
U-CODE HANDBOOK REV 2

P 4 0 0

HHJ

4-6-76

INDEX

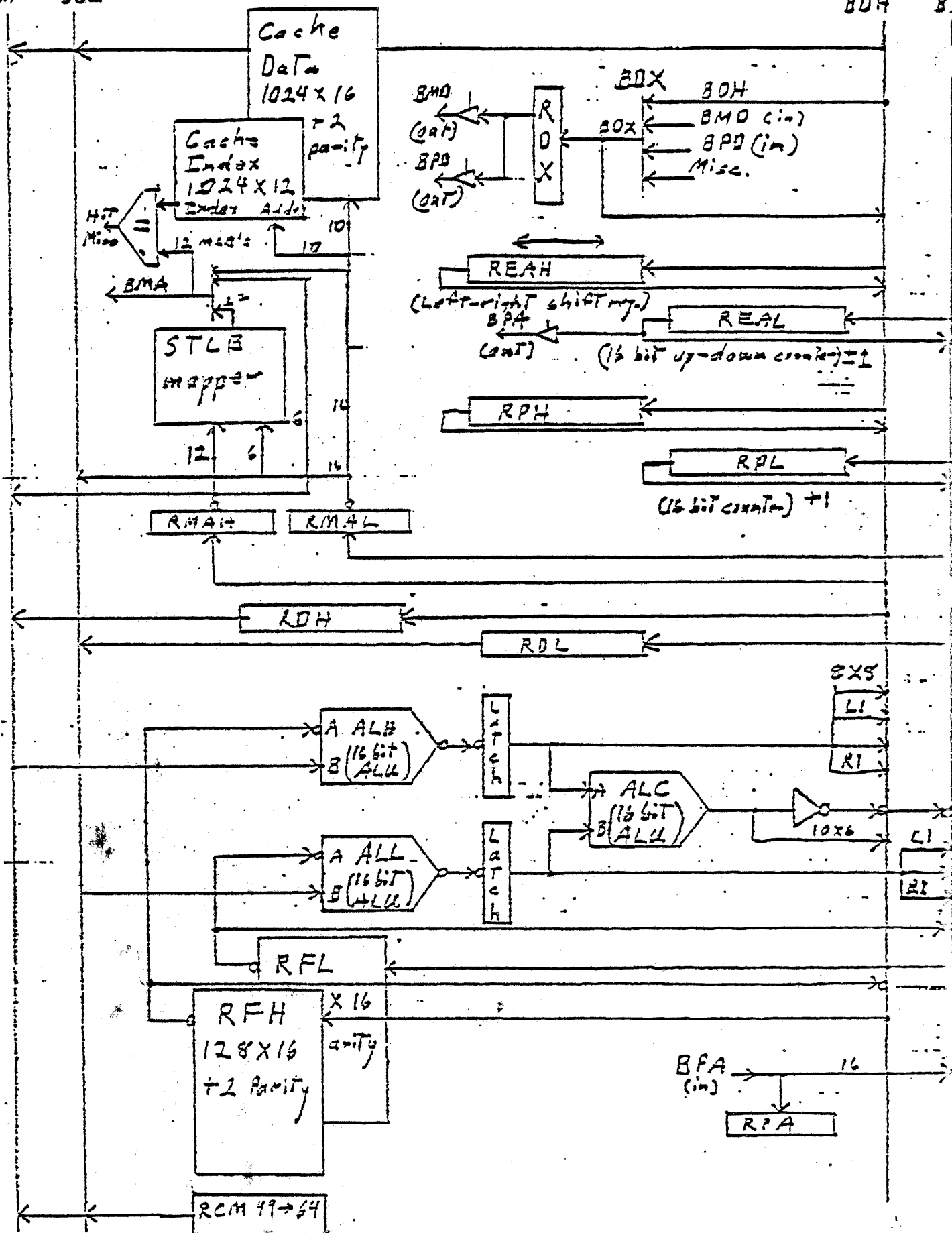
Topic

I	Index	1
II	Block Diagram	2-3
III	μ -Code Word and Field Description	
	1. μ -code Word	4-5
	2. Bus B	6
	3. Bus D (includes shifts-ends, addressable latch)	7-9
	4. Register File (includes Control Panel Operation)	10-13
	5. ALU Modes	14-17
	6. Destinations - Times	18-34
	7. IAC's includes Cbit	35-39
	8. Control Unit - Next Address (includes Jump Codes)	40-51
IV	Calculating μ -step Times	52-53
V	K400 - μ -code assembler definitions	54-65

1/29/76
M.H.
Rev 1 3/15/76
2 4/6/76
Page

B2H B3L

B0H B0L



P400 M-CODE WORD

BIT	Id	Default	Defined By	In need by	Remarks
1	} BBH BBL select	0	<BB SPEC>	<RF SPEC> if Left, Right side for RCD, RCM not obvious	8 combinations fixed available: see list
2		1			
3		1			
4	} ALH ALL ALC select and Mode	0	<OP SPEC>	RR Macro: others use specific ALH modes. Extensive error checking done on selected modes.	64 combinations FROM generated from the thousands possible. See associated list for the 64 defined modes.
5		CIN 0			
6		CIB 0	CIN =		
7		CALC 0	CIB =		
8		source 0	CALC =		
9		select 0			
10	} Inde- pendent Selects.	0	SETLATCH N	C = (bits 11-13)	Many different functions are gathered into this one field
11		0	RESETLATCH N	L = (bit 10)	
12		0	SETDLATCH N	BDX = (bits 10-11)	
13		0	RESETDLATCH N	MEMORY HSM, <MEM SPEC>	
14	} BDH BDL source	0	<BD SPEC>	E =	4 bits plus IAC SHIFT select one of 32 possible BD enables (see list).
15		0		note: also specifies REAN Lnd/Shift	
16		0		2. requires IACSHIFT for shifts	
17		1			
18	} Register Files select	0	<RF mnemonic>		See the associated list for the various direct and "indirect" addresses possible.
19		0			
20		0			
21		0			
22		0			
23		0			
24		0			
25	} BD Early Cache Control	0	if = 1, use RCD (cache Read, Mapped)		set if RCD is a source
26		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)

Bit	Field	Default	Defined By	Influenced by	Description
29 } 30 } 31 } 32 } 33 } 34 }	UA } UIS } IA }	0 0 0 0 0 0	<IAC SPEC>	<TIME SPEC> <SRENU SPEC> <ACT SPEC> <GEN DEST> (Others)	Several IAC's are often used at various times. See the IAC list for details.
35 } 36 } 37 } 38 }		0 0 0 0			
39 } 40 } 41 } 42 } 43 }	Clock	0 0 0 0 0	<GEN DEST> <TIME SPEC>	<IAC SPEC>	Each of the 128 codes selects a group of registers to be clocked (updated) and a time to be used.
44	DMX control 1 =		DMX	0 = Inhibit DMX	
46	Check (odd parity)			Parity bit is built by assembler-	
47 } 48 }	Control Select				
47			[=3] (EINIT) [=-1] SHORT JUMP	[=-0] LONG JUMP [=-2] Select BCY <CS spec>	
50 } 51 } 52 } 53 } 54 } 55 } 56 } 57 } 58 } 59 } 60 } 61 }	EINIT (Data value) = RCM= <DATA SPEC> etc.		{ 1 for SHORT JUMP, 0 for <ACT SPEC>= <CODECODE>, <CRTN>; 0 else. } NA bits 11, 12 { if 49=1 form } if 49=0 { 1 = CRTN NA13 like NA16. } NA bit 13 { if 49=1 form } if 49=0 { 2 = CODECODE NA14 like NA16. } NA bit 14 { if 49=1 form. } if 49=0 { select CODECODE NA15 like } on CRTN term NA16 } NA bit 15 { NA bit 16. Jump Condition select. NA16 = 61 if 59=60 = 0 } Select 1 of 8 conditions on 16 "JC" multi, exors	} Select 0 Short Jump 4 RCS BCY 1 RCH 5 BDH 2 DA 6 RCM 3 EAF 7 TA 1 1 P500 Relative Fetch Segmented Fetch Sectoral Fetch - Sectoral Fetch. P500 - P500 or Sectoral { = 1 for Sectoral, Fetch - unused for others.	} Unused except for Fetch } Long Jump } NA } <GOTO> } Data value } if } RTN.
62 } 63 } 64 }					

BUS B MHJ 1-28-76

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 MHJ 1-28-76

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.

