B1700 HARDWARE SPECIFICATIONS

DISTRIBUTION

DETROIT

D. Kosinski - CPM
R. Stackhouse - CPM
D. Bookwalter - CPM
C. Chamberlain - Tech Pub
J. Raker - Int'l
T. Allison - Int'l (2)
R. Gould - Int'l
H. Hayde - Int'l
J. Holton - Int'l
C. Nash - Int'l
E. Seif - Int'l
H. Nelson - BM&SS
J. Shifman - CSG
E. Rindels - BMG
W. Cunningham - BMG
E. Paulsen - TIO

PLYMOUTH

D. Calkins

DOWNINGTOWN

J. Berta

PAOLI

G. Smolnik

McLEAN

L. Guell
A. Kosla

PASADENA

D. Gale
J. Malson

MISSION VIEJO

T. Gibson

WESTLAKE

H. Smith
J. Dowling

GENNEVIELIERS

A. Isola

CUMBERNAULD

D. Wright

SENEFFE

J. Jeanty

GLENROTHES

B. Slattery

LIEGE

J. Wery
A. Beaujean

MIDDLESEX

W. Marshall

SANTA BARBARA

R. Bunker
Manufacturing (2)
Quality Assurance
TIO (2)
Software (30)
W. Keenan (5)
Hardware - J. Vigil
J. Trost
S. Bhagwan (2)

F. E. TRAINING DEV. GROUP

M. McCoy
3374 Rio Hondo Avenue
El Monte, California 91731

PATENT ATTORNEY

Mr. D. Bruce Prout
Christie, Parker & Hale
201 South Lake Avenue
Pasadena, California 91101

TIREMAN

F. Williams

P,S, #1913 1747

Burroughs Corporation

BUSINESS MACHINES GROUP
SMALL SYSTEMS PLANT

M-MEMORY PROCESSOR

PRODUCT SPECIFICATION

REVISIONS

REV LTR	REVISION ISSUE DATE	PAGES REVISED ADDED DELETED OR CHANGE OF CLASSIFICATION	PREPARED BY	APPROVED BY
M	1-8-76	Preface: Added Processor-2 Rel. Spec: Added Control Panel References Sec 1: Add * note stating CD bit indicates both S and M-Memory Parity errors. Table 1: Added PERR Table 3: Added variants to Read/Write MSM. Sec 2: Added M-Processor-2, M-Processor Adapters-1 & -2, M-Memory Adapter-2. Sec 3.2: Added PERR. Sec 3.3.10: Added address modulo 16 of micro-op must be zero for Processor-2. Sec 3.3.12: Added M-Memory-2 with parity. Sec 3.3.15: Added "data not accepted in time will be lost and U-Register may not be addressed after Cassette STOP. Sec 3.3.16: Added M-Memory-2 parity error and listed conditions which set Memory Parity Error bit. Sec 3.4.1 & 3.4.2: Added U excluded as source in STEP mode. Sec 3.4.6: Added note 2 on M as a destination register. Sec 3.4.17: Added G/B H/F and S/N variants. Sec 3.4.23 & 24 & 25: Added reference to Sec 3.3.16. Sec 3.4.29: Generalized purpose of dispatch operation. Changed processor must gain control of interrupt system to should gain. . . Described lockout operations when using the direct connect processor adapter. Sec 3.5: Specified concurrent operations for Processor-1 only. Sec 3.6: Corrected times for operations with A out-of-bounds. Deleted time for Move 24-bit to MSM. Sec 3.7: Added times for Processor-2.	WFK	C.E. Wymore 19 Sept 75 [illegible] 10-28-75 [illegible] Al Melkonian 1/5/76 [illegible] [illegible] 1-8-76

Burroughs Corporation

BUSINESS MACHINES GROUP
SMALL SYSTEMS PLANT

M-MEMORY PROCESSOR

PRODUCT SPECIFICATION

REVISIONS

REV LTR	REVISION ISSUE DATE	PAGES REVISED ADDED DELETED OR CHANGE OF CLASSIFICATION	PREPARED BY	APPROVED BY
M	1-8-76 Cont.	Reorganized, copy-edited for entry into Product Specification data base. New section sequence is as follows-revisions noted above are indicated by asterisks (*).	AAM	

WAS	IS
Preface	1.0*
New	1.1*
2.	1.2*
Un-numbered	1.3*
1.	2.0*
3.0	3.0
New	3.1
3.3.11	3.1.1
3.3.12	3.1.2*
3.3.10	3.1.3*
3.3.9	3.1.4
3.3.8	3.1.5
3.3.1	3.1.6
3.3.2	3.1.7
3.3.3	3.1.8
3.3.4	3.1.9
3.3.5	3.1.10
3.3.6	3.1.11
3.1.9	3.1.12
3.3.7	3.1.13
3.3.16	3.1.14*
New	3.1.14.1
New	3.1.14.2
3.3.13, 3.3.14	3.1.15
3.3.15	3.1.16*
3.3.17	3.1.17
3.3.18	3.1.18
3.3.19	3.1.19
3.3.20	3.1.20
3.3.21	3.1.21
3.1	3.2
3.1.1	3.2.1
3.1.2	3.2.2
3.1.3	3.2.3

Burroughs Corporation

BUSINESS MACHINES GROUP
SMALL SYSTEMS PLANT

M-MEMORY PROCESSOR

PRODUCT SPECIFICATION

REVISIONS

REV LTR	REVISION ISSUE DATE	PAGES REVISED ADDED DELETED OR CHANGE OF CLASSIFICATION	PREPARED BY	APPROVED BY
M	1-8-76 Cont.	WAS / IS 3.1.4 / 3.2.4 3.1.5 / 3.2.5 3.1.6 / 3.2.6 3.1.7 / 3.2.7 3.1.8 / 3.2.8 3.1.9 / 3.2.9 3.1.10 / 3.2.10 3.2 / 3.3* Numbering of Sections 3.4, 3.5, and 3.6 is unchanged. Section 3.7 is new.		

M - M E M O R Y P R O C E S S O R

P R O D U C T S P E C I F I C A T I O N

1 9 1 3 1 7 4 7

TABLE OF CONTENTS:

1.0	INTRODUCTION	1
1.1	PURPOSE	1
1.2	PRODUCT IDENTIFICATION	2
1.3	RELATED SPECIFICATIONS	2
2.0	GENERAL DESCRIPTION	3
3.0	PRODUCT DESCRIPTION	8
3.1	PROCESSOR REGISTERS	8
3.1.1	M	8
3.1.2	MSM	8
3.1.3	A, MBR AND TOPM REGISTERS	9
3.1.4	A STACK	10
3.1.5	TAS	10
3.1.6	X AND Y	11
3.1.7	L	11
3.1.8	T	12
3.1.9	FA	12
3.1.10	FB	13
3.1.11	SCRATCHPAD	14
3.1.12	FLCN	14
3.1.13	BR AND LR	14
3.1.14	C	15
3.1.14.1	CP (CYF, CPU, CPL)	15
3.1.14.2	CA, CB, CC, CD	15
3.1.15	MAXS, MAXM	18
3.1.16	U	18
3.1.17	DATA	18
3.1.18	CMND	19
3.1.19	NULL	19
3.1.20	READ	19
3.1.21	WRIT	19
3.2	24-BIT FUNCTION BOX	20
3.2.1	SUM	21
3.2.2	DIFF	21
3.2.3	XANY, XORY, XEOY	22
3.2.4	CMPX, CMPY	22
3.2.5	MSKX,MSKY	22
3.2.6	BICN	22
3.2.7	XYCN	23
3.2.8	XYST	24
3.2.9	INCN	25
3.2.10	PERR	25
3.3	4-BIT FUNCTION BOX	26
3.4	M-INSTRUCTIONS	27
3.4.1	REGISTER MOVE	27
3.4.2	SCRATCHPAD MOVE	28
3.4.3	SWAP F WITH DOUBLEPAD WORD	29
3.4.4	STORE F INTO DOUBLEPAD WORD	29
3.4.5	LOAD F FROM DOUBLEPAD WORD	29
3.4.6	MOVE 8-BIT LITERAL	30
3.4.7	MOVE 24-BIT LITERAL	30

3.4.8	SWAP MEMORY	31
3.4.9	READ/WRITE MEMORY	31
3.4.10	COUNT FA/FL	32
3.4.11	SCRATCHPAD RELATE FA	33
3.4.12	EXTRACT FROM REGISTER T	33
3.4.13	SHIFT/ROTATE REGISTER T LEFT	34
3.4.14	SHIFT/ROTATE REGISTER X/Y LEFT/RIGHT	35
3.4.15	SHIFT/ROTATE REGISTERS XY LEFT/RIGHT	35
3.4.16	NORMALIZE X	35
3.4.17	READ/WRITE MSM	36
3.4.18	CALL	37
3.4.19	BRANCH	38
3.4.20	BIAS	38
3.4.21	SET CYF	39
3.4.22	4-BIT MANIPULATE	39
3.4.23	BIT TEST BRANCH FALSE	40
3.4.24	BIT TEST BRANCH TRUE	40
3.4.25	SKIP WHEN	41
3.4.26	CLEAR REGISTERS	42
3.4.27	BIND	42
3.4.28	OVERLAY M-MEMORY	43
3.4.29	DISPATCH	43
3.4.30	CASSETTE CONTROL	45
3.4.31	HALT	45
3.4.32	NO OPERATION	46
3.4.33	MONITOR	46
3.5	CONCURRENT OPERATION (Processor-1 only)	46
3.6	M-INSTRUCTION (Processor-1)	52
3.7	M-INSTRUCTION TIMING (Processor-2)	55

1.0 INTRODUCTION

1.1 PURPOSE

This specification defines the functional requirements of two M-Memory Processors intended for general purpose data processing. Environmental conditions, safety conditions, reliability parameters, power requirements, etc., are specified in P.S. #1913 1739, B1700 CENTRAL SYSTEMS.

Both processors have I/O bus interfaces for communication with the I/O subsystem and control panel interfaces for communication with the control panel.

M-Processor-1 has a Port Adapter-Port Device Interface and requires the use of Port Adapter-1 in Port Interchange-1. Port Interchange-1 provides the communication paths to the system's main memory and to other port devices such as the B1700 Multiline Control.

M-Processor-2 has a Processor Adapter Interface that provides for acceptance of either Processor Adapter-1 or -2. Processor Adapter-1 provides a direct interface to main memory. No port interchange is used. Processor Adapter-2 provides a direct interface to Port Interchange-2 (no port adapter is used) which, in turn, provides the communication paths to main memory and to other port devices such as the Multiline Control.

1.2 PRODUCT IDENTIFICATION

2204 8839	M-MEMORY PROCESSOR-1
2212 8318	M-MEMORY PROCESSOR-2
2212 8326	M-PROCESSOR ADAPTER-1 (Direct Connect) *
2212 8334	M-PROCESSOR ADAPTER-2 (Port Connect) *
2204 8847	M-MEMORY ADAPTER-1 2KB **
2212 1982	M-MEMORY ADAPTER-2 8KB ***
2212 1990	M-MEMORY ADAPTER-2 6KB ***
2212 2006	M-MEMORY ADAPTER-2 4KB ***
2212 2014	M-MEMORY ADAPTER-2 2KB ***
2212 2022	M-MEMORY EXP KIT-2 2KB ***

* Used with M-Processor-2
** Used with M-Processor-1
*** Used with M-Processor-1 or -2

1.3 RELATED SPECIFICATIONS

P.S. #	NAME
1904 5681	B1700 SYSTEM INDEX
1913 1739	B1700 CENTRAL SYSTEM
2204 8623	B1700 I/O BUS SUBSYSTEM
1913 1754	B1700 MEMORY SUBSYSTEM
1913 1788	B1700 PORT INTERCHANGE
2209 7968	B1700 CONTROL PANEL

S.D.S. #	NAME
2200 2083	B1700 PORT ADAPTER - PORT DEVICE INTERFACE
2210 0143	B1700 CONTROL PANEL INTERFACE

2.0 GENERAL DESCRIPTION

The M-Memory Processor (or M-Processor) provides the combinatorial and arithmetic portion of the system along with other registers and hard storage that are appropriate for efficient operation.

The M-Processor provides hardware sensitivity to a set of low-level micro-functions which are used in a program string to fetch and execute instructions. This micro-program is contained in either a local high-speed Read-Write M-Memory or in the somewhat slower but larger main memory (B1700 S-Memory) or in both. M-Memory is modular in increments of 2K bytes up to a maximum of 8K bytes.

Included in the M-Processor are registers and pseudo registers which are addressable by the individual micro-operators.

The registers are normally addressed by a 4-bit group (row) number and a 2-bit select (column) number as shown in Table 1.

Some of the registers listed, such as the Pseudo Sum Register, can serve only as source registers while others are capable of serving both as source and destination registers. Also, some of the registers listed are actually subregisters which, although parts of larger registers, can be individually addressed and manipulated.

Table 2 summarizes the various conditions available by addressing particular pseudo source registers and actual registers; Figure 1 lists the micro-instructions and their variants; and Figure 2 is a diagram of the major registers.

