy Index into A
CXB	Copy Index into B
DBD	Decimal to Binary Conversion – Double Precision, Fixed Point
DBF	Decimal to Binary Conversion – Double Precision, Floating Point
DFS	Decimal to Binary Conversion – Single Precision, Floating Point
DIB	Decimal to Binary Conversion – Single Precision, Fixed Point
DIV	Divide
DPA	Double Precision Add
DPD	Double Precision Divide
DPM	Double Precision Multiply
DPN	Double Precision Negate
DPS	Double Precision Subtract
DSQ	Double Precision Square Root
EXF	Exponential of A – Floating Point
EXP	Exponential of A
FFF	Fixed-Floating Format Conversion
FLA	Floating Add – Double Precision
FLD	Floating Divide – Double Precision
FLM	Floating Multiply – Double Precision
FLN	Floating Negate – Double Precision
FLS	Floating Subtract – Double Precision
FSA	Floating Add – Single Precision
FSD	Floating Divide – Single Precision
FSM	Floating Multiply – Single Precision
FSN	Floating Negate – Single Precision
FSS	Floating Subtract – Single Precision
FSQ	Floating Square Root – Single Precision
LDE	Load Exponent
LDP	Load Double Precision
LGF	Logarithm of A – Floating Point
LOG	Logarithm of A
LQP	Load Quadruple Precision
LRSH	Logical Right Shift
LTP	Load Triple Precision
MUL	Multiply
SIN	Sine of A
SKB	Skip If M and B Do Not Compare Ones
SKD	Difference Exponents and Skip
SKE	Skip If A Equals M
SKR	Reduce M, Skip If Negative
SND	Sine of A – Double Precision
SNF	Sine of A – Floating Point
SQR	Square Root of A
STD	Store Double Precision
STE	Store Exponent
STP	Store Triple Precision
STQ	Store Quadruple Precision
XEE	Exchange Exponents
XMA	Exchange M and A
XXA	Exchange Index and A
XXB	Exchange Index and B

APPENDIX F CHANNEL MEMORY ACCESS PRIORITY

During each memory cycle the control unit interrogates each channel to determine if it needs access to memory. If only one channel requires memory access, the control unit permits the channel to proceed immediately. If more than one channel requires memory access, the determination of which one proceeds first is on the basis of a fixed and a variable priority. The fixed priority is in the order: Direct Access Channel, Time-Multiplexed Channel, and Central Processor. Time-Multiplexed Channels have fixed priority in the order: D, C, Y, and W. Direct Access Channels have variable priority that is normally determined by comparison of the Word Assembly Register in each channel. The channel whose Word Assembly Register has the fewest number of characters remaining to be filled is selected for memory access. For example, if the Word Assembly Register in Channel E has one character position unfilled and the Word Assembly Register in Channel F has three character positions unfilled, Channel E is selected. Thus, each channel increases its priority level as each character is read into the Word Assembly Register. If the contents of the register in two or more channels are equal in characters to be filled and no other channel in the set has fewer characters to be filled, priority is determined in sequence, with Channel E having top priority.

Note that the number of characters to be placed in the Word Assembly Register at any time is dependent on the characters per word count specified for the transmission. Consider, for example, that in Channel E the character count is three characters per word and in Channel F the character count is four characters per word. If both Channel F and Channel E need access to memory simultaneously, and if both have two characters filled in their respective Word Assembly Registers, then Channel E gets first memory access since it has only one character place to be filled.

APPENDIX G OPTIONAL EQUIPMENT

DATA MULTIPLEXING SYSTEM

The standard I/O systems provided with the XDS 925 Computer provide for operation with all standard XDS peripheral equipments and for high-performance special devices. The Data Multiplexing System provides an alternate I/O system that is of particular use in dealing with multiple source of data and for systems which may have very high data rates (see Figure G-1).

The XDS 925 Computer has essentially two major paths along which I/O data can flow to and from memory. The first path is the same that is used by the main frame itself. The PIN/POT operations use the first path. All Time-Multiplexed Communication Channels also use this path. In addition to this path, which is primarily under the control of the main frame, there is an optional second path that is completely under the control of the units attached to it. The second path has priority over the first for access to memory. This path is made available with the installation of the Multiple Access to Memory Feature.

MULTIPLE ACCESS TO MEMORY FEATURE (MAM)

The Multiple Access feature provides the necessary modules on both main frame and memory to permit memories to be accessed via the second path. A word can be transferred over the path in either direction in one cycle. If the computer is equipped with two or more memories and the main frame is communicating with one memory while some other device is using the second path to another memory, then there is no interference with computation. If both the main frame and an I/O device using the second path address the same memory, the second path has priority; the program loses one cycle while the second path transmits.

The Multiple Access feature is required for the attachment of Direct Access Communication Channels (DACC), Data Multiplex Channels (DMC), or Memory Interface Connections (MIC). These devices all incorporate a priority scheme for determining the assignment of the second path. (See Figure G-1.) Only four DACCs can be attached to one computer system; Memory Interface Connections, and Data Multiplex Channels, however, are unlimited in number.

A practically unlimited number of MICs in addition to the four DACCs and the Data Multiplexing System (DMS) can be attached to a computer system. Each MIC has the necessary priority control to operate with other MICs and DACCs and the DMS. Both MICs and DACCs can be arranged so as to produce any required configuration of priorities.

DATA MULTIPLEXING BASIC ELEMENTS

A Data Multiplexing System consists of two basic elements:

1. The Data Multiplex Channel (DMC) for communicating with several data sources/destinations and for synchronizing I/O operations with memory, MICs, DACCs, and other DMCs.

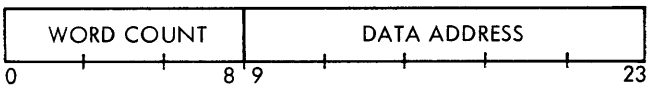
2. One or more Data Subchannels (DSC) for interfacing between peripheral devices and systems and the DMC.

Data Multiplex Channel (DMC)

The Data Multiplex Channel is the basic unit for the Data Multiplexing System. It connects to the second path to memory via the Multiple Access to Memory feature. A DMC consists of 24-bit register and control logic. All addresses and data are transmitted between the DMC and subchannels via a bus system. The data and address are connected to memory via the MAM only when a transfer is to be made. All program control required for a given I/O operation operates directly on the individual subchannel, not the DMC.

The DMC is equipped with an internal interlace feature. This feature allows a subchannel to specify the address of a word in memory where the data address and count are to be found. When operating with internal interlace, the subchannel supplies the address of its interlace word instead of the actual data address. The DMC reads out the interlace word, increments the address portion, decrements the count, restores the word and then accepts the data from or transmits the data to the subchannel. The DMC also supplies a signal to the subchannel if the decremented count is zero.

The format of the internal interlace word or word pair is as shown:



The 9-bit word count allows for block lengths to 512 words. With the 925, transmissions using internal interlace require 3 cycles per word.

The DMC also provides for automatic memory incrementing. The counting capability of the DMC register is such that the entire 24-bit register or either the upper 12 bits or the lower 12 bits may be incremented. When such a memory increment operation is to be performed, the subchannel signals the DMC with a special increment line and supplies the address. The DMC reads out the word, increments it and then restores. If the word was zero after the incrementing, the DMC signals the subchannel which may then interrupt the program. The maximum incrementing rate is 1 count every 2 cycles. Parity generation and detection are available.

Data Subchannels (DSC-N)

There are a number of subchannels which can be attached to the DMC. A full word, 24 bits plus parity, is available for the 925. Words (24 bits) are assembled in two 12-bit characters. Subchannels can control and generate program interrupts but do not include the interrupt levels themselves. The signals must be routed to optional interrupt levels if the interrupt features are to be used.

The subchannels use a priority scheme to determine which may transmit to the DMC at any given time. This is similar to the scheme used by the MICs, DMCs, and in transmitting to memory. Up to 128 DSCs can be connected to a DMC. A DSC can use the internal interlace feature of the DMC to control its transmission or it can be equipped with an external interlace (EIN).

A DSC using internal interlace has two words assigned to it. These two words are adjacent even/odd locations and are fixed for a given subchannel. The program can select either the even or odd location. If the even location is selected, the subchannel will automatically switch to the odd location when the count field of the even word is zero. The program can also select whether or not the subchannel will switch back to the even word when the count field of the odd word is zero. The subchannel will generate an interrupt signal when the count field of either word reaches zero. Transmission termination occurs when the odd word's count equals zero if the subchannel does not switch back to the even word.