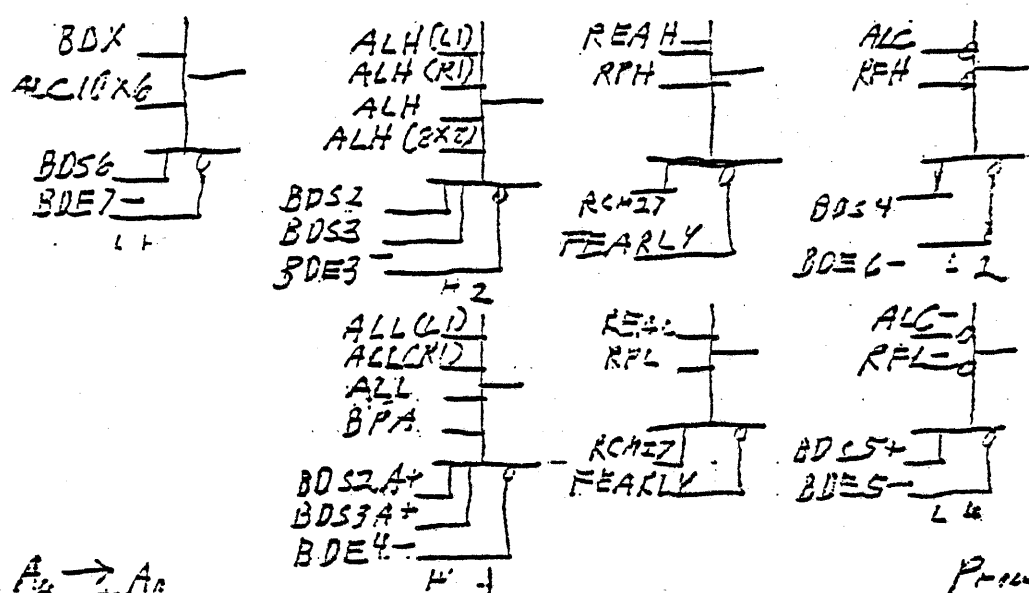
The u-code field which controls this also specifies
if cache is to be used this step or not. The encoding is:

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 controlled 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 0 Summary

1-00
MLH5
10/25/75



Pa-T3
= 825123

A₄ → A₀
B₁₅
LACS447+
RC414-17

Function
REAH or RP (Gate)

END!

Prim-Code - Octal
Dip Site → A24K → BDS2 → 56
A30K → GE44 → 45P+

- 0-0
- 0-1
- 0-2
- 0-3
- 0-4
- 0-5
- 0-6
- 0-7
- 1-0
- 1-1
- 1-2
- 1-3
- 1-4
- 1-5
- 1-6
- 1-7
- 2-0
- 2-1
- 2-2
- 2-3
- 2-4
- 2-5
- 2-6
- 2-7
- 3-0
- 3-1
- 3-2
- 3-3
- 3-4
- 3-5
- 3-6
- 3-7

ALH | ALL
RFL | ALC
RFL | RFL
ALH | ALC
RFL | ALL
BDX | ALC
RFL | ALL, LSI: 0
H₁ 81 BPA
ALC | ALC
ALC 10X6 | ALC
DMX
ALH | RFL
ALC | RFL
ALC | ALL
Disab: All
Left Shift
ALH | ALL (Shift REAH L-FT)
Right Shift

0
ALH16
LINK
ALH00
REAH01
ALL16
ALH01
0
ALH16
LINK
ALH00
REAH01
ALL16
ALH01
0

000 ✓
300 ✓
014 ✓
010 ✓
204 ✓
110 ✓
004 ✓
* 010 ✓
360 ✓
004 ✓
006 ✓
130 ✓
200 ✓
000 ✓
100 ✓
000 ✓
000 ✓
000 ✓
000 ✓
* 300 ✓
060 ✓
060 ✓
060 ✓
060 ✓
060 ✓
060 ✓
060 ✓
060 ✓

000 ✓
030 ✓
006 ✓
006 ✓
024 ✓
012 ✓
005 ✓
* 012 ✓
030 ✓
006 ✓
005 ✓
017 ✓
024 ✓
006 ✓
012 ✓
000 ✓
130 ✓
130 ✓
130 ✓
130 ✓
130 ✓
130 ✓
* 130 ✓
230 ✓
230 ✓
230 ✓
230 ✓
230 ✓
230 ✓
230 ✓
230 ✓

Interchanged
10/17/75

Prim # →

P400 μ -code
 Addressable Latch
 Cleared by IAC FETCH
 Loaded all during cycle with SETLATCH or
 RESETLATCH.

9. P400
 8/6/75
 MHT
 REV1. 1/28/76

Bits	Use	Name
0	if set, inhibit address Traps	EADRTTR-
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 IND32, CE4	ADL7

Possible Shifts and ENDS

Legal Requests:	RCM 14-17	RCM 10	Action
1. SHIFTRIGHT	10	1	LRL
E=2	10	1	LRL
E=LINK	12	1	Multiple
E=ALH00	13	1	MPV
E=ALH01	16	1	arithmetic
ROTATEDRIGHT	15	1	
ROTATEDSRIGHT	11	1	ARR
HOLE IN MIDDLE	RCM10 = 0		ARM
(all shifts above are legal)			LRS
2. SHIFTLEFT	0	0	LLL
E=0	0	0	LLL
E=LINK	2	0	Multiple
E=REAH01	4	0	48 bit
HOLE IN MIDDLE	RCM10 = 1		
(all shifts above are legal)			
SHIFTLEFT	1	0	ALS, ALL
ROTATEDLEFT	6	0	LLR
ROTATEDSLEFT	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	DSWSTATH
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	
C	M	M	M	Machine			R	E	E	Bup	RP Backup		D	IO		
I	C	P	M	Check Code			C	C	C	Inv	Count		M	Bus		
							M	C	C				X			
RESERVED				ECCU					DMX		RP BACKUP		MISS CHG			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	DSWSTATL
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	
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

0=Peripheral Data (BPD) Output

1=Peripheral Address (SPA) Input

2=Memory Data (BMD) Output

3=Cache Data (RCO)

4=Peripheral Address (BPA) Output

5=ROX-BPD Input

6=Memory Address (EMA)

