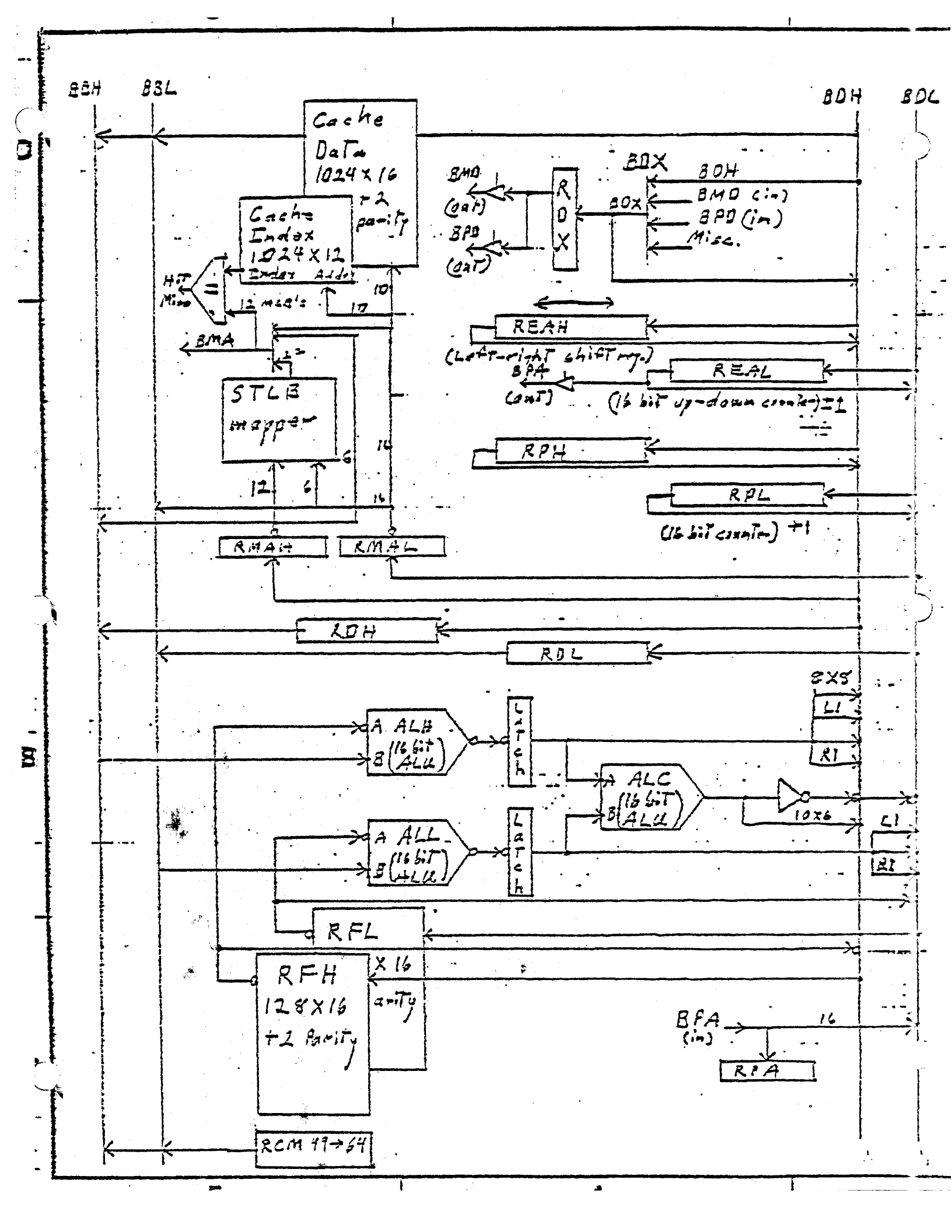
U-CODE HANDBOOK REV 2

P 4 0 0

HHJ 4-6-76

1/29/76
Rev1 3/15/76
2 4/6/76

INDEX


BSH
BSL
BDH
BDL
Cache Data 1024 x 16
+2 parity
Cache Index 1024 x 12
Index Addr
10
12
HIT Miss
12 MSB's
BMA
STLB mapper
6
16
12
6
RMAH
RMAL
BMD (out)
BPD (out)
RDX
BDX
BDH
BMD (in)
BPD (in)
Misc.
REAH
(Left-right shift reg.)
BPA (out)
REAL
(16 bit up-down counter) ±1
RPH
RPL
(16 bit counter) +1
RDH
RDL
A ALH
B (16 bit ALU)
Latch
A ALL
B (16 bit ALU)
Latch
SXS
LI
RI
ALC (16 bit ALU)
10x6
LI
RI
RFL
RFH 128 X 16 +2 Parity
X 16 parity
BFA (in)
16
RPA
RCM 49→64

P400 μ-CODE WORD

Bit	Id	Default	Defined By	Used by	Remarks
1	BBH	0	<BB SPEC>	<RF SPEC> if left, right side for RCD, RCM not obvious	8 combinations fixed available: see list
2	BBL	1			
3	select	1			
4	ALH	0	<OP SPEC>	RR Macro; others use specific ALH modes. Extensive error checking done on selected Modes.	64 combinations PROM generated from the thousands possible. See associated list for the 64 defined Modes.
5	ALL	0	CIN =		
6	ALC	0	CI6 =		
7	select	0	CALC =		
8	and Mode	0			
9	select	0			
10	Inde-	0	SETLATCH N	C= (bits 11→13)	<SREM SPEC> (bit 10)	Many different functions are gathered into this one field
11	pendent	0	RESETLATCH N	L= (bit 10)	REQPIO'
	Selects.		SETDLATCH N	BDX= (bits 10→11)	REQINT } Bits (12, 13)
12		0	RESETDLATCH N	MEMORY	REQCAI
13		0		IISM, <MEM SPEC>	
14	BDH	0	<BD SPEC>	E= note: 1. also specifies REAM Load/Shift 2. requires IACSHFT for shifts	4 bits plus IACSHFT select one of 32 possible BD enables (see list).
15	BDL	0			
16	source	0			
17		1			
18		0			
19	Register	0	<RF mnemonic>		See the associated list for the various direct and "indirect" addresses possible.
20	Files	0			
21	Select	0			
22		0			
23		0			
24		0			
25	BD Early	0	if =1, use RCD (cache Read, Mapped)		set if RCD is a source
26	Cache Control	1	if =0 use BD early MB One for other cases		set if EA$MA or P$MA
27		0	if = 1 select RP (early and late) if =0 select REA		

28 (See Clock)

P400 μ-CODE & IRD-

Bit	Field	Default	Referred By	Influenced by	Description
29	UA	0	<IAC SPEC>	<TIME SPEC> <SREAD SPEC> <ACT SPEC> <GEN DEST> (others)	Several IAC's are often used at one time. See The IAC list for details
30	UA	0			
31	UIS	0			
32	UIS	0			
33	UIS	0			
34	IA	0			
35	IA	0			
36	IA	1			
37	IA	0			
38	IA	0			
28	Clock	0	<GEN DEST> <TIME SPEC>	<IAC SPEC>	Each of the 128 cycles selects a group of registers to be clocked (updated) and a time to be used.
39	Clock	0			
40	Clock	0			
41	Clock	0			
42	Clock	0			
43	Clock	0			
45	Clock	0			
44	DMX control	1 = DMX 0 = Inhibit DMX			
46	Check	(odd Parity)	Parity bit is built by assembler		
47	Control Select	=3 (EMIT) =1 SHORT JUMP =0 LONG JUMP =2 Select BCY <CS spec>			
48	Control Select				

49	EMIT (Data value) = RCM= (<DATA SPEC> etc.	1 for Short Jump, 0 for <ACT SPEC>= <CDECODE>, <C RTN>; 0 else.		Long Jump NA <GOTO> Data (value) if RTN.	Select BCY: 0 ShortJump 4 RCS; 1 RCH 5 BDH; 2 DA 6 RCM; 3 EAF 7 TA
50		NA bits 11, 12			
51		if 49=1 form NAU like NA16	if 49=0: 1=CRTN, 0=CDECODE		1 1 P500 Relative Fetch
52		if 49=1 form NA14 like NA16	if 49=0: NA bit 13, Like 5 6, NA bit 14		Unused except for Fetch: Segmented Fetch, Sectored Fetch — Sectored Fetch P500 — P500 or Sectored
53		if 49=1 form NA15 Like NA16	if 49=0: select CDECODE or CRTN term, NA bit 15		=1 for Sectored Fetch — unused for others.
54		NA bit 16 Jump Condition select. NA16 = 61 if 59=60=0			
55		Select 1 of 8 conditions on 16 "JC" multiplexors			
56					
57					
58					
59					
60					
61					
62					
63					
64					

BUS B MHJ 1-28-76

Bus B consists of two 16 bit buses called BBH and BBL. Each has 4 sources. The 3 bit u-code field which selects Bus b inputs allows each High source to be selected with 2 Low sources. A table follows showing the possible combinations.

code	BBH	BBL
0	RCM	RCD
1	RCM	RDL
2	RDH	RCD
3	RDH	RDL
4	RMAH	RMAL
5	RMAH	RCM
6	RCD	RMAL
7	RCD	RCM

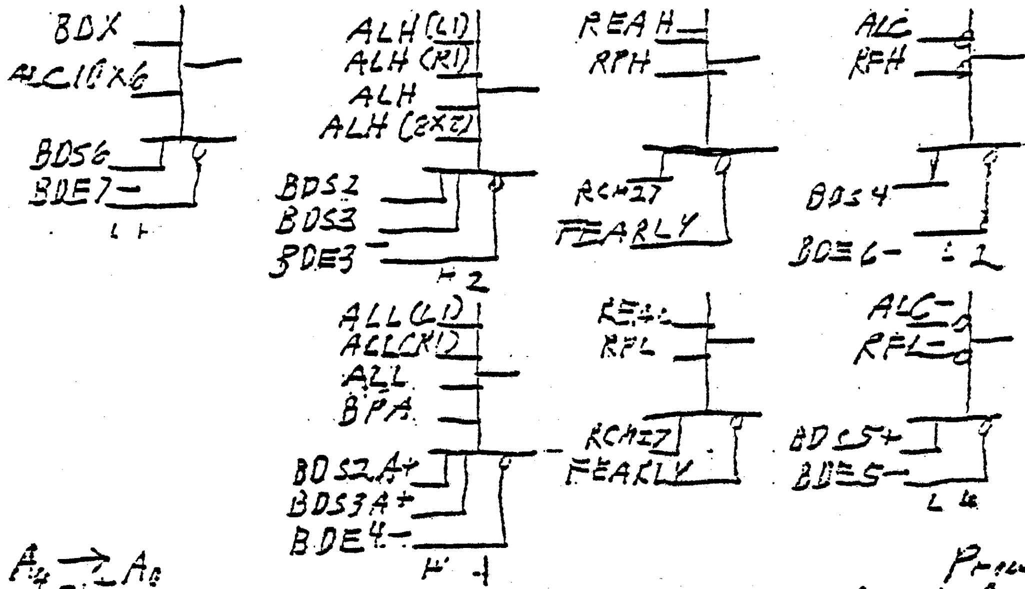
Bus D consists of two 16 bit buses called BDH and BDL. BDH has 10 different sources. BDL has 8 sources.

Bus D can be used two times in one u-code step. These uses are called Bus D Early (BDE) and Bus D Late (BDL). BDE is a limited capability as only REA or RP can be selected at this time. There is only one possible destination: RMA. The u-code field which controls this also specifies if cache is to be used this step or not. The encoding is:

0	REA	BD EARLY	
1	RP	BD EARLY	
2	REA	BD LATE	
2	NULL		
3	RP	BD LATE	
4	REA	BD EARLY	USE CACHE (RCD)
5	RP	BD EARLY	USE CACHE (RCD)
6	REA		USE CACHE (RCD)
7	RP		USE CACHE (RCD)

Bus D late selections are controled by a four bit u-code field and IACSHFT. The following table summerizes the possibilities. Codes 0-17 do not have IACSHFT, the others do. The next table shows the shifts which may be specified.

Bus D Summary



Pa-Ts = 825123

$A_4 \rightarrow A_0$ Bits IACSHIFT+ RC414-17	Function	ENDI	Dip site → PROM code – octal, A28K, $O_7 \rightarrow O_0$, BDS2 → S6	A30K, $O_7 \rightarrow O_0$, GEAH0 → [illegible]
0 0	REA or RP (Late)		000	000
0 1	ALH | ALL		300	030
0 2	RFH | ALC		014	006
0 3	RFH | RFL		010	006
0 4	ALH | ALC		204	024
0 5	RFH | ALL		110	012
0 6	BDX | ALC		004	005
0 7	RFH | ALL, LSHIFT 0		* 010	* 012
1 0	HSX8L BPA		360	030
1 1	ALC | ALC		004	006
1 2	ALC10X6 | ALC		006	005
1 3	DMX		130	017
1 4	ALH | RFL		200	024
1 5	ALC | RFL		000	006
1 6	ALC | ALL		100	012
1 7	Disable All		000	000
2 0	Left shift	0	000	130
2 1	↓	ALH16	000	130
2 2		LINK	000	130
2 3		ALH00	000	130
2 4		REAH01	000	130
2 5		ALL16	000	130
2 6		ALH01	000	130
* 2 7	ALH | ALL (shift REAH Left) =		* 300	* 130
3 0	Right shift	0	060	230
3 1	↓	ALH15	060	230
3 2		LINK	060	230
3 3		ALH06	060	230
3 4		REAH01	060	230
3 5		ALL16	060	230
3 6		ALH01	060	230
3 7		0	060	230

Unchanged 11/17/75

Prom # → 11[illegible] → 212

P400 μ-code

Addressable Latch

P400
8/6/75
MHJ
REV1 1/28/76

Cleared by IAC FETCH
Loaded all during cycle with SETLATCH or RESETLATCH.

