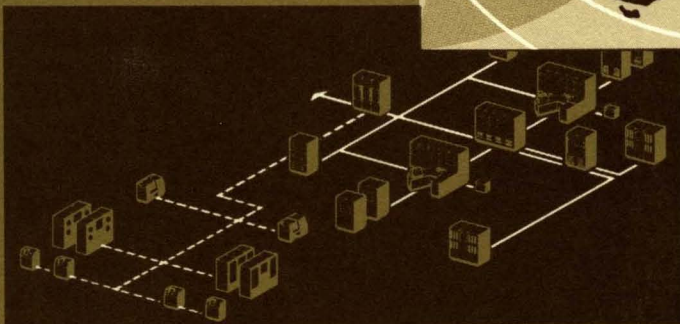
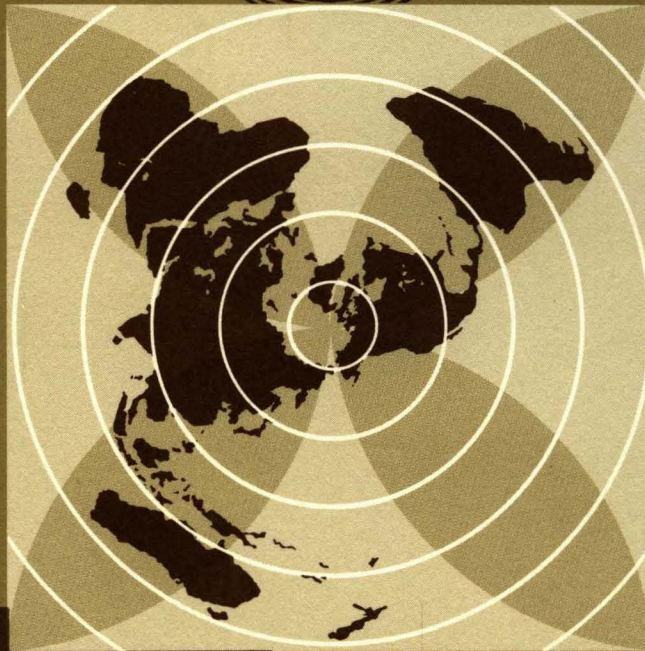
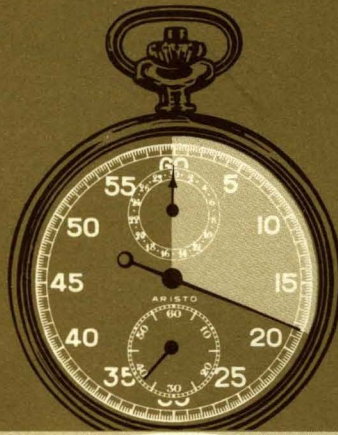


**PROGRAMMERS  
REFERENCE  
MANUAL**



**L-3055  
DATA PROCESSING  
SYSTEM**

**LIBRASCOPE DIVISION**

**GP GENERAL  
PRECISION**

**INFORMATION SYSTEMS GROUP**

808 WESTERN AVE • GLENDALE 1, CALIFORNIA

PROGRAMMER'S REFERENCE MANUAL  
FOR AN  
L-3055 DATA PROCESSING SYSTEM  
USED AS THE  
AN/FYQ-11 DATA PROCESSOR SET  
15 September 1963

General Precision, Inc.  
Information Systems Group  
Librascope Division  
808 Western Avenue  
Glendale 1, California

## LIST OF EFFECTIVE PAGES

Total number of pages in this publication is  
318 consisting of the following:

Page No.	Issue
Title	Original
A	Original
iii thru xvii	Original
1-0 thru 1-10	Original
2-1 thru 2-14	Original
3-1 thru 3-10	Original
4-1 thru 4-57	Original
5-1 thru 5-6	Original
6-1 thru 6-29	Original
7-1 thru 7-9	Original
8-1 thru 8-37	Original
9-1 thru 9-25	Original
10-1 thru 10-15	Original
11-1 thru 11-18	Original
12-1 thru 12-21	Original
13-1 thru 13-24	Original
14-1	Original
15-1 thru 15-5	Original
A-1 thru A-3	Original
B-1 thru B-4	Original
C-1 thru C-6	Original
D-1 thru D-3	Original
E-1 thru E-3	Original

\*The asterisk indicates pages changed,  
added or deleted by current change

## TABLE OF CONTENTS

Section I	INTRODUCTION	Page
1.1	SYSTEM FUNCTIONS . . . . .	1-1
1.2	SYSTEM FEATURES . . . . .	1-5
1.3	CENTRAL PROCESSOR . . . . .	1-6
1.4	CORE MEMORY. . . . .	1-6
1.5	DISC MEMORY . . . . .	1-7
1.6	REAL-TIME COMMUNICATION . . . . .	1-7
1.7	UNI-RECORD FACILITIES . . . . .	1-8
1.8	CHECKING FEATURES . . . . .	1-8
1.9	DUPLEXING . . . . .	1-9
1.10	PROGRAM INTERRUPTS . . . . .	1-9
Section II	CORE MEMORY AND WORK FORMAT	
2.1	MEMORY REGISTERS . . . . .	2-1
2.1.1	Memory Address Register . . . . .	2-1
2.1.2	Memory Access Register (M) . . . . .	2-1
2.2	SPECIAL FEATURES . . . . .	2-2
2.2.1	Cycle Demand Priority . . . . .	2-2
2.2.2	Individual Memory Trunkline . . . . .	2-2
2.2.3	Shared Memory Trunkline . . . . .	2-2
2.2.4	Module Address Patchboard . . . . .	2-3
2.2.5	Core Memory Twinning. . . . .	2-5
2.3	INFORMATION FORMAT. . . . .	2-5
2.3.1	Word . . . . .	2-5
2.3.2	Character . . . . .	2-6
2.3.3	Bit. . . . .	2-6
2.3.4	Alphanumeric Data Word . . . . .	2-7
2.3.5	Floating-Point Data Word . . . . .	2-8
2.4	INSTRUCTION WORD FORMAT . . . . .	2-9
2.4.1	Command Character (C) . . . . .	2-10
2.4.2	Select Characters (ZY) . . . . .	2-10
2.4.3	Index Character (X) . . . . .	2-11
2.4.4	Memory Address Characters (MMMM) . . . . .	2-12

## TABLE OF CONTENTS - Continued

	Page
Section II (Cont'd)	
2.4.5 Zone Bit Options . . . . .	2-12
Section III	
CENTRAL PROCESSOR	
3.1 ARITHMETIC SECTION . . . . .	3-1
3.1.1 R Register . . . . .	3-1
3.1.2 A Register . . . . .	3-1
3.1.3 Q Register . . . . .	3-2
3.1.4 Toggles . . . . .	3-2
3.1.5 Adder . . . . .	3-2
3.2 PROGRAM CONTROL SECTION . . . . .	3-2
3.2.1 Instruction Address Register (Ia) . . . . .	3-3
3.2.2 Operand Address Register (Oa) . . . . .	3-3
3.2.3 Index Holder (Xh) . . . . .	3-4
3.2.4 Command Holder (C) . . . . .	3-4
3.2.5 Z Holder (Z) . . . . .	3-4
3.2.6 Y Holder (Y) . . . . .	3-5
3.2.7 X Counter (X) . . . . .	3-5
3.2.8 Index Modify Register (B) . . . . .	3-5
3.2.9 Index Registers (Xr1-Xr17) . . . . .	3-5
3.2.10 Toggles and Switches . . . . .	3-6
3.3 PROGRAM CONTROL SECTION OPERATION . . . . .	3-6
3.3.1 MASM Operand Address Store . . . . .	3-6
3.3.2 Flagged Return Address . . . . .	3-7
3.3.3 Indirect Address . . . . .	3-7
3.3.4 PDPM Operation . . . . .	3-7
3.3.5 Index Modify . . . . .	3-8
3.4 PUSH-DOWN POP-UP MEMORY MODE (PDPM) . . . . .	3-8
3.5 MULTI-ADDRESS STACKING MEMORY MODE (MASM) . . . . .	3-9
Section IV	
INSTRUCTIONS	
4.1 ARITHMETIC INSTRUCTIONS . . . . .	4-2
4.1.1 Add . . . . .	4-4
4.1.2 Subtract . . . . .	4-6

## TABLE OF CONTENTS - Continued

Section IV (Cont'd)	Page
4.1.3 Add A to M, Place in R. . . . .	4-7
4.1.4 Subtract A from M, Place in R. . . . .	4-9
4.1.5 Multiply Long . . . . .	4-11
4.1.6 Multiply Short . . . . .	4-12
4.1.7 Divide Long . . . . .	4-13
4.1.8 Divide Short . . . . .	4-14
4.2 DATA MOVE INSTRUCTIONS. . . . .	4-16
4.2.1 Bring M to A. . . . .	4-17
4.2.2 Combine M with A . . . . .	4-18
4.2.3 Hold A to M . . . . .	4-19
4.2.4 Jerk M to R . . . . .	4-19
4.2.5 Keep R to M . . . . .	4-20
4.2.6 Load M in ZY . . . . .	4-21
4.2.7 Save ZY in M . . . . .	4-22
4.2.8 Copy Z to Y . . . . .	4-23
4.2.9 Left Shift . . . . .	4-25
4.2.10 Right Shift . . . . .	4-26
4.3 COMPARE INSTRUCTIONS . . . . .	4-28
4.3.1 Alphabetic Compare . . . . .	4-30
4.3.2 Numeric Compare . . . . .	4-31
4.3.3 Field Compare, Automatic Shift . . . . .	4-32
4.3.4 Index Register Compare . . . . .	4-33
4.4 SET-CONVERT INSTRUCTIONS . . . . .	4-33
4.4.1 Set-Reset Toggle . . . . .	4-34
4.4.2 Set-Reset Bit . . . . .	4-35
4.4.3 Logical OR . . . . .	4-35
4.4.4 Logical AND . . . . .	4-36
4.4.5 Convert . . . . .	4-36
4.5 PROGRAM CONTROL INSTRUCTIONS. . . . .	4-37
4.5.1 No Operation. . . . .	4-38
4.5.2 Unconditional Transfer . . . . .	4-38

TABLE OF CONTENTS - Continued

Section IV (Cont'd)	Page
4.5.3 Test . . . . .	4-38
4.5.4 Bit Compare . . . . .	4-39
4.5.5 Execute . . . . .	4-40
4.5.6 Modify Index Register. . . . .	4-40
4.5.7 Halt. . . . .	4-41
4.6 FLOATING-POINT INSTRUCTIONS. . . . .	4-41
4.6.1 Floating-Point Add. . . . .	4-43
4.6.2 Floating-Point Subtract. . . . .	4-45
4.6.3 Floating-Point Add A to M and place in R. . . . .	4-48
4.6.4 Floating-Point Subtract A from M and Place in R . . . . .	4-50
4.6.5 Floating-Point Multiply Long . . . . .	4-53
4.6.6 Floating-Point Divide Long . . . . .	4-55
4.6.7 Floating-Point Numeric Compare . . . . .	4-56
Section V I/O INTERFACE	
5.1 I/O BUFFER REGISTER . . . . .	5-1
5.2 CHARACTER HOLDER . . . . .	5-1
5.3 SCANNER. . . . .	5-1
5.4 I/O INSTRUCTION REGISTER . . . . .	5-2
5.5 I/O INSTRUCTION LOCATION REGISTER . . . . .	5-2
5.6 OUTPUT OPERATION . . . . .	5-2
5.7 INPUT OPERATION . . . . .	5-3
5.8 ERROR CONSIDERATION . . . . .	5-4
5.9 STATUS INDICATORS AND DEVICE ADDRESSES . . . . .	5-5
5.10 I/O INTERFACE INSTRUCTION FORMAT . . . . .	5-6
5.10.1 Input . . . . .	5-6
5.10.2 Output. . . . .	5-6
Section VI BUFFER PROCESSOR	
6.1 BUFFER PROCESSOR CONFIGURATION . . . . .	6-1
6.2 INPUT MESSAGE TRANSFER . . . . .	6-1
6.2.1 Data Link to Receiving L-119 . . . . .	6-1
6.2.2 Receiving L-119 to Central Processor . . . . .	6-4





TABLE OF CONTENTS - Continued

Section VII (Cont'd)	Page
7.8.3 Test . . . . .	7-8
7.9 TAPE TRANSPORT CONTROL INSTRUCTIONS . . . . .	7-8
7.9.1 Backspace Tape -( . . . . .	7-8
7.9.2 Forward-Space Tape -) . . . . .	7-8
7.9.3 Rewind Tape -[ . . . . .	7-9
7.9.4 Write File Mark - 7 . . . . .	7-9
Section VIII	UNI-RECORD CONSOLES
8.1 TAPE TRANSPORTS . . . . .	8-2
8.1.1 Tape Transport On-Line Read . . . . .	8-3
8.1.2 Tape Transport On-Line Write . . . . .	8-5
8.1.3 Tape Transport Mode Change Operation . . . . .	8-6
8.1.4 Tape Transport Control Operations . . . . .	8-7
8.1.5 Tape Transport Off-Line Operations . . . . .	8-8
8.1.6 Tape Transport Operation . . . . .	8-9
8.1.7 Tape Transport Operating Procedures . . . . .	8-12
8.2 CARD READER/PUNCH . . . . .	8-16
8.2.1 Console Description . . . . .	8-16
8.2.2 Card Reader . . . . .	8-17
8.2.3 Card Punch . . . . .	8-24
8.3 LINE PRINTER . . . . .	8-29
8.3.1 Line Printer On-Line . . . . .	8-30
8.3.2 Line Printer Off-Line . . . . .	8-34
Section IX	DISC MEMORY
9.1 ORGANIZATION OF DATA . . . . .	9-2
9.1.1 Character . . . . .	9-2
9.1.2 Block . . . . .	9-2
9.1.3 Data-Band . . . . .	9-2
9.1.4 Twin . . . . .	9-4
9.1.5 Flag Tracks . . . . .	9-4