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

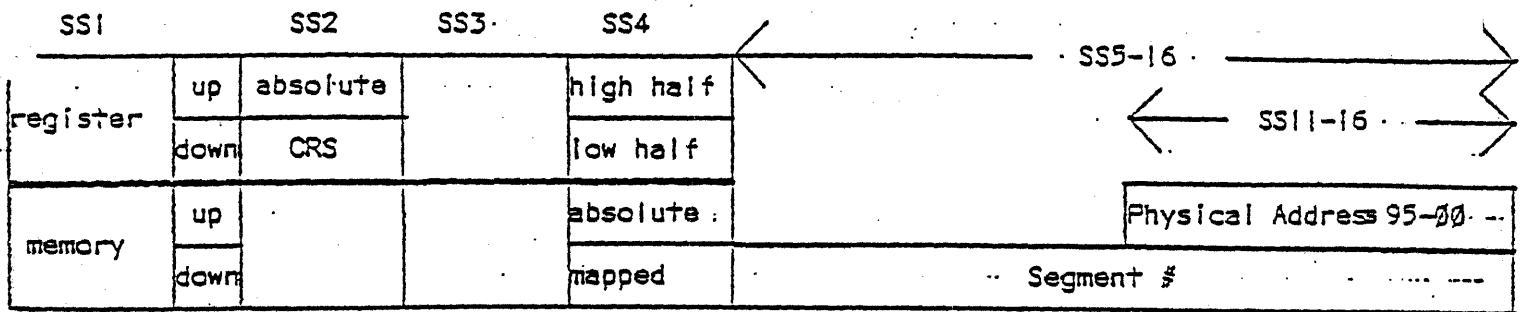
35 :41-42,56 If bit 42 set:51-55

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-300 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 0 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+I (STORE/FETCH) operates exactly as for memory with the address being pre-incremented.

Null Vector: In P-300 mode, if an external interrupt, fault, or check attempts to vector through a memory location containing a 0, the following action is taken:

HALT
data and address lights cleared
RP = address trapped
PBH = RPH
TR2L = address of vector

Figure 17.

DMA				Current Register Set (CRS)							
DMA				CRS							
addr	High	Low	Cell	High	Low	Addr	Cell	High	Low	Addr	Addr
0	TR0	-	0			40	0	GR0	-	100	140
1	TR1	-	1			41	1	GR1	-	101	141
2	TR2	-	2			42	2	GR2(1,A,LH)	-(2,B,LL)	102	142
3	TR3	-	3			43	3	GR3(EH)	-(EL)	103	143
4	TR4	-	4			44	4	GR4	-	104	144
5	TR5	-	5			45	5	GR5(3,S,Y)	-	105	145
6	TR6	-	6			46	6	GR6	-	106	146
7	TR7	-	7			47	7	GR7(0,X)	-	107	147
10	RDMX1	-	10			50	10	FR0(13)	-	110	150
11	RDMX2	-	11			51	11	-	-	111	151
12		RATMPL	12			52	12	FR1(4)	-(5)	112	152
13	RSGT1	-	13			53	13	-(6)	-	113	153
14	RSGT2	-	14			54	14	PB	-	114	154
15	RECC1	-	15			55	15	SB(14)	-(15)	115	155
16	RECC2	-	16			56	16	LB(16)	-(17)	116	156
17		REOIV	17			57	17	X8	-	117	157
20	ZERO	ONE	20	(20)	(21)	60	20	DTAR3(10)	-	120	160
21	PSSAVE	-	21			61	21	DTAR2	-	121	161
22		-	22	(22)	(23)	62	22	DTAR1	-	122	162
23		-	23			63	23	DTAR0	-	123	163
24		-	24	(24)	(25)	64	24	KEYS	(modals)	124	164
25		-	25			65	25	OWNER	-	125	165
26		-	26	(26)	(27)	66	26	FCCOE(11)	-	126	166
27		-	27			67	27	FADOR	-(12)	127	167
30	PSWPB	-	30	(30)	(31)	70	30	TIMER	-	130	170
31	PSWKEYS	-	31			71	31	-	-	131	171
32	BPA:PLA	PCBA	32	(32)	(33)	72	32		-	132	172
33	PS:PLB	PCBB	33			73	33		-	133	173
34	SWRMA	-	34	(34)	(35)	74	34		-	134	174
35	DSWSTAT	-	35			75	35		-	135	175
36	DSWPS	-	36	(36)	(37)	76	36		-	136	176
37		-	37			77	37		-	137	177

I numbers in parenthesis show P300 address Mapping

KEYSH

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

KEYSL (Modals)

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

FLEX=0 allows FLEX Faults

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

- 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

Register Files and Mapping

On Dispatcher
SD: Save Done

Figure 16.

RF select - P400 -1 Table 1

12. M#J
P400
6/6/75

Select RF
Function

Encode (in Rcm REI 18 → 24)

	18	19	20	21	22	23	24	Octal
1. μ -code addressed scratch locations (1's complement of RCH's 20 → 24) (TR0 = 137; TR1 = 136 etc.)	1	0	0	0	3	7	→	1.0.0 → 1.3.7
2. μ -code addressed Register Set locations (1's complement of RCH's 20 → 24) (A = 175)	1	1	0	0	3	7	→	1.4.0 → 1.7.7
2a. μ -code addressed switch from AXCI to RYCI								1.7.0
3. Base Registers (BR) of active Register Set (from BBH15, BBH16)	0	0	0	1	X	X	X	0.1.0
4. DIAR Registers of active Register Set (from BBH5, BBH6)	0	0	1	0	X	X	X	0.2.0
5. RPA addressed DMA channels	0	0	1	1	X	X	X	0.3.0
6. REAL address entire file (1's complement)	0	1	0	0	X	X	X	0.4.0
7. Address Trap Mapping	0	1	0	1	X	X	X	0.5.0
8. GR(CRS)	0	1	1	0	1	1	X	0.6.6
9. GR(CRSN)	0	1	1	0	1	0	X	0.6.4
10. GR(CFS)	0	1	1	0	0	1	X	0.6.2
11. GR(CFSN)	0	1	1	0	0	0	X	0.6.0
12. GR(CRD)	0	1	1	1	1	1	X	0.7.6
13. GR(CRDN)	0	1	1	1	1	0	X	0.7.4
14. GR(CFD)	0	1	1	1	0	1	X	0.7.2
15. GR(CFDN)	0	1	1	1	0	0	X	0.7.0

ALL use
current
Registers
set

ALU MODES MIJ 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.
TH	TRANSPORT THE H 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 BIT OPERATION.
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-							F u n c.	14.			1400 MHT 10/25/75 Rev 1.1/29/77		
CH4 Rcm8	H	L	C	CH	CL	CC		H	L	C	CH	CL	CC
0	A	A	TB	col	0		40	A/TA	S/TA		col	0	
1				col	cbiT		41				col	cbiT	
2	A	A	IA	0	0	1	42	IA	TA		0	0	1
3				cbiT			43				cbiT		
4	S	S		col	cbiT		44	S/TA	S/TA		col	cbiT	
5				col	1		45				col	1	
6	0	IA	TB	col	0		46	DA	DA	TA	col	0	
7				0	1		47				0	1	
10	-1	-1		0	0		50	IA	0	TA	0	0	
11				1	1		51				1	1	
12	S	TB	TB	col	1		52	IA	IA	S	col	1	
13				1	0	0	53				1	0	0
14	A OR B A OR E					0	54	TB	TB	A			0
15							55						
16	TB	A	A	0	0	0	56	A/TB	NotB	A/TA	0	0	0
17				0		1	57				0		1
20	A	TB	A	0	1	0	60	A	IA	TA	0	1	
21				1	0		61				1	0	
22	OR	OR	TB				62	0	AND	TB			
23							63						
24	A/S	A/S	TA	col	0/1		64	AND	AND	TA	col	0/1	
25				col	cbiT		65				col	cbiT	
26	XOR	XOR		col	cbiT		66	IA	IA	DA	col	cbiT	
27				1	0	0	67				1	0	0
30	NOTA	NOTA	IA			1	70	NOTB	NOTB	IA			1
31						0	71						0
32	TB	0	IA			1	72	0	TB	S			1
33						0	73						0
34	XOR	TA	S			1	74	A AND B	A AND B	OR			1
35						0	75						0
36	TB	TB	OR				76						
37							77						