Bits	Use	Name
0	if set, inhibit address traps	EADRTR-
1	S+ executed	FSPLUS
2	-S executed	FSMINUS
3	Source for BPA99.	BPA99
4	Source for BPA00.	BPA00
5	} Scratch	ADL5
6	}	ADL6
7	Scratch used by IN032, CE4	ADL7

Possible Shifts and ENDs

Legal Requests:	RCM 14→17	RCM 10	Action
1. SHIFT B RIGHT	10	1	LRL
E= 0	10	1	LRL
E= LINK	12	1	Multiple
E= ALH00	13	1	MPY
E= ALH01	16	1	arithmetic
ROTATE B RIGHT	15	1	
ROTATE S RIGHT	11	1	ARR
HOLE IN MIDDLE (all shifts above are legal)	RCM10 = 0		NRM LRS
2. SHIFT B LEFT	0	0	LLL
E= 0	0	0	LLL
E= LINK	2	0	Multiply
E= REAH01	4	0	48 bit
HOLE IN MIDDLE (all shifts above are legal)	RCM10 = 1		
SHIFT B LEFT	1	0	ALS ALL
ROTATE B LEFT	6	0	LLR
ROTATE S LEFT	1	1	ALR

Diagnostic Status Word (DSW)

80 bits, Registers '34,'35&'36 (named DSWRMA, DSWSTAT, and DSWPB)

Bits 1,32: DSWRMA
33,48: DSWSTATH
49,64: DSWSTATL
65,80: DSWPB

Valid on all checks except Power Fail as follows:

ssw1 = UP ssw2 = UP

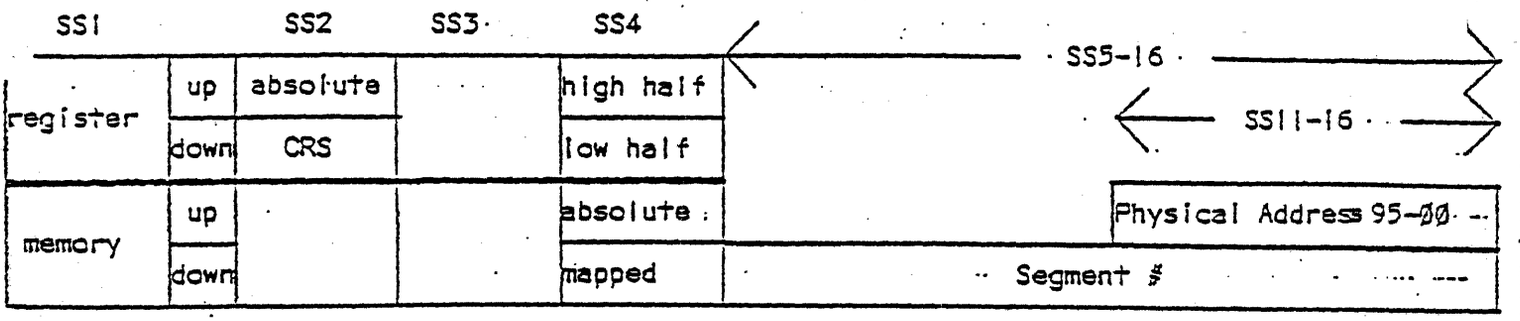
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	DSWSTATH
CI	MC	MP	MM	Machine Check Code			RCM	ECCU	ECCC DMX	Bup Inv	RP Backup Count			DMX	10 Bus	
RESERVED			ECCU							RP BACKUP			PAR		MISCH CHK	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	DSWSTATL
Reserved				ECCC Syndrome			Mod #	Reserved		u-Verify test #						

33: CI=Check Immediate
34: MC=Machine Check
35: MP=Memory Parity (ECC)
36: MM=Missing Memory
: Machine Check Code
Ø=Peripheral Data (BPD) Output
1=Peripheral Address (BPA) Input
2=Memory Data (BMD) Output
3=Cache Data (RCD)
4=Peripheral Address (BPA) Output
5=RDX-BPD Input
6=Memory Address (BMA)
7=Register File
40: Not RCM Parity (Reset for RCM Parity error - XCS only)
41: ECCU=ECC Uncorrectable Error
42: ECCC=ECC Corrected Error
43: Bup Inv=RP backup count (44-46) invalid
44,46: RP Backup Count-amount RPL (DSWPB) was incremented in current instruction
47: DMX, set if check occurred during DMX
48: IO Bus, set if check occurred during DMX, PIO or interrupt u-code
49,50: Reserved
51,55: ECCC Syndrome=5 syndrome bits on a corrected error
56: Mod #=Low order address bit of memory module causing the error
57,58: Reserved
59,64: u-Verify test # set on failure during Master Clear or VIRY instruction

Validity:
Always :1-33,43,47-48,59-80
If bit 34 set :37-40
35 :41-42,56 If bit 42 set:51-55
36 :56
If bit 43 reset:44-46

It is the responsibility of the check handling software to clear the DSW after a check has been processed.

Figure 13.



Notes: With all switches down, control panel works exactly as for the P-3ØØ following either a Master Clear or a HALT if not running in segmented mode. It is necessary to make mapped memory accesses if address traps are to be generated. If running segmented, memory accesses will be mapped to segment Ø unless an explicit segment number is entered in SS5-16.

Registers: Register address is in address register (switches down) For CRS, only low order 5 bits are used; for absolute, only low order 8 bits are used. Y+1 (STORE/FETCH) operates exactly as for memory with the address being pre-incremented.

Null Vector: In P-3ØØ mode, if an external interrupt, fault, or check attempts to vector through a memory location containing a Ø, the following action is taken:

HALT
data and address lights cleared
RP = $\overline{\text{address trapped}}$
PBH = RPH
TR2L = address of vector

Figure 17.

u-code scratch

RF0 addr	High	Low
0	TR0	-
1	TR1	-
2	TR2	-
3	TR3	-
4	TR4	-
5	TR5	-
6	TR6	-
7	TR7	-
10	RDMX1	-
11	RDMX2	-
12		RATMPL
13	RSGT1	-
14	RSGT2	-
15	RECC1	-
16	RECC2	-
17		REGIV
20	ZERO	ONE
21	PBSAVE	-
22		
23		
24		
25		
26		
27		
30	PSWPB	-
31	PSWKEYS	-
32	PPA:PLA	PCBA
33	PB:PLB	PCBB
34	SWRMA	-
35	OSWSTAT	-
36	OSWPB	-
37		

DMA

Cell	High	Low	RF1 Addr
0			40
1			41
2			42
3			43
4			44
5			45
6			46
7			47
10			50
11			51
12			52
13			53
14			54
15			55
16			56
17			57
20	(20)	(21)	60
21			61
22	(22)	(23)	62
23			63
24	(24)	(25)	64
25			65
26	(26)	(27)	66
27			67
30	(30)	(31)	70
31			71
32	(32)	(33)	72
33			73
34	(34)	(35)	74
35			75
36	(36)	(37)	76
37			77

Current Register Set (CRS)

CRS Cell	High	Low	RF2 Addr	RF3 Addr
0	GR0	-	100	140
1	GR1	-	101	141
2	GR2(I,A,LH)	-(2,B,LL)	102	142
3	GR3(EH)	-(EL)	103	143
4	GR4	-	104	144
5	GR5(3,S,Y)	-	105	145
6	GR6	-	106	146
7	GR7(0,X)	-	107	147
10	FR0(13)	-	110	150
11	-	-	111	151
12	FR1(4)	-(5)	112	152
13	-(6)	-	113	153
14	PB	-	114	154
15	SB(14)	-(15)	115	155
16	LB(16)	-(17)	116	156
17	XB	-	117	157
20	OTAR3(10)	-	120	160
21	OTAR2	-	121	161
22	OTAR1	-	122	162
23	OTAR0	-	123	163
24	KEYS	(modals)	124	164
25	OWNER	-	125	165
26	FCODE(11)	-	126	166
27	FADDR	-(12)	127	167
30	TIMER	-	130	170
31	-	-	131	171
32			132	172
33			133	173
34			134	174
35			135	175
36			136	176
37			137	177

[ ] numbers in parenthesis shows P300 address Mapping

KEYSH

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CB I	OP	L I N K	Adr Mode			F L E X	I E X	C L T	C E Q				I D	S O	

KEYSL (Modals)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
EN B	VI M							CRS			M I O	P X M	S E G	MCM	

Adr	Mode
0	16S
1	32S
2	64R
3	32R
4	32I
5	
6	64V
7	

FLEX=0 allows FLEX Faults

ENB: Set=enable interrupts
VIM: Set=Vectored interrupt mode
CRS: Current Register Set
MIO: Set=mapped I/O
PXM: Set=Process Exchange Mode
SEG: Set=Segmentation Mode
MCM: Machine Check Mode

n Dispatcher
SD: Save Done

Register Files and Mapping

Figure 16.

12. MHJ
P400
6/6/75
REI 1/29/76

RF select - P400 - 1

Table 1

Select RF Function	Encode (in Rcm 18→24) 18 19 20 21 22 23 24	octal	
1. µ-code addressed scratch locations (1's compliment of RCM's 20→24) (TRO = 137; TR1 = 136 etc.)	1 0 ← 0 ⇒ '37 →	100 → 137	
2. µ-code addressed Register Set locations (1's compliment of RCM's 20→24) (A = 175)	1 1 ← 0 ⇒ '37 →	140 → 177	
2a. µ-code addressed switch from RXC(1) To RYC(3)		170	
3. Base Registers (BR) of active Register Set (from BBH5, BBH6)	0 0 0 1 X X X	010	
4. DTAR Registers of active Register Set (from BBH5, BBH6)	0 0 1 0 X X X	020	
5. RPA addressed DMA channels	0 0 1 1 X X X	030	
6. REAL address entire file (1's compliment)	0 1 0 0 X X X	040	
7. Address Trap Mapping	0 1 0 1 X X X	050	
8. GR (RS)	0 1 1 0 1 1 X	066	All use current Register Set
9. GR (RSN)	0 1 1 0 1 0 X	064	
10. GR (CFS)	0 1 1 0 0 1 X	062	
11. GR (CFSN)	0 1 1 0 0 0 X	060	
12. GR (CRD)	0 1 1 1 1 1 X	076	
13. GR (CRDN)	0 1 1 1 1 0 X	074	
14. GR (FD)	0 1 1 1 0 1 X	072	
15. GR (FDN)	0 1 1 1 0 0 X	070	

ALU MODES MHJ 1-29-76

The three 16 bit ALU's in the Prime 400 have a limited number of combinations in which they may be used. These are summerized in the following table.

KEY:

A	ADD. A= ADD FOR ASSEMBLER.
TB	TRANSPORT THE B INPUT
IA	INCREMENT THE A INPUT IF C= 1, TRANSPORT IF C= 0. IA= INC FOR ASSEMBLER
S	SUBTRACT. IF C= 0, SUB-1. S= SUB FOR ASSEMBLER.
COL	CARRY OUT FROM ALL--ALLOWS ALH|ALL TO BE USED AS 32 BITOPERATION.
DA	DECREMENT THE A INPUT IF C= 0, TRANSPORT IF C= 1. IA= DEC FOR ASSEMBLER
/	INDICATES HARDWARE MODE SWITCH FROM ONE OPERATION TO ANOTHER. EXAMPLES INCLUDE: MPY, DIV, AND THE FETCH CYCLE EFFECTIVE ADDRESS FORMATION.

Other symbols are used but are felt to be self-explanatory.

ALU-

14.

FUNC	H	L	C	CH	CL	CC
0	A	A	TB	Col	0	
1				Col	Cbit	
2	A	A	IA	0	0	1
3				Cbit		
4	S	S		col	Cbit	
5				Col	1	
6	0	IA	TB	col	0	
7				0	1	
10	-1	-1		0	0	
11				1	1	
12	S	TB	TB	col	1	
13				1	0	0
14	A OR $\overline{B}$	A OR $\overline{B}$				0
15						
16	TB	A	A	0	0	0
17				0		1
20	A	TB	A-	0	1	0
21				1	0	
22	OR	OR	TB			
23						
24	A/S	A/S	TA	col	0/1	
25				col	Cbit	
26	XOR	XOR		col	Cbit	
27				1	0	0
30	NOTA	NOTA	IA			1
31						0
32	TB	0	IA			1
33						0
34	XOR	TA	S			1
35						0
36	TB	TB	OR			
37						
40	A/TA	A/TA		Col	0	
41				Col	Cbit	
42	IA	TA		0	0	1
43				Cbit		
44	S/TA	S/TA		col	cbit	
45				col	1	
46	DA	DA	TA	col	0	
47				0	1	
50	IA	0	TA	0	0	
51				1	1	
52	IA	IA	S	col	1	
53				1	0	0
54	TB	TB	A			0
55						
56	A/TB	NOTB	A/TA	0	0	0
57				0		1
60	A	IA	TA	0	1	0
61				1	0	
62	0	AND	TB			
63						
64	AND	AND	TA	col	0/1	
65				col	Cbit	
66	IA	IA	DA	col	Cbit	
67				1	0	0
70	NOTB	NOTB	IA			1
71						0
72	0	TB	S			1
73						0
74	A AND B	A AND B	OR			1
75						0
76						
77						

MHZ
10/25/75
Rev 1.1/24/7
CH CL CC

EMU RCM 8

* Added. 10/20/75

ALU OPERATIONS USED BY P400 U-CODE

The following list of ALU-operations is the total of those understood by the u-code assembler. They can be read as follows: A 16 or 32 in the name indicates that the assembler will take a non ALX= type statement. For example:

ALU A PLUS RDH => ...

uses the PLUS16 entry in the list.