The two-word internal interlace allows a subchannel to handle continuous data by alternately working from one memory area or another. By allowing the subchannel to switch automatically from one interlace word to the other, the program is relieved of the necessity for making real-time responses to the zero count condition. Using first the even then the odd interlace word allows maximum word count of 1024 for a pair of interlace words.

CHARACTER SUBCHANNEL (DSC-I)

The DSC-I contains a 12-bit data register that can assemble and disassemble two 6-bit characters, and transmit one or two 6-bit characters or one 12-bit character. It checks and generates the parity of characters to enable it to couple with standard XDS peripherals. The DSC-I has a unit address register. For the 925, it can be used for multiple typewriters or other character-oriented devices. However, it only uses 12 bits of the full 24-bit word.

The subchannel can operate with either internal or external interlace. It has one mode of output and two modes of input. During output, it transmits until the odd internal interlace word count is zero and then terminates if interlace cycling is not requested. The output can also be terminated if the device sends an END signal to the channel. This END signal may cause the DSC-I to generate an interrupt to the program.

Input, like output, can always be terminated due to an external END signal. The program can also specify if the DSC is to terminate and disconnect on zero count or disconnect only on the END signal. In either case, however, all transmission to memory is terminated after the odd interlace count reaches zero if interlace cycling is not requested.

FULL-WORD SUBCHANNEL (DSC-II)

The DSC-II is a general purpose subchannel designed to allow communication with word-oriented input/output units such as analog-digital and digital-analog converters. It contains no storage for data. The external device must be capable of holding the data during the transmission to/from the DMC. (An A-to-D converter would have such capability). Like the DSC-I, the DSC-II can operate with either internal or external interlace. Its operation in this respect is identical to that of the DSC-I. The DSC-II also contains control logic to facilitate memory increment operations in conjunction with the DMC.

EXTERNAL INTERLACE

The external interlace (EIN) can be attached to the DSC to control the transmission of its data to/from memory. The EIN consists of a 15-bit address register and a 9-bit count register. These registers are loaded automatically when the subchannel is activated, the information coming from the internal interlace memory locations. Once the EIN is set up, it will control the transmissions of the DSC at a maximum rate of 1 word per memory cycle. After each word is transmitted, the EIN increments its address register and decrements its count. When the count equals zero, the EIN signals the DSC, which can then generate a program interrupt and/or notify the external device. Transmission normally terminates on zero count. Sequencing of interlace words is identical to the sequence of operation performed for internal interlace, except that only two memory cycles are used for interlace word processing. The first is to access the interlace word initially; the second is to restore the interlace word when the count reaches zero.

PROGRAM CONTROL OF DATA SUBCHANNELS

Transmission of data between a DSC and computer memory is controlled by two 24-bit interlace control words unique to the DSC and wired into fixed adjacent locations in memory. During a transmission the DMC/DSC uses the two interlace control words for determination of transmission address and record length.

The DSCs are numbered from 0 to 0376 in even octal numbers; this permits a maximum of 128 subchannels. The memory locations of the interlace control word pairs associated with the DSCs are numbered X0000, X0001 for DSC-0, X0002, X0003 for DSC-2 ... X0376, X0377 for DSC-376. DSC-I numbering need not be contiguous, but DSC-II's are configured one or two in a module and are numbered with adjacent numbers. If a system contains multiple DSC-II modules (each with 1 or 2 subchannels), the module numbering need not be contiguous; 4, 0 and 0224, 0220 and 0314 is a typical possibility for five DSC-II subchannels. Transmissions to and from the DSC and memory may be under internal interlace control or, when so equipped, under external interlace control.

INTERNAL INTERLACE

During an internal interlace transmission, the DMC controls the interlacing operation in the following order:

1. Access Interlace Word

   The DMC accesses the interlace word assigned to the requesting subchannel.

2. Process Interlace Word

   The DMC increments the 15-bit address portion of the word and decrements the 9-bit word count.

3. Test for Zero and Set Indicator

   Next, the DMC tests the word count for zero and if it is zero, sets an indicator in the pertinent DSC.

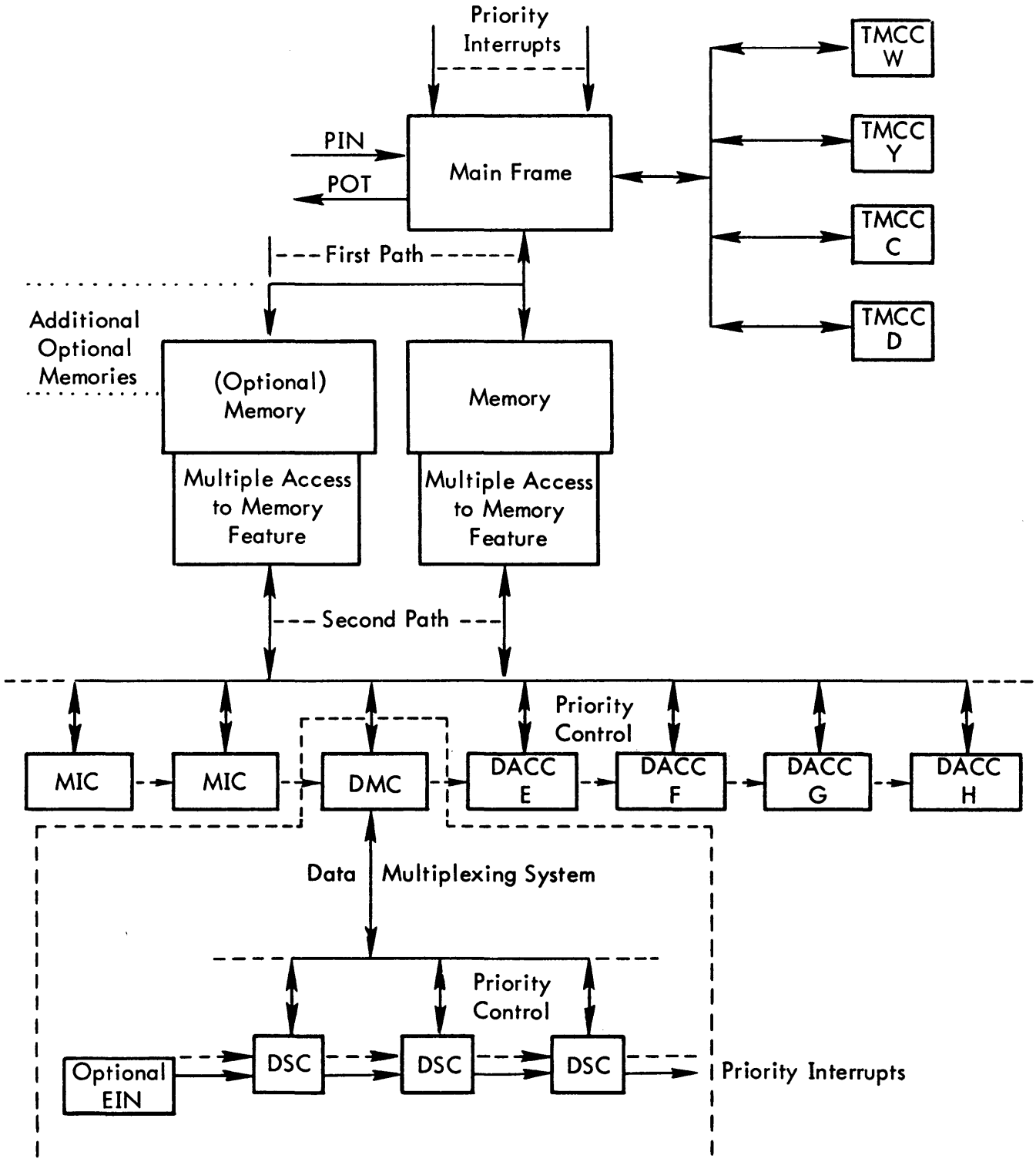


Figure G-1. XDS 925 Overall Computer Configuration

4. Restore

The DMC then places the new word count/address values back into memory using the assigned address of requesting subchannel.

5. Access/Store as Requested

The DMC accesses or stores the transmitted word as requested using the incremental address (see above).

6. Stop or Continue

The DSC checks its zero count indicator and

a. if zero and working on the even interlace word, the DSC continues operation using the odd interlace word,

b. if zero, working on the odd interlace word and the cycle bit is set, the DSC continues using the even interlace word,

c. if zero, working on the odd interlace word and the cycle bit is reset, the DSC terminates the operation on a DSC-II or responds as required by the function control on a DSC-I.

d. if not zero, the DSC returns operation to the DMC to continue at 1 (above).