Added. 10/20/77

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
MULL16	'00	MPYMPYXLCBIT	'41
ADD32	'00	TA32	'42
XADDX	'00	XTAX	'42
XADDTX	'00	TAXX	'42
MULL32	'00	TA16	'42
ADDALDXLCBIT	'01	THCXXHCBIT	'43
ADD32LCBIT	'01	MPYFSMPYFSX	'45
ADDADDX	'02	MPYFSMPYFSXLCBIT	'44
PLUS16	'02	DEC32	'46
ADD16	'02	DECXTA	'47
ADDDXX	'02	DECXTA	'47
ADDXXHCBIT	'03	DEC16	'47
ADD16HCBIT	'03	DAXX	'47
SUBSUBXLCBIT	'04	TAXEROX	'50
SUB32LCBIT	'04	TAXTA	'50
MINUS32	'05	INCXTA	'51
SUB32	'05	INC32	'52
ZERO-TATB	'06	INC16	'53
ZIFCXX	'06	INCXX	'53
XTA16	'06	TH32	'54
ZEROTAX	'06	XIBX	'54
MINUS1	'10	TOTEX	'54
MINUS1MINUS1X	'10	TH16	'54
ZERO	'11	TOTRADD	'54
ZEROGZEROX	'11	TAXX	'54
YIB16	'12	FETCHNOTFFETCH	'56
		FETCHXFFETCH	'56
		FETCHNOTFFETCHC1	'57
		FETCHXFFETCHC1	'57
		ADDXIA	'60

MINUS16	'13
SUB16	'13
AOEBOGTXX	'14
AOERROT	'14
THAGDADD	'16
THAFDADDC1	'17
ADDTBADD	'20
OR32	'22
ORORIP	'22
OR16	'22
ORXX	'22
LIVLIVIA	'24
DIVDIVX	'24
DIVDIVXLCBIT	'25
XOR32	'26
XOR16	'26
ANOT32	'30
NOTA32	'30
ANOT16	'30
NOTAXINC	'30
NOTA16	'30
NOTAXIA	'31
TSZERDINC	'32
TDXINC	'37
TS7LBOX	'32
TDXTA	'33
XORTASUBC1	'34
XORTASUBC0	'35
TDTEOR	'36

ADDINCTA	'6C
ADDIATAH1	'61
ADDXXH1	'61
XANDTB	'62
ZEROANDTB	'62
ANDXIA	'64
AND32	'64
ANDXX	'64
ANDXTA	'64
AND16	'64
INCINCLCBIT	'66
INCXDEC	'67
BNOT32	'70
BNOT16	'70
NOT6XX	'70
NOTBNOTBX	'70
NOTB16	'70
NOTB32	'70
XNOTBX	'70
NOTBXINC	'70
NOTBXIA	'71
ZCROTDX	'72
ZCROTBSUB	'72
ZCROTBSUBC0	'73
ANOTANDDANDOR	'74
ANDDANDDANDX	'74

CL INFORMATION REV 4 P400 4/6/76

18.

CLOCK INFORMATION REV 4 P400 4/6/76

SORTED CLOCK REV 4 P400 UCODE 4/6/76

160	16		
0	1	C	000000 J 160
0	1	C	000001 J 160
0	1	C	000020 J 160
0	1	C	000021 J 160
0	1	C	000040 J 160
0	1	C	000041 J 160
0	1	C	000060 J 160
0	1	C	000061 J 160
0	1	C	000100 J 160
0	1	C	000101 J 160
0	1	C	000120 J 160
0	1	C	000121 J 160
0	1	C	000140 J 160
0	1	C	000141 J 160
0	1	C	000160 J 160
0	1	C	000161 J 160

RDH		
RD		
RD	REAH	
RDL		
RDL	REAH	
REAH		
REA		
REA	RMA	RPL
REA	RMA	
REAL		
REAL	RMAL	
RPL		
RMA		
DXMRDY		
RED		

200	16		
0	1	C	000002 J 200
0	1	C	000003 J 200
0	1	C	000022 J 200
0	1	C	000023 J 200
0	1	C	000042 J 200
0	1	C	000043 J 200
0	1	C	000062 J 200
0	1	C	000063 J 200
0	1	C	000102 J 200
0	1	C	000103 J 200
0	1	C	000122 J 200
0	1	C	000123 J 200

RDH		
RD		
RDH	REAL	
RDL		
REAH		
REA	RMA	
REAL		
REAL	RMAL	
RFH		
RF		
RFL		

0	1	C	000142	J	200	RMA			
0	1	C	000143	J	200	MRDY	RFH		
0	1	C	000162	J	200	RCD	RF		
0	1	C	000163	J	200	MRDY	RCD	RMAL	

16

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

16

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

0	1	C	000146 J	280	RDH	RFL	
0	1	C	000147 J	280	RDH	RPL	
0	1	C	000166 J	280	RDL		
0	1	C	000167 J	280	RDL	REA	RMA

280A

16

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

280B

11

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

21,

320

15

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

360

11

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

DXMRDY

1

0	1	E	000160	J	160	DXMRDY			
---	---	---	--------	---	-----	--------	--	--	--

MRDY

7

22.

0	1	E	000143	200	MRDY	RFH	
0	1	E	000163	200	MRDY	RCD	RMAL
0	1	E	000033	280B	MRDY	RCD	
0	1	E	000052	280B	MRDY	RFH	
0	1	E	000053	280B	MRDY	RDH	
0	1	E	000073	280B	MRDY	RF	RMA
0	1	E	000112	280B	MRDY	RFH	RMAL

CD 14

0	1	E	000161	160	RCD		
0	1	E	000162	200	RCD	RF	
0	1	E	000163	200	MRDY	RCD	RMAL
0	1	E	000007	280	RCD		
0	1	E	000026	280	RCD	RDH	
0	1	E	000027	280	RCD	RDH	RDL
0	1	E	000046	280	RCD	REAL	
0	1	E	000047	280	RCD	RPL	
0	1	E	000013	280B	RCD	RF	
0	1	E	000033	280B	MRDY	RCD	
0	1	E	000072	280B	RCD	RDH	RPL
0	1	E	000130	320	RCD	RF	
0	1	E	000170	320	RCD	RDH	
0	1	E	000056	360	RCD	RDH	

12

0	1	E	000020	160	RD		
0	1	E	000021	160	RD	REAL	
0	1	E	000022	200	RD		
0	1	E	000024	240	RD		
0	1	E	000067	280	RD		
0	1	E	000106	280	RD	REAL	RMAL
0	1	E	000107	280	RD	RPL	
0	1	E	000011	320	RD		
0	1	E	000030	320	RD	REAL	RMAL
0	1	E	000131	320	RD	REAL	RPL
0	1	E	000150	320	RD	RF	