For the other operations, the form ALH= TA etc. is used. The format in the table is ALH|ALL|ALC|(H1) where an unspecified ALU is shown by X. The non-standard carry in conditions specified by CH= or CL= or CC= are appended to the ALU operators.

PLUS32	'00	MPYMPYX	'40
NULL16	'00	MPYMPYXLCOIT	'41
ADD32	'00	TA32	'42
XADDX	'00	XTAX	'42
XADDTE	'00	TAXX	'42
NULL32	'00	TA16	'42
ADDADDXLCOIT	'01	INCXXHCOIT	'43
ADD32LCOIT	'01	MPYFSMPYFSX	'45
ADDADDX	'02	MPYFSMPYFSXLCOIT	'44
PLUS16	'02	DEC32	'46
ADD16	'02	DECXTA	'47
ADDXX	'02	DECXTA	'47
ADDXXHCOIT	'03	DEC16	'47
ADD16HCOIT	'03	DAXX	'47
SUBSUBXLCHIT	'04	TAZEROX	'50
SUB32LCHIT	'04	TAXTA	'50
MINUS32	'05	INCXTA	'51
SUB32	'05	INC32	'52
ZERO-TATB	'06	INC16	'53
ZERUXX	'06	INCXX	'53
XTATB	'06	TD3A	'54
ZEROTAX	'16	XTBX	'54
MINUS1	'10	TOTBX	'54
MINUS1MINUS1X	'10	TB16	'54
ZERO	'11	IPTRADD	'54
ZERUZEROX	'11	TBXX	'54
YIELD	'12	FETCHNOTFFETCH	'56
		FETCHXFETCH	'56
		FETCHNOTHFETCHC1	'57
		FETCHXFETCHC1	'57
		ADDXTA	'60

MINUS16	'13
SUB16	'13
ADDBNOTXX	'14
ADDENOT	'14
TBADDADD	'16
TBADDADDC1	'17
ADDTBADD	'20
OR32	'22
ORORTP	'22
OR16	'22
ORXX	'22
DIVDIVTA	'24
DIVDIVX	'24
DIVDIVXLCBIT	'25
XOR32	'26
XOR16	'26
ANOT32	'30
NOTA32	'30
ANOT16	'30
NOTAXINC	'30
NOTA16	'30
NOTAXTA	'31
TBZERODEC	'32
TBXINC	'32
TBZLROX	'32
TBXTA	'33
XORTASUBC1	'34
XORTASUBCO	'35
TBTEOR	'36

ADDINCTA	'60
ADDTATAH1	'61
ADDXXH1	'61
XANDTB	'62
ZEROANDTO	'62
ANDXTA	'64
AND32	'64
ANDXX	'64
ANDXTA	'64
AND16	'64
INCINCXLCBIT	'66
INCXDEC	'67
BNOT32	'70
BNOT16	'70
NOTBXX	'70
NOTBNOTBX	'70
NOTB16	'70
NOTB32	'70
XNOTBX	'70
NOTBXINC	'70
NOTBXTA	'71
ZEROTBX	'72
ZEROTBSUB	'72
ZEROTBSUBCO	'73
ANOTANDBANDOR	'74
ANOTANDBANDX	'74

CL INFORMATION REV 4 P400 4/6/76

CLOCK INFORMATION REV 4 P400 4/6/76

SORTED CLOCK REV 4 P400 UCODE 4/6/76

```
160          16

  0    1  [  000000 ]  160
  0    1  [  000001 ]  160      RDH
  0    1  [  000020 ]  160      RD
  0    1  [  000021 ]  160      RD        REAH
  0    1  [  000040 ]  160      RDL
  0    1  [  000041 ]  160      RDL       REAH
  0    1  [  000060 ]  160      REAH
  0    1  [  000061 ]  160      REA
  0    1  [  000100 ]  160      REA       RMA       RPL
  0    1  [  000101 ]  160      REA       RMA
  0    1  [  000120 ]  160      REAL
  0    1  [  000121 ]  160      REAL      RMAL
  0    1  [  000140 ]  160      RPL
  0    1  [  000141 ]  160      RMA
  0    1  [  000160 ]  160      DXMRDY
  0    1  [  000161 ]  160      RCD

200          16

  0    1  [  000002 ]  200
  0    1  [  000003 ]  200      RDH
  0    1  [  000022 ]  200      RD
  0    1  [  000023 ]  200      RDH       REAL
  0    1  [  000042 ]  200      RDL
  0    1  [  000043 ]  200      REAH
  0    1  [  000062 ]  200      REA       RMA
  0    1  [  000063 ]  200      REAL
  0    1  [  000102 ]  200      REAL      RMAL
  0    1  [  000103 ]  200      RFH
  0    1  [  000122 ]  200      RF
  0    1  [  000123 ]  200      RFL
```

```
0    1  [   000142 ]  200        RMA
0    1  [   000143 ]  200        MRDY       RFH
0    1  [   000162 ]  200        RCD        RF
0    1  [   000163 ]  200        MRDY       RCD        RMAL

0         16

0    1  [   000004 ]  240
0    1  [   000005 ]  240        RDH
0    1  [   000024 ]  240        RD
0    1  [   000025 ]  240        RDH        RF
0    1  [   000044 ]  240        RDL        RF
0    1  [   000045 ]  240        REAH       RFH
0    1  [   000064 ]  240        REAH       RF
0    1  [   000065 ]  240        REAH       RFL
0    1  [   000104 ]  240        REAL
0    1  [   000105 ]  240        REAL       RFH        RMAL
0    1  [   000124 ]  240        RFH        RMA
0    1  [   000125 ]  240        REAL       RMAL
0    1  [   000144 ]  240        RF         RMA
0    1  [   000145 ]  240        RFH
0    1  [   000164 ]  240        RF
0    1  [   000165 ]  240        RFL

0         16

0    1  [   000006 ]  280
0    1  [   000007 ]  280        RCD
0    1  [   000026 ]  280        RCD        RDH
0    1  [   000027 ]  280        RCD        RDH        RDL
0    1  [   000046 ]  280        RCD        REAL
0    1  [   000047 ]  280        RCD        RPL
0    1  [   000066 ]  280        RDH
0    1  [   000067 ]  280        RD
0    1  [   000106 ]  280        RD         REAL       RMAL
0    1  [   000107 ]  280        RD         RPL
0    1  [   000126 ]  280        RDH        REAH
0    1  [   000127 ]  280        RDH        RFH
```

0	1	[	000146]	280	RDH	RFL	
0	1	[	000147]	280	RDH	RPL	
0	1	[	000166]	280	RDL		
0	1	[	000167]	280	RDL	REA	RMA

280A 16

0	1	[	000014]	280A	RDL	RMA	
0	1	[	000015]	280A	REA		
0	1	[	000034]	280A	REA	RMA	RPL
0	1	[	000035]	280A	RDL	RFL	
0	1	[	000054]	280A	REA	RPL	
0	1	[	000055]	280A	REAH	RFH	
0	1	[	000074]	280A	REAH	RF	
0	1	[	000075]	280A	REAL	RMAL	RPL
0	1	[	000114]	280A	RFH		
0	1	[	000115]	280A	RF		
0	1	[	000134]	280A	RF	RPL	
0	1	[	000135]	280A	RFH	RPL	
0	1	[	000154]	280A	RFL		
0	1	[	000155]	280A	RPL		
0	1	[	000174]	280A	RMA		
0	1	[	000175]	280A	RF	RMA	

280B 11

0	1	[	000012]	280B	RPL		
0	1	[	000013]	280B	RCD	RF	
0	1	[	000032]	280B	REA	RMA	
0	1	[	000033]	280B	MRDY	RCD	
0	1	[	000052]	280B	MRDY	RFH	
0	1	[	000053]	280B	MRDY	RDH	
0	1	[	000072]	280B	RCD	RDH	RPL
0	1	[	000073]	280B	MRDY	RF	RMA
0	1	[	000112]	280B	MRDY	RFH	RMAL
0	1	[	000113]	280B	RDL	RF	
0	1	[	000132]	280B	RDH	REAL	

21,

320 15

0	1	[	000010]	320				
0	1	[	000011]	320	RD			
0	1	[	000030]	320	RD	REAL	RMAL	RPL
0	1	[	000031]	320	RDH	REAL		
0	1	[	000050]	320	RDL	RFL		
0	1	[	000051]	320	REAH	RF		
0	1	[	000070]	320	REAL	RMAL		
0	1	[	000071]	320	RFH			
0	1	[	000110]	320	RF			
0	1	[	000111]	320	RFL			
0	1	[	000130]	320	RCD	RF		
0	1	[	000131]	320	RD	REAL	RPL	
0	1	[	000150]	320	RD	RF		
0	1	[	000151]	320	REA	RF	RMA	
0	1	[	000170]	320	RCD	RDH		

360 11

0	1	[	000016]	360				
0	1	[	000017]	360	RFH			
0	1	[	000036]	360	RF			
0	1	[	000037]	360	RFL			
0	1	[	000056]	360	RCD	RDH		
0	1	[	000057]	360	REA	RFL	RMA	
0	1	[	000076]	360	RDL	REAL		
0	1	[	000077]	360	RDL	RMA		
0	1	[	000116]	360	RPL			
0	1	[	000117]	360	RDL	REA	RMA	
0	1	[	000136]	360	RD	REAL	RMAL	

DXMRDY 1

0	1	[	000160]	160	DXMRDY

MRDY 7

0	1	[	000143]	200	MRDY	RFH		
0	1	[	000163]	200	MRDY	RCD	RMAL	
0	1	[	000033]	280B	MRDY	RCD		
0	1	[	000052]	280B	MRDY	RFH		
0	1	[	000053]	280B	MRDY	RDH		
0	1	[	000073]	280B	MRDY	RF	RMA	
0	1	[	000112]	280B	MRDY	RFH	RMAL	
CD			14					
0	1	[	000161]	160	RCD			
0	1	[	000162]	200	RCD	RF		
0	1	[	000163]	200	MRDY	RCD	RMAL	
0	1	[	000007]	280	RCD			
0	1	[	000026]	280	RCD	RDH		
0	1	[	000027]	280	RCD	RDH	RDL	
0	1	[	000046]	280	RCD	REAL		
0	1	[	000047]	280	RCD	RPL		
0	1	[	000013]	280B	RCD	RF		
0	1	[	000033]	280B	MRDY	RCD		
0	1	[	000072]	280B	RCD	RDH	RPL	
0	1	[	000130]	320	RCD	RF		
0	1	[	000170]	320	RCD	RDH		
0	1	[	000056]	360	RCD	RDH		
)			12					
0	1	[	000020]	160	RD			
0	1	[	000021]	160	RD	REAH		
0	1	[	000022]	200	RD			
0	1	[	000024]	240	RD			
0	1	[	000067]	280	RD			
0	1	[	000106]	280	RD	REAL	RMAL	
0	1	[	000107]	280	RD	RPL		
0	1	[	000011]	320	RD			
0	1	[	000030]	320	RD	REAL	RMAL	RPL
0	1	[	000131]	320	RD	REAL	RPL	
0	1	[	000150]	320	RD	RF		

CLOCK INFORMATION REV 4 . P400 4/6/76

0	1	[	000136]	360	RD	REAL	RMAL
DH			18				
0	1	[	000001]	160	RDH		
0	1	[	000003]	200	RDH		
0	1	[	000023]	200	RDH	REAL	
0	1	[	000005]	240	RDH		
0	1	[	000025]	240	RDH	RF	
0	1	[	000026]	280	RCD	RDH	
0	1	[	000027]	280	RCD	RDH	RDL
0	1	[	000066]	280	RDH		
0	1	[	000126]	280	RDH	REAH	
0	1	[	000127]	280	RDH	RFH	
0	1	[	000146]	280	RDH	RFL	
0	1	[	000147]	280	RDH	RPL	
0	1	[	000053]	280B	MRDY	RDH	
0	1	[	000072]	280B	RCD	RDH	RPL
0	1	[	000132]	280B	RDH	REAL	
0	1	[	000031]	320	RDH	REAL	
0	1	[	000170]	320	RCD	RDH	
0	1	[	000056]	360	RCD	RDH	
DL			14				
0	1	[	000040]	160	RDL		
0	1	[	000041]	160	RDL	REAH	
0	1	[	000042]	200	RDL		
0	1	[	000044]	240	RDL	RF	
0	1	[	000027]	280	RCD	RDH	RDL
0	1	[	000166]	280	RDL		
0	1	[	000167]	280	RDL	REA	RMA
0	1	[	000014]	280A	RDL	RMA	
0	1	[	000035]	280A	RDL	RFL	
0	1	[	000113]	280B	RDL	RF	
0	1	[	000050]	320	RDL	RFL	
0	1	[	000076]	360	RDL	REAL	
0	1	[	000077]	360	RDL	RMA	

```
 0    1  [   000117 ]  360        RDL        REA        RMA

REA           12

 0    1  [   000061 ]  160        REA
 0    1  [   000100 ]  160        REA        RMA        RPL
 0    1  [   000101 ]  160        REA        RMA
 0    1  [   000062 ]  200        REA        RMA
 0    1  [   000167 ]  280        RDL        REA        RMA
 0    1  [   000015 ]  280A       REA
 0    1  [   000034 ]  280A       REA        RMA        RPL
 0    1  [   000054 ]  280A       REA        RPL
 0    1  [   000032 ]  280B       REA        RMA
 0    1  [   000151 ]  320        REA        RF         RMA
 0    1  [   000057 ]  360        REA        RFL        RMA
 0    1  [   000117 ]  360        RDL        REA        RMA

REAH          11

 0    1  [   000021 ]  160        RD         REAH
 0    1  [   000041 ]  160        RDL        REAH
 0    1  [   000060 ]  160        REAH
 0    1  [   000043 ]  200        REAH
 0    1  [   000045 ]  240        REAH       RFH
 0    1  [   000064 ]  240        REAH       RF
 0    1  [   000065 ]  240        REAH       RFL
 0    1  [   000126 ]  280        RDH        REAH
 0    1  [   000055 ]  280A       REAH       RFH
 0    1  [   000074 ]  280A       REAH       RF
 0    1  [   000051 ]  320        REAH       RF

REAL          18

 0    1  [   000120 ]  160        REAL
 0    1  [   000121 ]  160        REAL       RMAL
 0    1  [   000023 ]  200        RDH        REAL
 0    1  [   000063 ]  200        REAL
 0    1  [   000102 ]  200        REAL       RMAL
```