Note that the first address used is the "address specified plus one" and the first word count is the "word count specified minus one". In particular, an initial word count of zero causes a 512-word block to be transmitted.

EXTERNAL INTERLACE CONTROL

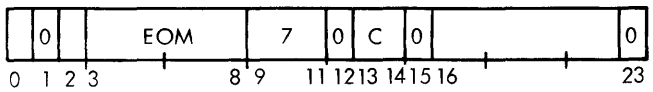
During transmissions utilizing external interlace control, the interlacing operation proceeds as described above except that when the DSC is activated, the DSC with external interlace (EIN) requests the DMC to access the desired interlace control word. The interlace control word is sent to the EIN. Thereafter, data transmissions to and from the DSC to memory utilize the interlace address and word count supplied by the EIN.

Data transmissions using the EIN require only one cycle while those data transmissions using internal interlace require three cycles. Should a transmission result in the EIN detecting a zero word count condition, the DSC-EIN will restore the external interlace word and will proceed according to 6 (above). Any termination of a DSC operation prior to zero word count due to any externally derived halt signal also causes a restoring of the EIN interlace control word.

DSC PROGRAMMING

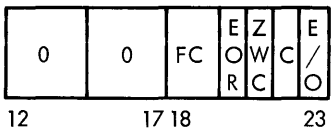
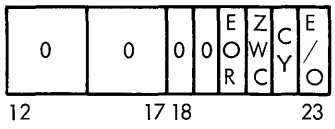
An EOM, POT sequence selects, alerts, and controls the subchannel; an EOM, SKS sequence selects and tests the status and conditions of the subchannel.

The EOM has the form:



and is referred to as the "select EOM".

Bit positions 16-23 contain the DSC number being alerted; these numbers are the even numbers from 0 to 0376 for the DSC and the C field (bits 13, 14) specifies one of three modes to which the DSC is alerted. When followed by a POT instruction, the modes have the following meaning:

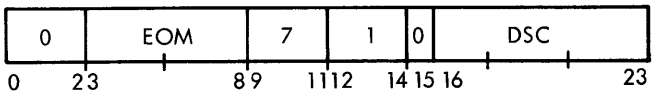
C	Effect
00	The subchannel decodes the lower 12 bits (12-23) of the "POTted" word as the lower 12 bits of a buffer control mode EOM. For DSC-I, this will select a device with the unit address field, set the character/word count, specify binary or BCD format, forward or reverse, and leader or no leader. For DSC-II, the 12 bits activate the subchannel and select the proper unit (if more than one is attached to the DSC).
01	The subchannel decodes the lower 12 bits of the "POTted" word as the lower 12 bits of an input/output control mode EOM. If bits 18 through 23 are zero, the "POTted" word addresses the selected DSC. For DSC-I, these bits perform such functions as rewind tape, space paper, etc. For DSC-II, these bits perform such functions as required by the selected device attached to the DSC.
10	The subchannel decodes the lower 12 bits of the "POTted" word for controlling the interlace and interrupts. The control type EOM should precede the buffer control EOM. For DSC-I the form is:  FC is a 2-bit function code similar to the TMCC/DACC terminal function codes. The remaining bits function as described below for DSC-II. For DSC-II, the form is: 

Bit Position	Function
20	A 1 in the EOR bit arms the End-of-Record interrupt for this channel.
21	A 1 in the ZWC bit arms the Zero Word Count interrupt.

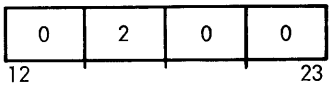
Bit Position	Function
22	A 1 in the CY bit (cycle) sets the cycle mode such that the interlace will switch from the odd word back to the even word at the zeroing of the odd word count. If ZWC and CY are set, a zero count interrupt is generated each time the interlace switches (to either word - even or odd). If CY is set to 0, the interlace will not proceed after the count of the odd word is zero; and a zero count interrupt occurs only when the count of the odd word is zero.
23	A 0 in the E/O bit selects the even interlace word as the first insterlace word in a transmission; note that when starting on the even word, the interlace always switches to the odd word for further control when the even word count goes to zero. A 1 in E/O sets the odd interlace word as the first interlace word in a transmission; the interlace ceases control when the odd word count reaches zero unless the C bit is set to cycle.

TERMINATING DSC INPUT/OUTPUT

Once the cycle bit has been set, the interlace continues to cycle back and forth between the even/odd interlace words. An EOM, POT sequence is used to terminate the cycle. The EOM is:



The lower 12 bits of the "POTted" word must be:

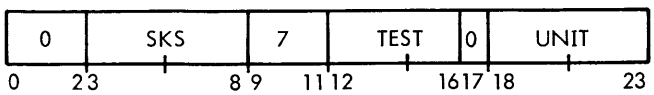


The interlace terminates the next time the count reaches zero in the odd interlace word.

For example, to terminate the cycle on DSC 4, use the following sequence:

```
EOM 071004
POT 010000
 :
 :
010000 00000200
```

The SKS to test subchannels has the form:



A select EOM with C equal to zero (C = 0) permits the SKS to be directed to the subchannel or to the device attached to it. The UNIT field specifies the device to be tested; the TEST field is defined for the particular device.

When testing the subchannel, the UNIT field is set to 00. The TEST field contains the same testing format as SKS for testing a TMCC.

For example, to test DSC 4 for error, use the following sequence:

```
EOM 070004
SKS 071000
```

MEMORY INTERFACE CONNECTION

Once a computer is equipped with a multiple access to memory feature, one or more memory interface connections (MIC) can be attached. The MIC is a general interface between the computer and the outside world that allows special devices to be connected to the computer. The MIC converts between the 4-volt logic levels used in the computer and the 8 volts used outside. It preserves the integrity of the memory by generating the parity of incoming data words. It will also check the parity of words read from memory to indicate memory failures. If incoming data is supplied with parity, the MIC will check for odd parity as it generates the internal memory parity and respond with a signal that indicates if the transmission was correct. The device that is connected to the MIC must store both the data and the address until the transmission to/from memory is completed.

AUTOMATIC POWER FAIL-SAFE SYSTEM

The computer core memory holds its information with all power removed, but information in the computer registers is destroyed by loss of power. Upon failure of main power to the computer, this system provides that the contents of all registers and other volatile information are automatically stored in core memory; also, further writing into core storage is inhibited during the decay period of the computer dc power supply outputs. Erroneous memory control is prevented during power-off and power-on operations. Power-off/-on interrupt routines permit proper resumption of a program, automatically, after power is restored. This solid-state system consists of ac power-sensing and memory-sequencing circuitry, two high-priority interrupt channels, and a "shut-down/start-up" programming sequence.

The SKIP IF SIGNAL NOT SET (SKS) instruction is an aid in programming this option. Its address is 024000. If the OFF interrupt (37) has just occurred, the computer executes the next instruction in sequence (does not skip).

MEMORY PARITY INTERRUPTS

XDS computers incorporate an extensive memory parity checking system. The inclusion of parity generation and checking circuitry assures the integrity of data and instructions transferred among the memory, the central processing unit, and input/output channels.

In normal operation a switch on the computer console specifies the action to be performed by the computer when a memory parity error is detected. Two actions are available: the computer halts with the parity indicator lighted; or the computer ignores the parity error and proceeds with the program.

In many real-time applications it is desirable to keep the computer running when a parity error is detected. Also, the program must be notified of the error without stopping computation.

An optional feature provides this capability by means of two levels of enabled interrupts. One interrupt level is associated with the central processor; the other interrupt level with the Direct Access Communication Channels and the Data Multiplexing System. Memory parity errors detected from these two sources produce a priority interrupt associated with the cause. The processing routine associated with the interrupt can then take appropriate action, such as reinitiate the failed operation, notify the operator, or enter a diagnostic routine. Such action allows memory parity errors to be recognized and handled properly without hindering the computer's performance of real-time or on-line calculations.

REAL-TIME CLOCK

The Real-Time Clock (RTC) provides a flexible time-orientation system for the XDS 925 Computer. It derives time pulses from the 60-cycle computer power supply. These pulses are then used to produce a timing mark every 16.67 milliseconds, or optionally every 8.33 milliseconds. The Real-Time Clock can also accept timing marks from a customer-supplied input, thereby allowing time measurement to any required resolution for special applications. These timing marks are supplied at standard XDS logic levels to the computer's RTC circuitry.