TABLE 1: M-PROCESSOR REGISTER SELECTION

		SELECT NUMBER 0	1	2	3
GROUP NUMBER	0	TA	FU	X	SUM
	1	TB	FT	Y	CMPX
	2	TC	FLC	T	CMPY
	3	TD	FLD	L	XANY
	4	TE	FLE	A	XEQY
	5	TF	FLF	M	MSKX
	6	CA	BICN	BR	MSKY
	7	CB	FLCN	LR	XORY
	8	LA	TOPM	FA	DIFF
	9	LB	UNASSIGNED	FB	MAXS
	10	LC	UNASSIGNED	FL	MAXM
	11	LD	PERR	TAS	U
	12	LE	XYCN	CP	MBR
	13	LF	XYST	MSM	DATA
	14	CC	INCN	READ	CMND
	15	CD	CPU	WRIT	NULL

TABLE 2: SUMMARY OF REGISTER CONDITIONS

BICN:	LSUY	CYF	CYD	CYL
XYCN:	MSBX	X=Y	X<Y	X>Y
XYST:	LSUX	INT	Y neq 0	X neq 0
FLCN:	FL=SFL	FL>SFL	FL<SFL	FL neq 0
INCN:	PORT DEVICE MISSING	PORT HI PRIORITY	PORT INTERRUPT	PORT LOCKOUT
CC:	CONTROL PANEL STATE LAMP FLIP-FLOP	TIMER INTERRUPT	I/O BUS INTERRUPT	CONTROL PNL INTERRUPT
CD:	MEMORY * READ DATA PAR. ERR. INTERRUPT	MEMORY WRT/SWAP ADDR OUT OF BOUNDS OVERRIDE	MEMORY READ ADDR OUT OF BNDS INTERRUPT	MEMORY WRT/SWAP ADDR OUT OF BOUNDS INTERRUPT

* Both S-Memory and M-Memory (See Section 3.3.16 for details)

	MICRO NAME	MC 15	14	13	12	MD 11	10	9	8	ME 7	6	5	4	MF 3	2	1	0
1C	REGISTER MOVE	0	0	0	1	REG 1 GROUP SOURCE REGISTER				REG 1 SELECT		REG 2 SELECT		REG 2 GROUP SINK REGISTER			
2C	SCRATCHPAD MOVE	0	0	1	0	REGISTER GROUP SOURCE OR SINK				REG SELECT		MOV DIR	DPW 1/2	DOUBLE PAD WORD ADDRESS			
3C	4 BIT MANIPULATE	0	0	1	1	REGISTER GROUP, 4 BIT SOR & SNK				REG SEL	MANIPULATE VARIANTS			4 BIT MANIP. LITERAL			
4C	BIT TEST REL BRANCH FALSE	0	1	0	0	REGISTER GROUP, 4 BIT SOURCE				REG SEL	TESTBIT NUMBER		DSP SGN	RELATIVE BRANCH DISPLACEMENT MAG.			
5C	BIT TEST REL BRANCH TRUE	0	1	0	1	REGISTER GROUP 4 BIT SOURCE				REG SEL	TESTBIT NUMBER		DSP SGN	RELATIVE BRANCH DISPLACEMENT MAG.			
6C	SKIP WHEN	0	1	1	0	REGISTER GROUP 4 BIT SOR & SNK				REG SEL	SKIP TEST VARIANTS			4 BIT TEST MASK			
7C	READ/WRITE MEMORY	0	1	1	1	R/W VAR	COUNT FA/FL VARIANTS			DATAREG CODE			TW SGN	DATA TRANSFER WIDTH MAGNITUDE			
8C	MOVE 8 BIT LITERAL	1	0	0	0	REGISTER GROUP; REG SEL IS 2.				ENTIRE 8 BITS OF 8 BIT LITERAL							
9C	MOVE 24 BIT LITERAL	1	0	0	1	REGISTER GROUP; REL SEL IS 2.				8 MOST SIGNIFICANT BITS OF FULL 24 BIT LITERAL							
10C	SHIFT/ROTATE T REG	1	0	1	0	SINK REGISTER GROUP				SNK REG SELECT			S/R VAR	LEFT SHIFT/ROTATE COUNT			
11C	EXTRACT FROM T REG	1	0	1	1	RIGHT BIT POINTER FOR EXTRACTION FLD				SNK REG CODE				EXTRACTION FIELD WIDTH			
123C	BRANCH RELATIVE	1	1	0	DSP SGN	RELATIVE DISPLACEMENT MAGNITUDE											
145C	CALL RELATIVE	1	1	1	DSP SGN	RELATIVE CALLED ADDRESS MAGNITUDE											
2D	SWAP MEMORY	0	0	0	0	0	0	1	0	DATAREG CODE			TW SGN	DATA TRANSFER WIDTH MAGNITUDE			
3D	CLEAR REGISTERS	0	0	0	0	0	0	1	1	L REG	T REG	Y REG	X REG	FA REG	FL REG	FU REG	CP REG
4D	SHIFT/ROTATE X OR Y	0	0	0	0	0	1	0	0	S/R DIR VARIANT		X/Y VAR	LEFT OR RIGHT, X OR Y SHIFT/ROTATE COUNT				
5D	SHIFT/ROTATE X AND Y	0	0	0	0	0	1	0	1	S/R,DIR VARIANT		LEFT OR RIGHT X AND Y SHIFT/ROTATE COUNT					
6D	COUNT FA/FL	0	0	0	0	0	1	1	0	COUNT FA/FL VARIANTS				COUNT SCALAR MAGNITUDE			
7D	EXCHANGE DPW	0	0	0	0	0	1	1	1	SINK DPW ADDRESS				SOURCE DPW ADDRESS			
8D	SCRATCHPAD RELATIVE FA	0	0	0	0	1	0	0	0				DSP SGN	LEFT HALF PAD WORD ADDRESS			
9D	MONITOR	0	0	0	0	1	0	0	1	LITERAL OCCURRENCE IDENTIFIER							
1E	DISPATCH	0	0	0	0	0	0	0	0	0	0	0	1	DISPATCH VARIANTS			SKP FLG
2E	CASSETTE CONTROL	0	0	0	0	0	0	0	0	0	0	1	0	CASSETTE MANIP VARIANTS			
3E	BIAS	0	0	0	0	0	0	0	0	0	0	1	1	BIAS VARIANTS			TST FLG
4E	STORE F INTO DPW	0	0	0	0	0	0	0	0	0	1	0	0	SINK DPW ADDRESS			
5E	LOAD F FROM DPW	0	0	0	0	0	0	0	0	0	1	0	1	SOURCE DPW ADDRESS			
6E	CARRY FF MANIPULATE	0	0	0	0	0	0	0	0	0	1	1	0	CYF CYD	CYF CYL	CYF 1	CYF 0
7E	EXERCISE MSM(L)	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	R/W VAR
1F	HALT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2F	OVERLAY M-STRING	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
3F	NORMALIZE X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
4F	BIND A, TOPM, MBR	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
ZERO	NO OPERATION	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VARIANTS	0 00 000	1 01 001	10 010	11 011	100	101	110	111
MOV DIR: 1/2 DPW:	P←R LEFT	R←P RIGHT						
MANIP. VARIANTS:	SET	AND	OR	EOR	INC	INC TEST	DEC	DEC TEST
DSP SIGN:	+	−						
DSP SIGN:	+	−						
SKIP TEST VARIANTS:	ANY CLR/	ALL CLR/	EQL CLR/	ALL CLR	ANY/ CLR/	ALL / CLR /	EQL / CLR /	ALL / CLR
R/W VAR:	READ	WRT						
CNT VAR:	NØP	FA↑	FL↑ T	FA↑ FL↓ L	FA↑ FL↑	FA↑	FL↓	FA↑ FL↑
REG SEL: TW SIGN:	X +	Y −						
S/R VAR:	SHFT	ROT						
SINK REG CODE:	X	Y	T	L				
DSP SIGN:	+	−						
DSP SIGN:	+	−						
TW SIGN; REG CODE:	+ X	− Y	T	L				
X/Y VAR:	X	Y						
S/R, DIR:	SFT←	SFT→	ROT←	ROT→				
S/R, DIR VARIANTS:	SFT←	SFT→	ROT←	ROT→				
COUNT FA/FL VAR:	NØP	FA↑	FL↑	FA↑ FL↓	FA↑ FL↑	FA↑	FL↓	FA↑ FL↓
DSP SIGN:	+	−						
SKP FLAG: DISP VAR:	FAIL LOCK	SUCC WRTLO	READ	R&C	WRTHI	ABSNT	UNDEF	UNDEF
CASSETTE MANIP:	START TAPE	STOP@ GAP	STOP ON X≠Y	UNDEF	UNDEF	UNDEF	STOP ON X=Y	UNDEF
TEST FLG: BIAS VAR:	TST/ UNIT	TEST F	S	FS	NØP	FCP	NØP	NØP
R/W VAR	READ	WRIT						

REGISTER SELECT

REGISTER GROUP	0	1	2	3
0	TA	FU	X	SUM
1	TB	FT	Y	CMPX
2	TC	FLC	T	CMPY
3	TD	FLD	L	XANY
4	TE	FLE	A	XEQV
5	TF	FLF	M	MSKX
6	CA	BICN	BR	MSKY
7	CB	FLCN	LR	XORY
8	LA	TOPM	FA	DIFF
9	LB	....	FB	MAXS
10	LC	....	FL	MAXM
11	LD	....	TAS	U
12	LE	XYCN	CP	MBR
13	LF	XYST	MSMA	DATA
14	CC	INCN	READ	CMND
15	CD	CPU	WRIT	NULL

INDIVIDUAL BITS OF SOME 4-BIT REGISTERS HAVE SPECIAL MEANINGS AS NOTED BELOW:

NOTE: BITS ARE NUMBERED HERE ACCORDING TO THE HARDWARE CONVENTION:

LA	LB	LC	LD	LE	LF
3 0.3	0.3	0.3	0.3	0.3	0.3 0
MSB 23			L		LSB 00

BIT #	BICN	XYCN	XYST	FLCN
0	CYL	X>Y	X≠0	FL≠0
1	CYD	X<Y	Y≠0	FL<SFL
2	CYF	X=Y	INT OR	FL>SFL
3	LSUY	MSBX	LSUX	FL=SFL

INT OR = ANY 1 OR MORE OF: CC0 CC1 CC2 CD0 CD3 INCN 1 INCN 3.

INCN
- 0 PORT DISPATCH LOCKOUT
- 1 PORT DISPATCH INTERRUPT
- 2 PORT HI PRIORITY DISPATCH INTERRUPT
- 3 INTERRUPT FOR MISSING PORT

CC & CD
- CC0 CONSOLE SOFT HALT INTERRUPT.
- CC1 I/O BUS SERVICE REQUEST INTERPT.
- CC2 REAL TIME CLOCK (100 MS) INTERPT.
- CC3 STATE FLAG (DISPLAYED ON CONSOLE).
- CD0 MEMORY WRITE/SWAP ADDRESS OUT OF BOUNDS INTERRUPT.
- CD1 MEMORY READ ADDRESS OUT OF BOUNDS INTERRUPT.
- CD2 MEMORY WRITE/SWAP ADDR OUT OF BOUNDS OVERRIDE FLAG.
- CD3 MEMORY READ DATA PARITY ERROR INTERRUPT.