CLOCK INFORMATION REV 4 P400 4/6/76

0	1	[	000104]	240	REAL			
0	1	[	000105]	240	REAL	RFH	RMAL	
0	1	[	000125]	240	REAL	RMAL		
0	1	[	000046]	280	RCD	REAL		
0	1	[	000106]	280	RD	REAL	RMAL	
0	1	[	000075]	280A	REAL	RMAL	RPL	
0	1	[	000132]	280B	RDH	REAL		
0	1	[	000030]	320	RD	REAL	RMAL	RPL
0	1	[	000031]	320	RDH	REAL		
0	1	[	000070]	320	REAL	RMAL		
0	1	[	000131]	320	RD	REAL	RPL	
0	1	[	000076]	360	RDL	REAL		
0	1	[	000136]	360	RD	REAL	RMAL	

20

0	1	[	000122]	200	RF		
0	1	[	000162]	200	RCD	RF	
0	1	[	000025]	240	RDH	RF	
0	1	[	000044]	240	RDL	RF	
0	1	[	000064]	240	REAH	RF	
0	1	[	000144]	240	RF	RMA	
0	1	[	000164]	240	RF		
0	1	[	000074]	280A	REAH	RF	
0	1	[	000115]	280A	RF		
0	1	[	000134]	280A	RF	RPL	
0	1	[	000175]	280A	RF	RMA	
0	1	[	000013]	280B	RCD	RF	
0	1	[	000073]	280B	MRDY	RF	RMA
0	1	[	000113]	280B	RDL	RF	
0	1	[	000051]	320	REAH	RF	
0	1	[	000110]	320	RF		
0	1	[	000130]	320	RCD	RF	
0	1	[	000150]	320	RD	RF	
0	1	[	000151]	320	REA	RF	RMA
0	1	[	000036]	360	RF		

14

CLOCK INFORMATION REV 4 P400 4/6/76

0	1	[	000103]	200	RFH			
0	1	[	000143]	200	MRDY	RFH		
0	1	[	000045]	240	REAH	RFH		
0	1	[	000105]	240	REAL	RFH	RMAL	
0	1	[	000124]	240	RFH	RMA		
0	1	[	000145]	240	RFH			
0	1	[	000127]	280	RDH	RFH		
0	1	[	000055]	280A	REAH	RFH		
0	1	[	000114]	280A	RFH			
0	1	[	000135]	280A	RFH	RPL		
0	1	[	000052]	280B	MRDY	RFH		
0	1	[	000112]	280B	MRDY	RFH	RMAL	
0	1	[	000071]	320	RFH			
0	1	[	000017]	360	RFH			
L			10					
0	1	[	000123]	200	RFL			
0	1	[	000065]	240	REAH	RFL		
0	1	[	000165]	240	RFL			
0	1	[	000146]	280	RDH	RFL		
0	1	[	000035]	280A	RDL	RFL		
0	1	[	000154]	280A	RFL			
0	1	[	000050]	320	RDL	RFL		
0	1	[	000111]	320	RFL			
0	1	[	000037]	360	RFL			
0	1	[	000057]	360	REA	RFL	RMA	
A			18					
0	1	[	000100]	160	REA	RMA	RPL	
0	1	[	000101]	160	REA	RMA		
0	1	[	000141]	160	RMA			
0	1	[	000062]	200	REA	RMA		
0	1	[	000142]	200	RMA			
0	1	[	000124]	240	RFH	RMA		
0	1	[	000144]	240	RF	RMA		

0	1	[	000167]	280	RDL	REA	RMA	
0	1	[	000014]	280A	RDL	RMA		
0	1	[	000034]	280A	REA	RMA	RPL	
0	1	[	000174]	280A	RMA			
0	1	[	000175]	280A	RF	RMA		
0	1	[	000032]	280B	REA	RMA		
0	1	[	000073]	280B	MRDY	RF	RMA	
0	1	[	000151]	320	REA	RF	RMA	
0	1	[	000057]	360	REA	RFL	RMA	
0	1	[	000077]	360	RDL	RMA		
0	1	[	000117]	360	RDL	REA	RMA	

RMAL 11

0	1	[	000121]	160	REAL	RMAL		
0	1	[	000102]	200	REAL	RMAL		
0	1	[	000163]	200	MRDY	RCD	RMAL	
0	1	[	000105]	240	REAL	RFH	RMAL	
0	1	[	000125]	240	REAL	RMAL		
0	1	[	000106]	280	RD	REAL	RMAL	
0	1	[	000075]	280A	REAL	RMAL	RPL	
0	1	[	000112]	280B	MRDY	RFH	RMAL	
0	1	[	000030]	320	RD	REAL	RMAL	RPL
0	1	[	000070]	320	REAL	RMAL		
0	1	[	000136]	360	RD	REAL	RMAL	

RPL 16

0	1	[	000100]	160	REA	RMA	RPL	
0	1	[	000140]	160	RPL			
0	1	[	000047]	280	RCD	RPL		
0	1	[	000107]	280	RD	RPL		
0	1	[	000147]	280	RDH	RPL		
0	1	[	000034]	280A	REA	RMA	RPL	
0	1	[	000054]	280A	REA	RPL		
0	1	[	000075]	280A	REAL	RMAL	RPL	
0	1	[	000134]	280A	RF	RPL		
0	1	[	000135]	280A	RFH	RPL		

```
0    1  [  000155 ]  280A     RPL
0    1  [  000012 ]  280B     RPL
0    1  [  000072 ]  280B     RCD       RDH       RPL
0    1  [  000030 ]  320      RD        REAL      RMAL      RPL
0    1  [  000131 ]  320      RD        REAL      RPL
0    1  [  000116 ]  360      RPL
```

28.a

COMBINATION LIST --- 117 ENTRIES

0	160			
0	160	RDH		
0	160	RD		
0	160	RD	REAH	
0	160	RDL		
0	160	RDL	REAH	
0	160	REAH		
0	160	REA		
0	160	REA	RMA	RPL
0	160	REA	RMA	
0	160	REAL		
0	160	REAL	RMAL	
0	160	RPL		
0	160	RMA		
0	160	DXMRDY		
0	160	RCD		
0	200			
0	200	RDH		
0	200	RD		
0	200	RDH	REAL	
0	200	RDL		
0	200	REAH		
0	200	REA	RMA	
0	200	REAL		
0	200	REAL	RMAL	
0	200	RFH		
0	200	RF		
0	200	RFL		
0	200	RMA		
0	200	MRDY	RFH	
0	200	RCD	RF	
0	200	MRDY	RCD	RMAL
0	240			
0	240	RDH		
0	240	RD		
0	240	RDH	RF	
0	240	RDL	RF	

1	0	280A	RFH	RPL		
1	0	280A	RFL			
1	0	280A	RPL			
1	0	280A	RMA			
1	0	280A	RF	RMA		
1	0	280B	RPL			
1	0	280b	RCD	RF		
1	0	2800	REA	RMA		
1	0	280B	MRDY	RCD		
1	0	280B	MRDY	RFH		
1	0	2800	MRDY	RDH		
1	0	280B	RCD	RDH	RPL	
1	0	280B	MRDY	RF	RMA	
1	0	2800	MRDY	RFH	RMAL	
1	0	280B	RDL	RF		
1	0	280B	RDH	REAL		
1	0	320				
1	0	320	RD			
1	0	320	RD	REAL	RMAL	RPL
1	0	320	RDH	REAL		
1	0	320	RDL	RFL		
1	0	320	REAH	RF		
1	0	320	REAL	RMAL		
1	0	320	RFH			
1	0	320	RF			
1	0	320	RFL			
1	0	320	RCD	RF		
1	0	320	RD	REAL	RPL	
1	0	320	RD	RF		
1	0	320	REA	RF	RMA	
1	0	320	RCD	RDH		
1	0	360				
1	0	360	RFH			
1	0	360	RF			
1	0	360	RFL			
1	0	360	RCD	RDH		
1	0	360	REA	RFL	RMA	
1	0	360	RDL	REAL		

1	0	240	REAH	RFH	
1	0	240	REAH	RF	
1	0	240	REAH	RFL	
1	0	240	REAL		
1	0	240	REAL	RFH	RMAL
1	0	240	RFH	RMA	
1	0	240	REAL	RMAL	
1	0	240	RF	RMA	
1	0	240	RFH		
1	0	240	RF		
1	0	240	RFL		
1	0	280			
1	0	280	RCD		
1	0	280	RCD	RDH	
1	0	280	RCD	RDH	RDL
1	0	280	RCD	REAL	
1	0	280	RCD	RPL	
1	0	280	RDH		
1	0	280	RD		
1	0	280	RD	REAL	RMAL
1	0	280	RD	RPL	
1	0	280	RDH	REAH	
1	0	280	RDH	RFH	
1	0	280	RDH	RFL	
1	0	280	RDH	RPL	
1	0	280	RDL		
1	0	280	RDL	REA	RMA
1	0	280A	RDL	RMA	
1	0	280A	REA		
1	0	280A	REA	RMA	RPL
1	0	280A	RDL	RFL	
1	0	280A	REA	RPL	
1	0	280A	REAH	RFH	
1	0	280A	REAH	RF	
1	0	280A	REAL	RMAL	RPL
1	0	280A	RFH		
1	0	280A	RF		
1	0	280A	RF	RPL	

CLOCK INFO..ATION REV 4 P400 4/6/76

3 D.a.

1	0	360	RDL	RMA	
1	0	360	RPL		
1	0	360	RDL	REA	RMA
1	0	360	RD	REAL	RMAL

CLOCKS SORTED NUMERICALLY

```
000000:     030000   160
000001:     030200   RDH,160
000002:     030000   200
000003:     030200   RDH,200
000004:     030000   240
000005:     030200   RDH,240
000006:     030000   280
000007:     030010   RCD,280
000010:     030000   320
000011:     030300   RD,320
000012:     020000   RPL,280B
000013:     036010   RCD,RF,280B
000014:     030160   RDL,RMA,280A
000015:     170000   REA,280A
000016:     030000   360
000017:     034000   RFH,360
000020:     030300   RD,160
000021:     130300   RD,REAH,160
000022:     030300   RD,200
000023:     070200   RDH,REAL,200
000024:     030300   RD,240
000025:     036200   RDH,RF,240
000026:     030210   RCD,RDH,280
000027:     030310   RCD,RDH,RDL,280
000030:     060320   RD,REAL,RMAL,RPL,320
000031:     070200   RDH,REAL,320
000032:     170060   REA,RMA,280B
000033:     030012   RCD,MRDY,280B
000034:     160060   REA,RMA,RPL,280A
000035:     032100   RDL,RFL,280A
000036:     036000   RF,360
000037:     032000   RFL,360
000040:     030100   RDL,160
000041:     130100   RDL,REAH,160
```