TABLE OF CONTENTS - Continued

Section IX (Cont'd)	Page
9.2 DISC MEMORY INSTRUCTION FORMAT . . . . .	9-5
9.2.1 The Interface Construction Word . . . . .	9-5
9.2.2 Disc Memory Operation Word . . . . .	9-6
9.2.3 Key/Data Address and Mask-Control Address Word . . . . .	9-6
9.2.4 Key Format . . . . .	9-7
9.2.5 Mask and Control Format . . . . .	9-8
9.3 DISC MEMORY COMPONENTS . . . . .	9-9
9.3.1 Disc Storage . . . . .	9-9
9.3.2 Core Buffer and Search Registers . . . . .	9-9
9.3.3 Disc Memory Operation Register . . . . .	9-9
9.3.4 MC Buffer Store . . . . .	9-10
9.3.5 CP Buffer Store . . . . .	9-11
9.4 DISC MEMORY INTERFACE . . . . .	9-11
9.4.1 Interface Instruction Register . . . . .	9-12
9.4.2 Interface Instruction Location Register . . . . .	9-12
9.4.3 Interface Address Processing Register . . . . .	9-12
9.4.4 Disc Memory Interface Operation . . . . .	9-12
9.5 DISC MEMORY OPERATION . . . . .	9-13
9.5.1 Write Function . . . . .	9-14
9.5.2 Read Function . . . . .	9-14
9.5.3 Obsolete Function . . . . .	9-15
9.5.4 Count Function . . . . .	9-15
9.5.5 Fixed Address Access . . . . .	9-15
9.5.6 Obsolete Access . . . . .	9-16
9.5.7 Content Access . . . . .	9-16
9.5.8 Compare Flag Access . . . . .	9-20
9.5.9 Multi-Block, First Qualify Access . . . . .	9-20
9.6 TWIN MODE . . . . .	9-22
9.7 STATUS TOGGLES . . . . .	9-22
9.7.1 Overload . . . . .	9-22
9.7.2 Twin Compare (ZY=/S) . . . . .	9-22



TABLE OF CONTENTS - Continued

Section X (Cont'd)	Page
10.4.2 CP Buffer Write . . . . .	10-11
10.5 PREPARE OPERATIONS . . . . .	10-11
10.5.1 Select Disc Memory Module . . . . .	10-11
10.5.2 Load Key, Mask and Control . . . . .	10-12
10.6 RETURN OPERATIONS . . . . .	10-12
10.6.1 Return Disc Memory Operation . . . . .	10-12
10.6.2 Return Key, Mask and Control Information . . .	10-13
10.6.3 Return Display Buffer Status . . . . .	10-13
 Section XI                                   PROGRAM INTERRUPTS	
11.1 INTERRUPT HIERARCHY . . . . .	11-1
11.2 IGNORE INTERRUPT TOGGLES . . . . .	11-2
11.3 INTERRUPT OPERATION . . . . .	11-2
11.4 INTERRUPT PROGRAM SELECTION . . . . .	11-3
11.5 INTERRUPT PROGRAM EXIT . . . . .	11-4
11.6 FLAGGED RETURN ADDRESSES . . . . .	11-4
11.7 INTERRUPT CLASS: I/O INTERFACE (ZY=42) . . . . .	11-4
11.7.1 Detail: Interface Not Busy (ZY=39) . . . . .	11-4
11.7.2 Detail: L-119 Message Available (ZY=2,) . . . .	11-5
11.7.3 Detail: Display Console Message Available (ZY=2#) . . . . .	11-5
11.7.4 Detail: Sending L-119 Available (ZY=2()) . . . .	11-5
11.8 INTERRUPT CLASS: REAL-TIME (ZY=43) . . . . .	11-5
11.8.1 Detail: Real-Time Clock (ZY=3()) . . . . .	11-5
11.8.2 Detail: Interval Timer (ZY=3,) . . . . .	11-5
11.9 INTERRUPT CLASS (AND DETAIL): OTHER PROCESSOR (ZY=44) . . . . .	11-5
11.10 INTERRUPT CLASS: DISC MEMORY INTERFACE (ZY=3[) . . . . .	11-5
11.10.1 Detail: Interface Not Busy (ZY=2Y) . . . . .	11-6
11.10.2 Detail: Disc Memory Unit Not Busy (ZY=2Z) . .	11-6
11.10.3 Detail: Display (ZY=35) . . . . .	11-6

TABLE OF CONTENTS - Continued

Section XI (Cont'd)	Page
11.11 INTERRUPT CLASS: UNI-RECORD INTERFACE (ZY=3W) . . . . .	11-6
11.11.1 Detail: Interface Not Busy (ZY=3Y) . . . . .	11-6
11.11.2 Detail: Operator (ZY=3Z) . . . . .	11-6
11.12 INTERRUPT CLASS: ERROR (ZY=41) -(COMPUTER, DATA TRANSFER) . . . . .	11-6
11.12.1 Detail: M Register Parity Error: (ZY=V1) . . . . .	11-7
11.12.2 Detail: R Register (ZY=V2) . . . . .	11-7
11.12.3 Detail: A Register (ZY=V3) . . . . .	11-7
11.12.4 Detail: Q Register (ZY=V4) . . . . .	11-7
11.12.5 Detail: Arithmetic (ZY=V5) . . . . .	11-7
11.12.6 Detail: R Repeater (ZY=V6) . . . . .	11-7
11.13 INTERRUPT CLASS: ERROR (ZY=41) -(COMPUTER, ARITHMETIC) . . . . .	11-7
11.13.1 Detail: Overflow (ZY=W1) . . . . .	11-8
11.13.2 Detail: F. P. Overflow (ZY=W2) . . . . .	11-8
11.13.3 Detail: Numerical (ZY=W3) . . . . .	11-8
11.13.4 Detail: Convert Overflow (ZY=W4) . . . . .	11-8
11.14 INTERRUPT CLASS: ERROR (ZY=41) -(COMPUTER, PROGRAM) . . . . .	11-8
11.14.1 Detail: Instruction (ZY=X1) . . . . .	11-8
11.14.2 Detail: Illegal Address (ZY=X2) . . . . .	11-9
11.14.3 Detail: End of Memory (ZY=X3) . . . . .	11-9
11.15 INTERRUPT CLASS: Error (ZY=41) - INTERFACE, DISC MEMORY) . . . . .	11-9
11.15.1 Detail: Vertical Parity Interval (ZY=/1) . . . . .	11-9
11.15.2 Detail: Horizontal Check Interval (ZY=/2) . . . . .	11-10
11.15.3 Detail: Flag Parity (ZY=/3) . . . . .	11-10
11.15.4 Detail: Fixed Address (ZY=/4) . . . . .	11-12
11.15.5 Detail: M Register (ZY=/5) . . . . .	11-12
11.15.6 Detail: Write (ZY=/6) . . . . .	11-12

TABLE OF CONTENTS - Continued

Section XI (Cont'd)	Page
11.15.7 Detail: Instruction (ZY=/7) . . . . .	11-13
11.15.8 Detail: Hang Up (ZY=/8) . . . . .	11-14
11.15.9 Detail: Transmission Parity (ZY=/9) . . . . .	11-14
11.15.10 Detail: Out of Temperature (ZY=/=) . . . . .	11-15
11.15.11 Detail: Slow Down (ZY=/') . . . . .	11-15
11.15.12 Detail: Operand Address (ZY=/>) . . . . .	11-15
11.15.13 Detail: Count (ZY=36) . . . . .	11-15
11.16 INTERRUPT CLASS: ERROR (ZY=41) - (INTERFACE, I/O) . . . . .	11-15
11.16.1 Detail: Parity (ZY=S1) . . . . .	11-15
11.16.2 Detail: M Register (ZY=S2) . . . . .	11-16
11.16.3 Detail: Hang Up (ZY=S3) . . . . .	11-16
11.16.4 Detail: Operand Address (ZY=S4) . . . . .	11-16
11.17 INTERRUPT CLASS: ERROR (ZY=41) - (INTERFACE, UNI-RECORD) . . . . .	11-16
11.17.1 Detail: On Line Data (ZY=T1) . . . . .	11-16
11.17.2 Detail: On Line Evesdrop (ZY=T2) . . . . .	11-16
11.17.3 Detail: On Line Mechanical (ZY=T3) . . . . .	11-17
11.17.4 Detail: M Register Parity Error (ZY=T4) . . . . .	11-17
11.17.5 Detail: Horizontal Check (ZY=T5) . . . . .	11-17
11.17.6 Detail: Instruction (ZY=T6) . . . . .	11-17
11.17.7 Detail: Overload (ZY=T7) . . . . .	11-17
11.17.8 Detail: Real World Parity (ZY=T8) . . . . .	11-17
11.17.9 Detail: Real World Mechanical (ZY=T9) . . . . .	11-17
11.17.10 Detail: Device Unavailable (ZY=T>) . . . . .	11-17
11.17.11 Detail: Operand Address (ZY=T<) . . . . .	11-18
Section XII	
DUPLEXING	
12.1 BUFFER PROCESSOR SWITCHING . . . . .	12-1
12.1.1 Receiving L-119 Module Controls . . . . .	12-1
12.1.2 Sending L-119 Module Controls . . . . .	12-4
12.1.3 L-119 Module Pair Controls . . . . .	12-4

TABLE OF CONTENTS - Continued

Section XII (Cont'd)	Page
12.1.4 Receiving L-119 Module Indicators . . . . .	12-5
12.1.5 Sending L-119 Module Indicators . . . . .	12-6
12.1.6 Testable L-119 Module Control Switches . . . . .	12-7
12.1.7 L-119 Module Operation from the Central Switching Console . . . . .	12-8
12.1.8 Audible Alarm . . . . .	12-9
12.1.9 Test Indicators Button . . . . .	12-10
12.2 AUTODIN REMOTE CONTROLS . . . . .	12-10
12.2.1 Sending Crypto Controls . . . . .	12-10
12.2.2 Receiving Crypto Controls . . . . .	12-11
12.2.3 Control Switch and Indicator Power . . . . .	12-11
12.2.4 Operation of Autodin Controls . . . . .	12-11
12.3 INTEGRATED CONSOLE SWITCHING . . . . .	12-13
12.3.1 Electronic Typewriter Switching . . . . .	12-13
12.3.2 Display and Console Printer Switching . . . . .	12-14
12.4 DISC MEMORY TRUNKLINE . . . . .	12-20
12.4.1 Control Status Toggle . . . . .	12-20
12.4.2 Disc Memory Trunkline Coupling Mode Toggle . . . . .	12-20
12.4.3 Other Disc Memory Busy . . . . .	12-20
Section XIII CONTROL CONSOLE AND CENTRAL PROCESSOR CONTROL PANEL	
13.1 CONTROL CONSOLE CONTROL PANEL . . . . .	13-1
13.2 CONTROL TYPEWRITER . . . . .	13-6
13.2.1 Manual Input . . . . .	13-6
13.2.2 Typewriter Instructions . . . . .	13-6
13.3 CENTRAL PROCESSOR CONTROL PANEL . . . . .	13-7
13.3.1 Left Section Indicators . . . . .	13-7
13.3.2 Center Section Indicators . . . . .	13-14
13.3.3 Right Section Controls and Indicators . . . . .	13-21

TABLE OF CONTENTS - Continued

Section XIV	REAL-DASH TIME CLOCK AND INTERVAL TIMER	Page
14.1	REAL-DASH TIME CLOCK . . . . .	14-1
14.2	INTERVAL TIMER . . . . .	14-1
Section XV	DISPLAY CONSOLES	
15.1	ELECTRONIC TYPEWRITER ET . . . . .	15-2
	15.1.1 Input Messages . . . . .	15-2
	15.1.2 Output Messages . . . . .	15-3
15.2	LOGIC KEYBOARD ASSEMBLY LKB . . . . .	15-4
15.3	MULTI COLOR DISPLAY MC . . . . .	15-4
15.4	CONSOLE PRINTER CP . . . . .	15-5
15.5	HARDCOPY ASSEMBLY HC . . . . .	15-5



## ILLUSTRATIONS

Figure		Page
1-1	Data Processor Set AN/FYQ-11 . . . . .	1-0
1-2	AN/FYQ-11 Data Flow and Switching Control Diagram . . .	1-3
2-1	Memory Address Selection Patchboard . . . . .	2-4
2-2	Information Word Format . . . . .	2-6
2-3	Instruction Word Format . . . . .	2-10
6-1	Buffer Processor Console Functional Configuration . . . . .	6-2
6-2	C or T Select Message Format . . . . .	6-3
6-3	Single Block Input Message Format . . . . .	6-5
6-4	Multi-Block Input Message Format . . . . .	6-6
6-5	Single Block Output Message Format . . . . .	6-8
6-6	Multi-Block Output Message Format . . . . .	6-9
6-7	L-119 Block Diagram . . . . .	6-12
6-8	Buffer Processor Maintenance Panel . . . . .	6-26
7-1	Uni-Record Device ZY Addresses . . . . .	7-5
8-1	Magnetic Tape Console Control Panel . . . . .	8-10
8-2	Card Flow Diagram . . . . .	8-18
8-3	Card Reader/Punch Control Panel . . . . .	8-25
8-4	Line Printer Control Panel . . . . .	8-35
9-1	Index-Data-Fixed Address-Flag Format . . . . .	9-3
9-2	Content Access - Example . . . . .	9-21
10-1	Display Buffer Status Word . . . . .	10-15
11-1	Interrupt Hierarchy . . . . .	11-19
12-1	System Switching Configuration . . . . .	12-2
12-2	Central Switching Console Control Panel, Center Section .	12-3

ILLUSTRATIONS - Continued

Figure		Page
12-3	Central Switching Console Control Panel, Right Section . .	12-16
12-4	Central Switching Console Control Panel, Left Section . . .	12-17
12-5	Switching Status Work (ZY17) . . . . .	12-19
13-1	Control Console Control Panel . . . . .	13-2
13-2	Central Processor Control Panel, Left Section . . . . .	13-8
13-3	Central Processor Control Panel, Center Section . . . . .	13-15
13-4	Central Processor Control Panel, Right Section . . . . .	13-22
E-1	Central Processor Functional Block Diagram . . . . .	E-3