FIGURE 1 M-PROCESSOR MICRO-INSTRUCTIONS AND VARIANTS

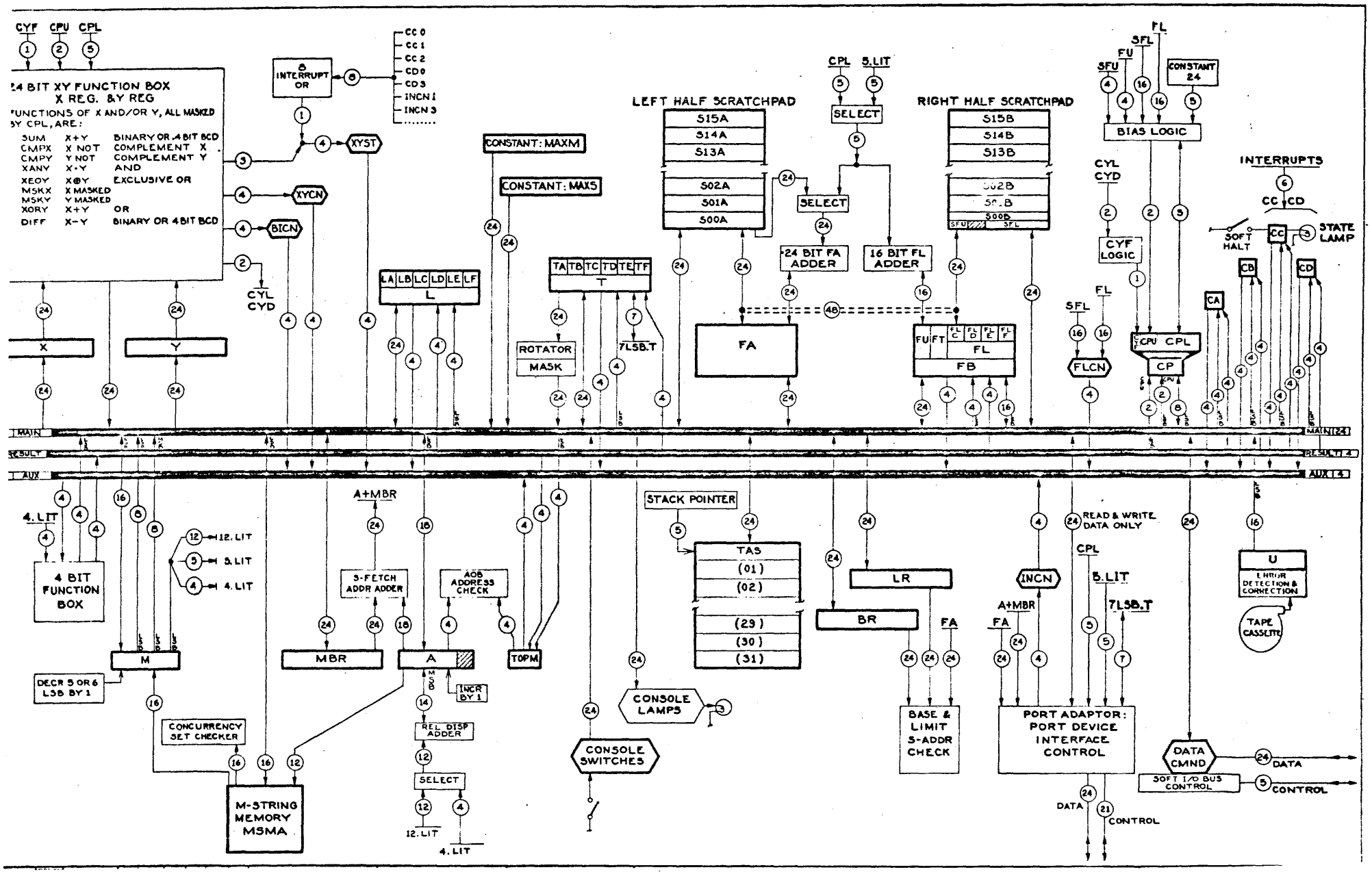


FIGURE 2 M-PROCESSOR REGISTERS

3.0 PRODUCT DESCRIPTION

3.1 PROCESSOR REGISTERS

3.1.1 M

The M Register (micro-register) is a 16-bit register used to hold the active micro-instruction (M-instruction or micro-operator or M-op). The state of this register is decoded to enable the different control signals within the processor to perform the operations called for by the M-Op. The M Register is broken into 4 fields for decoding that are structured such that 60 distinct M-ops can be decoded.

The M Register is addressable as a source and as a sink (destination). When used as a sink register, the source is bit-ORed with the upcoming M-op. Exception: In TAPE mode, the source is not bit-ORed with the upcoming M-op.

3.1.2 MSM

The MSM (micro-string memory or M-Memory) Register is a pseudo register that is addressable as a source when the control panel shows HALT, or as a sink when the control panel is switched to TAPE mode. As a source, MSM contains the micro-instruction that is pointed to by the A Register (See Section 3.1.3). It is addressable as a sink by the micro-instructions MOVE 24-BIT LITERAL and REGISTER MOVE in the TAPE mode. (See Section 3.4.1, REGISTER MOVE.)

M-Memory is a high-speed memory used to hold the sequences of micro-instructions that perform the macro-operations called for by S-language operators. The S-op's are normally located in main memory (B1700 S-Memory). M-Memory is available in increments of 1024 16-bit words up to a maximum of 4096 16-bit words. Any excess strings of micro-instructions which do not fit into the installed M-Memory are located in main memory.

Normal micro-programming should order micro-instruction sequences so that the fastest execution speed possible is achieved regardless of the actual size of the installed M-Memory. Significant improvements in throughput are

achieved when more and more of the important sequences of M-instructions are located in the M-Memory, since the M-Memory is at least five times faster than main memory. The M-Processor is capable of overlap operations with main memory.

M-Memory is addressable as a source and as a sink. The location accessed is usually determined by the contents of the A Register, which normally point to the next micro-operator. However, the micro-operator READ/WRITE MSM accesses the word determined by the contents of the L Register as does the MOVE SMEM TO MMEM (overlay) micro-operator.

M-Processor-2 provides an additional parity bit used to check parity. "Odd" denotes good parity.

3.1.3 A, MBR AND TOPM REGISTERS

The A Register is a 14-bit micro-program address register capable of addressing 16,384 micro-operators located in M-Memory and/or main memory.

The A Register is capable of having binary increments from 0 through 4095 added to or subtracted from it. A high-speed carry adder facilitates micro-program branching. The A Register is automatically incremented during RUN mode. Wrap-around can occur and is permitted.

The A Register can be addressed as a source and as a sink. When used as a source, the contents of the A Register are multiplied by 16. When used as a destination, the rightmost 4 bits of the source are lost.

Associated with the A Register is a 4-bit TOPM (Top of M-Memory) Register. This register, multiplied by 512, is compared with the A Register to determine from which memory (M-Memory or main memory) the micro-operator is addressed.

If the address in the A Register is equal to or is greater than the address denoted by (512) X (TOPM), the micro-operator is obtained from main memory, otherwise, it is obtained from the M-Memory.

To obtain the micro-operator from main memory, the A Address is multiplied by 16 and added to a 24-bit MBR (Micro-Instruction Base Register) to yield a bit address pointing to the micro-operator. (In S-Memory Processor-2, the Bit Address Modulo 16 must equal zero).

Both the TOPM and MBR Registers are addressable as sources and as sinks.

The TOPM Register is cleared to the binary value 1000 by the system clear signal.

The MBR, A, and TOPM Registers are addressed as destination registers by the micro-operator "BIND".

3.1.4 A STACK

The A Stack is 32 words deep by 24 bits wide, does not have automatic hard overflow, and operates as a push-down stack with a last-in, first-out type of structure. Using this stack, the microcoded routines operate in the normal software call-return type of programming. This allows for a highly shared microcode structure and reduces the associated memory requirements. Although the A Stack is not intended to be used as an operand stack, it has purposely been made 24 bits wide to allow limited capabilities for operand storage.

Wrap around of the TAS pointer is provided. That is, 32 consecutive pops or 32 consecutive pushes will cause the TAS pointer to contain its original contents.

3.1.5 TAS

The TAS (Top of Stack) Register is a 24-bit register which is the top of the A Stack. The TAS Register is addressable as a source and as a sink. References to TAS result in the stack being automatically pushed or popped.

3.1.6 X AND Y

The X Register and the Y Register are 24-bit general purpose registers used primarily to hold and to act as a source for the two operands of the arithmetic and combinatorial unit. Each register is addressable as a source and as a sink.

Both registers, along with the L Register and the T Register, are capable of read/write operations with main memory.

Both registers are capable of all SHIFT/ROTATE operations. The X Register is capable of the NORMALIZE and the READ/WRITE MSM operations.

3.1.7 L

The L Register is a 24-bit general purpose register used typically to hold logical flags for the micro-program code. The L Register, as well as each 4-bit group of L (denoted as LA, LB, LC, LD, LE, and LF), is addressable as a source and as a sink.

DISPATCH operations use the L Register as the source or sink for a 24-bit message (usually an address) which is stored in/fetched from S-Memory location zero.

OVERLAY M-MEMORY operations use the L Register as the source of the starting M-Memory address to be used in the M-Memory overlay operation.

The BIND operator uses the L Register as the source of the 24-bit value to be moved to the MBR.

Since the L Register is addressable in 4-bit groups, its contents are available for analysis and alteration via the 4-bit function box. MANIPULATE, SKIP and BIT-TEST-BRANCH instructions can operate on L data.

The L Register is one of 4 registers (X, Y, L, and T) capable of read/write operations with main memory.

The READ/WRITE MSM operator uses the L Register as the source of the M-Memory address to be used in the operation.

3.1.8 T

The T Register is a 24-bit general purpose register used primarily for the interpretation of S-Language instructions. The T Register, as well as each 4-bit group of T (denoted as TA, TB, TC, TD, TE, and TF), is addressable as a source and as a sink.

DISPATCH operations use the least significant seven bits of T as the source or sink for the port and channel information associated with the DISPATCH operation.

The BIND operator uses the T Register as the source of the 4-bit value to be moved to the TOPM Register and as the source of the 14-bit value to be moved to the A Register. The value to be moved to TOPM is contained in the rightmost (LSB) 4 bits of T while the value to be moved to A is contained in the leftmost (MSB) 20 positions of T. Note that when data is moved to A, it will be truncated on the left.

Since the T Register is addressable in 4-bit groups, its contents are available for analysis and alteration via the 4-bit function box. MANIPULATE, SKIP and BIT-TEST-BRANCH instructions can operate on T data.

The T Register is one of 4 registers (X, Y, L, and T) capable of read/write operations with main memory.

The T Register is also capable of the SHIFT/ROTATE and EXTRACT operations.

3.1.9 FA

The FA Register (Field Address) is a 24-bit register used primarily to hold an absolute bit address for main memory. It has the capability of directly addressing any bit in the memory starting at any point.

The FA Register is addressable as a source and as a sink.

The FA Register is capable of being counted up or down by a literal in a micro-instruction or by the value contained in CPL in order to facilitate iteration through a memory field. It can be incremented or decremented by a value in a left scratchpad word. It also has the capability of being loaded, stored, or swapped along with FB into a double scratchpad word.

Neither overflow nor underflow of FA is detected. The value of FA may go through its maximum value or its minimum value and wrap around.

3.1.10 FB

The FB Register is a 24-bit register which can be functionally divided into three portions: a 4-bit FU (Field Unit) register, a 4-bit FT (Field Type) register, and a 16-bit FL (Field Length) register.

The FB Register, as well as each 4-bit portion of FB denoted as FU, FT, FLC, FLD, FLE, and FLF is addressable as a source and as a sink. In addition, the 16-bit portion comprised of FLC, FLD, FLE, and FLF and denoted as FL is also addressable as source and as a sink.

The FU Register holds the length of the unit which makes up a field in memory. The FT Register holds field type information while the FL Register holds the total length of the field. FL is capable of describing fields up to 65,636 bits and can be adjusted up or down by a literal in a micro-instruction or by the value contained in CPL in order to facilitate iteration through a memory field.

Note: Overflow of FL is not detected. The value of FL will go through its maximum value and wrap around. Underflow of FL is detected and will not wrap around. The value of zero is left in FL.

Since the FB Register is addressable in 4-bit groups, its contents are available for analysis and alteration via the 4-bit function box. MANIPULATE, SKIP and BIT-TEST-BRANCH instructions can operate on FB data.

FB has the capability of being loaded, stored, or swapped along with FA into a double scratchpad memory word.

FU and FL along with corresponding portions of the first cell of scratchpad are used to set the various conditions of FLCN (see Section 3.1.12, FLCN) and the various conditions of the CP register (see Section 3.4.20, BIAS).

3.1.11 SCRATCHPAD

A scratchpad memory of sixteen 48-bit words is provided to hold field descriptors during the iteration of operands. Some cells may be used to hold S-Language stack pointers and other processor registers which are under constant manipulation.

The FU and FL portion of the FB Register and like portions of the first cell of the scratchpad are used to set the various conditions of FLCN (see Section 3.1.12, FLCN) and the various conditions of the CP register (see Section 3.4.20, BIAS).

3.1.12 FLCN

FLCN (field length condition) is a 4-bit pseudo register that holds the result of a static comparison of the FL portion of the FB register and the corresponding portion of the first scratchpad word. It is addressable as a source only.

FLCN:	FL=SFL	FL>SFL	FL<SFL	FL neq 0

3.1.13 BR AND LR

The LR and BR Registers are both 24-bit registers and are used for memory protection and for base relative addressing. Memory protection is provided by checking the memory address in FA with the BR (Base Register) and the LR (Limit Register) for all memory cycles. Any address outside these bounds is flagged in the CD Register. A memory READ operation is always allowed whether inside or

outside the boundary, but WRITE or SWAP operations are allowed outside the boundary only if the override bit of the CD register is true. A memory address equal to the BR or LR is considered in bounds. Note: The memory protection is provided only on the initial pointer and does not provide protection on those memory bits accessed when the Field Length is greater than one. The COUNT operation specified by the count variants will take place regardless of whether or not the memory cycle takes place.

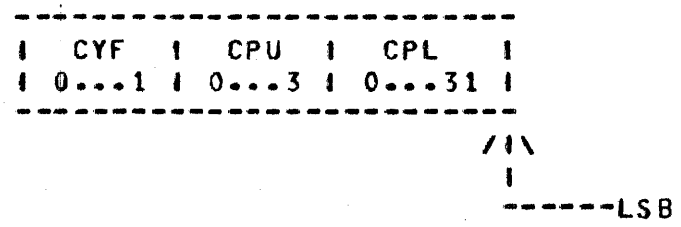
Each register is addressable as a source and as a sink.

3.1.14 C

The C (control) Register is a 24-bit register which is not addressable is an entity but which is functionally divided into one 8-bit section and four 4-bit sections.