CLOCK INFORMATION REV 4 P400 4/6/76

```
000042:   030100   RDL,200
000043:   130000   REAH,200
000044:   036100   RDL,RF,240
000045:   134000   REAH,RFH,240
000046:   070010   RCD,REAL,280
000047:   020010   RCD,RPL,280
000050:   032100   RDL,RFL,320
000051:   136000   REAH,RF,320
000052:   034002   RFH,MRDY,280B
000053:   030202   RDH,MRDY,280B
000054:   160000   REA,RPL,280A
000055:   134000   REAH,RFH,280A
000056:   030210   RDH,RCD,360
000057:   172060   RFL,REA,RMA,360
000060:   130000   REAH,160
000061:   170000   REA,160
000062:   170060   REA,RMA,200
000063:   070000   REAL,200
000064:   136000   REAH,RF,240
000065:   132000   REAH,RFL,240
000066:   030200   RDH,280
000067:   030300   RD,280
000070:   070020   REAL,RMAL,320
000071:   034000   RFH,320
000072:   020210   RCD,RDH,RPL,280B
000073:   036062   RF,RMA,MRDY,280B
000074:   136000   REAH,RF,280A
000075:   060020   REAL,RMAL,RPL,280A
000076:   070100   RDL,REAL,360
000077:   030160   RMA,RDL,360
000100:   160060   REA,RMA,RPL,160
000101:   170060   REA,RMA,160
000102:   070020   REAL,RMAL,200
000103:   034000   RFH,200
000104:   070000   REAL,240
000105:   074020   REAL,RFH,RMAL,240
000106:   070320   RD,REAL,RMAL,280
000107:   020300   RD,RPL,280
```

```
000110:     036000   RF,320
000111:     032000   RFL,320
000112:     034022   RFH,RMAL,MRDY,280D
000113:     036100   RF,RDL,280B
000114:     034000   RFH,280A
000115:     036000   RF,280A
000116:     020000   RPL,360
000117:     170160   REA,RMA,RDL,360
000120:     070000   REAL,160
000121:     070020   REAL,RMAL,160
000122:     036000   RF,200
000123:     032000   RFL,200
000124:     034060   RFH,RMA,240
000125:     070020   REAL,RMAL,240
000126:     130200   RDH,REAH,280
000127:     034200   RDH,RFH,280
000130:     036010   RF,RCD,320
000131:     060300   RD,REAL,RPL,320
000132:     070200   RDH,REAL,280B
000134:     026000   RF,RPL,280A
000135:     024000   RFH,RPL,280A
000136:     070320   REAL,RMAL,RD,360
000140:     020000   RPL,160
000141:     030060   RMA,160
000142:     030060   RMA,200
000143:     034002   RFH,MRDY,200
000144:     036060   RF,RMA,240
000145:     034000   RFH,240
000146:     032200   RDH,RFL,280
000147:     020200   RDH,RPL,280
000150:     036300   RF,RD,320
000151:     176060   RF,REA,RMA,320
000154:     032000   RFL,280A
000155:     020000   RPL,280A
000160:     030004   DXMRDY,160
000161:     030010   RCD,160
000162:     036010   RCD,RF,200
000163:     030032   RCD,MRDY,RMAL,200
```

```
000164:     036000    RF,240
000165:     032000    RFL,240
000166:     030100    RDL,280
000167:     170160    RDL,REA,RMA,280
000170:     030210    RCD,RDH,320
000174:     030060    RMA,280A
000175:     036060    RF,RMA,280A
```

```
                UIS         IS          IA    OTHER IAC'S
```

```
          UA        UIS         IS          IA    OTHER IAC'S
* P400 IAC'S    REV 03   MS-MHJ   4-6-76
*
          This listing shows all of the IAC's and the combinations
that may be used together.  Some cobinations are shown
explicitly, and the others may be determined as follows:
          (1) IA type IAC's may be used together as shown explicitly.
   They may also be used together with any others which do
   not use the IA field.
          (2) IAC's which have a UA field of 0 or 1 shown cannot
   be used with any other non-IA field IAC.
          (3) IAC's having a specified UIS field but no UA field specified
   may be used together with any other class (3) IAC having
   the same UIS field.  Or-ing the IS bits together
   requests the several IAC's.
          (4) UACC1 and UACC2 may be used with any class
   (1) or class (3) IAC (or both), but not with
   any class (2) IAC.
*         ALL NUMBERS ARE IN OCTAL
*
*
IAC         UA          UIS         IS          IA    OTHER IAC'S
---         --          ---         --          ---

ACKPE       1           0           13
ADRTR       0           0           17
BAL                                             22
DAL                                             26    INCREA
BDSW                    4           10
CHI         0           0           14
CRDXL                   5           02
DBD         1           0           14
DECREA                                          04    SHFT      SACC1
DECREA                                          06    SHFT
DECREA                                          17
EAE                     7           02
END                     4           04
ESCPN       0           0           12
```

	UA	UIS	IS	IA		OTHER MAC'S	
LSSTRB	0	0	13				
FRADP	1	0	04				
FETCH		7	10				
GATE	1	0	16				
ICPN	0	0	11				
IEX				03	SHFT	INCREA	
IEX				03	SETCC		
IEX				25	SETCC32		
IEX				31	SACC1	SETCC32	
IEX				36	SACC1	RTN	SETCC
INCREA				02	SHFT	SACC1	
INCREA				03	SHFT	IEX	
INCREA				05	SHFT		
INCREA				15			
INCREA				26	DAL		
IND		7	01				
IND16		7	04				
INTRP	1	0	03				
INH1	1	0	10				
INK	0	0	02				
INVCI	1	0	00				
LISTLD	1	0	12				
LCAL		2	17				

Cbit selections are encoded in the IS field as follows:

C= CBIT	00
C= ALH16	01
C= PLINK	02
C= ALLCOUT	03

(note: this is not staticized - do not clock RF or RMA on a step which uses this CBIT select,)

C= COUT	04
C= DDH01	05
C= ALHOV	06
C= SOVFL	07

LINK bit selections include th MSD of the IS field and shift information as well.

L= ALH01	MSD IS field = 0 or 1. Any SHIFT$LEFT BD select.
L= COUT	MSD IS = 0

	UA	UIS	IS	IA	OT	IAC'S			
	L= BDH03		MSB IS = 1 (IS = 10)						
	L= ALL16		MSB IS field = 0 or 1. Any SHIFT$RIGHT BD select.						
LDIAG		3	17	30					
LDRPL	0	0	03						
LDRP	0	0	07						
LDTARL	0	0	06						
LLATCH		3	17						
LMOD				10	SACC1				
LMOD				11	SETCC				
LMOD				23					
LMOD				37	SACC1	RTN	RXM		
LPID	1	0	05						
LSTLB	1	0	11						
NOP				24					
NOP	0	0	00						
ORDXL		6	10						
PFL	1	0	02						
POP	1	0	15						
RADE	0	0	05						
RCCPN	0	0	10						
RACPN				12	SSTRD				
RACPN				13					
RDATE		6	01						
IORETRY	0	0	15						
SSTEP	1	0	06						
RPIO		6	02						
RSTRD		4	01						
RSYSC	0	0	16						
RTN				34					
RTN				35	SACC1				
RTN				36	SACC1	IEX	SETCC		
RTN				37	SACC1	LMOD	RXM	(200 n.s. Min. Step)	
RXM				33				(200 n.s. Min. Step)	
RXM				37	SACC1	LMOD	RTN	(200 n.s. Min. Step)	
SACC1				01	SHFT				
SACC1				02	SHFT	INCREA			
SACC1				04	SHFT	DECREA			
SACC1				10	LMOD				

	UA	UIS	IS	IA	R IAC'S		
SACC1				14			
SACC1				31	SETCC32	IEX	
SACC1				35	RTN		
SACC1				36	IEX	RTN	SETCC
SACC1				37	LMOD	RTN	RXM
SADE	0	0	04				
SCPN		4	02				
SDATE		5	01				
SETCC				07	SHFT		
SETCC				11	LMOD		
SETCC				16	IEX		
SETCC				20			
SETCC				36	IEX	RTN	SACC1
SETCC32				21			
SETCC32				25	IEX		
SETCC32				31	IEX	SACC1	
SHFT				00			
SHFT				01	SACC1		
SHFT				02	SACC1	INCREA	
SHFT				03	INCREA	IEX	
SHFT				04	SACC1	DECREA	
SHFT				05	INCREA		
SHFT				06	DECREA		
SHFT				07	SETCC		
SPARE				25			
SPARE				31			
SPIO		6	04				
SSTRB				12	RACPN		
SSTRU				32			
UACC1	2						
UACC2	3						
UBDX		1	14				
UBDX		5	14				

(200 ns. Min Step)

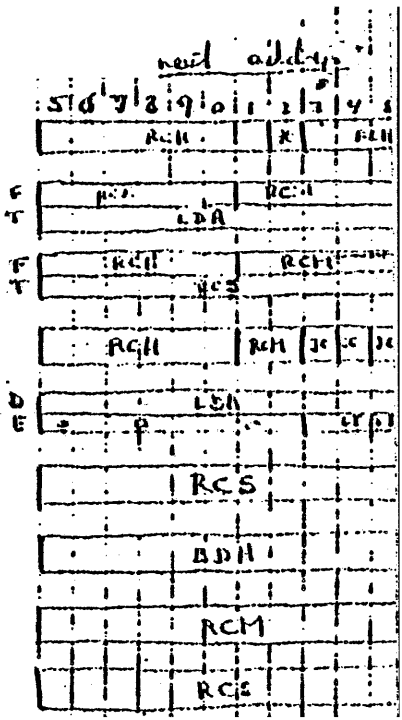
Any UBDX can use the two MSB's of the IS field to select BDX as follows:

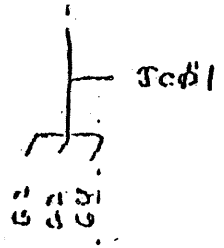
BDX= BMD	00
BDX= BPD	04
BDX= BDH	10

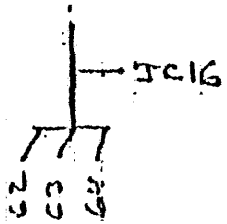
	UIS	IS	IA	OTHER AC'S
	BDX= MISC	14		
UHSM	1	03		
	Any uhsm can use the two LSB's of the IS field to select the Memory cycle as follows:			
	HSM= READ	00		
	HSM= INTREAD	01		
	HSM= WRITE	02		
	HSM= INTWRITE	03		
UPCI	1 0	01		
WKN			27	

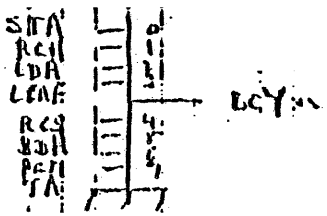
CS AU

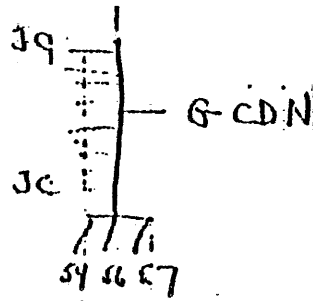
	47	24	46	10	11	12	13	14	15	16	17	18	19	20	21	22	23
emit	1	1					EMIT CONSTANT										
cond decode	0	1	0	BC11-12		0	BC13	GCDN cond	BC14	GCDN cond	BC15		JA6 cond			JA and GCDN cond	
cond return	0	1	0	BC11-12		1	BC13	RTN cond	BC14	RTN cond	BC15		JA6 cond			JA and RTN cond	
cond. branch	0	1	1	BC11-12		0 1	BC13 JA13	0	BC14 JA14	JA5 cond			JA6 cond			all JA's cond	
decode/EAF	1	0	0	1	0				misc control on DECODE/EAF								
RTN	1	0		0	0												
BD11 branch	1	0		0	1												
uncond branch	0	0							Brand address (BC03-16)								
init and IACRTN	0	0							EMIT constant								

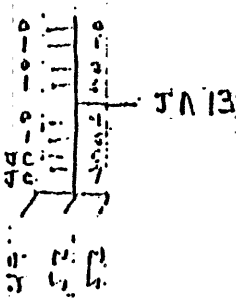


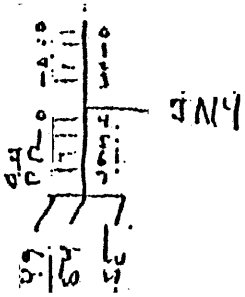


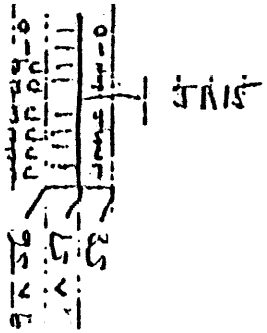


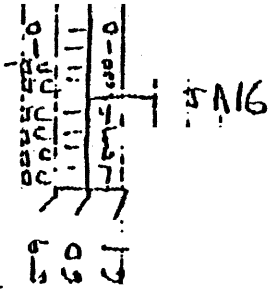


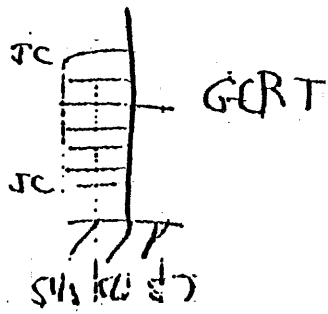












MHY
10/20/75
Rev 2 3/15/76

Jump Codes

JUMP codes are arranged by the LIT
They affect and the RCM 62→64
= which selects them. 4, 8, 12, and 16 way
branches are possible within
the same RCM JUMP code group).
For each code group there are
also 6 conditional returns available.
This table allows conditional stances to be
chosen.

Code 0 (RCM 62–64 = 0)

JC13	JC14	JC15	JC16	CRTN
REALO7	REAL14	REAL15	REAL16	1 ADLO6
	ADLO5	ADL06	ADLO7	5 ADLO5
IOBUS		DL01	MODNUM	7 NDL01
		FSPLUS	FSUIVS	3 ADLO7
		NADL06	NADLO7	4 NAL01
			CRS	2 NADL06

Code 1

JC13	JC14	JC15	JC16	CRTN
FPOSTX	BBH01	BBH04	BBH05	1 ALH0115
		ALH09	ALH01	2 NALH01
NEAC13	ALCCOUT	ALH0115	AL32NE	3 AL32NE
		REAH01	REAH05	4 AL32EQ
		NALH01	AL32EQ	5 ALCCOUT
		BDH10	RPH03	7 NBBH04

* If a JC16 conditional Return is used, only a branch may be chosen.