The timing marks are then used by the computer and its interrupt system to provide either an elapsed-time counter or a continuously incrementing time counter depending on the needs of the customer. The RTC operates in either mode depending only on the computer's stored program.

Location	Type	Computer	Description
074	Normal	925	CLOCK SYNC
075	Single Instruction	925	CLOCK PULSE

The Clock Pulse and Clock Sync interrupts function together to provide elapsed-time, event counter, or time-of-day clocks.

The Clock Pulse interrupt is a single-instruction interrupt. (Note: See Single Instruction Interrupts in Section 3.) An MIN instruction is usually placed in the Clock Pulse interrupt location. When MIN is used as a single-instruction interrupt subroutine, it causes the contents of the effective address to be incremented by one. Furthermore, if the new (incremented) contents of the effective address is 0000, a Clock Sync interrupt is generated. The Clock Sync interrupt can be generated in no other way.

ELAPSED-TIME CLOCK

The elapsed-time clock times the length of a program or subroutine, or initiates or discontinues processing at program-determined time intervals. An arbitrary memory location is reserved as a counter. When initialized, this cell contains the 2's complement of the number of time intervals to be counted. The Clock Pulse interrupt location contains an SKR instruction.

Each Clock Pulse interrupt results in decrementing the clock count by one. When the count is finished, an interrupt to the Clock Sync location occurs. A supervisory or other appropriate control program can then be entered to perform the customer-desired operation.

CONTINUOUSLY INCREMENTING CLOCK

The continuously incrementing clock maintains "time-of-day" for the computer. One memory location serves to count the timing marks. In this case, the Clock Pulse is used to increment this location. (The Clock Pulse interrupt location contains an MIN instruction.) A simple, straightforward subroutine can be entered to reconstruct the exact time-of-day from this twenty-four bit count.

ARM/DISARM

The Clock Pulse interrupt can be armed and disarmed with these instructions.

EOM Effective Address	Action
20200	Disarm Clock Pulse Interrupt
20100	Arm Clock Pulse Interrupt

The Clock Sync interrupt is always armed.

APPENDIX H XDS 925 INSTRUCTION LISTS

INSTRUCTION LIST - FUNCTIONAL CATEGORIES

Mnemonic	Instruction Code	Name	Function	Timing	Page Ref.
LOAD/STORE					
LDA	76	Load A	(M) → A	2	9
STA	35	Store A	(A) → M	3	9
LDB	75	Load B	(M) → B	2	9
STB	36	Store B	(B) → M	3	9
LDX	71	Load Index	(M) → X	2	9
STX	37	Store Index	(X) → M	3	9
EAX	77	Copy Effective Address into Index	Effective Address → X	2	10
ARITHMETIC					
ADD	55	Add M to A	(A) + (M) → A	2	10
MIN	61	Memory Increment	(M) + 1 → M	3	10
MDE	60	Memory Decrement	(M) - 1 → M	3	10
SUB	54	Subtract M from A	(A) - (M) → A	2	10
MUS	64	Multiply Step	Add or Subtract (M) from (A) based on B21-23; 1/4 AB → AB	2	10
DIS	65	Divide Step	2AB → AB $\overline{A}_o \rightarrow B_{23}$ If $(A_o) = (M_o)$, (A) - (M) → A; If $(A_o) \neq (M_o)$, (A) + (M) → A	2	11
LOGICAL					
ETR	14	Extract	(A) and (M) → A	2	11
MRG	16	Merge	(A) or (M) → A	2	11
EOR	17	Exclusive OR	(M)($\overline{A}$) or ($\overline{M}$)(A) → A	2	11
REGISTER CHANGE					
XAB	0 46 00000	Exchange A and B	(A) ⟷ (B)	1	12
BAC	0 46 10000	Copy B into A, Clear B	(B) → A, 0 → B	1	12
ABC	0 46 20000	Copy A into B, Clear A	(A) → B, 0 → A	1	12
CLR	0 46 30000	Clear AB	0 → AB	1	12
BRANCH					
BRU	01	Branch Unconditionally	M → P	1	12
BRX	41	Increment Index and Branch	(X) + 1 → X If X Neg., M → P If X Pos., P + 1 → P	1, 2	12

Mnemonic	Instruction Code	Name	Function	Timing	Page Ref.
BRANCH (cont.)					
BRM	43	Mark Place and Branch	(P) → M; M + 1 → P	2	13
BRR	51	Return Branch	(M) + 1 → P	2	13
TEST/SKIP					
SKS	40	Skip if Signal not Set	If Signal = 1, P + 1 → P If Signal = 0, P + 2 → P	2, 3	25,35, 36
SKG	73	Skip if A Greater than M	If (A) ≤ (M), P + 1 → P If (A) > (M), P + 2 → P	2, 3	13
SKM	70	Skip if A = M on B Mask	If (B)(A) ≠ (B)(M), P + 1 → P If (B)(A) = (B)(M), P + 2 → P	2, 3	13
SKN	53	Skip if M Negative	If (M) ≥ 0, P + 1 → P If (M) < 0, P + 2 → P	2, 3	14
SKA	72	Skip if M and A do not Compare Ones	If (A)(M) ≠ 0, P + 1 → P If (A)(M) = 0, P + 2 → P	2, 3	14
SHIFT					
RSH	0 66 000XX	Right Shift AB	AB Shift Right N Places	2-19	15
RCY	0 66 200XX	Right Cycle AB	AB Cycled Right N Places	2-19	15
LSH	0 67 000XX	Left Shift AB	AB Shift Left N Places	2-19	15
LCY	0 67 200XX	Left Cycle AB	AB Cycled Left N Places	2-19	16
NOD	0 67 100XX	Normalize and Decrement X	AB Left and X - 1 → X until $A_0 \neq A_1$, or N Shifts	2 + the number of places shifted	16
CONTROL					
HLT	00	Halt	Halts Computation	1	16
NOP	20	No Operation	--	1	16
EXU	23	Execute	Instruction M is performed, P unchanged	1	16
BREAKPOINT TESTS					
BPT 1	0 40 20400	Breakpoint No. 1 Test	Test Breakpoint Switch	1, 2	17
BPT 2	0 40 20200	Breakpoint No. 2 Test	Test Breakpoint Switch	1, 2	17
BPT 3	0 40 20100	Breakpoint No. 3 Test	Test Breakpoint Switch	1, 2	17
BPT 4	0 40 20040	Breakpoint No. 4 Test	Test Breakpoint Switch	1, 2	17
OVERFLOW					
OVT	0 40 20001	Overflow Indicator Test and Reset	Test Overflow Indicator	1, 2	17
ROV	0 02 20001	Reset Overflow	Turn Off Overflow Indication	1	17

Mnemonic	Instruction Code	Name	Function	Timing	Page Ref.
INTERRUPT					
EIR	0 02 20002	Enable Interrupt System		1	21
DIR	0 02 20004	Disable Interrupt System		1	21
IET	0 40 20004	Interrupt Enabled Test	Skip if Interrupt System Enabled	1, 2	21
IDT	0 40 20002	Interrupt Disabled Test	Skip if Interrupt System Disabled	1, 2	21
AIR	0 02 20020	Arm Interrupts		1	21
CHANNEL CONTROL					
ALC 0	0 02 50000	Alert Channel W	(For codes of other channels, see page 31.)	1	31
DSC 0	0 02 00000	Disconnect Channel W	(For codes of other channels, see page 31.)	1	31
ASC 0	0 02 12000	Alert to Store Address in Channel W	(For codes of other channels, see page 31.)	1	31
TOP 0	0 02 14000	Terminate Output on Channel W	(For codes of other channels, see page 31.)	1	31
CAT 0	0 40 14000	Channel W Active Test; Skip if Channel Inactive	(For codes of other channels, see page 35.)	2-3	35
CET 0	0 40 11000	Channel W Error Test; Skip if no Error	(For codes of other channels, see page 35.)	2-3	35
CZT 0	0 40 12000	Channel W Zero Count Test; Skip if Count = 0	(For codes of other channels, see page 36.)	2-3	36
CIT 0	0 40 10400	Channel W Interrecord Test	(For codes of other channels, see page 36.)	2-3	36
INPUT/OUTPUT					
MIW	12	M into W Buffer when Empty	(M) → W	2+wait	36
WIM	32	W Buffer into M when Full	(W) → M	3+wait	37
MIY	10	M into Y Buffer when Empty	(M) → Y	2+wait	37
YIM	30	Y Buffer into M when Full	(Y) → M	3+wait	37
POT	13	Parallel Output	(M) → Unit M in Parallel	3+wait	39
BPO	11	Block Parallel Output		3+N+wait	39
PIN	33	Parallel Input	(Unit M) → M in Parallel	4+wait	39
BPI	31	Block Parallel Input		4+N+wait	39
EOM	02	Energize Output M	1.75 μsec Pulse to Point(s) Addressed	1	24,29, 32,40
EOD	06	Energize Output to Direct Access Channels		1	25,31
BETW	0 40 20010	W Buffer Error Test		2, 3	35
BETY	0 40 20020	Y Buffer Error Test		2, 3	35