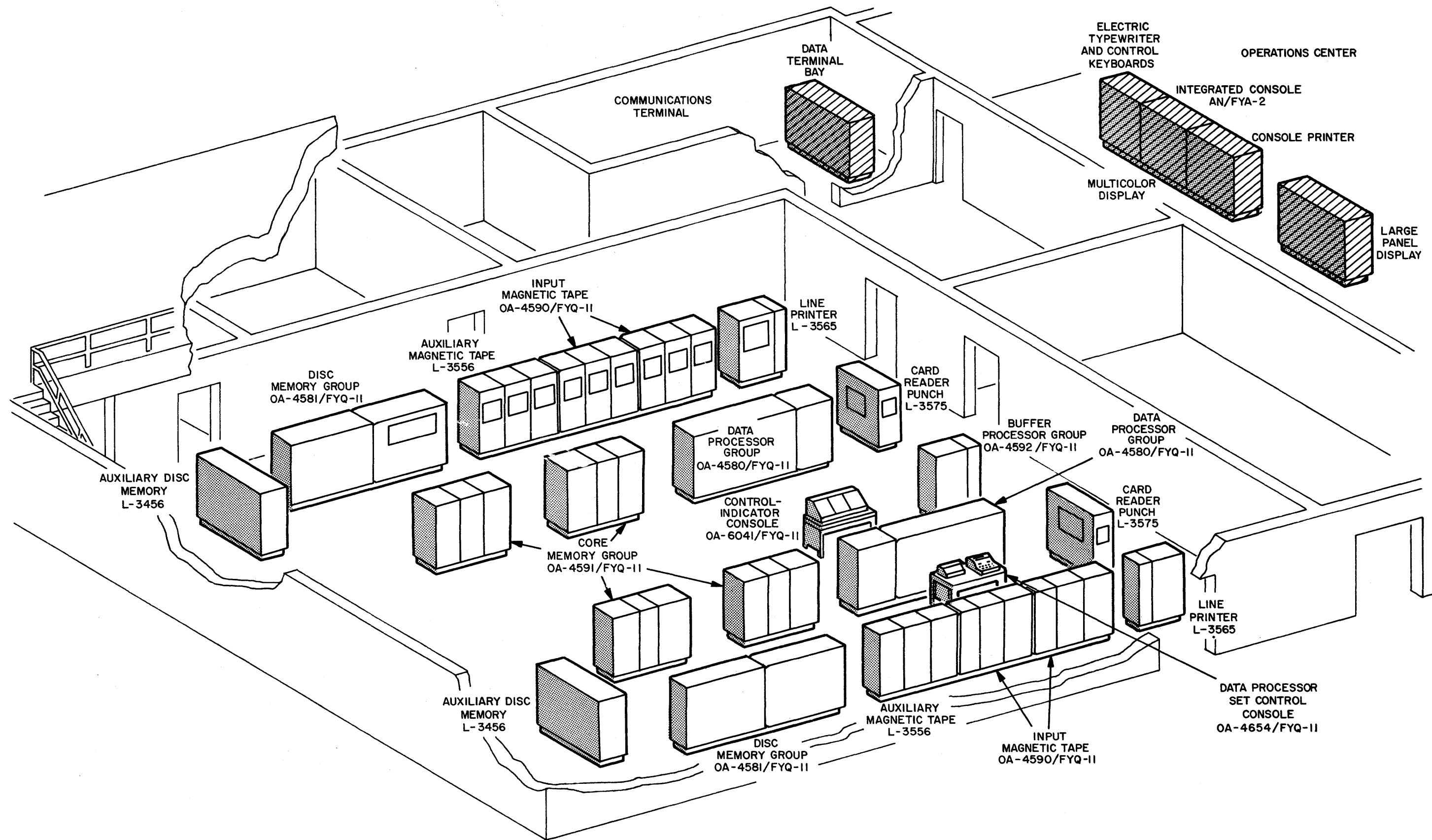


Figure 1-1. Data Processor Set AN/FYQ-11

## SECTION I

### INTRODUCTION

This publication is the Programmers Reference Manual for an L-3055 Data Processing System used as the Data Processor Set AN/FYQ-11, figure 1-1, a subsystem of the 473L Command and Control System. Content of the manual is orientated around system functional information that is needed for programming. This manual consists of 15 sections that are bound in one volume.

The system described is the Complete Operating Capability (COC) for Data Processor Set AN/FYQ-11. The Data Processing System operates with the Integrated Console Subsystem, Large Panel Display Subsystem and Data Communications Subsystem (AUTODIN) to provide large scale automatic processing, display and transmission of information. On-line and off-line magnetic tape, card reader/punch, and line printer operations are provided by the system.

#### 1.1 SYSTEM FUNCTIONS.

The Data Processing System consists of on-line, real-time general purpose alphanumeric data processing equipment that performs arithmetic, logical, storage, control, and retrieval operations on various inputs and stored data to produce outputs under program, operator, and automatic control. This data processing equipment maintains control of data by acquisition, filing, distribution, transmission, and monitoring.

The Data Processing System provides for inputs through digital data links, integrated console typewriters, a control console, a card reader, and magnetic tapes. Outputs are provided through integrated console typewriters, integrated console printer, integrated console displays, large panel displays, line printer, control console typewriter, card punch, magnetic tapes, and digital data links.

The basic functions of input-output, data processing and storage are performed by the four major elements of the system:

- a. Central Processor, Control Console, and Core Memory.
- b. Buffer Processors for real-time inputs from digital data links, and outputs to digital data links.

Section I  
Paragraph 1.1

c. Uni-Record devices consisting of Magnetic Tape Consoles, Card Reader/Punch, and Line Printer.

d. Disc Memory for fixed address or content-access operations and integrated console buffer storage.

Equipment items of the Data Processing System are listed in table 1-1.

Table 1-1. L-3055 Data Processing Equipment  
Used as Data Processor Set AN/FYQ-11

IOC	COC	NOMENCLATURE
1	2	Data Processor Group OA-4580/FYQ-11 (Central Processor L-3155)
1	2	Disc Memory Group OA-4581/FYQ-11 (Disc Memory L-3455)
1	1	Buffer Processor Group OA-4592/FYQ-11 (Buffer Processor Console, four L-119 Modules)
2	4	Input Magnetic Tape OA-4590/FYQ-11 (Magnetic Tape Console L-3555)
1	4	Core Memory Group OA-4591/FYQ-11 (Core Memory L-3755)
1	2	Console, Data Processor Set Control OA-4654/FYQ-11 (Control Console L-3855)
0	1	Console, Control-Indicator OA-6041/FYQ-11 (Central Switching Console L-3655)
1	2	Card Reader/Punch L-3575
1	2	Line Printer L-3565
1	2	Auxiliary Disc Memory L-3456
1	2	Auxiliary Magnetic Tape L-3556

Data furnished by Digital Data Links is available for computation, retrieval, and display under automatic control of stored programs and upon manual request. The system performs computations on the data, arranges the format, updates stored data, and presents the result to Integrated Consoles, Large Panel Displays, or

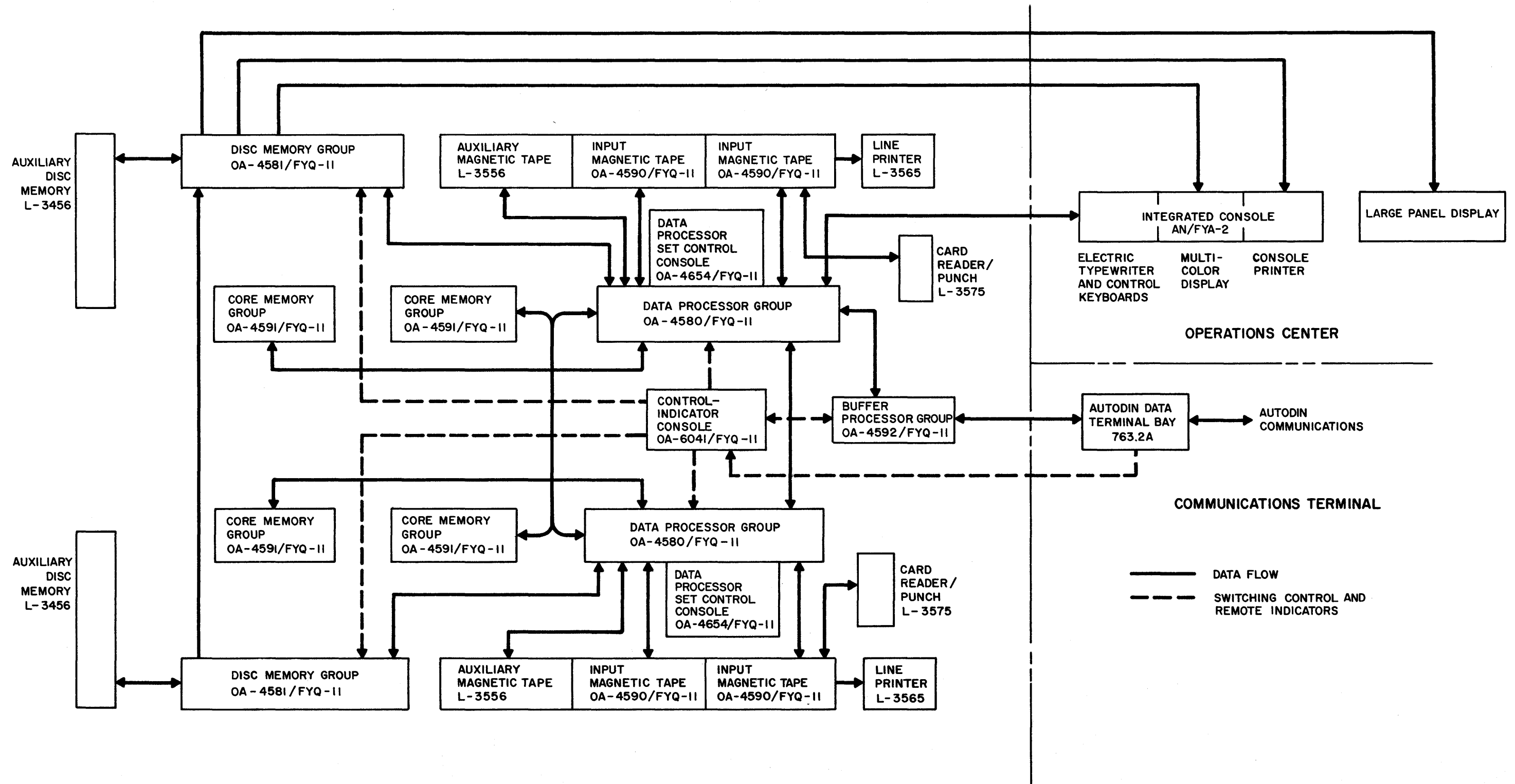


Figure 1-2. AN/FYQ-11 Data Flow and Switching Control Diagram

Digital Data Links. Programs control the execution of the functions required for the operational tasks. A control program supervises the reception of messages which are to be decoded and processed for transmission or display.

Rapid processing and retrieval of a wide variety of information as well as the execution of specific programs requires close interaction between manpower and the system. Considerable scheduling and system control is necessary to establish priorities, eliminate unnecessary delays, and perform the processing operations on a dynamic real-time basis. Characteristics of the units, the data peripheral devices, and programming must be understood by personnel responsible for the programming and operation.

The Data Processing System communicates electronically with units of the system and peripheral equipment by means of trunklines, buffers, and interfaces, figure 1-2. The Disc Memory Trunkline provides for communication between the Central Processor and Disc Memory. The Uni-Record Trunkline provides for data transfer and control between the Central Processor and the Magnetic Tape Console. The Tape Transports, Line Printer, Card Punch, and the Card Reader are the Uni-Record devices which are addressed by appropriate input or output instructions. The I/O Interface provides for data transfer between the Buffer Processor or an Integrated Console Electronic Typewriter, and the Central Processor. The I/O Interface in the Central Processor selects the particular Buffer Processor Module or Integrated Console Typewriter, and designates the operation to be performed.

## 1.2 SYSTEM FEATURES.

The Central Processor operates in parallel on words of eight alphanumeric characters. The magnetic Core Memory is modularly expandable up to 64,000 words. The Disc Memory, utilizing large discs, offers both fixed address and content access. The real-time interface provides time shared memory operation, as well as buffering, and separately programmed data-link operation. A Uni-Record Trunkline handles magnetic tapes, punched card equipment and a line printer.

All data transfers are parity checked, with provision for programmed error recovery. Multiplexing of all major system elements permits maintenance with reduction, but not loss, of capability.

The program interrupt feature gives the necessary response to real-time data transfer demands. Data transfers, both input/output and with Disc Memory, have their completion signaled to the program by an interrupt.



Section I  
Paragraphs 1.3 to 1.4

The entire Data Processing System has been organized to provide ease of programming, speed of operation, reliability, and general purpose flexibility.

1.3 CENTRAL PROCESSOR.

The Central Processor has an alphanumeric character organization with decimal memory addressing. The computer word of eight characters provides variable field flexibility with fixed word simplicity. Single address instructions specify a required field within any memory or register word for most operations.

There are 43 basic commands, including input and output. Control characters and flags within the instruction word extend the range of program operations. Automatic index modification is provided with all commands that specify an operand or next-instruction address. Two Index registers are static and 15 more are in the Core Memory. Indirect addressing and indirect instruction execution, with index modification, is provided.

The Central Processor may use the Core Memory in either of two special modes, PDPM (Push-Down, Pop-Up Memory), and MASM (Multi-Address Stacking Mode). Programmed memory comparisons may range over regions of Core Memory word by word, or on a character field basis, independent of word boundaries.

The Central Processor operates at a clock frequency of 1.5 megacycles. Data transfers with the Core Memory are full-word parallel, while most other operations between registers (including addition) are in half-word parallel. Typical operation times including instruction and operand access are: addition (non-indexed) 10.5 microseconds; full word multiplication 25.0 microseconds; indexing, an additional 2.0 microseconds (static); 5.0 microseconds (from core).

1.4 CORE MEMORY.

The five-microsecond cycle time magnetic Core Memory is provided in modules of 4000 words of 56 bits each. Module addresses are patched in the Central Processor. Each Central Processor may address 64,000 words of memory, of which 32,000 words can be shared between two Central Processors.

A special twinning feature allows duplicate writing in a pair of individual (non-shared) memory modules.

Transfers to and from the Core Memory are word parallel, and are parity checked. The memory is time shared within the Central Processor by the real-time, Disc

Memory, Uni-Record interfaces and the program control, permitting continued program execution while interface operations are in progress.

A special feature allows partial overlap of core memory cycles in separate modules, reducing the effective cycle time.

### 1.5 DISC MEMORY.

Disc storage is provided by central Disc Memory modules and by Auxiliary Disc Memory modules. Each disc module contains a storage capacity of about 20 million alphanumeric characters, organized into blocks of 128 characters (16 Data Processor words). Large discs rotate at approximately 14 revolutions per second, and utilize a full complement of fixed heads rather than moving heads. The access time is, therefore, a direct function of the revolution rate.

A Central Processor can be switched to one of two Disc Memory trunklines, and a trunkline can service up to seven central Disc Memory modules. Each module provides a Control Section and a fast buffer for transfers to and from the Central Processor. A Disc Memory module may, in turn, control up to seven Auxiliary Disc Memory modules, each of which also contains 20 million characters of storage but provide no Control Section or buffer storage. There may be a total of seven Central Modules and 49 Auxiliary Modules of disc storage with each Data Processor, permitting over 1 billion characters of file storage.