CLOCK INFORMATION REV 4 . P400 4/6/76

O	1	[	000136]	360	RD	REAL	RMAL
DH 18							
O	1	[	000001]	160	RDH		
O	1	[	000003]	200	RDH		
O	1	[	000023]	200	RDH	REAL	
O	1	[	000005]	240	RDH		
O	1	[	000025]	240	RDH	RF	
O	1	[	000026]	280	RCD	RDH	
O	1	[	000027]	280	RCD	RDH	RDL
O	1	[	000066]	280	RDH		
O	1	[	000126]	280	RDH	REAL	
O	1	[	000127]	280	RDH	RFH	
O	1	[	000146]	280	RDH	RFL	
O	1	[	000147]	280	RDH	RPL	
O	1	[	000053]	280B	MRDY	RDH	
O	1	[	000072]	280B	RCD	RDH	RPL
O	1	[	000132]	280B	RDH	REAL	
O	1	[	000031]	320	RDH	REAL	
O	1	[	000170]	320	RCD	RDH	
O	1	[	000056]	360	RCD	RDH	

DL	1	[	000040]	160	RDL		
O	1	[	000041]	160	RDL	REAL	
O	1	[	000042]	200	RDL		
O	1	[	000044]	240	RDL	RF	
O	1	[	000027]	280	RCD	RDH	RDL
O	1	[	000166]	280	RDL		
O	1	[	000167]	280	RDL	REA	RMA
O	1	[	000014]	280A	RDL	RMA	
O	1	[	000035]	280A	RDL	RFL	
O	1	[	000113]	280B	RDL	RF	
O	1	[	000050]	320	RDL	RFL	
O	1	[	000076]	360	RDL	REAL	
O	1	[	000077]	360	RDL	RMA	

O	1	C	000117 J	360	RDL	REA	RMA
REA 12							
0	1	C	000061 J	160	REA		
0	1	C	000100 J	160	REA	RMA	RPL
0	1	C	000101 J	160	REA	RMA	
0	1	C	000062 J	200	REA	RMA	
0	1	C	000167 J	280	RDL	REA	RMA
0	1	C	000015 J	280A	REA		
0	1	C	000034 J	280A	REA	RMA	RPL
0	1	C	000054 J	280A	REA	RPL	
0	1	C	000032 J	280B	REA	RMA	
0	1	C	000151 J	320	REA	RF	RMA
0	1	C	000057 J	360	REA	RFL	RMA
0	1	C	000117 J	360	RDL	REA	RMA

O	1	C	000021 J	160	RD	REAH
REAH 11						
0	1	C	000041 J	160	RDL	REAH
0	1	C	000060 J	160	REAH	
0	1	C	000043 J	200	REAH	
0	1	C	000045 J	240	REAH	RFH
0	1	C	000064 J	240	REAH	RF
0	1	C	000065 J	240	REAH	RFL
0	1	C	000126 J	280	RDH	REAH
0	1	C	000055 J	280A	REAH	RFH
0	1	C	000074 J	280A	REAH	RF
0	1	C	000051 J	320	REAH	RF

O	1	C	000120 J	160	REAL	
REAL 18						
0	1	C	000121 J	160	REAL	RMAL
0	1	C	000023 J	200	RDH	REAL
0	1	C	000063 J	200	REAL	
0	1	C	000102 J	200	REAL	RMAL

25,

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
J	1	[	000075]	280A	REAL	RMAL		RPL
J	1	[	000132]	280B	RDH	REAL		
0	1	[	000030]	320	RD	REAL		RMAL RPL
0	1	[	000031]	320	RDH	REAL		
J	1	[	000070]	320	REAL	RMAL		
J	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		
J	1	[	000025]	240	RDH	RF		
J	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			
J	1	[	000134]	280A	RF	RPL		
0	1	[	000175]	280A	RF	RMA		
0	1	[	000013]	280B	RCD	RF		
0	1	[	000073]	280B	MRDY	RF		RMA
J	1	[	000113]	280B	RDL	RF		
J	1	[	000051]	320	REAH	RF		
J	1	[	000110]	320	RF			
0	1	[	000130]	320	RCD	RF		
J	1	[	000150]	320	RD	RF		
J	1	[	000151]	320	REA	RF		RMA
J	1	[	000036]	360	RF			

O	1	C	000103	J	200	RFH		
O	1	C	000143	J	200	MRDY	RFH	
O	1	C	000045	J	240	REAH	RFH	
O	1	C	000105	J	240	REAL	RFH	RMAL
O	1	C	000124	J	240	RFH	RMA	
O	1	C	000145	J	240	RFH		
O	1	C	000127	J	280	RDH	RFH	
O	1	C	000055	J	280A	REAH	RFH	
O	1	C	000114	J	280A	RFH		
O	1	C	000135	J	280A	RFH	RPL	
O	1	C	000052	J	280B	MRDY	RFH	
O	1	C	000112	J	280B	MRDY	RFH	RMAL
O	1	C	000071	J	320	RFH		
O	1	C	000017	J	360	RFH		

L 1C

O	1	C	000123	J	200	RFL		
O	1	C	000065	J	240	REAH	RFL	
O	1	C	000165	J	240	RFL		
O	1	C	000146	J	280	RDH	RFL	
O	1	C	000035	J	280A	RDL	RFL	
O	1	C	000154	J	280A	RFL		
O	1	C	000050	J	320	RDL	RFL	
O	1	C	000111	J	320	RFL		
O	1	C	000037	J	360	RFL		
O	1	C	000057	J	360	REA	RFL	RMA

A 1B

O	1	C	000100	J	160	REA	RMA	RPL
O	1	C	000101	J	160	REA	RMA	
O	1	C	000141	J	160	RMA		
O	1	C	000062	J	200	REA	RMA	
O	1	C	000142	J	200	RMA		
O	1	C	000124	J	240	RFH	RMA	
O	1	C	000144	J	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
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	C	000155	J	280A	RPL			
0	1	C	000012	J	280B	RPL			
0	1	C	000072	J	280B	RCD	RDN	RPL	
0	1	C	000030	J	320	RD	REAL	RNAL	RPL
0	1	C	000131	J	320	RD	REAL	RPL	
0	1	C	000116	J	360	RPL			

28.1a

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	2800	RCD	RDH	RPL	
1	0	280B	MRDY	RF	RMA	
1	0	2800	MRDY	RFH	RMAL	
1	0	2800	RDL	RF		
1	0	2800	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	RHAL		
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	RFA		
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 INFORMATION REV 4 P400 4/6/76

30.a.

1	0	360	RDL	RHA	
1	0	360	RPL		
1	0	360	RDL	REA	RHA
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,REAL,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,REAL,160

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,2800
000113:	036100	RF,RDL,2800
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,REAL,280
000127:	034200	RDH,RFH,280
000130:	036010	RF,RCD,320
000131:	060300	RD,REAL,RPL,320
000132:	070200	RDH,REAL,2800
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

35.

UA UIS IS IA OTHER IAC'S
 * P400 IAC'S REV 03 PS-MHJ 4-6-76
 *

This listing shows all of the IAC's and the combinations that may be used together. Some combinations 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		
DBB	1	0	14		
DECREA				04	SHIFT SACC1
DECREA				06	SHIFT
DECREA				17	
EAF		7	02		
END		4	04		
ESCPN	0	0	12		

UA UIS IS IA

ER JAC'S

36.