Mnemonic	Instruction Code	Name	Function	Timing	Page Ref.
BRTW	0 40 21000	W Buffer Ready Test		2, 3	35
BRTY	0 40 22000	Y Buffer Ready Test		2, 3	35
TYPEWRITER					
RKB 0, 1, 4	0 02 00601	Read Keyboard		1	44
TYP 0, 1, 4	0 02 00641	Write Typewriter		1	44
PAPER TAPE					
RPT 0, 1, 4	0 02 02604	Read Paper Tape		1	47
PTL 0, 1, 4	0 02 00644	Punch Paper Tape with Leader		1	47
PPT 0, 1, 4	0 02 02644	Punch Paper Tape with no Leader		1	47
PUNCHED CARD					
CRT 0, 1	0 40 12006	Card Reader Ready Test		2, 3	51
CFT 0, 1	0 40 11006	Card Reader End-of-File Test		2, 3	51
RCD 0, 1, 4	0 02 02606	Read Card Decimal (Hollerith)		1	51
RCB 0, 1, 4	0 02 03606	Read Card Binary		1	51
CPT 0, 1	0 40 14046	Card Punch Ready Test		2, 3	51
PCD 0, 1, 4	0 02 02646	Punch Card Decimal (Hollerith)		1	51
PCB 0, 1, 4	0 02 03646	Punch Card Binary		1	51
FCT 0, 1	0 40 14006	First Column Test		2, 3	51
PBT 0, 1	0 40 12046	Punch Buffer Test		2, 3	51
SRC 0, 1	0 02 12006	Skip Remainder of Card		1	51
MAGNETIC TAPE					
TRT 0, n	0 40 1041n	Tape Ready Test		2, 3	55
FPT 0, n	0 40 1401n	File Protect Test		2, 3	55
BTT 0, n	0 40 1201n	Beginning of Tape Test		2, 3	55
ETT 0, n	0 40 1101n	End of Tape Test		2, 3	56
DT2, 0, n	0 40 1621n	Density Test, 200 BPI		2, 3	56
DT5, 0, n	0 40 1661n	Density Test, 556 BPI		2, 3	56
DT8, 0, n	0 40 1721n	Density Test, 800 BPI		2, 3	56
TFT 0	0 40 13610	Tape End-of-File Test		2, 3	56
TGT 0, n	0 40 1261n	Tape Gap Test		2, 3	56
WTB 0, n, 4	0 02 0365n	Write Tape in Binary		1	56
WTD 0, n, 4	0 02 0265n	Write Tape in Decimal (BCD)		1	56

Mnemonic	Instruction Code	Name	Function	Timing	Page Ref.
EFT 0, 4	0 02 0367n	Erase Tape Forward		1	56
ERT 0, n, 4	0 02 0767n	Erase Tape in Reverse		1	56
RTB 0, n, 4	0 02 0361n	Read Tape in Binary		1	56
RTD 0, n, 4	0 02 0261n	Read Tape in Decimal (BCD)		1	56
SFB 0, n, 4	0 02 0363n	Scan Forward in Binary		1	56
SFD 0, n, 4	0 02 0263n	Scan Forward in Decimal (BCD)		1	56
SRB 0, n, 4	0 02 0763n	Scan Reverse in Binary		1	56
SRD 0, n, 4	0 02 0663n	Scan Reverse in Decimal (BCD)		1	56
REW 0, n	0 02 1401n	Rewind		1	56
RTS 0	0 02 14000	Convert Read to Scan		1	56
SRR 0	0 02 13610	Skip Remainder of Record		1	56
	0 40 1021n	MAGPAK Test		2, 3	56
PRINTER					
PRT 0, 1	0 40 12060	Printer Ready Test		2, 3	61
EPT 0, 1	0 40 14060	End of Page Test		2, 3	61
PFT 0, 1	0 40 11060	Printer Fault Test		2, 3	61
POL 0, 1	0 02 10260	Printer Off-line		1	61
PSC 0, 1, N	0 02 1N460	Printer Skip to Channel N		1	61
PSP 0, 1, N	0 02 1N660	Printer Space N Lines		1	61
PLP 0, 1, 4	0 02 02660	Print Line Printer		1	61

INSTRUCTION LIST - NUMERICAL ORDER

Instruction Code	Mnemonic	Name	Page Ref.
00	HLT	Halt	16
01	BRU	Branch Unconditionally	12
02	EOM	Energize Output M	24, 29, 32, 40
0 02 00000	DSC 0	Disconnect Channel W	31
0 02 00601	RKB 0, 1, 4	Read Keyboard	44
0 02 00641	TYP 0, 1, 4	Write Typewriter	44
0 02 00644	PTL 0, 1, 4	Punch Paper Tape with Leader	47
0 02 02604	RPT 0, 1, 4	Read Paper Tape	47
0 02 02606	RCD 0, 1, 4	Read Card Decimal (Hollerith)	51
0 02 0261n	RTD 0, n, 4	Read Tape in Decimal (BCD)	56
0 02 0263n	SFD 0, n, 4	Scan Forward in Decimal (BCD)	56
0 02 02644	PPT 0, 1, 4	Punch Paper Tape with no Leader	47
0 02 02646	PDC 0, 1, 4	Punch Card Decimal (Hollerith)	51
0 02 0265n	WTD 0, n, 4	Write Tape in Decimal (BCD)	56
0 02 02660	PLP 0, 1, 4	Print Line Printer	61
0 02 03606	RCB 0, 1, 4	Read Card Binary	51
0 02 0361n	RTB 0, n, 4	Read Tape in Binary	56
0 02 0363n	SFB 0, n, 4	Scan Forward in Binary	64
0 02 03646	PCB 0, 1, 4	Punch Card in Binary	51
0 02 0365n	WTB 0, n, 4	Write Tape in Binary	56
0 02 0367n	EFT n, 4	Erase Tape Forward	56
0 02 0663n	SRD 0, n, 4	Scan Reverse in Decimal (BCD)	56
0 02 0763n	SRB 0, n, 4	Scan Reverse in Binary	56
0 02 0767n	ERT 0, n, 4	Erase Tape in Reverse	56
0 02 10260	POL 0, 1	Printer Off-line	39
0 02 12000	ASC 0	Alert to Store Address in Channel W	31
0 02 12006	SRC 0, 1	Skip Remainder of Card	56
0 02 13610	SRR 0	Skip Remainder of Record	56
0 02 14000	RTS 0	Convert Read to Scan	56
0 02 14000	TOP 0	Terminate Output on Channel W	31
0 02 1401n	REW 0, n	Rewind	56
0 02 1N460	PSC 0, 1, N	Printer Skip to Channel N	61
0 02 1N660	PSP 0, 1, N	Printer Space N Lines	61
0 02 20001	ROV	Reset Overflow	17
0 02 20002	EIR	Enable Interrupt System	21