The Disc Memory Control Section has the important capability of accessing data by content, in addition to the standard fixed address read and write operations. A search may be specified on any combination of bit, character, and field positions in each block. Once started, the search operation continues without Central Processor intervention, and up to 1350 blocks can be compared per disc revolution. Also the Disc Memory can tag specified blocks as obsolete, and new data may then be loaded into the first available obsolete spaces, thereby reducing store access time. These blocks can be retrieved with a search by content, which, under program option, can also provide the fixed address locations.

### 1.6 REAL-TIME COMMUNICATION.

Buffer Processors are provided to maintain communication with a Data Terminal Bay, and to buffer, code-convert, and format check data. These communicate with the Central Processor through an I/O Interface, which also handles data transfers to and from Display Console Electronic Typewriters.

Section I  
Paragraphs 1.7 to 1.8

The Buffer Processors execute their own internally stored programs. One, with appropriate program, is needed for incoming data, and another for outgoing data. Programs and data are stored on a 8000 revolution per minute magnetic disc.

The Display Consoles also have a sizeable buffer storage area as part of the Disc Memory for outgoing data.

The Central Processor I/O Interface has a control section that executes operations in parallel with program execution. This I/O Interface scans input devices, Buffer Processors and Display Consoles, signaling the Central Processor with an interrupt when a message is available.

Data transfers are bit serial at a rate (over 700,000 bits per second) set by the communicating devices, and are character parity checked.

#### 1.7 UNI-RECORD FACILITIES.

The Central Processor has a Uni-Record Interface which has a character-serial trunkline. This trunkline handles magnetic tape transports, punched card readers and punches, and line printers. The Uni-Record Interface has its own control section which can execute operations, once started, in parallel with program execution. Data transfers are character parity checked.

The Uni-Record Trunkline has the capability of off-line operations, namely, tape-to-card, card-to-tape, tape-to-line printer and card-to-line printer, under manual control.

#### 1.8 CHECKING FEATURES.

Reliability has been a primary aim in the data processing system design. When errors occur, they are caught by the character parity check that accompanies every data transfer by the Central Processor. Where possible, illegal characters are regarded as errors, such as in arithmetic operations and in memory addresses.

As part of the Disc Memory, I/O, and Uni-Record Interface operations, the Central Processor program receives an interrupt signal upon remote detection of parity error in data transfers either way. Under program control, an override of the automatic termination of data transfers upon error detection is provided as an aid in error corrections.

Blocks, as stored in the Disc Memory, have the character parity bits stripped from them, but longitudinal parity bits substituted, for reasons of speed and economy. Character parity is regenerated during transfer to the Central Processor.

A basic protection of stored data is provided by using the duplexed system to duplicate operations and stored data. This concept is carried to a finer level by the feature called "twinning", available both in Core Memory and Disc Memory. Information is simultaneously stored in duplicate when in twin mode. In event of failure of one read operation, the second copy is still available.

The normal result of a sensed error is a program interrupt, unless this interrupt is ignored. The interrupt program can test for the detailed source of the error to take remedial action.

### 1.9 DUPLEXING.

In a duplexed or multiplexed system, system functioning can be maintained by a single Central Processor. Program controlled cross connections are provided between Central Processors and Disc Memory Trunklines.

Each Display Console can be switched manually to either side of a duplex system. Similarly, Buffer Processors may be switched, and have duplicate standby equipment available.

Each Central Processor is able to test by program the state of each switch. In addition, special communication is available between Central Processors for programmed switching.

### 1.10 PROGRAM INTERRUPTS.

The Central Processor has an elaborate set of program interrupts. These provide a transfer of control to special programs which determine and deal with the causes of the interrupts and then resume normal program execution.

There are six classes of interrupts, each with an independent special program starting point. These classes are: Error, Disc Memory, Input, Real Time, Uni-Record, and Other Processor. Within each class are several detail interrupt sources which are program testable.

The Error class of interrupts has two divisions, Computer Error, which includes data transfer, arithmetic and program errors, and Interface Error, which includes Disc Memory, I/O Interface, and Uni-Record Interface errors.

The other classes of interrupts contain signals generated when specific action is to be taken as, for instance, when an interface finishes a data transfer operation.

All detail interrupt requests are stored in toggles which are reset by program individually or by class, and which can also be set by program.

Section I  
Paragraph 1.10

Other toggles, under program control, permit ignoring interrupt requests by class, subclass, or totally, but without losing the interrupt requests which remain program testable.

## SECTION II

### CORE MEMORY AND WORD FORMAT

Core Memory for the Data Processing System is provided in modules of 4000 words. These modules may be assigned to a Central Processor on an individual basis, or may be connected to two Central Processors via a shared memory trunkline. The assignment of module addresses is by a patchboard, which has the additional facility of pairing modules in the Twin mode.

#### 2.1 MEMORY REGISTERS.

##### 2.1.1 MEMORY ADDRESS REGISTER.

Words in each Core Memory module have decimal addresses of 0000 through 3999. Each module has an address register which is loaded in parallel via the trunk from the Central Processor, while the high order bits of the high order character of the address perform module selection through the patchboard. (In some units, two modules share a common address register.) The Memory Address register appears to the Central Processor as a single register. The Memory Address register receives word addresses in parallel from the Program Control Section and the Disc Memory, Uni-Record and I/O Interfaces. Parity is not checked, but an error is indicated when a non-existing memory module is addressed.

##### 2.1.2 MEMORY ACCESS REGISTER (M).

Each Core Memory module has an eight-character (full word) M register. (In some units, two modules share a common M register.) Transfers to or from Core Memory are full word parallel, via the trunkline which appears to the Central Processor as a single Memory Access Register, which is displayed on the Central Processor Control Panel. All transfers are character parity checked.

Memory cycle time is 5 microseconds, and data is on the trunkline 2.5 microseconds after the start of the cycle. For some operations Core Memory is half cycled for read only or write only. For instance, a half cycle write is performed when an address of four characters is to be stored into a word of memory, leaving the rest of the word unchanged. The effect is a saving of time where, otherwise, two full memory cycles would be needed; a read-restore, and a clear-write. The sequence is half cycle read, modify, half cycle write. Half cycling is under tight logical control

Section II  
Paragraphs 2.2 to 2.2.3

so that, even under error conditions, no information will ever be lost. Other than effects on timing, programming is not involved in half cycling by any possibilities of over-writing or inadvertently clearing portions of memory.

A special feature allows overlap of memory cycles in different Core Memory modules. The overlap gives an effective cycle time of about 4 microseconds. This cycle overlap is particularly effective in unrelated operations, such as the memory cycles requested by interfaces and their effect on normal program execution.

## 2.2 SPECIAL FEATURES.

### 2.2.1 CYCLE DEMAND PRIORITY.

The Disc Memory, I/O, and Uni-Record Interfaces operate independently of, and simultaneously with, the Program Control Section. Any or all may be requesting a memory cycle. A precedence occurs at logic level giving priorities as follows:

- a. I/O Interface.
- b. Disc Memory Interface.
- c. Uni-Record Interface.
- d. Program Control Section.

Central Processor program instruction execution can wait the most easily, and is given the lowest priority.

### 2.2.2 INDIVIDUAL MEMORY TRUNKLINE.

Each Central Processor has an individual Memory Trunkline which can service up to eight memory modules. The assignment of a module to a Trunkline is done physically by attaching cables.

Memory modules are contained in cabinets that hold up to four modules and two sets of Trunkline Drivers. Each Trunkline Driver can connect one Trunkline with up to three memory modules. Pairs of modules which share common Address and M registers cannot be connected to separate Trunklines.

### 2.2.3 SHARED MEMORY TRUNKLINE.

Each Central Processor has a Shared Memory Trunkline. A single Central Processor can service up to eight additional memory modules via this Trunkline. Two such Trunklines from two Central Processors can be connected together through a Shared Memory Switch built into the memory cabinet, and a total of up to eight

memory modules can be distributed on either side of the switch. When the Shared Memory Switch is closed, a single Shared Memory Trunkline is formed, servicing all of the memory modules on either side of the switch, and only one Central Processor has preemptive use of the Trunkline. That Central Processor also has exclusive program control of the switch. When the switch is open, each Central Processor can independently access the memory modules on its side of the switch. Either Central Processor can control the Shared Memory Switch, and have access to all of the Shared Memory, and that Central Processor has Control Status. Control Status is indicated to each Central Processor by its Control Status toggle, which can be program tested (ZY = 59) and reset, but not set. The Control Status toggles in the two Central Processors are cross connected so that, when one is reset by program, the other is set, which transfers Control Status. Control Status also has an effect on the control of Disc Memory Duplexing, (Section XII).

Each Central Processor has a program testable, settable, and resettable toggle (ZY = 31) that requests opening and closing of the Shared Memory Switch. Only the Request Memory Switch toggle in the Central Processor that has Control Status, affects the Shared Memory Switch. The Shared Memory Switch status is separately testable (ZY = 32). Switching will follow within 10 microseconds after changing the Request toggle.

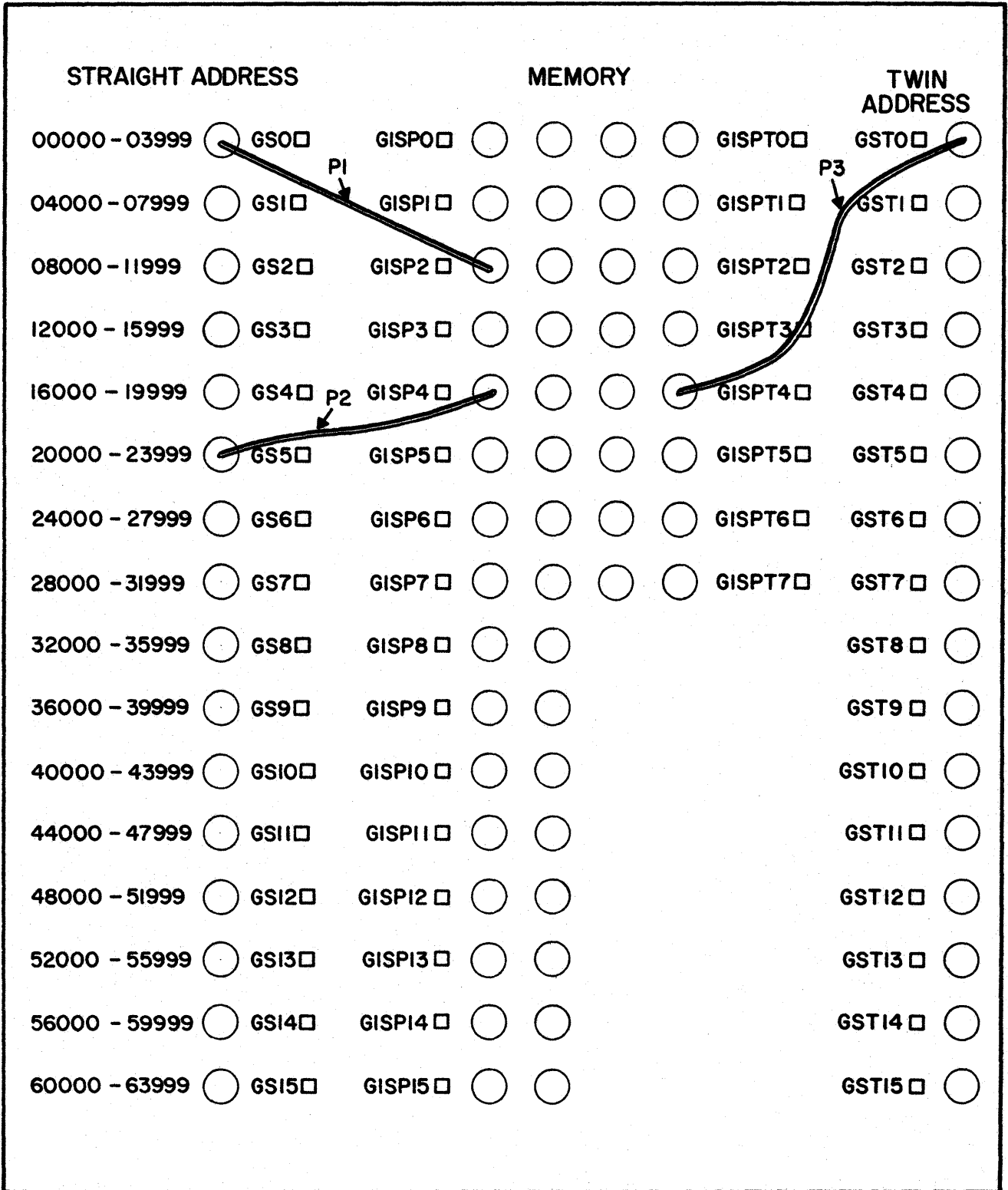
When one Central Processor is shut down, the other has full use of Shared Memory and the Shared Memory Switch remains closed. The Memory Switch Status will test closed (true). These conditions will persist until after power is reapplied to the shut down Central Processor.

#### 2. 2. 4 MODULE ADDRESS PATCHBOARD.

Each Central Processor has a small Patchboard, figure 2-1, with which its memory module addresses are established. Each of the sixteen groups of 4000 addresses (00000 through 63999) is presented as a selection signal that can be patched (connected) to any memory module, including all of those on the Shared Memory Trunkline. Memory modules are numbered 0 through 7 on the Individual Memory Trunkline and 8 through 15 on the Shared Memory Trunkline.

Shared Memory modules can be assigned module addresses independently by the Patchboards in the two Central Processors. That is, a given module may or may not have the same address patched to it in the two Central Processors.





PI, P2, P3: EXAMPLES OF PATCHES

Figure 2-1. Memory Address Selection Patch Board

### 2. 2. 5 CORE MEMORY TWINNING.

The Patchboard in each Central Processor which assigns module addresses has an additional capability of providing simultaneous write operations in pairs of memory modules on the individual Trunkline. Pairs which share common address and M registers cannot be twinned. For example, if memory module B is patched to module A, write operations addressed to A will also write in B (in corresponding words). However, write operations addressed to B will not affect A, unless a second patch of A to B is made. In any case, read operations are not affected by the twinning.

Twinning is under control of a toggle (ZY = 53) that can be program tested, set, or reset. When the Twin Mode toggle is reset, duplicate writing will not take place.