Code 2

JC13	JC14	JC15	JC16	CRTN
PXM	REAL11	REAL12	ALHI6	7 GFCTR
NCTAOK	FLEX	NGFCTR	GADRTR	5 FLEX
		ALHNE	ALHLT	1 ALHNE
		ALHEQ	ALHGE	2 ALHEQ
		AL32NE15	CCGE	3 ALHLT
		BBH09	NRFH01	4 ALHGE
				6 RFH02

Code 3

JC13	JC14	JC15	JC16	CRTN
CPCEXT	BPCMOD2	GAVIOL	VERY	7 NGAVIS
NGADRI	JCAP1	DAEQ20	NREADY	1 ALHGT
		ALHGT	AL32LE	2 ALHLE
		ALHLE	AL32GT	3 AL32LE
		ALLO1	ALLNE	4 AL32GT
		NBITTST	JCAP2	5 JCAP1

Code 4

JC13	JC14	JC15	JC16	CRTN
NPIOOK	QUOT	NSEGOK	NSEGTK	7 SEGOK
EAIIANI2		EAI2ORI5	GSKP	5 SOVFL
	SOVFL	CCGE15	CCNE	1 CCGE1
		CCLT	CCEQ	2 CCLT
		REALO1	REALO2	3 CCNE
				4 CCEQ

Code 5

JC13	JC14	JC15	JC16		CRTN
TMR		EXINT	NRUN	7	NEXINT
BDH13	BDH14	BDH15	BDH16	5	BDH01
	BDH01	CCLE	BDH02	1	CCLE
		CCGT	NBDH02	2	CCGT
		ALHCOUT	NRPLC1	4	NBDH02
				3	BDH02

Code 6

JC13	JC14	JC15	JC16		CRTN
			PCBIT	3	ALHOV
NFMC 2	AMO 2	AM1	AM2	4	NALHOV
BBH07	LINK 3	NXCS	ALHOV	1	NXCS
		ALCM1	CBIT	2	XCS
		XCS	NALHOV	5	LINK
		BBH08	ALHCOUT16	7	NRMAVLD
		RMAVLD			

Code 7

JC13	JC14	JC15	JC16		CRTN
FVIM		INLV	PFIT	7	NINLV
NBFCXB	MODO	MOD1	INMOD	0	NSREQ
	NSREQ	GRSCM1	ALL16	1	GRSCM1
		NGRSCM1	NALL16	2	NGRSCM1
		EBRP1	RBRP2	3	ALL16
			NREACM1	4	NALL16

JUMP CONDITIONS SORTED ALPHABETICALLY 4/6/76

JMP COND	FUNCTION	FOUND	CODE	CRTN
L05	ADRESSABLE LATCH	JA14-1	CODE 0	CRTN-5
ALL06	ADDRESSABLE LATCH	JA15-4	CODE 0	CRTN-1
ALL07	ADDRESSABLE LATCH	JA16-4	CODE 0	CRTN-3
AL32EQ	BOTH ALH AND ALL EQ ZERO	JA16-5	CODE 1	CRTN-4
AL32GT	ALH AND ALL > ZERO	JA16-5	CODE 3	CRTN-4
AL32LE	ALH AND ALL > OR = ZERO	JA16-4	CODE 3	CRTN-3
AL32NE	BOTH ALH AND ALL NE ZERO	JA16-4	CODE 1	CRTN-3
AL32NE15	ALH AND ALL NOT EQ 0	JA15-6	CODE 2	
ALCCOUT	ALC CARRY OUT	JA14-1	CODE 1	CRTN-5
ALCM1	ALC = MINUS ONE	JA15-6	CODE 6	
ALH01		JA16-3	CODE 1	
ALH0115		JA15-4	CODE 1	CRTN-1
ALH09		JA15-3	CODE 1	
ALH16		JA16-3	CODE 2	
ALHCOUT	ALH CARRY OUT	JA15-6	CODE 5	
ALHCOUT16	CARRY OUT OF ALH	JA16-7	CODE 6	
ALHEQ	ALH EQ ZERO	JA15-5	CODE 2	CRTN-2
ALHGE	ALH GREATER OR EQ ZERO	JA16-5	CODE 2	CRTN-4
ALHGT	ALH > ZERO	JA15-4	CODE 3	CRTN-1
ALHLE	ALH < OR = ZERO	JA15-5	CODE 3	CRTN-2
ALHLT	ALH LESS THAN ZERO	JA16-4	CODE 2	CRTN-3
ALHNE	ALH NOT EQ ZERO	JA15-4	CODE 2	CRTN-1
ALHOV	ARITHMETIC OVERFLOW	JA16-4	CODE 6	CRTN-3
ALL01		JA15-6	CODE 3	
ALL16		JA16-4	CODE 7	CRTN-3
ALLNE	ALL NOT = ZERO	JA16-6	CODE 3	
AM0	ADDRESS MODE 0	JA14-0	CODE 6	
AM1	ADDRESS MODE 1	JA15-3	CODE 6	
AM2	ADDRESS MODE 2	JA16-3	CODE 6	

JMP COND	FUNCTION	FOUND	CODE	CRTN
H01		JA14-0	CODE 1	
H04		JA15-2	CODE 1	
H05		JA16-2	CODE 1	
H07		JA13-1	CODE 6	
H08		JA15-7	CODE 6	
H09		JA15-7	CODE 2	
H10		JA15-7	CODE 1	
BH01		JA14-1	CODE 5	CRTN-5
BH02		JA16-4	CODE 5	CRTN-3
BH13		JA13-0	CODE 5	
BH14		JA14-0	CODE 5	
BH15		JA15-3	CODE 5	
BH16		JA16-3	CODE 5	
ICMOD2	I/O MODE LINE FOR MEM INC/PIO	JA14-0	CODE 3	
C IT		JA16-6	CODE 6	
C EQ	CONDITION CODE = ZERO	JA16-5	CODE 4	CRTN-4
C GE	CONDITION CODES >= ZERO	JA16-6	CODE 2	
C GE15	CONDITION CODE > OR = ZERO	JA15-4	CODE 4	CRTN-1
C GT	CONDITION CODE > ZERO	JA15-5	CODE 5	CRTN-2
C LE	CONDITION CODE < OR = ZERO	JA15-4	CODE 5	CRTN-1
C LT	CONDITION CODE < ZERO	JA15-5	CODE 4	CRTN-2
C NE	CONDITION CODE NOT = ZERO	JA16-4	CODE 4	CRTN-3
CRS	CURRENT REGISTER SET	JA16-7	CODE 0	
IAEQ20	REAL11-14 = '20 (PIO SPECIAL)	JA15-3	CODE 3	
DI01	DIAGNOSTIC LATCH	JA15-2	CODE 0	
IA11AN12	REAL 11 AND 12 = 1	JA13-0	CODE 4	
IA12OR15	REAL12 OR 15 =1	JA15-3	CODE 4	
I INT	POWER FAILURE PENDING	JA15-2	CODE 5	
I EX	FLOATING EXCEPTION MODE	JA14-1	CODE 2	CRTN-5
I OSTX	POST INDEXING NEEDED	JA13-1	CODE 1	
I MINUS	ADDRESSABLE LATCH 2	JA16-6	CODE 0	
I PLUS	ADDRESSABLE LATCH 1	JA15-6	CODE 0	
I IM	VECTORED INTERRUPT MODE	JA13-1	CODE 7	
I ADRTR	ADRESS TRAP	JA16-2	CODE 2	
I AVIOL	ACESS VIOLATION	JA15-2	CODE 3	
I CTR			CODE 2	CRTN-7
I CEXT	EXTERNAL PIO REQ	JA13-0	CODE 3	
I SCM1	REAL12-16 EQ MINUS ONE	JA15-4	CODE 7	CRTN-1

JMP [illegible] 0	FUNCTION	FOUND	CODE	CRTN
SKP	SKIP NET SAYS SKIP	JA16-3	CODE 4	
IRLV	MEMORY ACTIVE CAN INTERLEAVE	JA15-2	CODE 7	
IPMOD	DMX INPUT MODE LINE	JA16-3	CODE 7	
I[illegible]BUS	CHECK DURING DMX,PIO,INT	JA13-1	CODE 0	
JCAP1	ADRESS TRAP DECODE	JA14-1	CODE 3	CRTN-5
JCAP2	ADRESS TRAP DECODE	JA16-7	CODE 3	
JFP COND	FUNCTION	FOUND	CODE	CRTN
LINK	LINK FLIP-FLOP	JA14-1	CODE 6	CRTN-5
MODO	I/O BUS MODE LINE 0	JA14-0	CODE 7	
MOD1	I/O BUS MODE LINE 1	JA15-3	CODE 7	
M[illegible]ONUM	MEMORY ODD/EVEN MODULE	JA16-2	CODE 0	
MADL06	ADDRESSABLE LATCH	JA15-5	CODE 0	CRTN-2
MADL07	ADDRESSABLE LATCH	JA16-5	CODE 0	CRTN-4
MALH01		JA15-5	CODE 1	CRTN-2
MALHOV	NO ARITHMETIC OVERFLOW	JA16-5	CODE 6	CRTN-4
MALL16		JA16-5	CODE 7	CRTN-4
M[illegible]BH04			CODE 1	CRTN-7
M[illegible]DH02		JA16-5	CODE 5	CRTN-4
M[illegible]ITTST	MUX OUTPUT-SEL BDH FROM REAL	JA15-7	CODE 3	
M[illegible]PCXB	NO EXTENDER FOR DMX	JA13-0	CODE 7	
MCIAOK	CACHE INDEX NOT OK	JA13-1	CODE 2	
M[illegible]EL01			CODE 0	CRTN-7
[illegible]AC13	REAL13-16 NOT = -1	JA13-0	CODE 1	
M[illegible]EXINT			CODE 5	CRTN-7
N[illegible]MC	NOT MACHINE CHECK	JA13-0	CODE 6	
N[illegible]ADRI		JA13-1	CODE 3	
M[illegible]AVIOL			CODE 3	CRTN-7
N[illegible]FCTR	NO FETCH CYCLE TRAP	JA15-2	CODE 2	
[illegible]RSCM1	REAL12-16 NE MINUS ONE	JA15-5	CODE 7	CRTN-2
[illegible]INLV			CODE 7	CRTN-7
N[illegible]IDOK	PROCESS ID NOT OK	JA13-1	CODE 4	
[illegible]REACO1	CARRY OUT OF TOP OF REAL	JA16-7	CODE 7	
NREADY	NOT READY (PIO)	JA16-3	CODE 3	
N[illegible]FH01		JA16-7	CODE 2	
M[illegible]FL01		JA16-6	CODE 5	
RMAVLD	RMA VALID ON A CHECK		CODE 6	CRTN-7
N[illegible]UN	CONTROL PANEL SAYS STOP	JA16-2	CODE 5	
[illegible]EGOK	NO MATCH IN STLB	JA15-2	CODE 4	

JMP CON	FUNCTION	FOUND	CODE	CRTN
M:EGTR	NO STLB TRAP	JA16-2	CODE 4	
N:REQ	NO DMX REQUEST PENDING	JA14-1	CODE 7	CRTN-0
N:CS	NO EXTENDABLE CONTROL STORE	JA15-4	CODE 6	CRTN-1
PCBIT	PRE-C BIT (CBIT SOURCE)	JA16-2	CODE 6	
PFI	POWER FAILURE INTERRUPT	JA16-2	CODE 7	
PXM	PROCESS EXCHANGE MODE	JA13-0	CODE 2	
QUOT	QUOTIENT BIT	JA14-0	CODE 4	
R:RP1	RP BACK-UP COUNTER 1	JA15-6	CODE 7	
R:RP2	RP BACK-UP COUNTER 2	JA16-6	CODE 7	
REAH01		JA15-6	CODE 1	
REAH05		JA16-6	CODE 1	
REAL01		JA15-6	CODE 4	
REAL02		JA16-6	CODE 4	
REAL07		JA13-0	CODE 0	
REAL11		JA14-0	CODE 2	
REAL12		JA15-3	CODE 2	
REAL14		JA14-0	CODE 0	
REAL15		JA15-3	CODE 0	
REAL16		JA16-3	CODE 0	
RH02				CRTN-6
RAVLD	RMA VALID ON A CHECK	JA15-2	CODE 6	
RH03	RING ZERO IF RESET	JA16-7	CODE 1	
RGOK			CODE 4	CRTN-7
SOVFL	SHIFT OVERFLOW (ALH01 NOT= ALH	JA14-1	CODE 4	CRTN-5
TMR	CPU INTERNAL TIMER OVERFLOW	JA13-1	CODE 5	
VIRY	U-VERIFY ROUTINES IF TRUE	JA16-2	CODE 3	
XCS	EXTENDABLE CONTROL STORE	JA15-5	CODE 6	CRTN-2

48.