3.1.14.1 CP (CYF, CPU, CPL)

The 8-bit section, addressed as CP, is comprised of the arithmetic unit carry flip-flop (CYF), the 2-bit unit control for the arithmetic unit (CPU) and the 5-bit variable data length control (CPL). CPU is also addressable as a source and as a sink and when so addressed can be treated as if it were a 4-bit register whose leftmost 2 bits are always equal to zero.

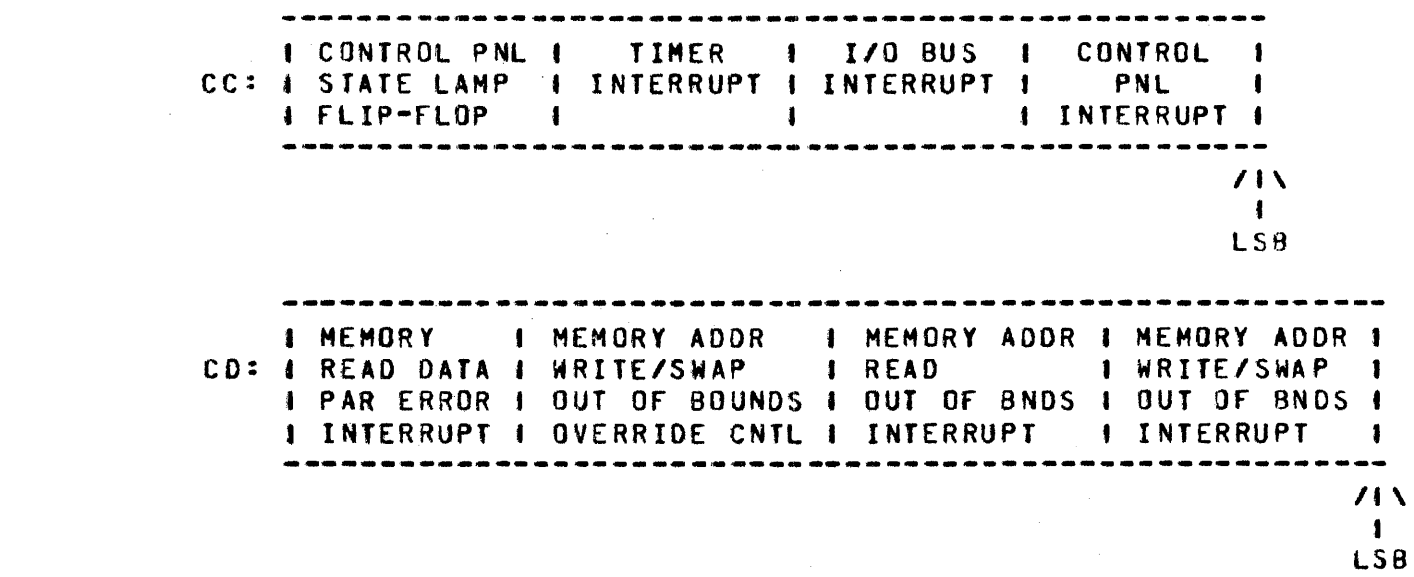


3.1.14.2 CA, CB, CC, CD

The remaining 16-bits of the C Register are addressable in 4-bit groups as CA, CB, CC and CD. Their contents are available for analysis and alteration via the 4-bit function box. MANIPULATE, SKIP and BIT-TEST-BRANCH instructions are applicable.

The 4-bit groups designated as CA and CB have no special functional assignment and are available as general purpose 4-bit storage registers.

The two 4-bit registers designated as CC and CD are used for the storage of various processor states and conditions as shown below:



The control panel state lamp flip-flop when true will cause the STATE lamp to light.

The timer interrupt signal is developed from the primary power frequency, field adjustable for either 50Hz or 60Hz. The interrupt signal is received and is used to set the appropriate CC bit once every 100 milliseconds.

The I/O Bus interrupt level is derived from the various I/O controls connected to the processor's I/O Bus. The level is the result of a service request by one or more controls and is used to set the interrupt bit every clock time.

The control panel interrupt level is derived from the on position of the control panel's INTERRUPT switch. The level from the switch is used to set the interrupt bit every clock time.

The memory read data parity error interrupt is set as a result of a parity error detected during:

(1) console READ of S-Memory by M-Processor-1,

(2) READ/SWAP/DISPATCH overlay from S-Memory,

(3) Micro-op fetch from M-Memory if H/F flip-flop = 1 (no halt),

(4) MOVE MSM TO REGISTER if H/F flip-flop = 1 (no halt),

(5) micro-op fetch from S-Memory by Processor-1 (program halts)

The S-Memory (main memory) parity error signal is received from the Port Interchange via the port-processor interface, or in the case of direct connection, from main memory. The M-Memory parity error signal is developed from logic in the processor.

The memory address out-of-bounds signals are derived from logic which compares the contents of the FA Register with the contents of the Base (BR) and Limit (LR) Registers on all memory accesses. The state of the out-of-bounds override control bit does not affect the setting of out-of-bounds interrupt bits but does affect the occurrence of the subsequent write operation. (See Section 3.3.7, Base and Limit Registers.)

No reaction occurs as a result of any interrupt until the micro-program tests the interrupt bit.

A micro-instruction or the control panel CLEAR pushbutton is capable of resetting a bit in the C Register. The bit being reset will be false for at least one clock period following the reset regardless of the continued existence of the condition to set the bit (e.g., control panel or service request interrupts). Any test micro-operator executed in this clock period will find the bit false. If the condition does not continue to exist beyond the reset time, a failure to set the bit may occur (e.g., timer interrupt).

3.1.15 MAXS, MAXM

MAXS and MAXM are both 24-bit pseudo registers that can be field-adjusted to give the actual size of the memories installed in the system. Both are addressable as sources only. MAXS, for main (S) memory, has 4K-byte resolution (least significant 15 bits are always zeroes). MAXM, for control (M) memory has a 1K-word resolution (least significant 10 bits are always zeroes).

3.1.16 U

The U Register is a 16-bit register used primarily to accumulate the bit-by-bit input from the control panel's tape cassette. The U Register is addressable as a source register only.

In RUN mode, if data is not yet available in the register, the micro-operator will be delayed. Data not accepted in time will be lost.

In TAPE mode, the register's contents are automatically moved to the M Register for execution except when executing micro-operators that reference the U Register as a source. In these cases the source data is moved directly from the U Register to the destination and will not be treated as an instruction to be executed. The next instruction from the tape will follow this data. For MOVE 24-BIT LITERAL, the 8 bits of the literal in the M Register are also moved to the destination. For CONDITIONAL BRANCH, the A Register may be changed but the next micro-operator read from the tape will also be executed.

Note: In RUN mode, the U Register may not be addressed after the issuance of a CASSETTE STOP micro.

3.1.17 DATA

DATA is a 24-bit pseudo register that can act as a source or as a destination. It is used to transfer data to and from the I/O Bus. When it is used as a source, the processor generates the RC (RESPONSE COMPLETE) signal to

the interface and accepts the 24-bits of data from the bus. When used as a destination, the processor generates the RC signal to the interface and the data from the designated source to the bus.

3.1.18 CMND

CMND (Command) is a 24-bit pseudo register that can act as a destination only. It is used to transfer commands to devices on the I/O Bus. The processor generates the CA signal to the interface and moves the data (or the command) from the designated source to the bus.

3.1.19 NULL

NULL is a 24-bit pseudo register that can act as a source and as a destination. When addressed as a source, all zeros are supplied to the destination. When addressed as a destination, the source data is not accepted. However, NULL is useful as a destination in order to pop the TAS Register without affecting other registers.

3.1.20 READ

READ is a 24-bit pseudo register that can act as a source only. When addressed as a source, information on the positions of the control panel switches is supplied to the destination.

3.1.21 WRIT

WRIT is not permitted to be addressed as a source or as a destination, but is used in conjunction with a memory WRITE operation from the control panel.

3.2 24-BIT FUNCTION BOX

The 24-bit "function box" is composed of a 24-bit arithmetic unit and a 24-bit combinatorial unit. It has as data inputs the contents of the X and Y Registers and the carry flip-flop (CYF). It also uses CPU (control for the arithmetic unit) and CPL (the 5-bit variable operand length) from the CP portion of the C Register.

All results from the combinatorial section are generated immediately and are continuously available to the micro-programmer. A move to one of the input registers or an alteration of a value in the CP portion of the C Register immediately generates a new result. The results are available to the next micro-instruction and are accessed by moving the contents of a result register to a destination register or by testing one of the 4-bit condition registers.

The results are most of the commonly used functions between two operands. These include the And, Or, Exclusive-Or, sum, carry-out, difference, and borrow functions and the set of equal-to, greater-than, and less-than relationals. The results of the unary operations of complementation and masking are also available.

The results of the arithmetic unit are under control of the CPU and the CPL Registers as follows.

CPU	UNIT TYPE	POSSIBLE CPL VALUES	DATA TYPE
00	1-bit operands	1 to 24	Binary
00	4-bit operands	4,8,12,16,20,or 24	4-bit binary
10	Undefined		
11	8-bit operands	8, 16, 24	EBCDIC

For valid arithmetic operations, the operand length (as specified by CPL) must be an exact multiple of the length of the unit specified by CPU.

The contents of each of the registers described in the following subsections are immediately available to the micro-programmer.

3.2.1 SUM

SUM is a pseudo register equal to the sum of the X, Y, and CYF Registers (X + Y + CYF). Zero bits in the more significant bit positions of the 24-bit result are produced when the length as determined by CPL (5-bit variable data length control) is less than 24. Results are not defined for CPL values 25 through 31. The carry-out level is generated from the bit position of the output specified by CPL. If CPL = 0, the carry-out level is equal to CYF. If CPL = 1, the carry-out level is generated from the rightmost bit of X, Y and CYF. See Section 3.1.14.1, CP (CYF, CPU, CPL).

If CPU (2-bit arithmetic unit control) = 00, the binary sum is produced. See Section 3.1.14.1, CP (CYF, CPU, CPL).

If CPU = 01, the decimal sum is produced by considering the X and Y inputs to be comprised of six 4-bit units. Results are not defined for non-Binary Coded Decimal (BCD) units. CPL must be a multiple of four.

If CPU = 10 or 11, the sum is undefined.

3.2.2 DIFF

DIFF is a pseudo register equal to the difference of the X, Y, and CYF Registers (X - Y - CYF). Zero bits appear in the more significant bit positions of the 24-bit result when the length as determined by CPL is less than 24. Results are not defined for CPL values 25 through 31. The borrow-out level, generated from the static comparison of all 24-bits of X & Y, is true if X < Y or if X = Y and CYF is true.

If CPU = 00, the binary difference is produced.

If CPU = 01, the decimal difference is produced by considering the X & Y inputs to be comprised of six 4-bit units. Results are not defined for non-BCD units. CPL must be a multiple of four.

If CPU = 10 or 11, the difference is undefined.

A negative result is in 2's complement form in the binary case and in 10's complement form in the decimal case.

3.2.3 XANY, XORY, XEOY

XANY (X and Y), XORY (X or Y), and XEOY (X exclusive-or Y) are 24-bit pseudo registers that hold the results of the specified logical functions. Zero bits are produced in the more significant bit positions of the 24-bit result when the length as determined by CPL is less than 24. Results for the CPL values 25 through 31 are not defined.

3.2.4 CMPX, CMPY

CMPX (complement of X) and CMPY (complement of Y) are 24-bit pseudo registers that hold the 1's complements of the specified registers. Zero bits are produced in the more significant positions of the 24-bit result when the length as determined by CPL is less than 24. CPL values 25 through 31 have undefined results.

3.2.5 MSKX, MSKY

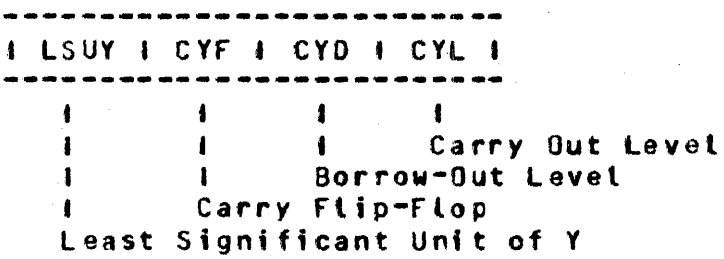
MSKX, (mask of X) and MSKY (mask of Y) are 24-bit pseudo registers that hold the mask of the appropriate register (X or Y). Beginning with LSB of X or Y, the number of bit positions included in the mask is determined by the value of CPL. Zero bits are produced in the more significant bit positions of the 24-bit result when the length as determined by CPL is less that 24. Results are not defined for CPL values 25 through 31.

3.2.6 BICN

BICN (binary conditions) is a 4-bit pseudo register that holds the following binary conditions, considered as a 4-bit group, and addressable as a source only:

NOTE: CYF is also addressed by the SET CYF M-Instruction as well as being available in the (8-bit) group addressed as CP.

BICN:



The carry-out level is a function of X, Y, CPL, and CPU. See Section 3.2.1. The borrow-out level is a function of X, Y, and CYF. See Section 3.2.2, DIFF.