Operations in which there is a write half cycle in Twin mode will always have a corresponding full clear and write cycle in the twin module. The clear occurs at the time of the read half cycle that preceded the write half cycle. No time is lost, but the information preserving properties of Twin mode are lost in half-cycle operations.

Figure 2-1 shows how the Patchboard can be used for module address selection. Patch wire P1 establishes addresses 00000 to 03999 for Core Memory module 2. Patch wire P2 establishes addresses 20000 to 23999 for Core Memory module 4. Patch wire P3 establishes twin addresses 00000 to 03999 in Core Memory module 4. When the Twin Mode toggle is ON, write operations addressed to locations 00000 to 03999 in module 2 will write also in corresponding locations 20000 to 23999 in module 4. Read operations addressed 00000 to 03999 will read only from module 2 and, similarly, read operations addressed 20000 to 23999 will read only from module 4. Write operations addressed 20000 to 23999 will write only in module 4 and will not affect module 2. Modules 2 and 4 do not share common address and M registers.

## 2. 3 INFORMATION FORMAT.

### 2. 3. 1 WORD.

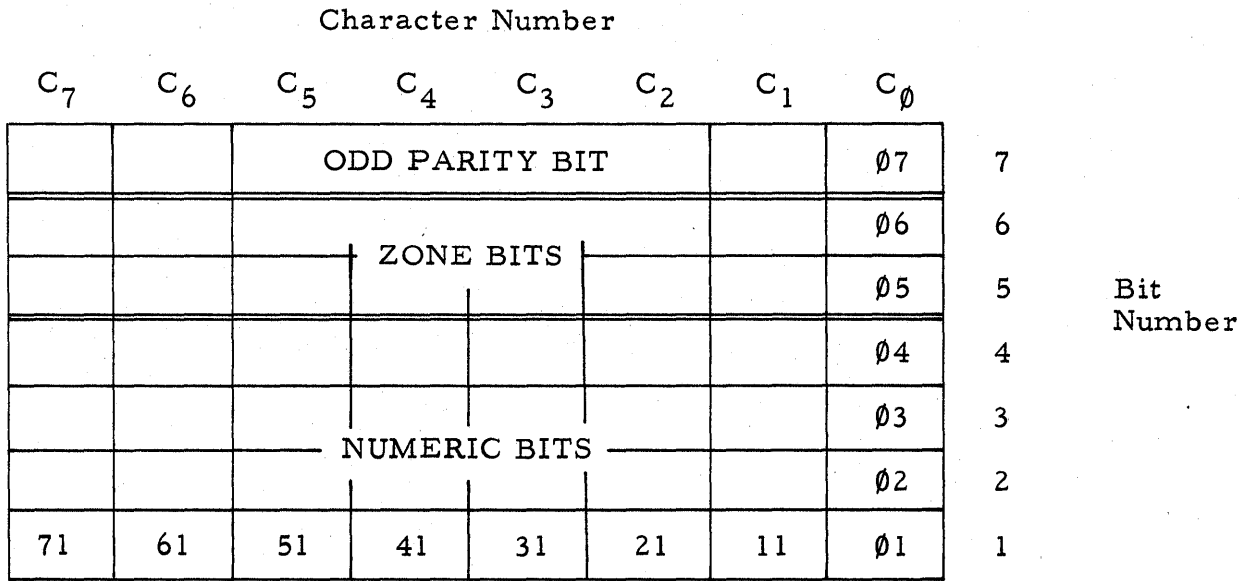
The basic storage element is the Central Processor word, which may be assigned specific Core Memory addresses. A word consists of eight 7-bit characters, figure 2-2, and may contain either an instruction, alphanumeric data, or a floating-point number. The characters in a word are numbered zero through seven, beginning with the least significant character position.

Section II

Paragraphs 2. 3. 2 to 2. 3. 3

2. 3. 2 CHARACTER.

A character contains seven bits. These bits are numbered 1 through 7, beginning with the least significant bit position. Bits 1 through 4 are designated the numeric bits, bits 5 and 6 are the zone bits, and bit 7 is an odd parity check bit. The following diagram presents the character and bit assignment for a Central Processor word.



\*  $C_7$  most significant     $C_0$  least significant

Figure 2-2. Information Word Format

A short two-digit notation will sometimes be used in this manual to designate a bit position in the Central Processor word. The first digit will provide the character number, and the second digit will specify the bit number within that character. Bit 75, for example, designates bit number 5 for the most significant character position.

2. 3. 3 BIT.

A bit may have a binary value of 1 or  $\emptyset$ , in accordance with the position of a toggle, or the direction of magnetization of a single core in Core Memory or of an area on either magnetic disc or magnetic tape. These bits may be transferred from Core Memory to the Program Control, Disc Memory Control, or Arithmetic Sections, where the binary value is recognized and used to access instructions, transfer data, and perform computation. The 1 status of a toggle is also sometimes considered its Set, ON, or True state, while a  $\emptyset$  status is the Reset, OFF, or False state.

The four numeric bits plus the two zone bits are sufficient to permit 64 alphanumeric character code combinations. Refer to Appendix A for a list of Central Processor, Typewriter/Printer and Punched Card character codes. A seventh bit has been added to each 6-bit character code to provide an odd number of 1 bits. This odd parity status is checked on every data transfer to or from Core Memory, Arithmetic Section registers, Disc Memory, Uni-Record Trunkline devices, and other input-output units. In the event of a parity error, an interrupt request is generated and the Central Processor will enter an interrupt program (unless errors are being ignored under program control).

#### 2. 3. 4 ALPHANUMERIC DATA WORD.

An alphanumeric data word may contain several fields of data, each containing fixed-point numeric, alphabetic, or alphanumeric (any combination of symbols, alphabetic, and fixed-point numeric) information. A field consists of any consecutive number of characters within the word, and may be specified by many Central Processor instructions. Floating-point instructions, however, process only full-word operands.

The following example presents a Gross Pay field, containing \$110.25, and a Rate of Pay field, specifying \$2.25 per hour, within an alphanumeric data word.

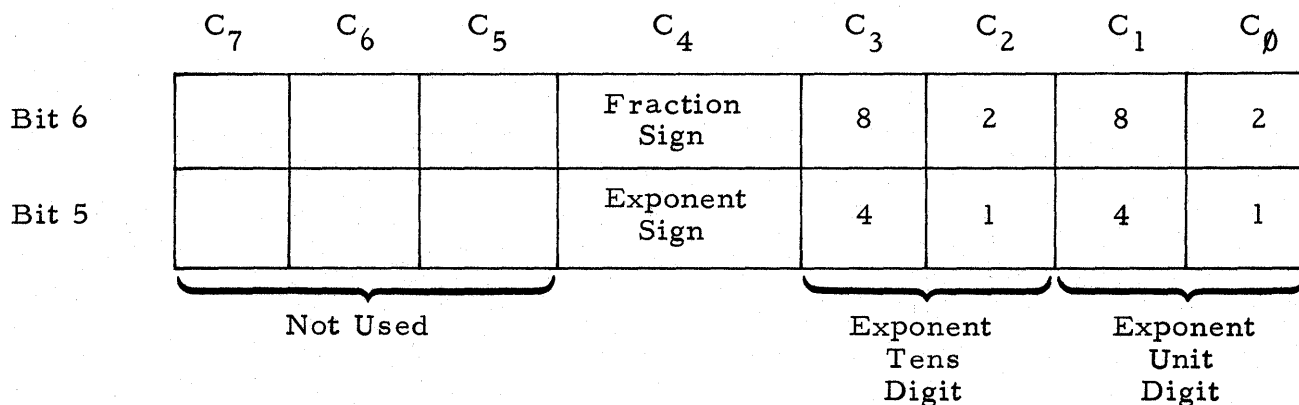
**EXAMPLE:**

C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>
1	1	0	2	5	2	2	5
Gross Pay field					Rate/Pay Field		

The programmer must locate the decimal point of numeric fields for fixed-point operations. The sign of each fixed-point field within the word is carried in the sixth bit of the least significant character position. A positive sign is indicated by a 0 in bit position six, while a negative sign is provided by a 1. The fifth bit of the least significant character of the field, as well as the zone bits of the other numeric characters, are normally 0. Alphabetic characters and special symbols contain combinations of 1 and 0 in the zone bit positions.

2.3.5 FLOATING-POINT DATA WORD.

Floating-point execution may be specified as an option by all full-word arithmetic instructions and by a numeric compare instruction, paragraph 2.4.5. This option is provided to extend the range of the operands that participate in these operations. Floating-point numbers carry and automatically process the decimal point position. A floating-point data word presents the number as a signed decimal fraction times a signed integral power of ten. The sign of the fraction and the exponent with its sign are carried in the zone bits of characters  $C_4$  through  $C_0$  in the following format. A 1 bit in a sign position indicates a minus.



The exponent is carried as two binary-coded decimal digits in the zone bits of the four least significant characters of the data word.

Floating-point numbers can range from  $.00000000 \times 10^{-99}$  (equals zero) to  $.99999999 \times 10^{99}$ . There is no minus zero in either the mantissa or exponent.

The following example presents the bit configuration for the floating-point operand,  $+ .0162 \times 10^{-14}$ , which could be represented in the computer word and in the example as the normalized quantity,  $+ .16200000 \times 10^{-15}$ . A floating-point number is said to be normalized when it is positioned so that the machine decimal point is to the left of the most significant (non-zero) digit of the fraction.

EXAMPLE:

	C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>	
Floating-Point Data Word (as it would appear in a register)	∅	1	∅	∅	1	∅	∅	∅	B <sub>7</sub>
	∅	∅	∅	∅(+)	∅	∅	∅	∅	B <sub>6</sub>
	∅	∅	∅	1(-)	∅	1	1	1	B <sub>5</sub>
	∅	∅	∅	∅	∅	∅	∅	∅	B <sub>4</sub>
	∅	1	∅	∅	∅	∅	∅	∅	B <sub>3</sub>
	∅	1	1	∅	∅	∅	∅	∅	B <sub>2</sub>
	1	∅	∅	∅	∅	∅	∅	∅	B <sub>1</sub>
Quantity: + . 1	6	2	∅	∅	∅	∅	∅	∅	x 10 <sup>-15</sup>
Printout on Line Printer or Typewriter:	1	6	2	+	∅	+	+	+	

## 2.4 INSTRUCTION WORD FORMAT.

Each Central Processor or input-output instruction is contained in one 8-character word, figure 2-3. Disc Memory instructions, which require three consecutive words, are discussed separately in Sections IX and X. The most significant character (C) of the instruction word specifies the command, the next two characters (Z and Y) are used normally for field select and address select, but may also specify command modification. The next most significant character (X) selects one of 17 Index registers. The least significant four characters (MMMM) normally provide the unmodified operand or next instruction address. Zone bits 5 and/or 6 of the X, ZY, and three least-significant MMMM characters may be used to modify operand access or instruction execution.

Zone bit positions that are used with specific instructions to provide some other control of instruction execution, are described individually in Section IV. In subsequent sections of this manual, a character in the instruction word may be designated either by its position in the word or by the instruction function. Bit 5 of the Z character, for example, may be designated as bit 65 or as bit Z5.

Section II  
 Paragraphs 2. 4. 1 to 2. 4. 2

Character No.	C7	C6	C5	C4	C3	C2	C1	C0
Instruction Characters	C	Z	Y	X	M	M	M	M
Zone Bit 6		Tagged Halt	Floating Point	Flag Return		MASM Hold	MASM O. A. -1	Indirect Address
Zone Bit 5		Absolute	Normalized			MASM Bring	MASM O. A. +1	

Figure 2-3. Instruction Word Format

2. 4. 1 COMMAND CHARACTER (C).

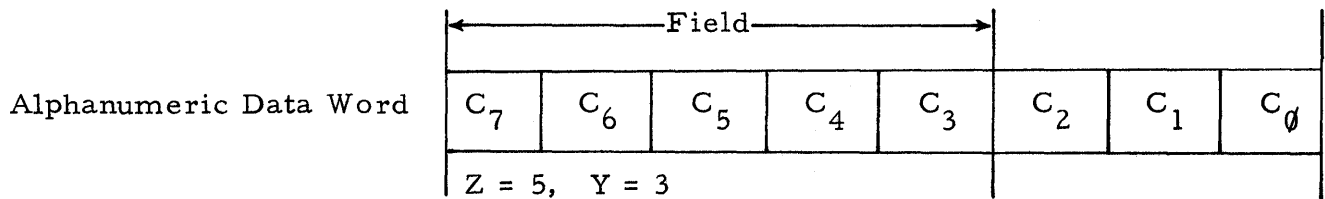
The command character (C) specifies one of 34 basic Central Processor commands, 6 input-output commands, or designates Disc Memory operation. The ZY characters and bit options of the instruction word, however, expand the program range to hundreds of individual and distinct Central Processor and input-output operations. The different Disc Memory operations are specified by a three-word Disc Memory instruction.

The command character normally provides easy to remember codes in machine language. (A for Add, M for Multiply, I for Input, etc.) Programs for this Data Processing System, however, will seldom be written directly in machine language. The programmer will normally use LAP, the Librascope Assembly Program, to provide the actual Central Processor instructions. The LAP system permits the use of mnemonic terms for the command code and the different bit options. It allows Index registers and ZY select characters to be defined symbolically, facilitates access to closed subroutines, provides the programmer with a variety of system macros, permits him to create additional macros, etc. Refer to Appendix D for LAP mnemonics.

2. 4. 2 SELECT CHARACTERS (ZY).

a. Field Select. Central Processor fixed-point arithmetic, transfer, and compare instructions specify field select operation. The decimal value of the numeric bits of Z then determine the number of characters in the field, and the decimal value of the numeric bits of Y designate the position of the least significant character. The sum of the decimal values of Z and Y must be less than or equal to 8 for results to be defined. The following example illustrates the ZY function.

EXAMPLE:



b. Address Select. Input-Output, Test, Set, Save, Load, and Copy instructions specify a Trunkline or other input-output device, or designate a signal line, toggle, or Central Processor register. The least significant five bits of both the Z and Y characters normally are used for addressing, and are capable of selecting up to 1024 positions or devices. The ZY addresses assigned to Central Processor instructions are listed in Appendices B and C.

c. Instruction Modify. The ZY characters also control execution of operations such as the Shift and the Modify Index register instructions. These instructions are discussed individually in Section IV.

2. 4. 3 INDEX CHARACTER (X).