JUMP CONDITIONS -- SORTED BY CODE GROUP 4/6/76

*
*

JMP COND	FUNCTION	FOUND	CODE	CRTN
NDL01			CODE 0	CRTN-7
REAL07		JA13-0	CODE 0	
IOBUS	CHECK DURING DMX/PIO/INT	JA13-1	CODE 0	
REAL14		JA14-0	CODE 0	
ADL05	ADRESSABLE LATCH	JA14-1	CODE 0	CRTN-5
DL01	DIAGNOSTIC LATCH	JA15-2	CODE 0	
REAL15		JA15-3	CODE 0	
ADL06	ADDRESSABLE LATCH	JA15-4	CODE 0	CRTN-1
NADL06	ADDRESSABLE LATCH	JA15-5	CODE 0	CRTN-2
FSPLUS	ADDRESSABLE LATCH 1	JA15-6	CODE 0	
MODNUM	MEMORY ODD/EVEN MODULE	JA16-2	CODE 0	
REAL16		JA16-3	CODE 0	
ADL07	ADDRESSABLE LATCH	JA16-4	CODE 0	CRTN-3
NADL07	ADDRESSABLE LATCH	JA16-5	CODE 0	CRTN-4
FSMINUS	ADDRESSABLE LATCH 2	JA16-6	CODE 0	
CRS	CURRENT REGISTER SET	JA16-7	CODE 0	
NBBH04			CODE 1	CRTN-7
NEAC13	REAL13-16 NOT = -1	JA13-0	CODE 1	
FPOSTX	POST INDEXING NEEDED	JA13-1	CODE 1	
BBH01		JA14-0	CODE 1	
ALCCOUT	ALC CARRY OUT	JA14-1	CODE 1	CRTN-5
BBH04		JA15-2	CODE 1	
ALH09		JA15-3	CODE 1	
ALH0115		JA15-4	CODE 1	CRTN-1
NALH01		JA15-5	CODE 1	CRTN-2
REAH01		JA15-6	CODE 1	
BBH10		JA15-7	CODE 1	
BBH05		JA16-2	CODE 1	
ALH01		JA16-3	CODE 1	
AL32NE	BOTH ALH AND ALL NE ZERO	JA16-4	CODE 1	CRTN-3
AL32EQ	BOTH ALH AND ALL EQ ZERO	JA16-5	CODE 1	CRTN-4
RPH03	RING ZERO IF RESET	JA16-7	CODE 1	
REAH05		JA16-6	CODE 1	

MP COND	FUNCTION	FOUND	CO	CRTN
GFCTR			CODE 2	CRTN-7
PXM	PROCESS EXCHANGE MODE	JA13-0	CODE 2	
NCIAOK	CACHE INDEX NOT OK	JA13-1	CODE 2	
REAL11		JA14-0	CODE 2	
FLEX	FLOATING EXCEPTION MODE	JA14-1	CODE 2	CRTN-5
NGFCTR	NO FETCH CYCLE TRAP	JA15-2	CODE 2	
REAL12		JA15-3	CODE 2	
ALHNE	ALH NOT EQ ZERO	JA15-4	CODE 2	CRTN-1
ALHEQ	ALH EQ ZERO	JA15-5	CODE 2	CRTN-2
AL32NE15	ALH AND ALL NOT EQ 0	JA15-6	CODE 2	
BBH09		JA15-7	CODE 2	
GADRTR	ADRESS TRAP	JA16-2	CODE 2	
ALH16		JA16-3	CODE 2	
ALHLT	ALH LESS THAN ZERO	JA16-4	CODE 2	CRTN-3
ALHGE	ALH GREATER OR EQ ZERO	JA16-5	CODE 2	CRTN-4
CCGE	CONDITION CODES >= ZERO	JA16-6	CODE 2	
NRFH01		JA16-7	CODE 2	
RFH02				CRTN-6
NGAVIOL			CODE 3	CRTN-7
GPCEXT	EXTERNAL PIO REQ	JA13-0	CODE 3	
NGADRI		JA13-1	CODE 3	
BPCMOD2	I/O MODE LINE FOR MEM INC,PIO	JA14-0	CODE 3	
JCAP1	ADRESS TRAP DECODE	JA14-1	CODE 3	CRTN-5
GAVIOL	ACESS VIOLATION	JA15-2	CODE 3	
DAEQ20	REAL11-14 = '20 (PIO SPECIAL)	JA15-3	CODE 3	
ALHGT	ALH > ZERO	JA15-4	CODE 3	CRTN-1
ALHLE	ALH < OR = ZERO	JA15-5	CODE 3	CRTN-2
ALL01		JA15-6	CODE 3	
NBITTST	MUX OUTPUT-SEL BDH FROM REAL	JA15-7	CODE 3	
VIRY	U-VERIFY ROUTINES IF TRUE	JA16-2	CODE 3	
NREADY	NOT READY (PIO)	JA16-3	CODE 3	
AL32LE	ALH AND ALL > OR = ZERO	JA16-4	CODE 3	CRTN-3
AL32GT	ALH AND ALL > ZERO	JA16-5	CODE 3	CRTN-4
ALLNE	ALL NOT = ZERO	JA16-6	CODE 3	
JCAP2	ADRESS TRAP DECODE	JA16-7	CODE 3	
SEGOK			CODE 4	CRTN-7
EA11AN12	REAL 11 AND 12 = 1	JA13-0	CODE 4	
NPIDOK	PROCESS ID NOT OK	JA13-1	CODE 4	

MP CO	FUNCTION	FOUND	CODE	CRTN
QUDT	QUOTIENT BIT	JA14-0	CODE 4	
SOVFL	SHIFT OVERFLOW (ALH01 NOT= ALH	JA14-1	CODE 4	CRTN-5
NSEGOK	NO MATCH IN STLB	JA15-2	CODE 4	
EA12OR15	REAL12 OR 15 =1	JA15-3	CODE 4	
CCGE15	CONDITION CODE > OR = ZERO	JA15-4	CODE 4	CRTN-1
CCLT	CONDITION CODE < ZERO	JA15-5	CODE 4	CRTN-2
REAL01		JA15-6	CODE 4	
NSEGTR	NO STLB TRAP	JA16-2	CODE 4	
GSKP	SKIP NET SAYS SKIP	JA16-3	CODE 4	
CCNE	CONDITION CODE NOT = ZERO	JA16-4	CODE 4	CRTN-3
CCEQ	CONDITION CODE = ZERO	JA16-5	CODE 4	CRTN-4
REAL02		JA16-6	CODE 4	
NEXINT			CODE 5	CRTN-7
BDH13		JA13-0	CODE 5	
TMR	CPU INTERNAL TIMER OVERFLOW	JA13-1	CODE 5	
BDH14		JA14-0	CODE 5	
BDH01		JA14-1	CODE 5	CRTN-5
EXINT	POWER FAILURE PENDING	JA15-2	CODE 5	
BDH15		JA15-3	CODE 5	
CCLE	CONDITION CODE < OR = ZERO	JA15-4	CODE 5	CRTN-1
CCGT	CONDITION CODE > ZERO	JA15-5	CODE 5	CRTN-2
ALHCOUT	ALH CARRY OUT	JA15-6	CODE 5	
NRUN	CONTROL PANEL SAYS STOP	JA16-2	CODE 5	
BDH16		JA16-3	CODE 5	
BDH02		JA16-4	CODE 5	CRTN-3
NBDH02		JA16-5	CODE 5	CRTN-4
NRFL01		JA16-6	CODE 5	
NRMAVLD	RMA VALID ON A CHECK		CODE 6	CRTN-7
NFMC	NOT MACHINE CHECK	JA13-0	CODE 6	
BBH07		JA13-1	CODE 6	
AM0	ADDRESS MODE 0	JA14-0	CODE 6	
LINK	LINK FLIP-FLOP	JA14-1	CODE 6	CRTN-5
RMAVLD	RMA VALID ON A CHECK	JA15-2	CODE 6	
AM1	ADDRESS MODE 1	JA15-3	CODE 6	
NXCS	NO EXTENDABLE CONTROL STORE	JA15-4	CODE 6	CRTN-1
XCS	EXTENDABLE CONTROL STORE	JA15-5	CODE 6	CRTN-2
ALCM1	ALC = MINUS ONE	JA15-6	CODE 6	
BBH08		JA15-7	CODE 6	

MP COND	FUNCTION	FOUND	COD	CRTN
PCBIT	PRE-C BIT (CBIT SOURCE)	JA16-2	CODE 6	
AM2	ADDRESS MODE 2	JA16-3	CODE 6	
ALHOV	ARITHMETIC OVERFLOW	JA16-4	CODE 6	CRTN-3
NALHOV	NO ARITHMETIC OVERFLOW	JA16-5	CODE 6	CRTN-4
CBIT		JA16-6	CODE 6	
ALHCOUT16	CARRY OUT OF ALH	JA16-7	CODE 6	
NINLV			CODE 7	CRTN-7
NOPCXB	NO EXTENDER FOR DMX	JA13-0	CODE 7	
FVIM	VECTORED INTERRUPT MODE	JA13-1	CODE 7	
MOD0	I/O BUS MODE LINE 0	JA14-0	CODE 7	
NSREQ	NO DMX REQUEST PENDING	JA14-1	CODE 7	CRTN-0
INLV	MEMORY ACTIVE CAN INTERLEAVE	JA15-2	CODE 7	
MOD1	I/O BUS MODE LINE 1	JA15-3	CODE 7	
GRSCM1	REAL12-16 EQ MINUS ONE	JA15-4	CODE 7	CRTN-1
NGRSCM1	REAL12-16 NE MINUS ONE	JA15-5	CODE 7	CRTN-2
RBRP1	RP BACK-UP COUNTER 1	JA15-6	CODE 7	
PFI	POWER FAILURE INTERRUPT	JA16-2	CODE 7	
INMOD	DMX INPUT MODE LINE	JA16-3	CODE 7	
ALL16		JA16-4	CODE 7	CRTN-3
NALL16		JA16-5	CODE 7	CRTN-4
RBRP2	RP BACK-UP COUNTER 2	JA16-6	CODE 7	
NREACO1	CARRY OUT OF TOP OF REAL	JA16-7	CODE 7	

P400 U-CODE TIMING

The time for any u-step to complete is the longest path of the several required for that step.

1. If RCD is used, the minimum time is 280 ns.

2. The times for data path operations can be computed by adding up the times for each section.
   a. Stable ALH or ALL output:
      126 ns for logical operations.
      144 ns for 16 bit arithmetic operations
      169 ns for 32 bit arithmetic operations

   b. Stable ALC output:
      +24 ns for ALC logical operations added to ALH or ALL time
      +42 ns for ALC arithmetic operations

   c. Stable Cache data
      +65 ns at output of ALU's with no accellerate.
      +0 ns at output of ALU's with any accelleration

   d. Bus D to Destination
      +100 ns to Register File.
      +34 ns to any other destination.
      160 ns from begining of cycle if ALU's not used and destination is not Register Files.
      200 ns from beginning of cycle if ALU's not used and destination is a Register File.

   e. Bus B to Destination
      +0 ALU selection is slower and they must be used for transport.

3. Conditional Branches--Returns
   a. Stable conditions at the beginning of the cycle.

160 ns for a conditional or unconditional
branch.
200 ns for a conditional return
160 ns for an unconditional return

b. Other conditions
134 ns after stability for conditional branches.
(If test is ALUNE, take 144 ns for stable ALU data
and add 134 ns for the branch giving 278 => 280 ns
for the cycle.)
147 ns after stability for conditional returns.

4. If all Traps are disabled, then 280 ns is the minimum.

5. Accelleration.
a. Cache data because of pre-loaded RMA.
Ignore cache in calculation.
b. ALU to RF -- previous step is like this one.
160 ns (accellerated -- request 240) for logical
and 16 bit arithmetic operations.
200 ns (request 280) for 32 bit arithmetic operations.

The u-code assembler does a good but conservative job of calculating data path times. It does not attempt to compute the times for the live conditional branches. The use of T= XXX statements to the assembler for any conditional branch on a 'live' condition is reccommended. The assembler selected times are almost always far to fast.

MACROS FOR P400***** USE PMA64 ONLY *****

A MODIFIED BNF TYPE LANGUAGE IS USED TO DESCRIBE THE P400 ASSEMBLY LANGUAGE.

[...] BRACKETS INDICATE OPTIONAL PARAMETERS

! SEPARATES ALTERNATIVE CHOICES

<...> METASYMBOL, ITEM FITTING THE DEFINITION OF THE SYMBOL MUST BE SUBSTITUTED BFOR THE SYMBOL.

<...>;= METASYMBOL DEFINITION

MACRO DESCRIPTIONS

--- IDNT MACRO --- THIS MACRO IS USED FOR IDENTIFICATION

```
        IDNT  (<STRING>),(<STRING>),...
```

<STRING>:= A STRING OF UP TO 30 CHARACTERS. THE STRING MAY NOT INCLUDE SEMI-COLONS,COLONS OR PARENTHESIS.