LSUY is true if the least significant unit of Y is equal to 1 and CPU = 00 or 10, or if the least significant unit of Y is equal to 1001 and CPU = 01 or 11.

3.2.7 XYCN

XYCN (XY condition) is a 4-bit pseudo register that holds the following relational conditions, considered as a 4-bit group and addressable as a source only:

XYCN:

MSBX is true if the bit in X referenced by CPL is 1. CPL = 1 references the rightmost bit of X while CPL = 24 references the leftmost bit. MSBX = 0 if CPL = 0.

The relational results are based on the binary value of all 24-bits of X and Y.

3.2.8 XYST

XYST (XY states) is a 4-bit pseudo register that holds the following relational conditions, considered as a 4-bit group and addressable as a source only:

XYST:	LSUX	INT	Y neq 0	X neq 0

LSUX is true when the least significant unit of X is equal to 1 and CPU = 00 or 10, or when the least significant unit of X is equal to 1001 and CPU = 01 or 11.

The relational results are based on the binary value of all 24 bits of X or Y.

INT is true if any of the following conditions as reflected in INCN, CC, and CD are true. (See Section 3.1.10, INCN and 3.1.14.2, CA, CB, CC, CD.)

1. Missing Port Device
2. Port Interrupt
3. Timer Interrupt
4. Bus I/O Interrupt
5. Control Panel Interrupt
6. Memory Parity Error Interrupt
7. Memory Write/Swap Address Out of Bounds Interrupt

3.2.9 INCN

INCN (interrupt conditions) is a 4-bit pseudo register, source only, that reflects the state of certain interface lines between the processor and the port interchange.

	PORT MISSING DEVICE	PORT HI PRIORITY INTERRUPT	PORT INTERRUPT	PORT LOCKOUT
INCN:				

3.2.10 PERR

PERR (parity error) is a 4-bit register, source only, that reflects error conditions as follows:

NO ASSIGNMENT	M-MEMORY PARITY ERROR	S-MEMORY PARITY ERROR	CASSETTE PARITY ERROR

The true state of any bit causes a console lamp to be illuminated and a processor halt.

The PERR Register is cleared by:

(1) a general CLEAR
(2) control panel START pushbutton depressed,
(3) console READ of S-Memory or M-Memory i.e., READ/WRITE or INC pushbutton depressed.

The M-Memory parity error bit is set if a parity error is detected during:

(1) console READ of the MSM Register, i.e., READ/WRITE or INC pushbutton depressed,
(2) fetch of a micro-operator from MSM if halt/flag (H/F flip flop) = 0 (HALT),
(3) REGISTER MOVE of MSM if halt flag = 0,
(4) SCRATCHPAD MOVE of MSM is halt flag = 0.

The S-Memory parity error bit is set if a parity error is detected during:

(1) console READ of S-Memory, i.e., READ pushbutton depressed,
(2) fetch of a micro-operator from S-Memory.

The cassette parity error bit is set if a parity error is detected during a CASSETTE READ.

PERR is present in M-Processor-2 only. If addressed as a source by a M-Processor-1, zero bits are returned.

3.3 4-BIT FUNCTION BOX

The 4-bit function box (4-bit arithmetic and combinatorial section of the processor) can accept, as one of its inputs, the contents of any of the following 4-bit registers and pseudo registers. The second input is obtained from the M-Instruction itself.

TA	TB	TC	TD	TE	TF
LA	LB	LC	LD	LE	LF
FU	FT	FLC	FLD	FLE	FLF
CA	CB	CC	CD	TOPM	CPU
BICN	XYCN	XYST	FLCN	INCN	PERR

Outputs include the result of most of the commonly used functions between two operands; for example: set, and, or, exclusive-or, and binary sum and difference (both modulo 16). Outputs are directed back to the source register.

The sum and difference output can be tested for overflow and underflow respectively and, based on the test, a skip of one instruction can be made.

The 4-bit function box also provides for the selective testing of one of the bits of a four-bit group and relative branching based on the result of the test. A skip of one instruction based on the result of testing on a combination of up to four bits in the group is also provided.

BICN, XYCN, XYST, FLCN, and INCN are not actually registers but can be sourced as if they were. They can be changed

only as a result of changing the condition which they reflect.

CPU is actually a 2-bit register but can be addressed as if it were a 4-bit register whose leftmost 2 bits are always equal to zero.

3.4 M-INSTRUCTIONS

3.4.1 REGISTER MOVE

FORMAT:

OP CODE 0001	SOURCE REGISTER GROUP # 0 -15	SOURCE REGISTER GROUP # 0...3	DESTINATION REGISTER SELECT # 0...3	DESTINATION REGISTER GROUP # 0...15

Move the contents of the source register to the destination register. If the source register is smaller than the destination register, data are right justified with left (most significant) zero bits supplied. If the source register is larger than the destination register, data are truncated from the left.

The contents of the source register are unchanged unless it is also the destination register.

Exceptions:

1. When M is used as a destination register in RUN or STEP mode, the operation is changed to a bit-OR which modifies the next micro-operation. It does not modify the instruction as stored in the memory. In TAPE mode, no bit-OR takes place.

2. WRIT and CMND are excluded as source registers.

3. BICN, FLCN, XYCN, XYST, INCN, READ, WRIT, SUM, CMPX, CMPY, XANY, XEOY, MSKX, MSKY, XORY, DIFF, MAXS, MAXM, and U are excluded as destination registers.

4. When DATA is designated as a source, CMND and DATA are excluded as destinations.

5. When U or DATA is designated as a source and when the next micro is to be obtained from main memory, M is excluded as a destination.

6. U is excluded as a source in STEP mode.

3.4.2 SCRATCHPAD MOVE

FORMAT:

OP CODE 0010	RGSTR GRP # 0...15	RGSTR SLCT# 0...3	DIRECTION 0 TO SCRTCHPD 1 FROM SCRTCHPD	SCRTCHPD WORD 0-LFT WRD 1-RT WRD	SCRTCHPD WORD ADDRESS 0...15

Move the contents of the register (SCRATCHPAD) to SCRATCHPAD (register). If the move is between registers of unequal lengths, the data is right justified with left (most significant) zero bits supplied or with data truncated from the left, whichever is appropriate.

The contents of the source register are unchanged.

Exceptions:

1. When M is used as a destination register, the operation is changed to a bit-or which modifies the next micro-operation. It does not modify the instruction as stored in the memory.

2. WRIT and CMND are excluded as source registers.

3. BICN, FLCN, XYCN, XYST, INCN, READ, WRIT, SUM, CMPX, CMPY, XANY, XORY, XEQY, MSKX, MSKY, DIFF, MAXS, MAXM and U are excluded as destination registers.

4. U is excluded as a source in STEP mode.

3.4.3 SWAP F WITH DOUBLEPAD WORD

FORMAT:

OP CODE	DESTINATION 48-BIT SCRATCHPAD WORD 0...15	SOURCE 48-BIT SCRATCHPAD WORD 0...15
0000 0111		

Move the contents of the FA and FB registers to a holding register. Move the contents of the left and right word of the source scratchpad word to the FA and FB register respectively. Move the contents of the holding register to the left and right word of the destination scratchpad word.

3.4.4 STORE F INTO DOUBLEPAD WORD

FORMAT:

OP CODE	SCRATCHPAD WORD ADDRESS 0...15
0000 0000 0100	

Move the contents of the FA and FB register to the left and right word respectively of the designated scratchpad word.

The contents of the source registers are unchanged.

3.4.5 LOAD F FROM DOUBLEPAD WORD

FORMAT:

OP CODE	SCRATCHPAD WORD ADDRESS 0...15
0000 0000 0101	

Move the contents of the left and right word of the designated scratchpad word to the FA and FB register respectively.

The contents of the source registers are unchanged.

3.4.6 MOVE 8-BIT LITERAL

	OP CODE	DESTINATION REGISTER	LITERAL
FORMAT:	1000	GROUP # 0...15	0...255

Move the 8-bit literal given in the instruction to the destination register. If the move is between registers of unequal lengths, the data is right justified with left (most significant) zero bits supplied. The register select number is assumed to be 2.

Exceptions:

1) READ and WRIT are excluded as destination registers.

2) When M is used as a destination register, the operation is changed to a bit-or which modifies the next micro instruction. It does not modify the instruction as stored in the memory.

3.4.7 MOVE 24-BIT LITERAL

	OP CODE	DESTINATION REGISTER	24-BIT LITERAL
FORMAT:	1001	GROUP # 0...15	0...MAX

Move the 24-bit literal given in the instruction to the destination register. If the move is between registers of unequal lengths, the literal is truncated from the left. The register select number is assumed to be 2.

Exceptions:
---

1) READ, WRIT, M and MSM (except in TAPE mode) are excluded as destination registers.

3.4.8 SWAP MEMORY

FORMAT:

OP CODE	REGISTER #	FIELD DIRECTION	MEMORY FIELD
0000 0010	00 = X 01 = Y 10 = T 11 = L	0 - POSITIVE 1 - NEGATIVE	LENGTH 0...24

Swap data from main memory with the data in the specified register. If the value of the memory field is less than 24, the data from memory is right justified with left (most significant) zero bits supplied while the data from the register is truncated from the left.

Register FA contains the bit address of the memory field while the field direction sign and field length is given in the instruction.

If the value of the memory field length as given in the instruction is zero, the value given in CPL is used.

3.4.9 READ/WRITE MEMORY

FORMAT:

OP CODE	DIRECTION	COUNT VRNTS	RGSTR #	FIELD DIRECTION	MEMORY FIELD
0111	0 to RGSTR 1 TO MEMORY	0...7	00 = X 01 = Y 10 = T 11 = L	0 - POSITIVE 1 - NEGATIVE	LENGTH 0...26

Move the register's (memory's) contents to the memory (register). If the value of the memory field length is less than 24, the data from memory is right justified with left (most significant) zero bits supplied while the data

from the register is truncated from the left.

The contents of the source is unchanged.

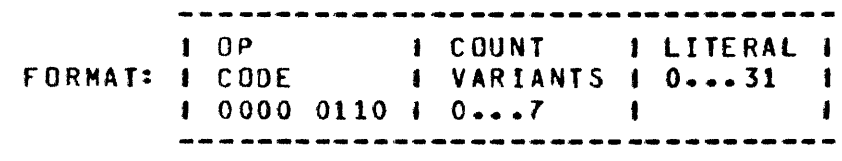
Register FA contains the bit address of the memory field while the memory field direction sign and memory field length is given in the instruction.

If the value of the memory field length as given in the instruction is zero, the value in CPL is used.

Memory field length values (or CPL values if MFL = 0) of 25 and 26 are truncated to the value 24. When used on a WRITE operation, the value 25 and 26 cause odd and even parity respectively to be written into memory regardless of the parity of the read data.

For a description of the count variants, see Section 3.4.10, COUNT FA/FL.

3.4.10 COUNT FA/FL



Increment (decrement) binarily the designated register(s) by the value of the literal contained in the instruction or by the value of CPL if the value of the literal is zero.

Neither overflow nor underflow of FA is detected. The value of FA may go through its maximum value or its minimum value and wrap around.

Overflow of FL is not detected. The value of FL will go through its maximum value and wrap around. Underflow of FL is detected and will not wrap around. The value of zero is left in FL.

Literal values (or CPL values if LIT = 0) of 25 through 31 are truncated to the value 24.

Count variants are as follows:

V = 000 No count
001 Count FA Up
010 Count FL Up
011 Count FA Up and FL Down
100 Count FA Down and FL Up
101 Count FA Down
110 Count FL Down
111 Count FA Down and FL Down

3.4.11 SCRATCHPAD RELATE FA

FORMAT:

OP CODE	RESERVED	SIGN OFF SPAD WORD	LEFT SCRATCHPAD WORD ADDRESS
0000 1000	000	0-POSITIVE 1-NEGATIVE	0...15

Replace the contents of the FA Register by the binary sum of the FA Register and specified scratchpad register.

Neither overflow nor underflow of FA is detected. The value of FA may go through its maximum value or its minimum value and wrap around.

3.4.12 EXTRACT FROM REGISTER T

FORMAT:

OP CODE	ROTATE BIT COUNT	DESTINATION REGISTER	EXTRACT BIT COUNT
1011	0...24	00 - X 01 - Y 10 - T 11 - L	0...24

Rotate register T left by the number of bits specified and then extract from the right the number of bits specified. Move this result to the destination register supplying left most (most significant) zero bits if the extract count is less than 24.

The contents of the source register are unchanged unless it is also the destination register.

A rotate value of 24 is equivalent to 0.

3.4.13 SHIFT/ROTATE REGISTER T LEFT

FORMAT:

OP CODE	DSTNATN REGISTR GROUP #	DSTNATN REGISTR SELECT#	S/R VARIANT	S/R BIT COUNT
1010	0...15	0...3	0 - SHIFT 1 - ROTATE	0...24

SHIFT (ROTATE) Register T left by the number of bits specified and then move the 24-bit result to the destination register. If the move is between registers of equal lengths, the data is right justified with data truncated from the left.

The contents of the source register are unchanged unless the source register is also the destination register.

Zero fill on the right and truncation on the left occurs for the SHIFT operation.