Most of the Central Processor, input-output, and Disc Memory instructions may specify index modification of the operand address. Bits 1 through 5 of the X character in the instruction word will select one of 17 Index registers. An X value of 0 means no index operation. Binary values 1 through 15 specify operand address modification by one of 15 Index registers, XR1 through XR15, stored in Core Memory. These occupy the four least significant character positions of memory addresses 0001 through 0015, respectively. Binary values of either 16 or 17 designate address modification by one of two static Index registers, XR16 or XR17. Core memory positions 0016 and 0017 are normal storage locations. Binary values other than 1 through 17 in bits X1 through X5 specify no index modification.

The indexing function will automatically add the content of the designated Index register to the MMMM operand address of the instruction word. This action precedes instruction execution and provides an effective operand or next instruction address equal to MMMM + (X), defined as the value of MMMM plus the content of the X designated Index register. Address modification by XR1 through XR15 increases instruction operation time by one 5-microsecond Core Memory cycle. Static Index register modification by XR16 or XR17 is accomplished in one clock time.



## Section II

### Paragraphs 2. 4. 4 to 2. 4. 5. 1

A Modify Index Register instruction ( $\square$ ) is available to increment or decrement the content of the Z designated Index register by the Y specified decimal value of from 1 through 9. If the content of the selected Index register then attempts to become negative or exceed core memory capacity, normal sequential instruction access continues; otherwise the next instruction will be accessed at the effective operand address in the  $\square$  instruction. The indexing option, in conjunction with the  $\square$  instruction, facilitates the repetitive use of program instructions and routines.

#### 2. 4. 4 MEMORY ADDRESS CHARACTERS (MMMM).

The MMMM characters of the instruction word in most instructions specify the unmodified or original Core Memory location of the operand or the next instruction address. When the address portion of the instruction word is not required, it may be used for other data storage.

The numeric bits of the three least significant characters and bits 1 and 2 of the most significant character in MMMM specify locations 0000 through 3999 for each Core Memory module. Bits 3 through 6 of the most significant MMMM character provide a binary value of 0 through 15, to select one of 16 Core Memory modules. It should be noted that the Central Processor arithmetic instructions provide decimal operations. A Convert instruction (P) is available to translate between the four character MMMM binary-decimal address and an equivalent five character decimal address. This instruction is intended to be used in conjunction with arithmetic operations which modify or provide memory addresses in excess of 9999.

The indexing function, the Modify Index Register instruction, and any zone-bit-specified MMMM increment-decrement action automatically will provide proper binary-decimal address modification. These operations process only the numeric bits of the three least significant MMMM address characters, and all six data bits of the most significant character. All other bit positions in the Central Processor word are ignored.

#### 2. 4. 5 ZONE BIT OPTIONS.

2. 4. 5. 1 INDIRECT ADDRESSING. Further operand or next instruction address modification is provided if bit 6 of the least significant character in the instruction word is a 1. Then the MMMM address is incremented by the content of the specified Index register to provide an indirect effective address. The word at this indirect location again will contain MMMM and X characters, and another indirect address bit designation. Now the new MMMM+(X) will provide the effective operand

or next instruction address if the indirect address bit is a 0, or specify the next indirect effective memory address if the bit is again a 1.

Assume that an Add instruction word provides an operand address MMMM of 0090, specifies an X Index register 13 whose content is 0010, and the indirect-bit designator is 1. Since the indirect effective address is 0100, the Central Processor will access this location. Assume that the word at 0100 contains an MMMM of 2050, designates an X Index register 9 whose content is 0125, and the indirect bit designator is 0. The effective operand address is then 2175, and this location will supply the required operand for the Add operation. It should be noted that during the indirect addressing operation all but the MMMM, X, and indirect-bit positions at any indirect address location will be ignored. All other bit options, as well as the ZY characters, are provided by the original instruction word.

2. 4. 5. 2 FLOATING-POINT. Floating-point operation may be specified by all full word arithmetic instructions (A, S, +, -, M, and D) and by the Numeric Compare instruction (Q) if bit Y6 in the instruction word is a 1. Fixed-point execution is selected if the bit is a 0. This Y6 bit option will be ignored for all other instructions. Floating-point instructions may be indexed, and may specify indirect addressing.

The operands and/or result of a floating-point operation will automatically be normalized if bit Y5 in the instruction word is a 1. This bit option will be ignored for all other instructions and for fixed-point operation.

The actual normalize process varies for the several floating-point instructions. These instructions are described individually in Section IV.

A normalized floating-point operand has its fraction shifted left, with proper modification of the exponent, until the most significant digit position in the Central Processor word is non-zero. Floating Point Instructions are described in paragraph 4. 6.

2. 4. 5. 3 ABSOLUTE. Absolute execution is specified for the A, S, +, -, M, G, D, / and Q instructions, during either fixed-point or floating-point operation, if bit Z5 in the instruction word is a 1. This bit option will be ignored for all other instructions. Absolute operation considers the specified field, word, or fraction in Core Memory to contain a positive sign. Standard instruction execution is then completed, using the algebraic operand held by the relevant Central Processor register.

## Section II

### Paragraphs 2. 4. 5. 4 to 2. 4. 5. 5

2. 4. 5. 4 TAGGED HALT. Tag control is available with all instructions. A Tag button on the Control Console can be depressed to set a corresponding Central Processor toggle, which may be program tested, (ZY=5 J). The ON position of the Tag button and bit Z6 in the instruction word equal to 1 will cause the Central Processor to Halt after access of the instruction word.

The Central Processor ignores the tag bit if the Tag button is OFF. Pressing the Start button or the Step button on the Control Console, following the Halt, permits the program to continue with normal instruction execution. The tag control assists in program debugging by providing an optional Halt preceding execution of any tagged instructions.

2. 4. 5. 5 FLAG RETURN. The address of the instruction word is placed automatically in memory when bit 6 of the X character is a 1. If an interrupt does occur, the stored instruction address represents a return point for the program.

An X6 flagged instruction has its address stored in the most significant half of one of four Core Memory locations, depending on the type of instruction. The least significant half of these locations is not modified.

- a. Memory Address 0001 - Central Processor instruction
- b. Memory Address 0002 - Disc Memory instruction
- c. Memory Address 0003 - Uni-Record Interface instruction
- d. Memory Address 0004 - Input-Output Interface instruction

The utility and operation of the X6 flag return option is described further in Section XI.

## SECTION III

### CENTRAL PROCESSOR

This section describes the Arithmetic and Program Control Sections of the Central Processor. Instruction execution is described in Section IV. Later sections describe other portions of the Central Processor; such as the Core Memory, Disc Memory, I/O, Uni-Record Interfaces, and the Real-Time Clock.

#### 3.1 ARITHMETIC SECTION.

All arithmetic and many logic operations are carried out in the Arithmetic Section, which also participates in the transfer of data between registers, figure E-1.

The Arithmetic Section contains the Arithmetic registers and a high-speed, four-character-parallel adder-subtractor. By use of an auxiliary four-character Sum register, the adder can add two four-character numbers in two clock times, and two eight-character numbers in three clock times.

##### 3.1.1 R REGISTER.

The R register is a one-word static register which functions as a lower accumulator in arithmetic operations and as a distributor in data transfers. The R register communicates in full-word parallel with the M register, half-word parallel with the A register via the adder, and half-word parallel with other registers for data transfers. Most instructions requiring a memory cycle for execution involve the R register on the way to and from the M register, so as to free the M register for data transfer through the Interfaces. The R register is displayed on the Central Processor Control Panel.

##### 3.1.2 A REGISTER.

The A register is a one-word static register that participates in arithmetic and logical operations. Normally the A register is used as an accumulator. It communicates half-word parallel with the adder and, through the adder, with the R register. The A register is displayed on the Central Processor Control Panel.

Section III  
Paragraphs 3. 1. 3 to 3. 2

### 3. 1. 3 Q REGISTER.

The Q register is a one-word static register that is not accessible to the programmer, but which generates a 2-multiple and 5-multiple in such functions as multiply and divide. For these instructions, the Q register communicates full-word parallel with the M register.

### 3. 1. 4 TOGGLES.

The Arithmetic Section has four toggles that are set true by program errors. These are:

- a. Overflow (fixed-point Add, Subtract, and Divide) (ZY = W1)
- b. Floating-Point Overflow (ZY = W2)
- c. Convert Overflow (ZY = W3)
- d. Non-Numeric Operand (numeric bits not in the range 0 to 9) (ZY = W4)

The specific conditions which cause these toggles to be set and the testing and re-setting of them are presented in paragraph 11. 13.

### 3. 1. 5 ADDER.

The Arithmetic Section has a four-character parallel adder that develops four characters of a sum or difference in one Central Processor clock time. The result is stored temporarily in a Sum register; a second clock time is required to transfer the result into the result register (A or R). The transfer to the result register will overlap the second sum generation in the case of a field of five digits or greater, giving a full word add in three clock times.

Also the adder is employed in index modification of operand addresses. A special feature handles the most significant character, which is not a decimal digit, but carries a value of 0 to 63.

The Adder has a special self-checking feature that generates an Arithmetic Data Transfer Error Interrupt request (ZY = V5). Any single malfunction of the Adder that leads to an incorrect result during an addition or subtraction operation will be detected.

## 3. 2 PROGRAM CONTROL SECTION.

The Program Control Section provides the registers and control circuitry that enable the Central Processor to carry out the specified instructions. Each

instruction is obtained from Core Memory in the location specified as the instruction address and is held and processed in the Instruction register which is described in terms of its component parts.

### 3.2.1 INSTRUCTION ADDRESS REGISTER (Ia).

The Instruction Address register (Ia) is a four-character static register that provides the address for instruction access. The normal address format is as follows: The numeric bits of the three least significant characters, and bits 1 and 2 of the fourth character, specify locations 0000 through 3999 for each Core Memory module. Bits 3 through 6 of the fourth character provide a binary value of 0 through 15 that select one of 16 Core Memory modules. Since the fourth character carries a binary value of 0 to 63, Core Memory addresses can be thought of as running from 00000 to 63999.

The Ia register counts up by one after instruction execution, unless its content was affected by a program jump, paragraph 4.5. A program jump places the fully modified operand address in the Ia register to be used for the next instruction access.

The Ia register content can be changed by a Load ( $\oplus$ ) or Copy (=) instruction, paragraphs 4.2.6 and 4.2.8. In case of Ia register change, one is added to the resulting Ia content before the next instruction access. Transfers to and from Ia are in four character parallel.

The Ia register content will count up from 63999 to 00000, but instruction execution is halted and an End of Memory Error Interrupt request is generated. Should the Ia register count up to a non-existent address (one for which there is no Core Memory module), the Core Memory control rejects the address and generates an Illegal Address Interrupt request. The content of the Instruction Address register is displayed on the Central Processor Control Panel.

### 3.2.2 OPERAND ADDRESS REGISTER (Oa).

The Operand Address register (Oa) is a four-character static register that holds the operand address through the execution of an instruction. The address is held in normal format, paragraph 3.2.1.

The Operand Address register is first loaded with the address portion, MMMM, of the instruction word, paragraph 2.4.4. Options that may modify the operand address then take place. These are: Index Modify, paragraph 2.4.3, Indirect

### Section III

#### Paragraphs 3.2.3 to 3.2.5

Addressing, paragraph 2.4.5.1, and PDPM, paragraph 3.4. MASM, paragraph 3.5, does not modify Oa. Transfers to and from Oa are in four character parallel.

If the modified operand address specifies a non-existent memory module, and a memory cycle is required, an Illegal Address Interrupt request is generated (unless the memory cycle is required, no interrupt is requested). Interrupt requests may occur also during execution of instructions that require multi-word Core Memory accesses. The latter may also generate End of Memory Interrupt requests.

The Operand Address register content can be transferred to another register by a Copy instruction. In this case, it is the original address portion of the Copy instruction itself that is transferred. The Operand Address register can be transferred into by a Copy instruction, but will have its content replaced by the next instruction. The content of the Operand Address register is displayed on the Central Processor Control Panel.

#### 3.2.3 INDEX HOLDER (Xh).

The Index holder (Xh) is a one-character static register that holds the six-bit index character of each instruction and selects Index register operation. For index operation, refer to paragraph 2.4.3. The content of the Index holder is displayed on the Central Processor Control Panel.

#### 3.2.4 COMMAND HOLDER (C).

The Command holder (C) is a one-character static register that controls instruction execution. It is loaded with the six-bit command character of each instruction word. An Instruction Error Interrupt request is generated if C does not provide an assigned command character. The content of the Command holder is displayed on the Central Processor Control Panel.

#### 3.2.5 Z HOLDER (Z).

The Z holder is a one-character static register that controls command execution. It is loaded with the Z character of each instruction word and is displayed on the Central Processor Control Panel.

### 3.2.6 Y HOLDER (Y).

The Y holder is a one-character static register that controls command execution. It is loaded with the Y character of each instruction word and is displayed on the Central Processor Control Panel.

### 3.2.7 X COUNTER (X).

The X counter is a part of the phase control that participates in the execution of (character-serial) instructions. It is of interest after a Field Compare instruction, when it contains the number of comparisons completed, modulo 8, paragraph 4.3.3. The X Counter content can be transferred into the Core Memory by a Save (\$) instruction. It will appear as a decimal value, from 1 to 8, in the least significant character of the word in which it is stored, which otherwise is not altered.

### 3.2.8 INDEX MODIFY REGISTER (B).

The Index Modify register is a four-character static register that receives the content of the selected Index register prior to memory address modification or prior to execution of a Modify Index Register instruction. The Index Modify register is not accessible to the program; however, it is displayed on the Central Processor Control Panel. The Index Modify register content does not change in response to Save, Load, or Copy instructions that specify an Index register.

### 3.2.9 INDEX REGISTERS (Xr1-Xr17).

The Central Processor is provided with a total of 17 Index registers, two of which are four-character static registers. The other 15 are the four least significant character positions of memory locations 00001 through 00015.

The Adder is used in a special mode to add the selected Index register content to the operand address of instructions. Index modification from a static Index register requires two extra clock times in addition to the time for the instruction-procure memory cycle. Index modification from a memory index location requires two extra clock times in addition to the time for the Index-register-read memory cycle.

Index register content can be altered by the Modify Index register, paragraph 4.5.6, Load, paragraph 4.2.6, and Copy, paragraph 4.2.8, instructions. Copy and Save instructions, paragraph 4.2.7, can transfer Index register content into another register or into Core Memory. Transfers are in four-character parallel.

Refer to paragraph 2.4.3 for a description of the indexing function.



Section III

Paragraphs 3.2.10 to 3.3.1

3.2.10 TOGGLES AND SWITCHES.

Addresses of all toggles, switches, and status tests are listed in Appendix C.