--- ALU MACRO ---

```
[<LABEL>]  ALU  [<BDEARLY SPEC>]  <RF SPEC>  <OP SPEC>
               <AB SPEC> => [<BD SOURCE>]  <GENERIC DEST>  [<TIME SPEC>]
                [<IAC SPEC>]  [<ACT SPEC>]
```

--- OPERATOR MACROS ---

```
[<LABEL>]  <MACRO NAME> [<BDEARLY SPEC>]  <GEN1 SOURCE> =>  [<BD SOURCE>]
                        <GENERIC DEST>  [<TIME SPEC>]  [<IAC>] [<ACT SPEC>]

<MACRO NAME>:=INC!NOT!DEC!CON

<GEN1 SOURCE>:=<RF SPEC>  FOR' INC,DEC
              :=<CON SPEC> FOR CON
              :=<GEN2 SOURCE>  FOR NOT

  <CON SPEC>:=ZERO!MINUS1
```

--- TEST MACRO ---

```
[<LABEL>]  TEST  [<BDEARLY SPEC>]  <GEN2 SOURCE>  [<TIME SPEC>]
                 [<IAC SPEC>]  [<ACT SPEC>]
```

```
--- NOP MACRO ---

[<LABEL>] NOP [<BDEARLY SPEC>] [<TIME SPEC>] [<IAC SPEC>] [<ACT SPEC>]


--- RR MACRO ---

[<LABEL>]  RR  [<BDEARLY SPEC>]  <GEN3 SOURCE>  =>  [<BD SOURCE>]
               <GENERIC DEST> [<TIME SPEC>] [<IAC SPEC>] [<ACT SPEC>]


<GEN2 SOURCE>:=<RF SPEC>!<BB SPEC>
<GEN3 SOURCE>:=<RF SPEC>!<BB SPEC>|BMD|BPD|REA|REAL|REAH|RP|
               RPL|RPH|BPA|BDX
```

```
<DDEARLY SPEC>:=EA>MA ! P>MA

<RF SPEC>:=<RF MNEMONIC> <LENGTH>                    (2)   (1)

  <RF MNEMONIC>:=,!
                 X!A!B!S!Y!VSC!E!EH!EL!
                 GR(RD)!GR(RS)!GR(FD)!GR(FS)!
                 GR(RDN)!GR(RSN)!GR(FDN)!GR(FSN)!
                 GR(BR)!GR(DTAR)!
                 GRH(RD)!GRH(RS)!GRH(FD)!GRH(FS)!
                 GRH(RDN)!GRH(RSN)!GRH(FDN)!GRH(FSN)!
                 GRH(BR)!GRH(DTAR)!
                 GRL(RD)!GRL(RS)!GRL(FD)!GRL(FS)!
                 GRL(RDN)!GRL(RSN)!GRL(FDN)!GRL(FSN)!
                 GRL(BR)!GRL(DTAR)!
                 LB!SB!XB!PB!
                 LBH!SBH!XBH!PBH!
                 LBL!SBL!XBL!PBL!
                 DTAR2!DTAR3!DTAR0!DTAR1!
                 TR0!TR1!TR2!TR3!TR4!TR5!TR6!TR7!RDMX1!RDMX2!
                 TR0H!TR1H!TR2H!TR3H!TR4H!TR5H!TR6H!TR7H!RDMX1H!RDMX2H!
                 TR0L!TR1L!TR2L!TR3L!TR4L!TR5L!TR6L!TR7L!RDMX1L!RDMX2L!
                 RF(DMA)!
                 RF(REAL)!RFH(REAL)!RFL(REAL)!
                 RF(AMAP)!RFH(AMAP)!RFL(AMAP)!
                 KEYS!OWNER!FCODE!FADDR!TIMER!
                 KEYSH!OWNERH!FCODEH!FADDRH!TIMERH!
                 KEYSL!OWNERL!FCODEL!FADDRL!TIMERL!
                 PSWPB!PSWKEYS!PPA!PPB!
                 PSWPBH!PSWKEYSH!PPAH!PPBH!
                 PSWPBL!PSWKEYSL!PPAL!PPBL!
                 DSWRMA!DSWSTAT!DSWSTATH!DSWSTATL!
                 RSGT1!RSGT2!RSGT1H!RSGT2H!RSGT1L!RSGT2L!
                 RECC1!RECC2
```

<LENGTH>:= 16| 32|LIH | NULL (1) (2)

```
<OP SPEC>:=<OP1 SPEC>!<OPS SPEC>

  <OP1 SPEC>:=PLUS!MINUS!AND!OR!XOR!TA!TB!INC!DEC

  <OPS SPEC>:=<ALH SPEC> <ALL SPEC> <ALC SPEC> <CH SPEC> <CL SPEC> <CC SPEC>

    <ALH SPEC>:= NULL !  ALH=  (<A SPEC> <OP2 SPEC> <B SPEC>)   (1)

    <ALL SPEC>:= NULL !  ALL= (<A SPEC> <OP2 SPEC> <B SPEC>)   (1)

    <ALC SPEC>:= NULL !  ALC= (<A SPEC> <OP2 SPEC> <B SPEC>)   (1)

    <CH SPEC>:= NULL ! (CH= COL!1!CDIT!0)

    <CL SPEC>:= NULL ! (CL= 0!1!CDIT)

    <CC SPEC>:= NULL ! (CC= 0!1).

      <OP2 SPEC>:=<OP1 SPEC>!ADD!SUB!NOT!MPY!DIV!MPYFS!FETCH!ZERO!MINUS1

      <A SPEC>:=A ! ANOT!NULL

      <B SPEC>:=B ! BNOT!NULL
```

```
<BB SPEC>:=<RCM SPEC>!<RCD SPEC> !
          (<RCM SPEC>,<RCD SPEC>)!(<RCM SPEC>,RDL)!(RDH,<RCD SPEC>)!RD!
          RMA!(RMAH,RMAL)!(RMAH,<RCM SPEC>)!(<RCD SPEC>,RMAL)!(RDH,RDL)!
          (<RCD SPEC>,<RCM SPEC>)! ,                                  (1)  (6)

 <RCM SPEC>:= RCM <LENGTH>!= <EXPRESSION> <LENGTH>  (1)
 <RCD SPEC>:= RCD <LENGTH>          (1)

<BD SOURCE>:= <BD SPEC>!<SREND SPEC>

 <BD SPEC>:= '[ ' HL!HC!CC!CL!RFHC!RFHL!HRFL!CRFL!
          REA!RP!DMX!BPA!H8X8BPA!DISABLE!BDXC!RFHRFL!
          H8X8!!RFHLL1!BDX!C10X6C!C10X6  ']' ! NULL      (5)  (8)

 <SREND SPEC>:= '[ '<SHIFT SPEC>' ]' <END SPEC> !
                '[ '<ROTATE SPEC>' ]' <END SPEC>     (5)

   <SHIFT SPEC>:=SHIFT$<LR1 SPEC> !NULL

     <LR1 SPEC>:=LEFT!RIGHT!SLEFT!REALEFT

  <ROTATE SPEC>:=ROTATE$<LR2 SPEC>! NULL

     <LR2 SPEC>:=SLEFT!LLEFT!SRIGHT!LRIGHT

   <END SPEC:= NULL! E=  0!LINK!ALH00!REAH01!ALH01
```

```
<GENERIC DEST>:= (<RF SPEC>,<DEST SPEC>)                    (3)

  <DEST SPEC>:=REAH!REAL!RPH!RPL!RMAH!RMAL!RDH!RDL!RDX!MEMORY!
               REA!RP!RMA!RD!RCD

<TIME SPEC>:=<ACCEL SPEC> <VAL SPEC>

  <VAL SPEC>:= 160!200!240!280!320!360

  <ACCEL SPEC>:=UA1!UA2!T=

<IAC SPEC>:= ACKPE!ADRTR!IACBAL!BDSW!CHI!CRDXL!DBB!DECREA!EAF!ENB!
             ESCPN!ESSTRB!LDRP!LPID!RMC!RMMOD!RSYSC!
             FBADP!FETCH!GATE!IEX!INCREA!IND!IND16!INH1!INK!ICPN!
             INVCI!LISTLB!LCAL!LDIAG!LDRPL!LSTLB!LDTARL!
             LLATCH!LMOD!NTRAP!ORDXL!PFL!POP!RADE!RACPN!RCCPN!
             RDATE!RPIO!RSTRB!RTN!RXM!SACC1!SADE!SCPN!SDATE!
             SETCC!SETCC32!SHFT!SPIO!SSTRB!UACC1!UACC2!UBDX!
             UHSM!UPCI!WKN!SHFTREA!INCRP!MRDY!NTRAP!NOP

<ACT SPEC>:=<RDEC SPEC>!<JUMP SPEC>!<GOTO SPEC>!
            <BAL SPEC>!<CS SPEC>!RTN!NULL

  <RDEC SPEC>:= CRTN <EXPRESSION> !
                CRTN <EXPRESSION> ELSE <GOTO SPEC>!
                CRTN <EXPRESSION> <JUMP SPEC>!
                CDECODE <EXPRESSION>!
                CDECODE <EXPRESSION> ELSE <GOTO SPEC>!
                CDECODE <EXPRESSION> <JUMP SPEC>!
                NULL

  <JUMP SPEC>:=JUMP  <COND SPEC> TO <ADDR SPEC> ! NULL    (9)

    <COND SPEC>:=(<JA13>,<JA14>,<JA15>,<JA16>)          (4)  (1)

      JA13,JA14,JA15,JA16:=JUMP NET ASSOCIATED WITH BITS 13,14,15, OR 16

    <ADDR SPEC>:= (ADDR1, ADDR2, ... ADDRN)             (4)  (7)  (9)
```

```
<GOTO SPEC>:=GOTO <EXPRESSION> ! NULL

<BAL SPEC>:= BAL <COND SPEC> TO <ADDR SPEC> !BAL <EXPRESSION>

<CS SPEC>:= CS= (0,<CS0 SPEC>)!CS= (1,<CS1 SPEC>)!CS =(2,<CS2 SPEC>)!
            CS= (3,<CS3 SPEC>)!NULL   (1)

  <CS2 SPEC>:=BDH!RCS!EAF!DECODE <EXPRESSION>

  <EXPRESSION>:=ANY VALID PMA EXPRESSION

----------------------
COMMENT:  TO DEACTIVATE ADDRESS CHECKING ON 2 AND 4-WAY BRANCHES
          (NONE EXISTS FOR 8 AND 16-WAY BRANCHES) , THE VALUE
          OF NOCK$ IN SET$ MACRO MUST BE SET TO 0.
----------------------
```

NOTES:

(1). IF MORE THAN ONE ARGUMENT IS USED IN ANY FIELD, PARENTHESES MUST BE USED TO ENCLOSE THE ENTIRE ARGUMENT LIST. IE. (A,32) WOULD BE AN EXAMPLE OF <RF SPEC>. ALSO, SEE NOTE 7 BELOW.

(2). <LENGTH>:USED ONLY WHEN THE DEFAULT LENGTH OF A REGISTER IS TO BE OVERRIDDEN

(3). <GENERIC DEST>:DESTINATIONS CAN BE CONCATENATED AS (RPH,REAH)

(4). <COND/ADDR SPEC>:NUMBER OF ADDRESSES MUST EQUAL 2**(NUMBER OF CONDITIONS) (SEE 10. BELOW)

(5). <BD SPEC>/<SHIFT SPEC>/<ROTATE SPEC>: IF ANY ONE OF THESE EXISTS, IT MUST BE ENCLOSED IN BRACKETS []. IE. [HL] AND [SHIFT$LEFT]. NOTE THAT THERE MUST BE A SPACE AFTER [

(6). <BD SPEC>: CONCATENATION CAN BE SPECIFIED AS (RDH,RCD).

(7). <ADDR SPEC>: SEPARATE THE ITEMS IN THE LIST

(8). <BD SPEC>: THE DEFAULT IS HL

(9). THERE IS A MAXIMUM OF 32 CHARACTERS WITHIN PARENTHESIS.

(10). FOR 8 AND 16-WAY BRANCHES, MERELY SPECIFY THE FIRST ADDRESS, SINCE PMA IS UNABLE TO HANDLE THE LONG STRING THAT WOULD BE GENERATED BY LISTING THEM ALL. THE USER IS THEN RESPONSIBLE FOR ALLOCATING ALL 8 AND 16-WAY BRANCHES PROPERLY.

NOISE WORDS =>,],MIDDLE,ON,ELSE

```
---  ----------------
SITIONAL NOISE WORDS
----------------------

        (ALH=)=99,(ALL=)=100,(ALC=)=101,;
        (E)=102,(E=)=103,(HOLE)=104,(TR=)=105,;
        (L=)=106,(CH=)=107,(CL=)=108,(CC=)=109,(SETLATCH)=110,;
        (RESETLATCH)=111,(SETDLTCH)=112,(RESETDLTCH)=113,;
        (C=)=114,(=)=115,(UA1)=116,(UA2)=117,(T=)=118,;
        (JUMP)=120,(CDECCDE)=121,(CRTN)=122,(GOTO)=123,(BAL)=124,;
        (BDX=)=127,(HSM=)=128,(CS=)=129,(TO)=130,(ALSO)=131


--------------------------
PECIAL SYMBOLS   (RESERVED WORDS)
--------------------------


         SEE <RCM SPEC>
R= NONE!TDMX!NX!ALL   (DEFAULT IS ALL)
=  CBIT!COUT!PLINK!ALHOV!ALH16!BDH01!SOVFL!ALLFCOUT
OLE IN MIDDLE  *(USED WITH <SHIFT SPEC> AND <ROTATE SPEC>
=     SEE <END SPEC>
=  ALH01!ALL16!COUT!BDH03
LH=      SEE <ALH SPEC>
LL=      SEE <ALL SPEC>
LC=      SEE <ALC SPEC>
H=COL!1!CBIT!0    NOTE**
L=0!1!CBIT         IF CH,CL OR CC IS USED <OPS SPEC> MUST BE USED
C=0!1              TO SPECIFY <OP SPEC>
ETLATCH N
ESETLATCH N
ETDLTCH N
ESETDLTCH N
SM=  <MEM SPEC>
 <MEM SPEC>:=READ!WRITE!INTREAD!INTWRITE
BX= <BDX SPEC>
```

```
  <BDX SPEC>:=BMD!BPD!BDH!MISC
UA1          USE ACCELERATE 1
UA2          USE ACCELETATE 2
T=           TIME=
JUMP
GOTO
BAL
CRTN
CDECODE
TO
=>
----------------
```