If the value of the SHIFT/ROTATE count as given in the instruction is zero, the value given in CPL is used.

Exceptions:

1. When M is used as a destination register, the operation is changed to a bit-or which modifies the next micro-operation. It does not modify the instruction as stored in the memory.

2. BICN, FLCN, XYCN, XYST, INCN, READ, WRIT, SUM, CMPX, CMPY, XANY, XEQY, MSKX, MSKY, XORY, DIFF, MAXS, MAXM, and U are excluded as destination registers.

3.4.14 SHIFT/ROTATE REGISTER X/Y LEFT/RIGHT

FORMAT:

OP CODE 0000 0100	S/R VARIANT 0-SHIFT 1-ROTATE	L/R VARIANT 0-LEFT 1-RIGHT	X/Y VARIANT 0-X REG 1-Y REG	S/R BIT COUNT 0...24

SHIFT (ROTATE) Register X or Register Y left (right) by the number of bits specified.

Zero fill on the right and truncation on the left occurs for the left shift. Zero fill on the left and truncation on the right occurs for the right shift.

3.4.15 SHIFT/ROTATE REGISTERS XY LEFT/RIGHT

FORMAT:

OP CODE 0000 0101	S/R VARIANT 0-SHIFT 1-ROTATE	L/R VARIANT 0-LEFT 1-RIGHT	S/R BIT COUNT 0...48

SHIFT (ROTATE) Register X and Y left (right) by the number of bits specified. The register X is the leftmost (more significant) half of the concatenated 48-bit XY register.

Zero fill on the right and truncation on the left occurs for the left shift. Zero fill on the left and truncation on the right occurs for the right shift.

3.4.16 NORMALIZE X

FORMAT:

OP CODE 0000 0000 0000 0011

SHIFT the X Register left while counting FL down, until FL=0 or until the bit in X referenced by CPL = 1. Zeros are shifted into the rightmost end of X.

CPL = 1 references the rightmost bit of X while CPL = 24 references the leftmost bit of X. CPL is undefined.

3.4.17 READ/WRITE MSM

FORMAT:

OP CODE	VARIANTS			R/W VARIANT
0000 0000 0111	G/B	H/F	S/N	0 TO X 1 FROM X

Move the contents of the X Register to the M-Memory word specified by the address contained in the L Register if the R/W variant bit = 1. The data is truncated from the left.

Move the contents of the M-Memory word specified by the address contained in the L Register to the X Register if the R/W variant bit = 0. The data is right justified with left (most significant) zero bits supplied.

The lower 4 bits and the upper 8 bits of the address in L are ignored.

READ/WRITE MSM causes the A register to be moved to the TAS Register and the L Register to be moved to the A Register before the instruction is executed. The TAS is restored to A after the READ/WRITE MSM operation is completed.

The S Variant is used to enable the set/reset of the G/B and the H/F flip flops. If S = 1, the G/B and H/F flip flops are set/reset by the G/B and H/F variants. If S = 0, no change is made in the G/B and H/F flip flops.

If the G/B flip flop is true, all READ/WRITE MSM operations will force bad parity in the addressed word. If the G/B flip flop is false, all READ/WRITE MSM operations will force good parity in the addressed word.

If the H/F flip flop is true, the processor upon reading an M-Memory word containing a parity error will flag the error condition by setting a CD bit true. It will not halt. If the H/F flip flop is false, the processor upon detection of a parity error in reading an M-Memory word will flag the error condition by setting PERR bit 1 true and then halt. Reading an M-Memory word occurs when fetching a M-oo from M-Memory or when moving an M-Memory word to any destination.

The H/F and G/B flip flops are cleared to zero (false) with the CLEAR signal. If S = 1, the G/B and H/F flip flops are set/reset prior to the execution of the READ/WRITE MSM portion of the operation.

3.4.18 CALL

FORMAT:

OP CODE	DISPLACEMENT SIGN	DISPLACEMENT VALUE
111	0=POSITIVE 1=NEGATIVE	0...4095

Push the address of the next in-line micro-instruction into the A Stack and then fetch the next micro-instruction from the location obtained by adding the signed displacement value given in the instruction to the address of the next in-line micro-instruction.

A displacement value indicates the number of 16-bit words.

Note: When the A Address is stored in the A Stack, it is multiplied by 16 and stored as a bit address.

Note: Exit is accomplished by employing the REGISTER MOVE instruction with the TAS as the source register and A as the destination register.

3.4.19 BRANCH

FORMAT:	OP CODE	DISPLACEMENT SIGN	DISPLACEMENT VALUE
	110	0=POSITIVE 1=NEGATIVE	0...4095

Fetch the next micro-instruction from the location obtained by adding the signed displacement value given in the instruction to the address of the micro-instruction next-in-line.

A displacement value indicates the number of 16-bit words.

3.4.20 BIAS

FORMAT:	OP CODE	VARIANTS	TEST CPL NEQ 0 FLAG
	0000 0000 0011	0...7	0 - NO TEST 1 - TEST CPL RESULT

Set CPU to the value 1 if the value of FU is 4 or 8 and to 0 otherwise, unless V = 2. If V = 2, the CPU value is determined by SFU in lieu of FU.

Set the value of CPL to the smallest of the values denoted on each line in the following table.

V	VALUES
0	FU
1	24 and FL
2	24 and SFL
3	24 and FL and SFL
4	CPL
5	24 and CPL and FL
6	CPL
7	CPL

If test flag equals 1 and final value of CPL is not zero, the next 16-bit micro-instruction is skipped.

3.4.21 SET CYF

FORMAT:

OP CODE	VARIANTS
0000 0000 0110	1,2,4,8

Set the carry flip-flop as specified by the variants.

V = 1 Set CYF to 0
2 Set CYF to 1
4 Set CYF to CYL
8 Set CYF to CYD

Note CYD = (X<Y) + (X=Y)CYF.

3.4.22 4-BIT MANIPULATE

FORMAT:

OP CODE	REGISTER GROUP #	REGISTER SELECT #	VARIANTS	LITERAL
0011	0...15	0...1	0...7	0...15

Perform the operation specified by the variants on the designated register.

V = 0 Set the register to the value of the literal.
1 Set the register to the logical And of the register and literal.
2 Set the register to the logical Or of the register and literal.
3 Set the register to the logical Exclusive-Or of the register and literal.
4 Set the register to the binary sum (modulo 16) of the register and literal.
5 Set the register to the binary sum (modulo 16) of the register and literal, and skip the next M-Instruction if a carry is produced.

6 Set the register to the binary difference (modulo 16) of the register and literal.

7 Set the register to the binary difference (modulo 16) of the register and literal, and skip the next M-Instruction if a borrow is produced.

Exception

BICN, FLCN, XYCN, XYST, and INCN, when specified as operand registers, are not changed as a result of this operation. However, the carry and borrow outputs are produced and a skip can result.

3.4.23 BIT TEST BRANCH FALSE

FORMAT:

OP CODE	REG GROUP #	REGISTER SELECT #	REG BIT #	DSPLCMNT SIGN	DSPLCMNT VALUE
0101	0...15	0...1	0...3	0-POS 1-NEG	0...15

Test the designated bit within the specified register and branch relative to the next instruction by the signed displacement value if the bit is zero. If the bit is one, a displacement value of zero is assumed and control passes to the next in-line M-instruction. A displacement value indicates the number of 16-bit words from the next in-line instruction.

See Section 3.1.14 for information on the reset of bits in the C Register.

3.4.24 BIT TEST BRANCH TRUE

FORMAT:

OP CODE	RGSTR GROUP #	RGSTR SELECT #	RGSTR BIT #	DSPLCMNT SIGN	DSPLCMNT VALUE
0101	0...15	0...1	0...3	0-POSITIVE 1-NEGATIVE	0...15

Test the designated bit within the specified register and branch relative to the next instruction by the signed

displacement value if the bit is one. If the bit is zero, a displacement value of zero is assumed and control passes to the next in-line M-instruction. A displacement value indicates the number of 16-bit words from the next in-line instruction.

See Section 3.1.14.2, CA, CB, CC, CD, for information on the reset of bits in the C Register.

3.4.25 SKIP WHEN

FORMAT:

OP CODE	REGISTER GROUP #	REGISTER SELECT #	VARIANTS	MASK
0110	0...15	0...1	0...7	0...15

Test only the bits in the register that are referenced by the "1" bits in the mask, ignoring all others unless V = 2 or V = 6. If so, compare all bits for an equal condition. Then perform the action as specified below.

V = 0 If any of the referenced bits is a "1", skip the next M-instruction.

1 If all of the referenced bits are "1", skip the next M-instruction.

2 If the register is equal to the mask, skip the next M-instruction.

3 Same as V = 1, but also clear the referenced bits to zero without affecting the non-referenced bits.

4 If any of the referenced bits is a "1", do not skip the next M-instruction.

5 If all of the referenced bits are "1" do not skip the next M-instruction.

6 If the register is equal to the mask, do not skip the next instruction.

7 Same as V = 4, but also clear the referenced bits to zero without affecting the non-referenced bits.

Note: If the mask equals 0000, the "ANY" result is false. The skip is not made for V = 0 and is made for V = 4. If the mask equals 0000, the "ALL" result is true. The skip is made for V = 1 and V = 3 and is not made for V = 5 and V = 7.

Exceptions:

1. BICN, FLCN, XYCN, XYST, and INCN cannot be cleared with V = 3 or 7. However, they can be tested.

2. See Section 3.1.14 related to the reset of bits in the C Register.

3.4.26 CLEAR REGISTERS

FORMAT:

OP CODE	REGISTER FLAGS 8 BITS							
0000 0011	L	T	Y	X	F A	F L	F U	C P

Clear the specified register(s) to zero if the respective flag bit is a one.

3.4.27 BIND

FORMAT:

OP CODE
0000 0000 0000 0100

Move the 24-bit value from the L Register to the MBR Register. Move the least significant 4 bits from the T Register to the TOPM Register. Move the most significant 20 bits from the T Register to the A Register truncating the left most 6 bits of the source.

3.4.28 OVERLAY M-MEMORY

FORMAT:

OP CODE
0000 0000 0000 0010

Overlay M-Memory from main memory.

The starting addresses of both M-Memory and the main memory are taken from the L Register and the FA Register respectively. The length of the data overlay in bits is taken from the FL Register.

The execution of the instruction proceeds as follows:

1. A is moved to TAS with a stack push.

2. L is moved to A.

3. The first 16 bits are read from main memory and stored in the M-Memory. Register FL is decremented and FA and A are incremented.

4. Step three is repeated until FL = 0 at which point the process terminates with a move of TAS to A.

3.4.29 DISPATCH

FORMAT:

OP CODE	VARIANTS	SKIP VARIANT
0000 0000 0001	000-LOCKOUT	0-SKIP IF ALREADY LOCKED
	001-WRITE LOW	1-SKIP IF NOT ALREADY LOCKED
	010-READ	(Applies only to lockout variant)
	011-READ & CLEAR	
	100-WRITE HIGH	
	101-PORT ABSENT	

Dispatch operations are used to send/receive interrupt and interrupt information to/from other ports.

Since the interrupt system is shared by all ports, the processor should gain control of the interrupt system by successfully completing a LOCKOUT prior to a DISPATCH WRITE.

LOCKOUT sets the lockout bit in the DISPATCH Register and allows, via the skip variant, skipping or not skipping the next 16-bit instruction based upon success or failure (already set) of the LOCKOUT.

WRITE (High or Low) DISPATCH sets the Lockout and Interrupt flip flops in the port interchange. It also stores the contents of the L Register into memory 0 through 23 and the contents of the least significant 7 bits of the T Register (designating the destination port # and channel #) into the appropriate port interchange register. In addition, it sets (Write High) or resets (Write Low) the high Interrupt flip flop in the port interchange.

READ DISPATCH stores the contents of memory locations 0 through 23 into the L Register and the contents of the Port Channel register into the least significant 7 bits of the T Register. The other 17 bits of the T Register are unaffected.

READ AND CLEAR DISPATCH in addition to performing the READ DISPATCH operation clears the lockout flip flop, the two interrupt flip flops and the Port Device Absent flip flop in the port interchange. It does not clear any memory locations.

PORT ABSENT is executed by the processor when necessary to return a Port Device Absent level signal to another port indicating the absence of the designated channel.

Dispatch operations in the case of Processor-2 and Processor Adapter-1 (direct connect to memory) are limited to the following:

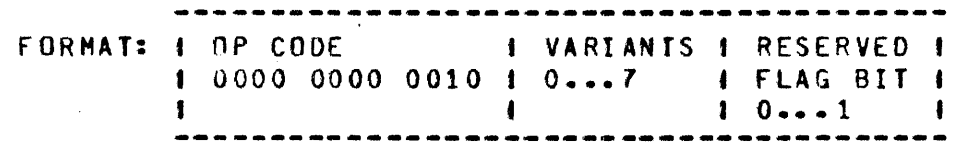
1. LOCKOUT + SKIP-IF-NOT-ALREADY-LOCKED: always skips.