LSSTRB	0	0	13
FRADP	1	0	04
FETCH		7	10
GATE	1	0	16
ICPN	0	0	11
IEX			
IEX			
IEX			
IEX			
IEX			
INCREA			
INCREA			
INCREA			
INCREA			
INCREA			
IND		7	01
IND16		7	04
INTRP	1	0	03
INH1	1	0	10
INK	0	0	02
INVC1	1	0	00
LISTLO	1	0	12
LCAL		2	17

03	SHIFT	INCREA
03	SETCC	
25	SETCC32	
31	SACC1	SETCC32
36	SACC1	RTH SETCC
02	SHIFT	SACC1
03	SHIFT	IEX
05	SHIFT	
15		
26	DAL	

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= ALH0V	06
C= SOVFL	07

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

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

UA UIS IS IA OT IAC'S

37

L= BDH03
L= ALL16

MSB IS = 1 (IS = 10)
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				
RP10		6	02				
RSTRD		4	01				
RSYSC	0	0	16				
RTN				34			
RTN				35	SACC1		
RTN				36	SACC1	IEX	SETCC
RTN				37	SACC1	LMOD	RXM
RXM				33			
RXM				37	SACC1	LMOD	RTN
SACC1				01	SHIFT		
SACC1				02	SHIFT	INCREA	
SACC1				04	SHIFT	DECREA	
SACC1				10	LMOD		

(200 n.s. Min. Step)
(200 n.s. Min. Step)
(200 n.s. Min. Step)

UA

UIS

IS

IA

R. IAC'S

38.

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	SHIFT		
SETCC				11	LMOD		
SETCC				16	IEX		
SETCC				20			
SETCC				36	IEX	RTN	SACC1
SETCC32				21			
SETCC32				25	IEX		
SETCC32				31	IEX	SACC1	
SHIFT				00			
SHIFT				01	SACC1		
SHIFT				02	SACC1	INCREA	
SHIFT				03	INCREA	IEX	
SHIFT				04	SACC1	DECREA	
SHIFT				05	INCREA		
SHIFT				06	DECREA		
SHIFT				07	SETCC		
SPARE				25			
SPARE				31			
SP10		6	04				
SSTR0				12	RACPN		
SSTR0				32			
UACC1	2						
UACC2	3						
UBDX		1	14				
UBDX		5	14				

(200 ns. Min Step)

Any UBDX can use the two MSR'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

39.

UHSN BDX= MISC 14
 1 03
 Any uhsn 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

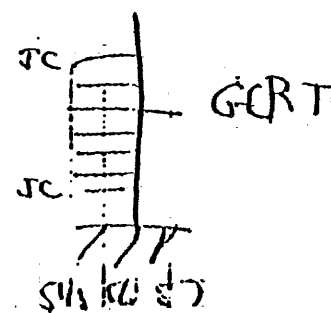
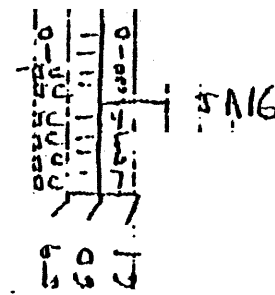
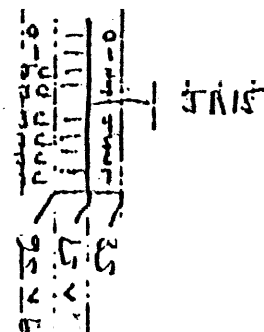
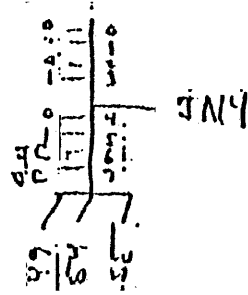
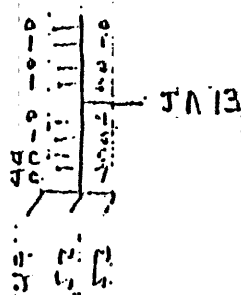
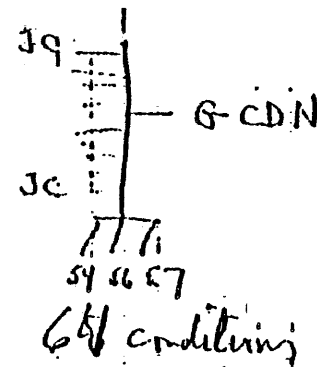
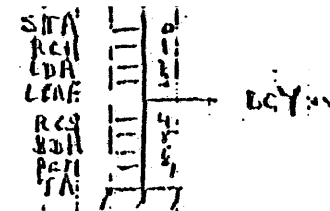
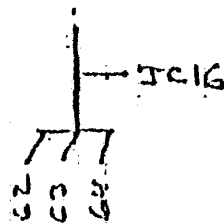
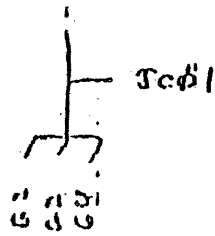
DU

40.

emit
 cond decide
 cond return
 cond. branch
 decode/IAF
 RTN
 BDH branch
 uncond branch
 emit and IACRTN

47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
1	1					EMIT	CONSTANT										
0	1	0	BCY11-12	0	BCY13	BCDN cond	BCY14	BCDN cond	BCY15	JN16 cond		JN16 cond		JN16 cond	JN16 and GCDN cond		
0	1	0	BCY11-12	1	BCY13	RTN cond	BCY14	RTN cond	BCY15	JN16 cond		JN16 cond		JN16 cond	JN16 and RTN cond		
0	1	1	BCY11-12	0	BCY13	0	BCY14	JN15 cond		JN16 cond		JN16 cond		JN16 cond	all JN's cond		
1	0	0	1	0				misc control on	DECODE/IAF								
1	0		0	0													
1	0		0	1													
0	0							Branch address (BCY03-16)									
0	0							EMIT constant									

next address																
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
RCH								RCH								
RCH								RCH								
RCH								RCH								
RCH								RCH								
RCH								RCH								
RCH								RCH								
RCH								RCH								
RCH								RCH								
RCH								RCH								



MHJ
10/29/75
Rev 2 3/15/76

Jump Codes

JUMP codes are arranged by the bit they affect and the RCM 62-64 which selects them. 4, 8, 12, and 16 way branches are only possible within the same RCM 62-64 Code group. For each code group there are also conditional returns available. This table allows conditional branches to be chosen.

Code 0 (RCM 62-64 = 0)

<u>JC13</u>	<u>JC14</u>	<u>JC15</u>	<u>JC16</u>	<u>CRTN</u>
REAL07	REAL14	REAL15	REAL16	1 ADL06
	ADL05	ADL06	ADL07	5 ADL05
IOBUS		DL01	MODNUM	7 NDLO1
		FSPLUS	FSHINS	3 ADL07
		NADL06	NADL07	4 NAL01
			CRS	2 NADL06

Code 1 <u>JC13</u>	<u>JC14</u>	<u>JC15</u>	<u>JC16</u>	<u>CRTN</u>
FPOSTX	BBH01	BBH04	BBH05	1 ALH015
		ALH09	ALH01	2 NALH01
NEAC13	ALCCOAT	ALH015	AL32NE	3 AL32NE
		REAH01	REAH05	4 AL32EG
		NALH01	AL32EQ	5 ALCCOAT
		BDH10	RPH03	7 NBBH04

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

Code 2
JC13

PXM
NGAOK

JC14

REAL11
FLEX

JC15

REAL12
NGFCTR
ALHNE
ALHEQ
AL32NE15
BBH09

JC16

ALH16 7 GFCTR
GADPTR 5 FLEX
ALHLT 1 ALHNE
ALHGE 2 ALHEQ
CCGE 3 ALHLT
NRFH01 4 ALHGE
6 RFH02

CR11

Code 3
JC13

CPCEXT
NGADRI

JC14

BPCMOD2
JCAPI

JC15

GAVIOL
DAEQ20
ALHGT
ALHLE
ALLOI
NBTTST

JC16

VERY 7 NGAVID
NREADY 1 ALHGT
AL32LE 2 ALHLE
AL32GT 3 AL32LE
ALLNE 4 AL32GT
JCAP2 5 JCAPI

CR11

Code 4
JC13

NPIDOK
EAIHAN12

JC14

QUOT
SOVFL

JC15

NSEGOR
EAIZORIS
CCGE15
COLT
REAL01

JC16

NSEGTR 7 SEGOR
GSKP 5 SOVFL
CCNE 1 CCGE1
CCCEQ 2 COLT
REAL02 3 CCNE
4 CCCEQ

CR11

43.