The ten Program Memo toggles can be individually or jointly tested, and can be set or reset under program control. The ON or OFF states of the toggles are indicated on the Central Processor Control Panel.

The nine individual Breakpoint switches on the Control Console, Section XIII, can be individually tested by program, but not altered.

For toggles used in MASM and PDPM modes, see paragraphs 3.4 and 3.5.

For toggles involved in Search mode and the Compare instructions, see paragraph 4.3.

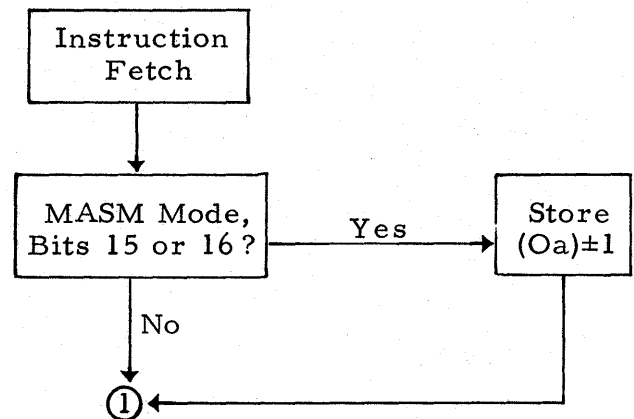
For a summary of Program Interrupts, see paragraph 1.10. For a detailed description, refer to Section XI.

3.3 PROGRAM CONTROL SECTION OPERATION.

After an Instruction Fetch memory cycle, a series of steps may take place modifying parts of the instruction before an effective operand address is generated and instruction execution ensues. The optional steps may be called out by bits in the original or the modified instruction word, paragraph 2.4.5, or by PDPM or MASM modes, paragraphs 3.4 and 3.5. The sequence of operations leading to instruction execution, Section IV, starting with Instruction Fetch is given in the following paragraphs.

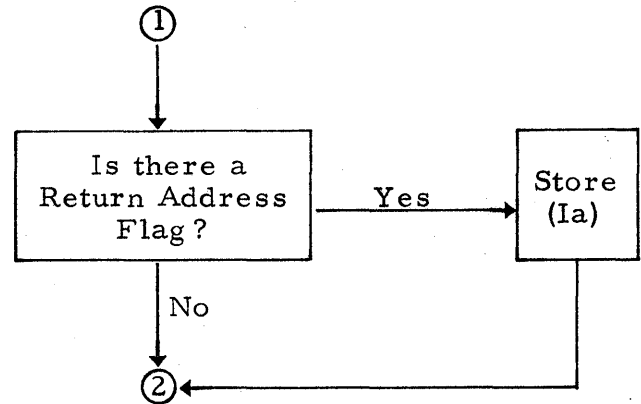
3.3.1 MASM OPERAND ADDRESS STORE.

If MASM mode is set, and if it is applicable to the instruction, bits 15 or 16 specify storing the operand address incremented or decremented by 1 back into the instruction in Core Memory.

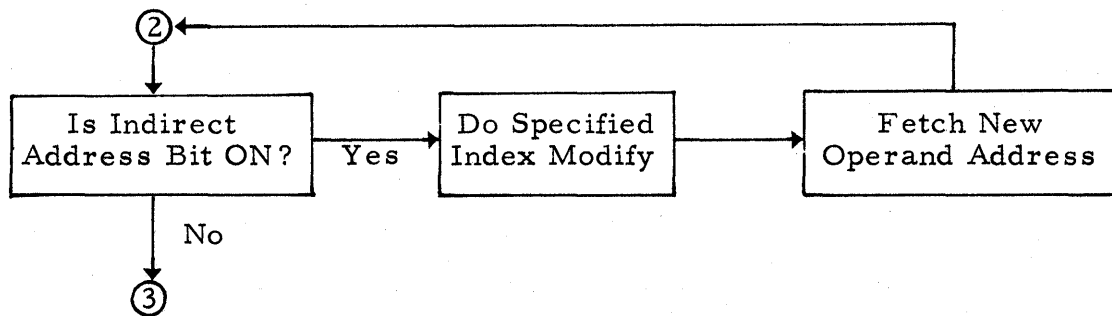


### 3.3.2 FLAGGED RETURN ADDRESS.

If there is a Return Address flag, the Instruction Address register content is stored into the proper Return Address register in the Core Memory, paragraphs 2.4.5.5 and 11.6.



### 3.3.3 INDIRECT ADDRESS.

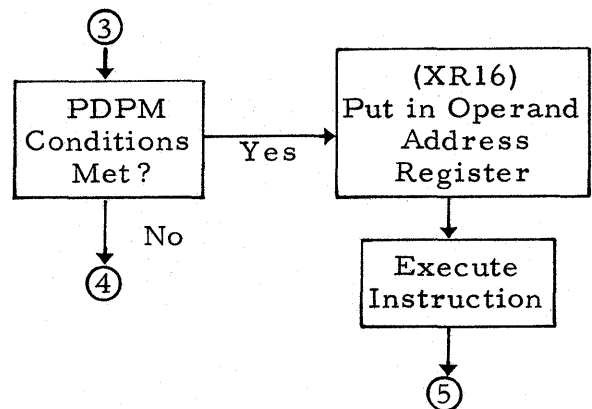


If the Indirect Address bit is a 1, and if the index character is significant, it is examined for Index Modification. A memory cycle is necessary to fetch a memory Index register. When Index Modification is complete, the modified operand address is used to fetch a new operand address and index character. Again, indirect addressing can be specified, with the Index Modify option. Indirect addressing continues until a non-indirect address is found.

If a MASM Mode Bring operation is specified, paragraph 3.5, it is done at 3.

### 3.3.4 PDPM OPERATION.

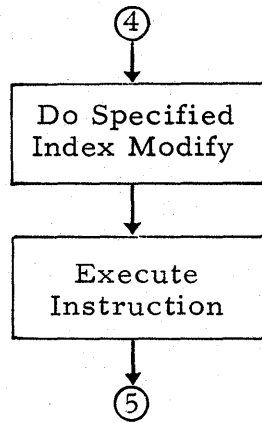
If the PDPM mode is set, the instruction has a PDPM option, and the Operand Address register content is 0000, PDPM will operate. Then Index register 16 is decremented or incremented by 1, according to whether the instruction requires an Operand Access or Store. The content of XR16 is put into the Operand Address register before decrementing, but after incrementing, paragraph 3.4.



3.3.5 INDEX MODIFY.

Except in the case where PDPM operates, a final Index Modification may be specified. A memory cycle is necessary to fetch a memory Index register. The Index register content is transferred to the Index Modify register (B) for display on the Central Processor Control Panel, paragraph 3.2.8.

If a MASM Hold operation is specified, it is done at ⑤ .



3.4 PUSH-DOWN POP-UP MEMORY MODE (PDPM).

The PDPM mode facilitates the sequential storage and last-in first-out access to a stack of Core Memory word locations.

A PDPM toggle may be set ON, reset OFF, or tested by program (ZY = 51). With this toggle ON, the Central Processor will enter the PDPM mode. The A, S, +, -, M, G, D, /, B, C, H, J, K, X, \$, and ⊕ instructions then will be executed as PDPM instructions if either: the address provided by the address bits of MMMM in the instruction word is 0000 and the indirect address bit (bit 06) is 0, or the final indirect addressed location, paragraph 3.3.3, contains an MMMM of 0000. The PDPM operand address will be supplied by the content of static Index register 16, as described in the following paragraphs. The Central Processor will provide normal instruction execution if the final effective MMMM address is other than 0000, and will ignore the PDPM toggle for all but potential PDPM instructions. The address supplied by XR16 during PDPM operation ignores the index option.

The A, S, +, -, M, G, D, /, B, C, J and ⊕ instructions require an Operand Access. The memory location for PDPM operation of these instructions is specified by the content of XR16, which is then decremented by 1.

The H, K, and \$ instructions require an Operand Store. PDPM operation for these instructions will first increment the content of XR16 by 1, and this modified address specifies the memory store location for instruction execution.

The Execute instruction (X) may operate in the PDPM mode, and XR16 then will specify the location of the instruction to be executed. The Instruction Pickup itself will be treated as an Operand Access, and the content of XR16 will then be decremented by 1. The new instruction may also designate PDPM operation. Whether this provides another Operand Access or an Operand Store will depend on the instruction, following the rules in the above two paragraphs. If the executed instruction requires an Operand Access, the word accessed will be the word following the executed instruction. If this executed instruction requires an Operand Store, the store location will be the location of the executed instruction.

### 3.5 MULTI-ADDRESS STACKING MEMORY MODE (MASM).

The MASM mode facilitates the manipulation of up to three tables in Core Memory. A single MASM instruction can specify three operand locations, saving additional instruction access cycles. Many instructions may operate simultaneously in both the PDPM and MASM modes.

A MASM toggle may be set, reset, or tested by the Central Processor program (ZY = 50). With this toggle ON, the Central Processor will enter the MASM mode. The A, S, +, -, M, G, D, /, B, C, H, J, E, Q, :, and K instructions then will be executed as MASM instructions if bit 25 and/or bit 26 in the instruction word is a 1, and/or if bit 15 or bit 16 is a 1.

The MASM instruction itself will be executed in a normal manner, and may specify standard indexing and indirect addressing. The MASM bit options, however, can designate additional Bring and/or Hold functions, and/or operand address modification in the instruction, as described in the following paragraphs. The Central Processor will ignore the MASM toggle for all but potential MASM instructions, or if none of the required MASM bit options is a 1, i. e., bit 15, bit 16, bit 25 or bit 26.

a. Bit 25 = 1. Specifies a full-word Bring (B) operation prior to normal execution of the actual MASM instruction. The Bring operand location is specified by the content of static XR16. MASM Count toggles +16(ZY = 5( ) and -16(ZY = 5, ) may be set, reset, or tested by the Central Processor program. Toggle +16 will also be reset automatically when -16 is set, and -16 will be reset when +16 is set. Execution

Section III  
Paragraph 3.5

of a MASM Bring operation, when specified by bit 25, will then increment the content of XR16 by 1 when the +16 toggle is ON, and will decrement XR16 by 1 when the -16 toggle is ON.

b. Bit 26 = 1. Specifies a full-word Hold (H) operation following normal execution of the actual MASM instruction. The Hold operand location is specified by the content of static XR17. MASM Count toggle +17(ZY = 5#) and -17(ZY = 5°) may be set, reset, or tested by the Central Processor program. Toggle +17 also will be reset automatically when -17 is set, and -17 will be reset when +17 is set. Execution of a MASM Hold operation, when specified by bit 26, then will increment the content of XR17 by 1 when the +17 toggle is ON, and will decrement XR17 by 1 when the -17 toggle is ON. Note that both a Bring and a Hold operation may be executed in addition to the MASM instruction. Also, the transfers specified by bit 25 and bit 26 do not modify the content of the R register, which is not true of normal B and H instructions.

c. Bit 15 = 1. Specifies that the current MASM instruction word is stored back in memory with its original operand address MMMM incremented by 1, paragraph 3.3.1.

d. Bit 16 = 1. Specifies that the current MASM instruction word is stored back in memory with its original operand address MMMM decremented by 1, paragraph 3.3.1. If bit 15 and bit 16 are both 1, an Instruction Error Interrupt request is generated.

It should be noted that any MASM instruction accessed during execution of an Execute instruction (X) can carry out the MASM Bring and Hold functions. However, any MASM instruction accessed through the use of the X instruction will not execute the MASM operand address increment-decrement operation.

## SECTION IV

## CENTRAL PROCESSOR INSTRUCTIONS

In the detailed descriptions of Central Processor instruction execution that follow, instructions have been grouped under the following general categories:

4.1 ARITHMETIC INSTRUCTIONS

- 4.1.1 Add (A)
- 4.1.2 Subtract (S)
- 4.1.3 Add A to M (+)
- 4.1.4 Subtract A from M (-)
- 4.1.5 Multiply Long (M)
- 4.1.6 Multiply Short (G)
- 4.1.7 Divide Long (D)
- 4.1.8 Divide Short (/)

4.2 DATA MOVE INSTRUCTIONS

- 4.2.1 Bring M to A (B)
- 4.2.2 Combine M with A (C)
- 4.2.3 Hold A to M (H)
- 4.2.4 Jerk M to R (J)
- 4.2.5 Keep R to M (K)
- 4.2.6 Load M in ZY ( $\oplus$ )
- 4.2.7 Save ZY in M ( $\$$ )
- 4.2.8 Copy Z to Y (=)
- 4.2.9 Left Shift (L)
- 4.2.10 Right Shift (R)

4.3 COMPARE INSTRUCTIONS

- 4.3.1 Alphabetic Compare (E)
- 4.3.2 Numeric Compare (Q)
- 4.3.3 Field Compare (:)
- 4.3.4 Index Register Compare (Y)

4.4 SET-CONVERT INSTRUCTIONS

- 4.4.1 Set-Reset Toggle (V)
- 4.4.2 Set-Reset Bit (#)
- 4.4.3 Logical Or ( $\vee$ )
- 4.4.4 Logical And ( $\wedge$ )
- 4.4.5 Convert (P)

4.5 PROGRAM CONTROL INSTRUCTIONS

- 4.5.1 No Operation (N)
- 4.5.2 Unconditional Transfer (U)
- 4.5.3 Test (T)
- 4.5.4 Bit Compare (\*)
- 4.5.5 Execute (X)
- 4.5.6 Modify Index Register ( $\square$ )
- 4.5.7 Halt (Z)

4.6 FLOATING-POINT INSTRUCTIONS

- 4.6.1 Add (A)
- 4.6.2 Subtract (S)
- 4.6.3 Add A to M (+)
- 4.6.4 Subtract A from M (-)
- 4.6.5 Multiply Long (M)
- 4.6.6 Divide Long (D)
- 4.6.7 Numeric Compare (Q)

Section IV  
Paragraph 4. 1

Each instruction has listed the permissible options for operand access and instruction execution. These may include index, indirect-address absolute, floating-point, normalize, tagged halt, flag return, PDPM, and MASM operation. Examples are provided with the instruction description to clarify execution. These examples do not demonstrate the options for fixed-point instructions, except for a few illustrations of absolute operation. A description of allowable floating-point instructions is presented separately, and provides examples of both unnormalized and normalized operation.

Index modification is specified by the X character of the instruction word as described in paragraph 2.4.3. The indirect-address, absolute, floating-point, normalize, tagged halt, and flag return options are designated by the zone bit positions of the instruction word, as described in paragraph 2.4.5. Further instruction modification is available to specific instructions by operating in the PDPM and/or MASM mode, as described in paragraphs 3.4 and 3.5.