Instruction Code	Mnemonic	Name	Page Ref.
0 02 20004	DIR	Disable Interrupt System	21
0 02 20020	AIR	Arm Interrupts	21
0 02 50000	ALC 0	Alert Channel W	31
06	EOD	Energize Output to Direct Access Channels	25, 31, 32
10	MIY	M into Y Buffer when Empty	37
11	BPO	Block Parallel Output	39
12	MIW	M into W Buffer when Empty	36
13	POT	Parallel Output	39
14	ETR	Extract	11
16	MRG	Merge	11
17	EOR	Exclusive OR	11
20	NOP	No Operation	16
23	EXU	Execute	16
30	YIM	Y Buffer into M when Full	37
31	BPI	Block Parallel Input	39
32	WIM	W Buffer into M when Full	37
33	PIN	Parallel Input	39
35	STA	Store A	9
36	STB	Store B	9
37	STX	Store Index	9
40	SKS	Skip if Signal not Set	25, 40
0 40 1021n		MAGPAK Test	56
0 40 10400	CIT	Channel W Interrecord Test	36
0 40 1041n	TRT 0, n	Tape Ready Test	55
0 40 11000	CET 0	Channel W Error Test; Skip if No Error	35
0 40 11006	CFT 0, 1	Card Reader End-of-File Test	51
0 40 1101n	ETT 0, n	End of Tape Test	56
0 40 11060	PFT 0, 1	Printer Fault Test	61
0 40 12000	CZT 0	Channel W Zero Count Test; Skip if Count = 0	36
0 40 12006	CRT 0, 1	Card Reader Ready Test	51
0 40 1201n	BTT 0, n	Beginning of Tape Test	55
0 40 12046	PBT 0, 1	Punch Buffer Test	51
0 40 12060	PRT 0, 1	Printer Ready Test	61
0 40 1261n	TGT 0, n	Tape Gap Test	56
0 40 13610	TFT 0	Tape End-of-File Test	56
0 40 14006	FCT 0, 1	First Column Test	51
0 40 1401n	FPT 0, n	File Protect Test	55
0 40 14046	CPT 0, 1	Card Punch Ready Test	51

Instruction Code	Mnemonic	Name	Page Ref.
0 40 14060	EPT 0, 1	End of Page Test	61
0 40 1621n	DT2, 0, n	Density Test, 200 BPI	56
0 40 1661n	DT5 0, n	Density Test, 556 BPI	56
0 40 1721n	DT8 0, n	Density Test, 800 BPI	56
0 40 20001	OVT	Overflow Indicator Test and Reset	17
0 40 20002	IDT	Interrupt Disabled Test	21
0 40 20004	IET	Interrupt Enabled Test	21
0 40 20010	BETW	W Buffer Error Test	35
0 40 20020	BETY	Y Buffer Error Test	35
0 40 20040	BPT 4	Breakpoint No. 4 Test	17
0 40 20100	BPT 3	Breakpoint No. 3 Test	17
0 40 20200	BPT 2	Breakpoint No. 2 Test	17
0 40 20400	BPT 1	Breakpoint No. 1 Test	17
0 40 21000	BRTW	W Buffer Ready Test	35
0 40 22000	BRTY	Y Buffer Ready Test	35
41	BRX	Increment Index and Branch	12
43	BRM	Mark Place and Branch	13
0 46 00000	XAB	Exchange A and B	12
0 46 10000	BAC	Copy B into A, Clear B	12
0 46 20000	ABC	Copy A into B, Clear A	12
0 46 30000	CLR	Clear AB	12
51	BRR	Return Branch	13
53	SKN	Skip if M Negative	14
54	SUB	Subtract M from A	10
55	ADD	Add M to A	10
60	MDE	Memory Decrement	10
61	MIN	Memory Increment	10
64	MUS	Multiply Step	10
65	DIS	Divide Step	11
0 66 00XXX	RSH	Right Shift AB	15
0 66 20XXX	RCY	Right Cycle AB	15
0 67 00XXX	LSH	Left Shift AB	15
0 67 10XXX	NOD	Normalize and Decrement X	16
0 67 20XXX	LCY	Left Cycle AB	16
70	SKM	Skip if A = M on B Mask	13
71	LDX	Load Index	9
72	SKA	Skip if M and A do not Compare Ones	14
73	SKG	Skip if A Greater than M	13
75	LDB	Load B	9
76	LDA	Load A	9
77	EAX	Copy Effective Address into Index	10

INSTRUCTION LIST - ALPHABETICAL ORDER

Mnemonic	Instruction Code	Name	Page Ref.
ABC	0 46 20000	Copy A into B, Clear A	12
ADD	55	Add M to A	10
AIR	0 02 20020	Arm Interrupts	21
ALC 0	0 02 50000	Alert Channel W	31
ASC 0	0 02 12000	Alert to Store Address in Channel W	31
BAC	0 46 10000	Copy B into A, Clear B	12
BETW	0 40 20010	W Buffer Error Test	35
BETY	0 40 20020	Y Buffer Error Test	35
BPI	31	Block Parallel Input	39
BPO	11	Block Parallel Output	39
BPT 1	0 40 20400	Breakpoint No. 1 Test	17
BPT 2	0 40 20200	Breakpoint No. 2 Test	17
BPT 3	0 40 20100	Breakpoint No. 3 Test	17
BPT 4	0 40 20040	Breakpoint No. 4 Test	17
BRM	43	Mark Place and Branch	13
BRR	51	Return Branch	13
BRTW	0 40 21000	W Buffer Ready Test	35
BRTY	0 40 22000	Y Buffer Ready Test	35
BRU	01	Branch Unconditionally	12
BRX	41	Increment Index and Branch	12
BTT 0, n	0 40 1201n	Beginning of Tape Test	55
CAT 0	0 40 14000	Channel W Active Test; Skip if Channel Inactive	35
CET 0	0 40 11000	Channel W Error Test; Skip if no Error	35
CFT 0, 1	0 40 11006	Card Reader End-of-File Test	51
CIT 0	0 40 10400	Channel W Interrecord Test	36
CLR	0 46 30000	Clear AB	12
CPT 0, 1	0 40 14046	Card Punch Ready Test	51
CRT 0, 1	0 40 12006	Card Reader Ready Test	51
CZT 0	0 40 12000	Channel W Zero Count Test; Skip if Count = 0	36
DIR	0 02 20004	Disable Interrupt System	21
DIS	65	Divide Step	11
DSC 0	0 02 00000	Disconnect Channel W	31
DT2 0, n	0 40 1621n	Density Test, 200 BPI	56
DT5 0, n	0 40 1661n	Density Test, 556 BPI	56
DT8 0, n	0 40 1721n	Density Test, 800 BPI	56

Mnemonic	Instruction Code	Name	Page Ref.
EAX	77	Copy Effective Address into Index	10
EFT n, 4	0 02 0367n	Erase Tape Forward	56
EIR	0 02 20002	Enable Interrupt System	21
EOD	06	Energize Output to Direct Access Channels	25, 31, 32
EOM	02	Energize Output M	24,29,32,40
EOR	17	Exclusive OR	11
EPT 0, 1	0 40 14060	End of Page Test	61
ERT 0, n, 4	0 02 0767n	Erase Tape in Reverse	56
ETR	14	Extract	11
ETT 0, n	0 40 1101n	End of Tape Test	56
EXU	23	Execute	16
FCT 0, 1	0 40 14006	First Column Test	51
FPT 0, n	0 40 1401n	File Protect Test	55
HLT	00	Halt	16
IDT	0 40 20002	Interrupt Disabled Test	21
IET	0 40 20004	Interrupt Enabled Test	21
IORD		Input/Output of a Record and Disconnect	33
IORP		Input/Output of a Record and Proceed	33
IOSD		Input/Output until Signal then Disconnect	33
IOSP		Input/Output until Signal then Proceed	34
LCY	0 67 20XXX	Left Cycle AB	16
LDA	76	Load A	9
LDB	75	Load B	9
LDX	71	Load Index	9
LSH	0 67 00XXX	Left Shift AB	15
MDE	60	Memory Decrement	10
MIN	61	Memory Increment	10
MIW	12	M into W Buffer when Empty	36
MIY	10	M into Y Buffer when Empty	37
MRG	16	Merge	11
MUS	64	Multiply Step	10
NOD	0 67 10XXX	Normalize and Decrement X	16
NOP	20	No Operation	16
OVT	0 40 20001	Overflow Indicator Test and Reset	17
PBT 0, 1	0 40 12046	Punch Buffer Test	51
PCB 0, 1, 4	0 02 03646	Punch Card Binary	51
PCD 0, 1, 4	0 02 02646	Punch Card Decimal (Hollerith)	51
PFT 0, 1	0 40 11060	Printer Fault Test	61