2. WRITE LOW: Always sets Port Device Absent level true (true indicates absence).

3. READ & CLEAR: Always sets the Port Device Absent Level false (false indicates present).

No changes occur in the T and L Registers. In the INCN Register only the Port Device Absent bit can change. The Lockout, the Interrupt, and High Priority bits will always be false. No other dispatch operations are defined.

3.4.30 CASSETTE CONTROL

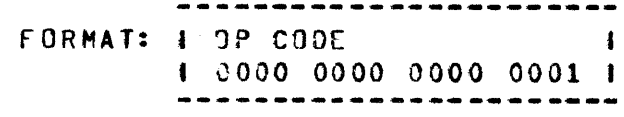


Perform the indicated operation on the tape cassette.

V = 0 Start Tape
1 Stop Tape (The processor also halts if it is in TAPE mode.)
2 Stop Tape if X neq Y (The processor also halts if it is in TAPE mode.)
3 Reserved
4 Reserved
5 Reserved
6 Stop Tape is X = Y (The processor also halts it it is in TAPE mode.)
7 Reserved

Note: All Stop Tape variants cause the tape to halt in the next available gap.

3.4.31 HALT



Stop execution of the micro-instructions. In RUN mode the next micro to be executed is fetched and stored in the M Register and the A Register points to the next following micro. In TAPE mode the next micro is not fetched and stored in the M Register but the HALT micro is left in the M Register.

3.4.32 NO OPERATION

FORMAT:

OP CODE
0000 0000 0000 0000

Skip to the next sequential instruction.

3.4.33 MONITOR

FORMAT:

OP CODE	VARIANTS							
0000 1001	7	6	5	4	3	2	1	0

Skip to the next sequential instruction.

During the time this micro-operator is executing the operator and the last two bits (0 and 1) are decoded, AND-ed with the system clock and are present in the backplane as follows:

MONITOR	0	True for the OP Code
MONITOR	00R0	True if last two bits are 00
MONITOR	01R0	True if last two bits are 01
MONITOR	02R0	True if last two bits are 10
MONITOR	03R0	True if last two bits are 11

At the backplane, the monitors are one-half clock from leading edge to trailing edge.

3.5 CONCURRENT OPERATION (Processor-1 only)

In order to achieve maximum utilization of the processor during processor memory cycles, the micro-instructions are classified into two classes -- a concurrent set and a non-concurrent set. The non-concurrent set is that set which must wait until the memory cycle is completed (data has been accepted or released) before proceeding with its execution. The concurrent set is that set which can overlap its execution with memory cycle. The overlap starts with the clock period following Address Accept and

can extend up to and beyond the time data is accepted.

For a WRITE operation, the processor presents data during the clock period immediately following Address Accept and is released for a non-concurrent operation during the very next clock period.

For a READ operation, the processor accepts data during the first clock period in which data is present and is released for a non-concurrent operation during the clock period immediately following this acceptance.

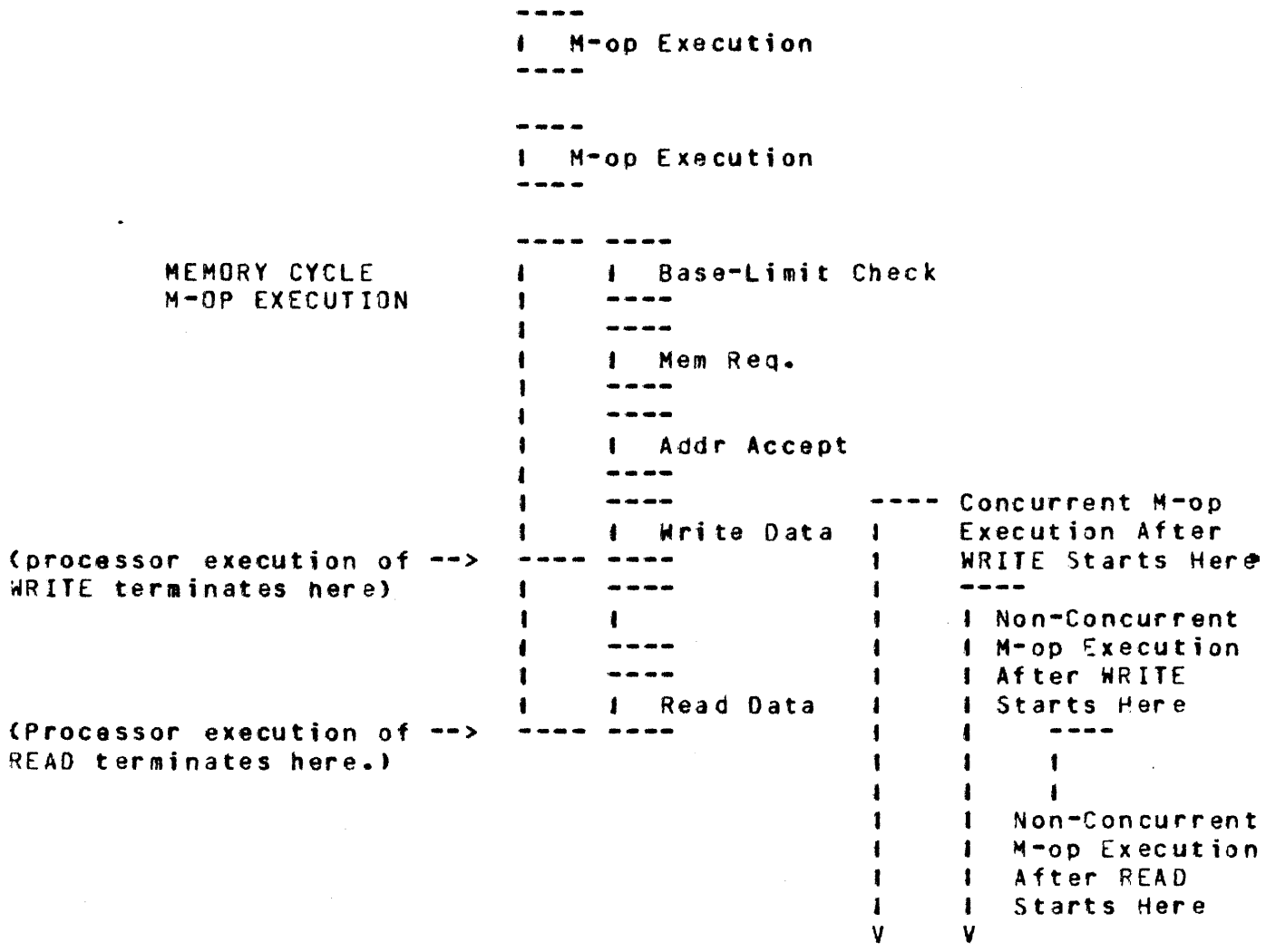
The M-instructions comprising the concurrent set are:

READ MEMORY*	COUNT FA/FL**	SCRATCHPAD RELATE FA
WRITE MEMORY*	BIAS**	SWAP F WITH DOUBLEPAD WORD**
SWAP MEMORY*	MONITOR**	STORE F INTO DOUBLEPAD WORD**
	BRANCH**	STORE F FROM DOUBLEPAD WORD**

* A memory cycle operation overlaps with another only during that portion of the cycle comprising the base-limit checking. The actual memory request is made during the clock period following acceptance of data.

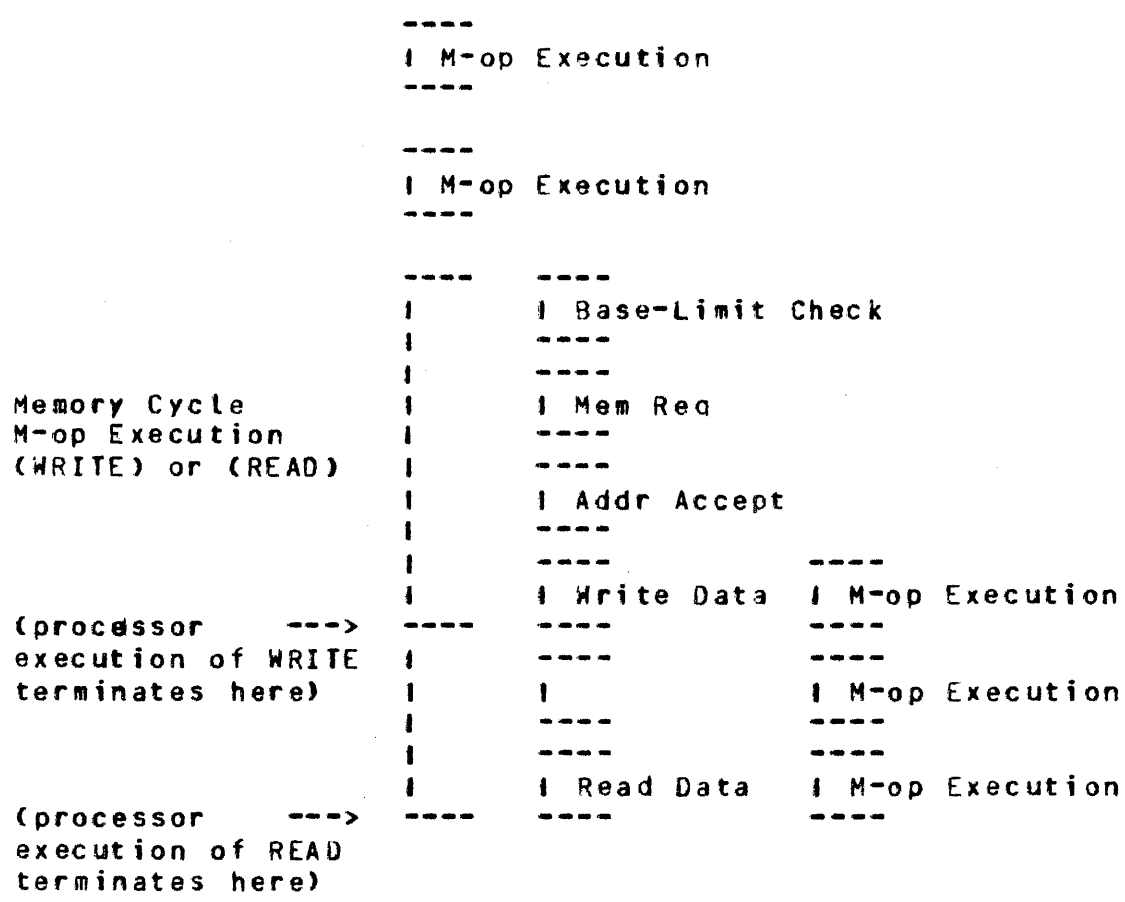
** A 1-clock concurrent micro-operator and the 2-clock branch micro-operator has a 1-clock NOP (no operation) placed after it by the hardware.

Timing diagrams showing execution times for consecutive processor M-instructions are shown below. The diagrams assume that the instructions are obtained from M-Memory and that the processor receives the memory cycle immediately after requesting it.



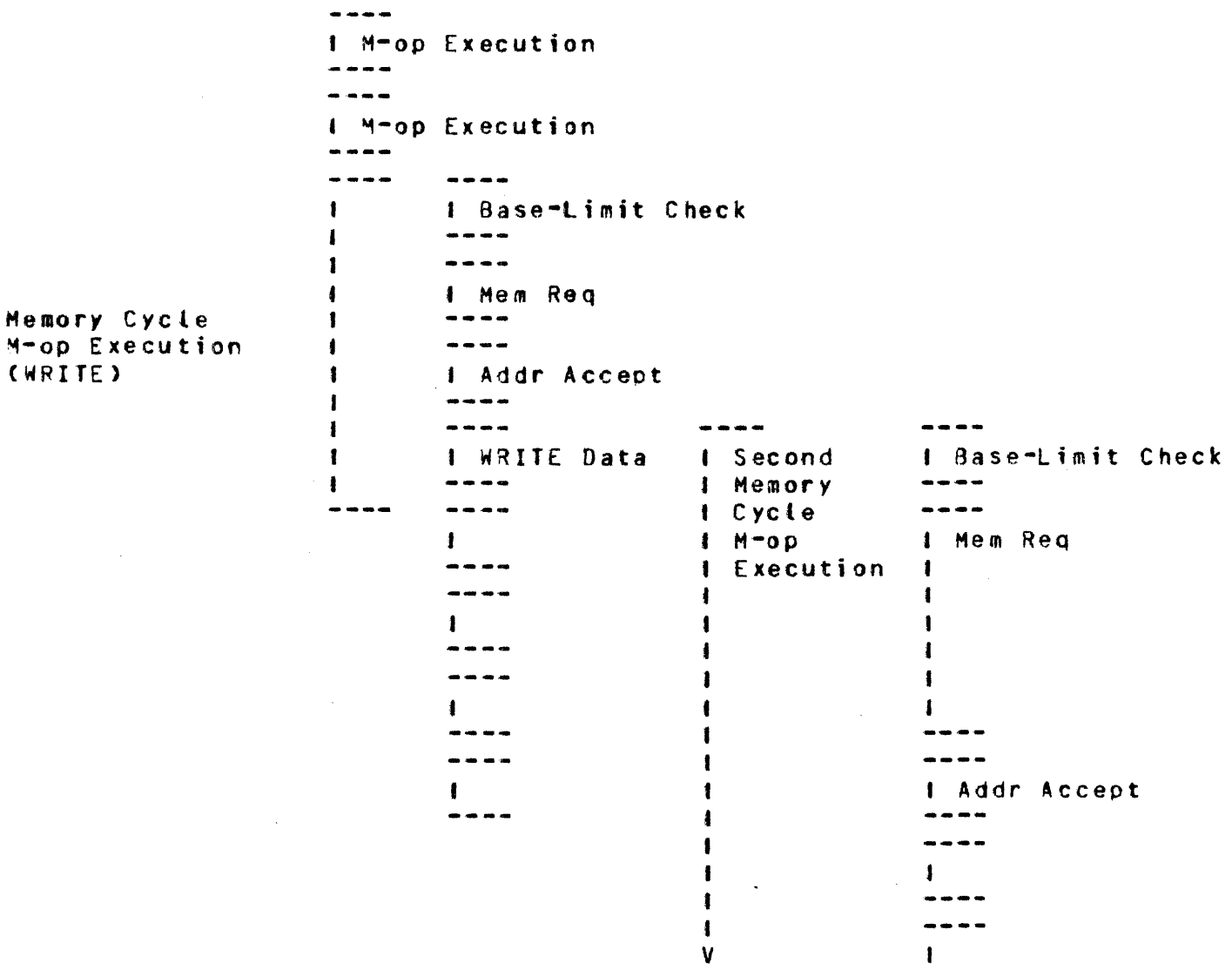
EXAMPLE ONE

(General)