Code 5

<u>IC13</u>	<u>IC14</u>	<u>IC15</u>	<u>IC16</u>	<u>CRTN</u>
TMR		EXINT	NRUN	7 NEXINT
BDH13	BDH14	BDH15	BDH16	5 BDH01
	BDH01	CCL	BDH02	1 CCL
		CCGT	NBDH02	2 CCGT
		ALHCOU7	NRFLO	4 NBDH02
				3 BDH02

Code 6

<u>IC13</u>	<u>IC14</u>	<u>IC15</u>	<u>IC16</u>	<u>CRTN</u>
NFMC 2	AMO 2	AMI	PGBIT	3 ALHOU
BBH07	LINK 3	NXCS	AM2	4 NALHOU
		ALCM1	ALHOU	1 NXCS
		XCS	CBIT	2 XCS
		BBH08	NALHOU	5 LINK
		RMAVLD	ALHCOU16	7 NRMALD

Code 7

<u>IC13</u>	<u>IC14</u>	<u>IC15</u>	<u>IC16</u>	<u>CRTN</u>
FVIM		INLV	PFI'Z	7 NINLV
NEPCXB	MOD0	MOD1	INMOD	0 NSREQ
	NSREQ	GRSCM1	ALL16	1 GRSCM1
		NGRSCM1	NALL16	2 NGRSCM1
		LBRPI	RBRP2	3 ALL16
			NREACM1	4 NALL16

JUMP CONDITIONS SORTED ALPHABETICALLY

4/6/76

44.

JUMP CONDITIONS SORTED ALPHABETICALLY

4/6/76

JMP COND	FUNCTION	FOUND	CODE	CRTN
LO5	ADRESSABLE LATCH	JA14-1	CODE 0	CRTN-5
ALLO6	ADDRESSABLE LATCH	JA15-4	CODE 0	CRTN-1
ALLO7	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
ALLO1		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	
PH01		JA14-1	CODE 5	CRTN-5
PH02		JA16-4	CODE 5	CRTN-3
PH13		JA13-0	CODE 5	
PH14		JA14-0	CODE 5	
PH15		JA15-3	CODE 5	
PH16		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
C RS	CURRENT REGISTER SET	JA16-7	CODE 0	
LA EQ20	REAL11-14 = *20 (PIO SPECIAL)	JA15-3	CODE 3	
DI01	DIAGNOSTIC LATCH	JA15-2	CODE 0	
LA11AN12	REAL 11 AND 12 = 1	JA13-0	CODE 4	
LA12OR15	REAL12 OR 15 = 1	JA15-3	CODE 4	
C INT	POWER FAILURE PENDING	JA15-2	CODE 5	
EX	FLOATING EXCEPTION MODE	JA14-1	CODE 2	CRTN-5
POSTX	POST INDEXING NEEDED	JA13-1	CODE 1	
MINUS	ADDRESSABLE LATCH 2	JA16-6	CODE 0	
PLUS	ADDRESSABLE LATCH 1	JA15-6	CODE 0	
IM	VECTORED INTERRUPT MODE	JA13-1	CODE 7	
DRTR	ADDRESS TRAP	JA16-2	CODE 2	
AVIOL	ACCESS VIOLATION	JA15-2	CODE 3	
ICTR			CODE 2	CRTN-7
ICEXT	EXTERNAL PIO REQ	JA13-0	CODE 3	
SCM1	REAL12-16 EQ MINUS ONE	JA15-4	CODE 7	CRTN-1

46.

JMP	FUNCTION	FOUND	CODE	CRTN
SKP	SKIP NET SAYS SKIP	JA16-3	CODE 4	
INLV	MEMORY ACTIVE CAN INTERLEAVE	JA15-2	CODE 7	
IPMOD	DMX INPUT MODE LINE	JA16-3	CODE 7	
I/O BUS	CHECK DURING DMX,PIO,INT	JA13-1	CODE 0	
JCAP1	ADDRESS TRAP DECODE	JA14-1	CODE 3	CRTN-5
JCAP2	ADDRESS TRAP DECODE	JA16-7	CODE 3	
JFP COND	FUNCTION	FOUND	CODE	CRTN
LINK	LINK FLIP-FLOP	JA14-1	CODE 6	CRTN-5
MOD0	I/O BUS MODE LINE 0	JA14-0	CODE 7	
MOD1	I/O BUS MODE LINE 1	JA15-3	CODE 7	
MODNUM	MEMORY ODD/EVEN MODULE	JA16-2	CODE 0	
NADL06	ADDRESSABLE LATCH	JA15-5	CODE 0	CRTN-2
NADL07	ADDRESSABLE LATCH	JA16-5	CODE 0	CRTN-4
NALH01		JA15-5	CODE 1	CRTN-2
NALHOV	NO ARITHMETIC OVERFLOW	JA16-5	CODE 6	CRTN-4
NALL16		JA16-5	CODE 7	CRTN-4
NBBDH04			CODE 1	CRTN-7
NBDH02		JA16-5	CODE 5	CRTN-4
NITTSY	MUX OUTPUT-SEL BDH FROM REAL	JA15-7	CODE 3	
NIPCXB	NO EXTENDER FOR DMX	JA13-0	CODE 7	
NCAOK	CACHE INDEX NOT OK	JA13-1	CODE 2	
NEL01			CODE 0	CRTN-7
LAC13	REAL13-16 NOT = -1	JA13-0	CODE 1	
PEXINT			CODE 5	CRTN-7
EMC	NOT MACHINE CHECK	JA13-0	CODE 6	
HEADRI		JA13-1	CODE 3	
HCAVIOL			CODE 3	CRTN-7
HFCYTR	NO FETCH CYCLE TRAP	JA15-2	CODE 2	
H RSCM1	REAL12-16 NE MINUS ONE	JA15-5	CODE 7	CRTN-2
INLV			CODE 7	CRTN-7
HIDOK	PROCESS ID NOT OK	JA13-1	CODE 4	
HREACO1	CARRY OUT OF TOP OF REAL	JA16-7	CODE 7	
NREADY	NOT READY (PIO)	JA16-3	CODE 3	
HBFH01		JA16-7	CODE 2	
HFFLO1		JA16-6	CODE 5	
HMAVLD	RMA VALID ON A CHECK		CODE 6	CRTN-7
HFUN	CONTROL PANEL SAYS STOP	JA16-2	CODE 5	
LEGOK	NO MATCH IN STLU	JA15-2	CODE 4	