Mnemonic	Instruction Code	Name	Page Ref.
PIN	33	Parallel Input	39
PLP 0, 1, 4	0 02 02660	Print Line Printer	61
POL 0, 1	0 02 10260	Printer Off-line	62
POT	13	Parallel Output	39
PPT 0, 1, 4	0 02 02644	Punch Paper Tape with no Leader	47
PRT 0, 1	0 40 12060	Printer Ready Test	61
PSC 0, 1, N	0 02 1N460	Printer Skip to Channel N	61
PSP 0, 1, N	0 02 1N660	Printer Space N Lines	61
PTL 0, 1, 4	0 02 00644	Punch Paper Tape with Leader	47
RCB 0, 1, 4	0 02 03606	Read Card Binary	51
RCD 0, 1, 4	0 02 02606	Read Card Decimal (Hollerith)	51
RCY	0 66 20XXX	Right Cycle AB	15
REW 0, n	0 02 1401n	Rewind	56
RKB 0, 1, 4	0 02 00601	Read Keyboard	44
ROV	0 02 20001	Reset Overflow	17
RPT 0, 1, 4	0 02 02604	Read Paper Tape	47
RSH	0 66 00XXX	Right Shift AB	15
RTB 0, n, 4	0 02 0361n	Read Tape in Binary	56
RTD 0, n, 4	0 02 0261n	Read Tape in Decimal (BCD)	56
RTS 0	0 02 14000	Convert Read to Scan	56
SFB 0, n, 4	0 02 0363n	Scan Forward in Binary	56
SFD 0, n, 4	0 02 0263n	Scan Forward in Decimal (BCD)	56
SKA	72	Skip if M and A do not Compare Ones	14
SKG	73	Skip if A Greater than M	13
SKM	70	Skip if A = M on B Mask	13
SKN	53	Skip if M Negative	14
SKS	40	Skip if Signal not Set	25, 40
SRB 0, n, 4	0 02 0763n	Scan Reverse in Binary	56
SRC 0, 1	0 02 12006	Skip Remainder of Card	51
SRD 0, n, 4	0 02 0663n	Scan Reverse in Decimal (BCD)	56
SRR 0	0 02 13610	Skip Remainder of Record	56
STA	35	Store A	9
STB	36	Store B	9
STX	37	Store Index	9
SUB	54	Subtract M from A	10
TFT 0	0 40 13610	Tape End-of-File Test	56
TGT 0, n	0 40 1261n	Tape Gap Test	56
TOP 0	0 02 14000	Terminate Output on Channel W	31

Mnemonic	Instruction Code	Name	Page Ref.
TRT 0, n	0 40 1041n	Tape Ready Test	55
TYP 0, 1, 4	0 02 00641	Write Typewriter	44
WIM	32	W Buffer into M when Full	37
WTB 0, n, 4	0 02 0365n	Write Tape in Binary	56
WTD 0, n, 4	0 02 0265n	Write Tape in Decimal (BCD)	56
XAB	0 46 00000	Exchange A and B	12
YIM	30	Y Buffer into M when Full	37

INDEX

A

Address modification, 5
Arithmetic instructions, 10
Arming feature, 21
Arm interrupts, 21
Automatic power fail-safe option, A-26

B

Backspace magnetic tape, 55
Block parallel input/output, 39
Branch instructions, 12
Breakpoint tests, 17

C

Card punch, 51
Card reader, 50
Central processor registers, 3
Character
    codes, A-1
    subchannel, A-23
Communication channels, 23, 26
Compatible mode, 32, 41
Computer configuration, 2
Control
    console, 42, 43
    instructions, 16

D

Data
    Multiplex System, 23, A-22
    Multiplex Channel, A-22
    subchannel, A-22
    subchannel programming, A-25
Direct Access Channels
Disable interrupt, 20
Displays, 24

E

Enable interrupt, 20
Extended mode, 22, 32
External interlace, A-23

F

Fixed-point format, 5
Floating-point
    format, 5
    operations, 17
Format
    card, 50
    control characters, 63
    instruction word, 4
    magnetic tape, 54
    memory word, 4
    paper tape, 47
Full-word subchannel, A-23

G

General Description, 1

H

Halt, 16, 42

I

Indexing, 5
Indirect addressing, 6
Input/Output
    card, 50
    communication channel, 26
    direct parallel, 24, 39
    EOM, 29, 40
    instructions, 24
    magnetic tape, 54
    paper tape, 47
    single-bit, 40
    typewriter, 38
Interlace register, 28
Internal interlace, A-23
Interrupt
    control, 20
    enable/disable, 20
    location assignments, 20
    priority, 19
    single-instruction, 19

L

Line Printer, 60
Load/Store instructions, 9
Logical instructions, 11

M

Magnetic tape
    format, 54
    reading, 54
    writing, 55
Memory
    Interface connection, A-26
    parity interrupts, A-26
    SDS 925, 4
    word formats, 4
Multiple Access to Memory feature, A-22

N

Non-interruptible instructions, 20
Normalize, 16

O

Off-line printing, 61
Overflow, 6, 17

P

Paper tape reader, 47
Paper tape punch, 47
Peripheral equipment, 47
Printing, off-line, 61
Priority
    Channel memory access, A-21
    interrupt, 19
Programmed Operators, 6, A-19, A-20

R

Reading magnetic tape, 54
Real-time clock, A-27
Register change instructions, 12
Registers
    interlace, 28
    SDS 925, 3
    Time-Multiplexed Channel, 27

S

Scan magnetic tape, 55, 56
Shift instructions, 14
Single-word
    operations, 37
    transfer, 36
SKS
    channel and device, 35
    standard, 35
Special characteristics, 5
Store, 9
Subroutine execution, 7
Switches, 42

T

Terminal functions, 33
Tests
    card reader, 51
    card punch, 51
    line printer, 61
    magnetic tape, 55, 56
Test/Skip instructions, 13
Time-Multiplexed Communication Channels, 23, 27
Typewriter, 38

U

Unit address codes, 30

W

Writing magnetic tape, 55

XDS 925 INPUT/OUTPUT INSTRUCTIONS

Mnemonics and octal codes given are for Channel W, device number 1 (or n), and the 4 characters/word mode.

Mnemonic	Octal Code	Name	Page Ref.
CHANNEL CONTROL			
EOM A	02	Energize Output M	24, 29, 32, 40
EOD A	06	Energize Output to Direct Access Channel	25, 31
ALC 0	0 02 50000	Alert Channel W	31
DSC 0	0 02 00000	Disconnect Channel W	31
ASC 0	0 02 12000	Alert to Store Address in Channel W	31
TOP 0	0 02 14000	Terminate Output on Channel W	31
CHANNEL TESTS			
SKS A	40	Skip if Signal Not Set	35
BRTW	0 40 21000	W Buffer Ready Test	35
BRTY	0 40 22000	Y Buffer Ready Test	35
BETW	0 40 20010	W Buffer Error Test	35
BETY	0 40 20020	Y Buffer Error Test	35
CAT 0	0 40 14000	Channel W Active Test; Skip if Inactive	35
CET 0	0 40 11000	Channel W Error Test; Skip if No Error	35
CZT 0	0 40 12000	Channel W Zero Count Test; Skip if Count = 0	36
CIT 0	0 40 10400	Channel Inter-record Test; Skip if Inter-record Indicator Is Set	36
SINGLE-WORD DATA TRANSFER			
MIW A, T	12	Memory Into Channel W When Empty	36
MIY A, T	10	Memory Into Channel Y When Empty	37
WIM A, T	32	Channel W Into Memory When Full	37
YIM A, T	30	Channel Y Into Memory When Full	37
DIRECT PARALLEL INPUT/OUTPUT			
PIN A, T	33	Parallel Input	39
BPI A, T	31	Block Parallel Input	39
POT A, T	13	Parallel Output	39
BPO A, T	11	Block Parallel Output	39