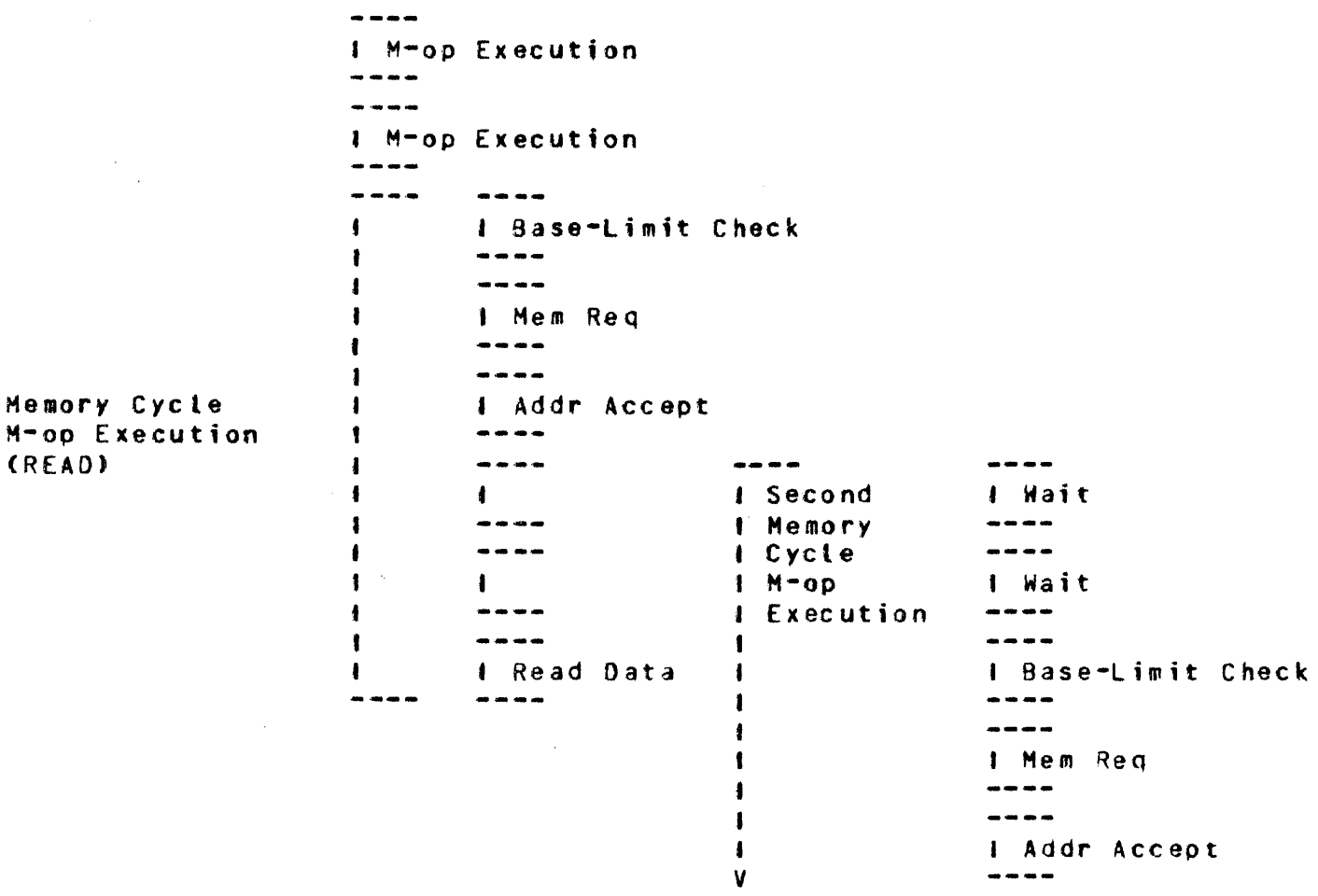
EXAMPLE TWO

(Concurrent M-op's overlapped with WRITE or READ Cycle)



EXAMPLE THREE

(2nd Memory Cycle overlapped with WRITE Cycle)



EXAMPLE FOUR

(2nd Memory Cycle overlapped with Read Cycle)

3.6 M-INSTRUCTION (Processor-1)

The following instruction times (execute plus fetch of next M-Instruction) are given for the case where the next M-Instruction is contained in M-Memory. In general, five clocks are added to the basic time if the M-instruction is fetched from S-Memory (main,memory).

M-Instruction	Clocks
REGISTER MOVE	1*
(MSM, SUM, DIFF as source)	2*
(DATA as source)	3*
(U as source)	U*
(M as destination with A out-of-bounds	1
*Add one additional clock if MSM, TOPM, A is destination of if MBR is destination with A out-of-bounds. Add two if DATA, CMND is destination.	
SCRATCHPAD MOVE	1*
(MSM, SUM, DIFF as source)	2
(Data as source)	3
(U as source)	u
(MSM, TOPM, A as destination)	2*
(DATA, CMND as destination)	3*
(MBR as destination with A out-of-bounds)	2*
(M as destination with A out-of-bounds)	1*
*Add one additional clock if previous op was WRITE INTO SCRATCHPAD	
SWAP F WITH DOUBLEPAD WORD	1
STORE F INTO DOUBLE WORD	1
LOAD F FROM DOUBLEPAD WORD	1
MOVE 8-BIT LITERAL	1
(A, MSM as destination)	2
MOVE 24-BIT LITERAL	3
(TAS as destination with A out-of-bounds)	8
(Others as destination with A out-of-bounds)	7
(TAPE mode)	u+1
SWAP MEMORY	
(Followed by Non-concurrent M-op)	10
(Followed by Memory Cycle)	9
(Followed by 7 concurrent M-op's)	3

M-Instruction	Clocks
WRITE MEMORY	
(Followed by non-concurrent M-op)	4
(Followed by Memory Cycle)	5
(Followed by 1 concurrent M-op)	3
(If outside base-limit & inhibited)	1
READ MEMORY	
(Followed by non-concurrent M-op)	6
(Followed by Memory cycle)	5
(Followed by 3 concurrent M-op's)	3
DISPATCH LOCKOUT	Same as READ MEMORY + 1
(Skip taken)	Same as READ MEMORY + 2
DISPATCH WRITE	Same as WRITE MEMORY
DISPATCH READ	Same as READ MEMORY + 1
DISPATCH READ AND CLEAR	Same as READ MEMORY + 1
DISPATCH PORT ABSENT	1
COUNT FA/FL	1
SCRATCHPAD RELATE FA	2
EXTRACT FROM REGISTER T	1
SHIFT/ROTATE REGISTER T LEFT	1
(A, IOPM, MSM as destinations)	2
(DATA, CMND as destinations)	3
(MBR as destination with A out-of-bounds)	2
SHIFT/ROTATE REGISTER X/Y L/R	1 plus S/R count
SHIFT/ROTATE REGISTER XY L/R	1 plus S/R count
NORMALIZE	1 plus # of bits shifted
READ/WRITE MSM	6
CALL	2
BRANCH	2
BIAS	1
(Skip taken)	2
SET CYF	1
4-BIT MANIPULATE	1
(Skip taken)	2
(BICN, FLCN, XYCN, XYST)	2
(BICN, FLCN, XYCN, XYST, and SKIP taken)	3

M-Instruction	Clocks
BIT TEST BRANCH FALSE	1
(Branch taken)	2
BIT TEST BRANCH TRUE	1
(Branch taken)	2
SKIP WHEN	1
(Skip taken)	2
(BICN, FLCN, XYCN, XYST)	2
(BICN, FLCN, XYCN, XYST and SKIP taken)	3
CLEAR REGISTER	1
BIND	3
Overlay M-Memory	4+(6)(# of words moved)
(FL = 0 initially)	1
CASSETTE CONTROL	1
HALT	1
NO OPERATION	1
Monitor	1

3.7 M-INSTRUCTION TIMING (Processor-2)

The following instruction times (execute plus fetch of next M-Instruction) are given for the case where the next M-Instruction is contained in M-Memory. In general, five or six clocks are added to the basic time if the next M-Instruction is fetched from S-Memory; five clocks for the direct-connect to S-Memory case and six clocks for the port-connect case. When consecutive M-Instructions are fetched from two contiguous S-Memory locations and the second location is located on an odd 16-bit boundary, one clock is added, for the second fetch, to the five (or six) required for the first fetch.

M-Instruction	Clocks
REGISTER MOVE	1*
(MSM, SUM, DIFF as source)	2*
(DATA as source)	3*
(U as source)	U*
(M as destination with A out-of-bounds)	1
*Add one additional clock if MSM, TOPM, A, is destination or if MBR is destination with A out-of-bounds. Add two if DATA, CMND is destination.	
SCRATCHPAD MOVE	1*
(MSM, SUM, DIFF as source)	2
(DATA as source)	3
(U as source)	u
(MSM, TOPM, A as destination)	2*
(DATA, CMND as destination)	3*
(MBR as destination and A out-of-bounds)	2*
(M as destination and A out-of-bounds)	1
*Add one additional clock if previous op was WRITE into SCRATCHPAD	
SWAP F WITH DOUBLEPAD WORD	1
STORE F INTO DOUBLEPAD WORD	1
LOAD F FROM DOUBLEPAD WORD	1
MOVE 8-BIT LITERAL	1
(A, MSM as destination)	2

M-Instruction	Clocks
MOVE 24-BIT LITERAL	3
(TAS as destination with A out-of-bounds)	3+ (5 or 6)
(Others as destination with A out-of-bounds)	2+ (5 or 6)
(TAPE mode)	U+1
(First portion of instruction aligned on even 16-bit S-Memory boundary)	3
(First portion of instruction aligned on odd 16-bit S-Memory boundary)	2+ (5 or 6)
SWAP (DIRECT CONNECT)	8**
SWAP (PORT CONNECT)	9**
WRITE (DIRECT CONNECT)	3*
WRITE (PORT CONNECT)	4*
READ (DIRECT CONNECT)	7
READ (PORT CONNECT)	8
*Add 4 clocks if next op is a memory cycle	
**Add 6 clocks if next op is a memory cycle	
COUNT FA/FL	1
SCRATCHPAD RELATE FA	2
EXTRACT FROM REGISTER T	1
SHIFT/ROTATE REGISTER T LEFT	1
(A, TOPM, MSM as destinations)	2
(DATA, CMND as destinations)	3
(MBR as destination with A out-of-bounds)	2
SHIFT/ROTATE REGISTER X/Y L/R	1 plus S/R count
SHIFT/ROTATE REGISTER XY L/R	1 plus S/R count
NORMALIZE	1 plus # of bits shifted
READ/WRITE MSM	6
CALL	2
BRANCH	2
BIAS	1
(Skip taken)	2
SET CYF	1
4-BIT MANIPULATE	1
(Skip taken)	2
(BICN, FLCN, XYCN, XYST)	2
(BICN, FLCN, XYCN, XYST and SKIP taken)	3

M-Instruction	Clocks
BIT TEST BRANCH FALSE	1
(Branch taken)	2
BIT TEST BRANCH TRUE	1
(Branch taken)	2
SKIP WHEN	1
(Skip taken)	2
(BICN, FLCN, XYCN, XYST)	2
(BICN, FLCN, XYCN, XYST and SKIP taken)	3
CLEAR REGISTERS	1
BIND	3
OVERLAY M-MEMORY (DIRECT CONNECT)	4+4 (# of words moved)*
OVERLAY M-MEMORY (PORT CONNECT)	4+4.5 (# of words moved)**
(FL = 0 initially)	1
*Add 2 if # of words is odd	
**Add 2.5 if # of words is odd	
CASSETTE CONTROL	1
HALT	1
NO OPERATION	1
MONITOR	1