47

JMP CON	FUNCTION	FOUND	CODE	CRTN
MS EGR	NO STLB TRAP	JA16-2	CODE 4	
MS REQ	NO DMX REQUEST PENDING	JA14-1	CODE 7	CRTN-0
NACS	NO EXTENDABLE CONTROL STORE	JA15-4	CODE 6	CRTN-1
PCBIT	PRE-C BIT (CUT SOURCE)	JA16-2	CODE 6	
PTI	POWER FAILURE INTERRUPT	JA16-2	CODE 7	
PXM	PROCESS EXCHANGE MODE	JA13-0	CODE 2	
QOT	QUOTIENT BIT	JA14-0	CODE 4	
RERP1	RP BACK-UP COUNTER 1	JA15-6	CODE 7	
RERP2	RP BACK-UP COUNTER 2	JA16-6	CODE 7	
REAL01		JA15-6	CODE 1	
REAL05		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	
RU02				CRTN-6
RAVLD	RMA VALID ON A CHECK	JA15-2	CODE 6	
RU03	RING ZERO IF RESET	JA16-7	CODE 1	
RGOK			CODE 4	CRTN-7
ROVFL	SHIFT OVERFLOW (ALH01 NOT= ALH)	JA14-1	CODE 4	CRTN-5
ITR	CPU INTERNAL TIMER OVERFLOW	JA13-1	CODE 5	
UVR	U-VERIFY ROUTINES IF TRUE	JA16-2	CODE 3	
UCS	EXTENDABLE CONTROL STORE	JA15-5	CODE 6	CRTN-2

JUMP CONDITIONS -- SORTED BY CODE GROUP

4/6/76

48.

JUMP CONDITIONS -- SORTED BY CODE GROUP

4/6/76

JMP COND	FUNCTION	FOUND	CODE	CRTN
NDLO1			CODE 0	CRTN-7
REAL07		JA13-0	CODE 0	
IOBUS	CHECK DURING DMX,PIO,INT	JA13-1	CODE 0	
REAL14		JA14-0	CODE 0	
ADLOS	ADRESSABLE LATCH	JA14-1	CODE 0	CRTN-5
DLO1	DIAGNOSTIC LATCH	JA15-2	CODE 0	
REAL15		JA15-3	CODE 0	
ADLO6	ADDRESSABLE LATCH	JA15-4	CODE 0	CRTN-1
NADLO6	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	
ADLO7	ADDRESSABLE LATCH	JA16-4	CODE 0	CRTN-3
NADLO7	ADDRESSABLE LATCH	JA16-5	CODE 0	CRTN-4
FSMINUS	ADDRESSABLE LATCH 2	JA16-6	CODE 0	
CRS	CURRENT REGISTER SET	JA16-7	CODE 0	
NEBH04			CODE 1	CRTN-7
REAC13	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	

49.

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	
EBH09		JA15-7	CODE 2	
GADRT	ADDRESS 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	ADDRESS TRAP DECODE	JA14-1	CODE 3	CRTN-5
GAVIOL	ACCESS 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
ALLO1		JA15-6	CODE 3	
NBITTST	MUX OUTPUT-SEL EDH 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	ADDRESS 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

LOVE

CRTN

50.

QUOT	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	
HRUN	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	
NRMVLD	RMA VALID ON A CHECK		CODE 6	CRTN-7
NFMC	NOT MACHINE CHECK	JA13-0	CODE 6	
BDH07		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	
BDH08		JA15-7	CODE 6	

MP COND

FUNCTION

FOUND

COL

CRTN

51.

MP COND	FUNCTION	FOUND	COL	CRTN
PCBIT	PRE-C BIT (CDIT 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
CHIT		JA16-6	CODE 6	
ALHCOU16	CARRY OUT OF ALH	JA16-7	CODE 6	
NINLV			CODE 7	CRTN-7
NDPCXB	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
RDRP1	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
VALL16		JA16-5	CODE 7	CRTN-4
RDRP2	RP BACK-UP COUNTER 2	JA16-6	CODE 7	
VREACO1	CARRY OUT OF TOP OF REAL	JA16-7	CODE 7	

P400 U-CODE TIMING

52.

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 acceleration.
 - +0 ns at output of ALU's with any acceleration
 - d. Bus D to Destination
 - +100 ns to Register File.
 - +34 ns to any other destination.
 - 160 ns from beginning 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 D 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 ALINE, 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. Acceleration.

a. Cache data because of pre-loaded RMA.

Ignore cache in calculation.

b. ALU to RF -- previous step is like this one.

160 ns (accelerated -- 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 recommended. The assembler selected times are almost always far to fast.

K400

1/29/76

54.

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

A MODIFIED DNF 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 FOR 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>
 <RB 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

<BDEARLY SPEC>:=EA>MA ! P>MA

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

<RF MNEMONIC>:=,!

X!A!D!S!Y!VSC!E!EH!EL!
 GR(KD)!GR(RS)!GR(FD)!GR(FS)!
 GR(RDN)!GR(RSN)!GR(FDN)!GR(FSN)!
 GR(DR)!GR(DTAR)!
 GRH(KD)!GRH(RS)!GRH(FD)!GRH(FS)!
 GRH(RDN)!GRH(RSN)!GRH(FDN)!GRH(FSN)!
 GRH(DR)!GRH(DTAR)!
 GRL(KD)!GRL(RS)!GRL(FD)!GRL(FS)!
 GRL(RDN)!GRL(RSN)!GRL(FDN)!GRL(FSN)!
 GRL(DR)!GRL(DTAR)!
 LD!SD!XD!PD!
 LDH!SDH!XBH!PBH!
 LDL!SDL!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!
 PSWPD!PSWKEYS!PPA!PPB!
 PSWPDH!PSWKEYSH!PPAH!PPBH!
 PSWPBL!PSWKEYSL!PPAL!PPBL!
 DSWRMA!DSWSTAT!DSWSTATH!DSWSTATL!
 RSGT1!RSGT2!RSGT1H!RSGT2H!RSGT1L!RSGT2L!
 RECC1!RECC2

58,

<LENGTH>:= 161 32111H 1 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!CBIT!0)

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

<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

<BD 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!H8XBPA!DISABLE!DDXC!RFH!RFL!
 H8XB!RFHLL!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>:=ROTATES\$<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!ADRTY!IACDAL!DDSH!CHI!CRDXL!DDB!DECREA!EAF!ENB!
ESCPN!ESSTRB!LDRP!LPID!RMC!RMOD!RSYSC!
FBADP!FETCH!GATE!IEX!INCREA!IND!IND16!INH1!INK!ICPN!
INVC1!LISTLB!LCAL!LDIAG!LDRPL!LSTLB!LDTARL!
LLATCH!LMOD!NTRAP!ORDXL!PFL!POP!RADE!RACPN!RCCPN!
RDATE!RPIO!RSTRB!RTN!RXM!SACC1!SADE!SCPNI!SDATE!
SETCC!SETCC32!SHIFT!SPIO!SSTRB!UACC1!UACC2!UBDX!
UNSM!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 <EXPRESSTON> ! NULL

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

<CS SPEC>:= CS= (0,<CS0 SPEC>)ICS= (1,<CS1 SPEC>)ICS =(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 $2^{**}(\text{NUMBER OF CONDITIONS})$ (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

ADDITIONAL NOISE WORDS

(ALH=)=99, (ALL=)=100, (ALC=)=101,
 (E)=102, (F=)=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

SPECIAL SYMBOLS (RESERVED WORDS)

SEE <RCM SPEC>
 R= NONE!TDMX!NX!ALL (DEFAULT IS ALL)
 = CBIT!COUT!PLINK!ALHOV!ALH16!BDH01!SOVFL!ALLFCOUT
 IOLE 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
 DM= <BDX SPEC>

65.

```
<BDX SPEC>:=BMDIRPD!BDHIMISC
UA1      USE ACCELERATE 1
UA2      USE ACCELETATE 2
T=       TIME=
JUMP
GOTO
DAL
CRTN
CDECODE
TO
=>
```