Peripheral Device Instructions and Tests

Mnemonic	Octal Code	Name	Page Ref.
PAPER TAPE			
RPT 0, 1, 4	0 02 02604	Read Paper Tape	47
PPT 0, 1, 4	0 02 02644	Punch Paper Tape Without Leader	47
PTL 0, 1, 4	0 02 00644	Punch Paper Tape With Leader	47
CARD			
RCB 0, 1, 4	0 02 03606	Read Card Binary	51
RCD 0, 1, 4	0 02 02606	Read Card Decimal (Hollerith)	51
SRC 0, 1	0 02 12006	Skip Remainder of Card	51
CRT 0, 1	0 40 12006	Card Reader Ready Test	51
FCT 0, 1	0 40 14006	First Column Test	51
CFT 0, 1	0 40 11006	Card Reader EOF Test	51
PCD 0, 1, 4	0 02 02646	Punch Card Decimal (Hollerith)	51
PCB 0, 1, 4	0 02 03646	Punch Card Binary	51
CARD (cont.)			
PBT 0, 1	0 02 12046	Punch Buffer Test	51
CPT 0, 1	0 02 14046	Card Punch Ready Test	51
LINE PRINTER			
PLP 0, 1, 4	0 02 02660	Print Line Printer	61
POL 0, 1	0 02 10260	Printer Off-line	61
PSC 0, 1, n	0 02 1n460	Printer Skip to Format Channel n	61
PSP 0, 1, n	0 02 1n660	Printer Upspace n Lines	61
PFT 0, 1	0 40 11060	Printer Fault Test	61
PRT 0, 1	0 40 12060	Printer Ready Test	61
EPT 0, 1	0 40 14060	End of Page Test	61
TYPEWRITER			
RKB 0, 1, 4	0 02 02601	Read Keyboard	44
TYP 0, 1, 4	0 02 02641	Write Typewriter	44
MAGNETIC TAPE			
TRT 0, n	0 40 1041n	Tape Ready Test	55
FPT 0, n	0 40 1401n	File Protect Test	55
BTT 0, n	0 40 1201n	Beginning of Tape Test	55
ETT 0, n	0 40 1101n	End of Tape Test	56
DT2 0, n	0 40 1621n	Density Test, 200 BPI	56
DT5 0, n	0 40 1661n	Density Test, 556 BPI	56
DT8 0, n	0 40 1721n	Density Test, 800 BPI	56
TFT 0	0 40 13610	Tape End-of-File Test	56
TGT 0	0 40 12610	Tape Gap Test	56
	0 40 1021n	MAGPAK Test	56
RTB 0, n, 4	0 02 0361n	Read Tape Binary	56
RTD 0, n, 4	0 02 0261n	Read Tape Decimal (BCD)	56
SFB 0, n, 4	0 02 0363n	Scan Forward Binary	56
SFD 0, n, 4	0 02 0263n	Scan Forward Decimal (BCD)	56
SRB 0, n, 4	0 02 0763n	Scan Reverse Binary	56
SRD 0, n, 4	0 02 0663n	Scan Reverse Decimal (BCD)	56
REW 0, n	0 02 1401n	Rewind	56
RTS 0	0 02 14000	Convert Read to Scan	56
SRR 0	0 02 13610	Skip Remainder of Record	56
WTB 0, n, 4	0 02 0365n	Write Tape Binary	56
WTD 0, n, 4	0 02 0265n	Write Tape Decimal (BCD)	56
EFT 0, n, 4	0 02 0367n	Erase Tape Forward	56
ERT 0, n, 4	0 02 0767n	Erase Tape in Reverse	56

OCTAL CODE CHANNEL SELECTION

Channel	EOM (02)	SKS (40)
W	00000000	00000000
Y	00000100	00000100
C	20000000	20000000
D	20000100	20000100
E	00400000	00040000
F	00400100	00040100
G	20400000	20040000
H	20400100	20040100

Legend: A = address; T = tag field; n = number ($0 \leq n \leq 7$)

XDS

Xerox Data Systems 701 South Aviation Blvd. / El Segundo, California 90245 (213) 772-4511 / Cable SCIDATA / Telex 674839 / TWX 910-325-6908

EASTERN TECHNOLOGY CENTER
12150 Parklawn Drive
Rockville, Maryland 20852
(301) 933-5900

PRINTED CIRCUITS DEPT.
600 East Bonita Avenue
Pomona, Calif. 91767
(714) 624-8011

TECHNICAL TRAINING
5250 West Century Blvd.
Los Angeles, Calif. 90045
(213) 772-4511

INTERNATIONAL MANUFACTURING SUBSIDIARY

Scientific Data Systems Israel, Ltd.
P.O. Box 5101
Haifa, Israel
04-530253, 04-64589
Telex 922 4474

SALES OFFICES

Western Region

Building Arts Bldg.
Suite G100
5045 N. 12th St.
Phoenix, Arizona 85014
(602) 264-9324

1360 So. Anaheim Blvd.
Anaheim, Calif. 92805
(714) 774-0461

5250 West Century Blvd.
Los Angeles, Calif. 90045
(213) 772-4511

Vista Del Lago Office Center
122 Saratoga Avenue
Santa Clara, Calif. 95050
(408) 246-8330

13701 Riverside Drive
Sherman Oaks, Calif. 91403
(213) 986-5510

*Wells Fargo Building
Suite 410
Del Amo Financial Center
21535 Hawthorne Blvd.
Torrance, Calif. 90503
(213) 542-5561

3333 South Bannock
Suite 400
Englewood, Colo. 80110
(303) 761-2645

320 Ward Avenue
Honolulu, Hawaii 96814
(808) 531-8257

Fountain Professional Bldg.
9004 Menaul Blvd., N.E.
Albuquerque, N.M. 87112
(505) 298-7683

El Paso Natural Gas Bldg.
Suite 201
315 E. 2nd South Street
Salt Lake City, Utah 84111
(801) 322-0501

400 Building
Suite 415
400 108th Avenue N.E.
Bellevue, Wash. 98004
(206) 454-3991

Midwestern Region

*International Towers Building
6th Floor
8550 West Bryn Mawr Avenue
Des Plaines, Illinois 60018
(312) 693-6060

Clausen Bldg., Suite 310
16000 W. Nine Mile Road
Southfield, Michigan 48075
(313) 353-7360

4410 Woodson Road
Suite 111
St. Louis, Missouri 63134
(314) 423-6200

One Erieview Plaza
13th Floor
Cleveland, Ohio 44114
(216) 522-1850

Seven Parkway Center
Suite 238
Pittsburgh, Pa. 15220
(412) 921-3640

Southern Region

State National Bank Bldg.
Suite 620
200 W. Court Square
Huntsville, Alabama 35801
(205) 539-5131

Orlando Executive Center
1080 Woodcock Road
Orlando, Florida 32803
(305) 841-6371

2964 Peachtree Road, N.W.
Suite 350
Atlanta, Georgia 30305
(404) 261-5323

Jefferson Bank Bldg.
Suite 720
3525 N. Causeway Blvd.
Metairie, Louisiana 70002
(504) 837-1515

4920 S. Lewis Avenue
Suite 103
Tulsa, Oklahoma 74105
(918) 743-7753

One Turtle Creek Village
Dallas, Texas 75219
(214) 528-6580

*2300 West Loop South
Suite 150
Houston, Texas 77027
(713) 623-0510

Eastern Region

10227 Wincopin Circle
Suite 716
Columbia, Maryland 21043
(301) 730-4900

20 Walnut Street
Wellesley Hills, Mass. 02181
(617) 237-2300

Brearley Office Building
190 Moore Street
Hackensack, N. J. 07601
(201) 489-0100

The Fortune Building
280 North Central Avenue
Hartsdale, New York 10530
(914) 948-2929

*1301 Avenue of the Americas
New York City, N.Y. 10019
(212) 765-1230

673 Panorama Trail West
Rochester, New York 14625
(716) 586-1500

P.O. Box 168
535 Pennsylvania Ave.
Ft. Washington Industrial Park
Ft. Washington, Pa. 19034
(215) 643-4250

Kogerama Building
Suite 212
No. 1 Tidewater
Executive Center
Norfolk, Virginia 23502
(703) 497-6811

Washington (D.C.) Operations

*2351 Research Blvd.
Rockville, Maryland 20850
(301) 948-8190

Canada

864 Lady Ellen Place
Ottawa 3, Ontario
(613) 722-8387

1009 7th Avenue, S.W.
Calgary 2, Alberta
(403) 265-8134

280 Belfield Road
Rexdale 605, Ontario
(416) 677-8422

1901 North Service Road
Trans-Canada Highway
Dorval, P.Q.
(514) 683-3755

INTERNATIONAL OFFICES & REPRESENTATIVES

European/African Headquarters

Scientific Data Systems
York House, Empire Way
Wembley, Middlesex
HA 9 OQB
England
(01) 903-2511, Telex 27992

Sweden

Nordisk Elektronik AB
Stureplan 3
Stockholm 7
(08) 24 83 40

Denmark

A/S Nordisk Elektronik
Danasvej 2
Copenhagen V
EVA 8285/EVA 8238

Norway

Nordisk Elektronik (Norge) A/S
Middelthunsgt. 27
Oslo 3
(2) 60 25 90

France

Compagnie Internationale
pour l'Informatique, C.I.I.
66, Route de Versailles
78-Louveciennes
Yvelines
951 86 00 (Paris area)

Israel

Elbit Computers Ltd.
Subsidiary of Elron
Electronic Industries Ltd.
88 Hagiborim Street
Haifa
6 4613

*Regional Headquarters