#### 4.1 ARITHMETIC INSTRUCTIONS.

The eight arithmetic instructions specify and control the algebraic addition, subtraction, long and short multiplication, and long and short division operations. One of the operands is located in the specified field or word in the effective operand address, M. The other operand and algebraic result is contained in the specified field or word in the A register and/or the R register. The Arithmetic Section processes only the binary-coded decimal digits contained in the numeric bit positions, paragraph 2.3.4. The zone bits, other than the sign positions (and exponents, for floating point), are ignored in the operands and not modified in the result registers.

All arithmetic instructions may specify absolute operation. The sign of the operand in M then is considered positive, but the operand in the A register and/or the R register will use its actual sign for algebraic execution. All but the Short Multiply and the Short Divide instructions may specify floating-point operation, paragraph 4.6. All arithmetic instructions may also designate indexing, indirect addressing, and both PDPM and MASM execution.

The Add instruction (A) algebraically adds the specified ZY field in the designated Core Memory word to the content of the 8-Y most significant character positions of the A register. The Subtract instruction (S) algebraically subtracts the ZY field. The result of the A or S operation is retained in the 8-Y character positions of the

A register. The Add A to M, and Place in R instruction (+) algebraically adds the specified ZY field of the A register to the designated Core Memory word in the 8-Y most significant character positions of the R register. The Subtract A from M and place in R instruction (-) algebraically subtracts the ZY field of the A register. The result of the + or - operation is retained in the 8-Y character positions of the R register. The A, S, + and - instructions all require that the least significant character of both operands occur in corresponding character positions of the word. Any of these instructions can produce an addition or subtraction operation, depending on the sign of both operands. An addition may overflow the most significant character position of the A or R result register and set ON the Arithmetic Overflow Error Interrupt toggle (ZY=W1). A subtraction will subtract the specified ZY field from the 8-Y operand, retaining the sign of 8-Y. Subtracting a larger number from a smaller number provides an underflow past the most significant character position of the result register. An automatic complement cycle then reverses the sign, and subtracts the 8-Y result character positions from zero to produce the true algebraic magnitude and sign.

The Multiply Long instruction (M) multiplies the content of the A register by the specified ZY field in the designated memory word. This instruction essentially provides a fractional multiply operation that left-justifies an 8+Z character algebraic product in the 16-character combined A and R registers. The decimal point is effectively to the left of the A register for the multiplicand, to the left of the ZY field for the multiplier, and to the left of the A register for the product.

The Multiply Short instruction (G) multiplies the content of the four least significant character positions of the A register by the up-to-four characters ZY field in the designated memory word. This instruction essentially provides an integer multiply operation that right-justifies an algebraic product from zero to eight characters long in the A register. The machine decimal point is effectively to the right of the A register for the multiplicand, to the right of the ZY field for the multiplier, and to the right of the A register for the product.

The Divide Long instruction (D) divides the 16-character dividend in the combined A and R registers by the specified ZY field in the designated memory word. This instruction essentially provides a fractional divide operation that left-justifies the quotient in the A register and the remainder in the R register. The decimal point is effectively to the left of the A register for the dividend, to the left of the ZY field for the divisor, to the left of the A register for the quotient, and to the left of the



Section IV  
Paragraph 4. 1. 1

R register for the remainder. If the Z-character divisor is not greater in absolute value than the content of the Z most significant character positions of the dividend, the Arithmetic Overflow Error Interrupt toggle is set ON to terminate the divide operation.

The Divide Short instruction (/) divides the content of the A register by the ZY field in the designated memory word. This instruction essentially provides an integer divide operation that right-justifies the quotient in the A register and the remainder in the R register. The decimal point is effectively to the right of the ZY field for the divisor, to the right of the A register for the quotient, and to the right of the R register for the remainder. There is no requirement on the magnitude of the ZY divisor field relative to the dividend for the integer Divide Short instruction.

The Non-Numeric Error Interrupt Toggle (ZY = W3) is set ON during any arithmetic operation when either one of the following conditions exists:

- a. The specified field contains a character whose numeric bits are other than decimal 0 through 9.
- b. A character with numeric bits other than decimal 0 through 9 is placed in the result register (a malfunction).

4. 1. 1 ADD.

Command Character	A
Instruction	AZYXMMMM
Options	Index, Indirect-Address, Absolute, Floating-Point, Normalize, Tagged Halt, Flag Return, PDPM, MASM
Mnemonic	ADD, ADM

Add the characters of the specified ZY field in the effective operand address M to the content of the 8-Y most significant character positions in the A register. The algebraic sum remains in the A register at the 8-Y most significant positions. The Y least significant character positions of the A register are not modified. The R register contains the word from the effective operand address.

The word in the effective operand address is transferred to the R register. If the signs of the operands are the same, an addition is specified. The sum remains in the 8-Y most significant character positions of the A register. Overflow of the A register word will set ON the Arithmetic Overflow Error Interrupt toggle.

If the signs of the operands are different, a subtraction is specified. The ZY field in the R register is subtracted from the content of the 8-Y most significant character positions of the A register. Underflow will reverse the sign bit in the A register, and provide a complement cycle that produces a correct magnitude.

EXAMPLES:

(1)(a) Memory address M and the R and A registers before execution of instruction A34XMMMM:

	Z = 3				Y = 4			
M	2	3	1	4	1	2	3	4
R	/ / / / / / / /							
A	4	7	7	4	6	6	6	6
	8 - Y				Y			

(b) Memory address M and the R and A registers after instruction execution:

M	2	3	1	4	1	2	3	4
R	2	3	1	4	1	2	3	4
A	5	∅	8	8	6	6	6	6

(2)(a) Memory address M and the R and A registers before execution of instruction A34XMMMM:

	Z = 3				Y = 4			
M	4	5	7	5	6	6	6	6
R	/ / / / / / / /							
A	2	∅	∅	$\bar{\emptyset}$	8	8	8	8
	8 - Y				Y			

Section IV  
 Paragraph 4. 1. 2

(b) Memory address M and the R and A registers after instruction execution:

M	4	5	7	5	6	6	6	6
R	4	5	7	5	6	6	6	6
A	1	4	2	5	8	8	8	8

4. 1. 2 SUBTRACT.

Command Character S

Instruction SZYXMMMM

Options Index, Indirect-Address, Absolute, Floating-Point, Normalize, Tagged Halt, Flag Return, PDPM, MASM

Mnemonic SUB, SUBM

Subtract the characters of the specified ZY field in the effective operand address M from the content of the 8-Y most significant character positions in the A register. The algebraic difference remains in the A register at the 8-Y most significant character positions. The Y least significant character positions of the A register are not modified. The R register contains the word from the effective operand address.

The word in the effective operand address is transferred to the R register. If the signs of the operands are the same, a subtraction is specified. The ZY field in the R register is subtracted from the content of the 8-Y most significant character positions in the A register. The difference, with the sign of the operand initially in the A register, remains in the 8-Y most significant character positions of the A register. Underflow will reverse the sign bit in the A register, and provide a complement cycle that produces a correct magnitude.

If the signs of the operands are different, an addition is specified. The sum, with the sign of the operand initially in the A register, remains in the 8-Y most significant character positions of the A register. Overflow of the A register word will set ON the Arithmetic Overflow Error Interrupt toggle.

EXAMPLES:

(1)(a) Memory address M and the R and A registers before execution of instruction S42XMMMM:

	Z = 4						Y = 2	
M	9	9	7	7	6	6	3	3
R								
A	7	7	6	9	8	8	2	2
	8 - Y						Y	

(b) Memory address M and the R and A registers after instruction execution:

M	9	9	7	7	6	6	3	3
R	9	9	7	7	6	6	3	3
A	7	6	9	2	2	2	2	2

(2)(a) Memory address M and the R and A registers before execution of instruction S31XMMMM:

	Z = 3						Y = 1	
M	A	B	C	D	7	7	8	G
R								
A	∅	∅	∅	∅	5	5	5	6
	8 - Y						Y	

(b) Memory address M and the R and A registers after instruction execution:

M	A	B	C	D	7	7	8	G
R	A	B	C	D	7	7	8	G
A	∅	∅	∅	1	3	3	3	6

4.1.3 ADD A TO M, PLACE IN R.

Command Character +

Instruction +ZYXMMMM

Section IV  
 Paragraph 4.1.3

Options                    Index, Indirect-Address, Absolute, Floating-Point,  
                               Normalize, Tagged Halt, Flag Return, PDPM, MASM

Mnemonic                 RAD, RADM

Add the characters of the specified ZY field in the A register to the content of the 8-Y most significant character positions of the word in the effective operand address M. The algebraic sum remains in the R register at the 8-Y most significant character positions of the R register. The Y least significant positions of the R register contain the corresponding characters of the word in M. The A register is not modified.

The word in memory address M is transferred to the R register. If the signs of the operands are the same, an addition is specified. The sum remains in the 8-Y most significant character positions of the R register. Overflow of the R register word will set ON the Arithmetic Overflow Error Interrupt toggle.

If the signs of the operands are different, a subtraction is specified. The ZY field in the A register is subtracted from the content of the 8-Y most significant character positions in the R register. The difference, with the sign of the operand initially in the R register, remains in the 8-Y most significant character positions of the R register. Underflow will reverse the sign bit in the R register and provide a complement cycle that produces a correct magnitude.

EXAMPLES:

(1)(a) Memory address M and the R and A registers before execution of instruction +24XMMMM:

	8 - Y				Y			
M	2	3	4	3	A	B	C	D
R	/ / / / / / / /							
A	1	5	7	5	1	2	3	4
			Z = 2			Y = 4		

(b) Memory address M and the R and A registers after instruction execution:

M	2	3	4	3	A	B	C	D
R	2	4	1	8	A	B	C	D
A	1	5	7	5	1	2	3	4

(2)(a) Memory address M and the R and A registers before execution of instruction +35XMMMM:

	8 - Y			Y				
M	1	3	8	6	9	6	6	9
R								
A	1	5	5̄	A	B	C	D	E
	Z = 3			Y = 5				

(b) Memory address M and the R and A registers after instruction execution (required complement cycle):

M	1	3	8	6	9	6	6	9
R	∅	1	7̄	6	9	6	6	9
A	1	5	5̄	A	B	C	D	E

#### 4.1.4 SUBTRACT A FROM M, PLACE IN R.

Command Character -

Instruction -ZYXMMMM

Options Index, Indirect-Address, Absolute, Floating-Point, Normalize, Tagged Halt, Flag Return, PDPM, MASM

Mnemonic MINUS, MINM

Subtract the characters of the specified ZY field in the A register from the content of the 8-Y most significant character positions of the word in the effective operand address M. The algebraic difference remains in the R register at the 8-Y most

Section IV  
 Paragraph 4.1.4

significant positions. The Y least significant positions of the R register contain the corresponding characters of the word in M. The A register is not modified.

The word in memory address M is transferred to the R register. If the signs of the operands are the same, a subtraction is specified. The ZY field in the A register is subtracted from the content of the 8-Y most significant character positions in the R register. The difference, with the sign of the operand initially in the R register, remains in the 8-Y most significant character positions of the R register. Underflow will reverse the sign bit in the R register, and provide a complement cycle that produces a correct magnitude.

If the signs of the operands are different, an addition is specified. The sum remains in the 8-Y most significant character positions of the R register. Overflow of the register word will set ON the Arithmetic Overflow Error Interrupt toggle.

EXAMPLES:

(1)(a) Memory address M and the R and A registers before execution of instruction -23XMMMM:

	8 - Y					Y		
M	∅	8	6	4	2	4	5	6
R								
A	2	3	4	5	6	7	8	9
					Z = 2		Y = 3	

(b) Memory address M and the R and A registers after instruction execution:

M	∅	8	6	4	2	4	5	6
R	∅	8	5	8	6	4	5	6
A	2	3	4	5	6	7	8	9

(2)(a) Memory Address M and the R and A registers before execution of instruction -42XMMMM:

	8 - Y						Y	
M	7	5	6	5	4	5	A	B
R	/ / / / / / / /							
A	5	4	6	8	8	$\bar{8}$	7	7
			Z = 4				Y = 2	

(b) Memory address M and the R and A registers after instruction execution:

M	7	5	6	5	4	5	A	B
R	7	6	3	4	3	3	A	B
A	5	4	6	8	8	$\bar{8}$	7	7

#### 4.1.5 MULTIPLY LONG.

Command Character    M  
 Instruction            MZYXMMMM  
 Options                Index, Indirect-Address, Absolute, Floating-Point,  
                           Normalize, Tagged Halt, Flag Return, PDPM, MASM  
 Mnemonic              LMULT, LMULTM

Multiply the 8-character content of the A register by the specified ZY field in the effective operand address M. The ZY field is padded automatically with 8-Z least significant decimal zeros to provide an 8-character multiplier. The 16-character product then remains in the combined A and R registers. The most significant half of the result is contained in the A register, and the least significant half is retained in the R register. The proper algebraic sign of the product is placed in the least significant character position of both the A register and the R register.

#### EXAMPLE:

(1)(a) Memory Address M and the R and A registers before execution of instruction M32XMMMM:



Section IV  
Paragraph 4.1.6

		Z = 5					Y = 2	
M	∅	9	4	∅	2	$\bar{3}$	5	7
R								
A	∅	∅	3	6	4	2	1	$\bar{0}$

(b) Memory Address M and the R and A registers after instruction execution:

M	∅	9	4	∅	2	$\bar{3}$	5	7
R	1	6	8	3	∅	∅	∅	∅
A	∅	∅	3	4	2	4	4	1

4.1.6 MULTIPLY SHORT.

Command Character G

Instruction GZYXMMMM

Options Index, Indirect-Address, Absolute, Tagged Halt, Flag Return, PDPM, MASM

Mnemonic MULT, MULTM

Multiply the content of the four least significant character positions in the A register by the specified ZY field ( $Z \leq 4$ ) in the effective operand address M. The ZY field is padded automatically with 4-Z more significant decimal zeros to provide a 4-character multiplier. The 8-character product then remains in the A register. The proper algebraic sign of the product is placed in the least significant character position of the A register, and the R register contains the word in M.

EXAMPLE:

(1)(a) Memory address M and the R and A registers before execution of instruction G32XMMMM:

		Z = 3				Y = 2		
M	∅	9	4	∅	2	$\bar{3}$	5	7
R								
A	5	2	3	6	4	2	1	$\bar{0}$

(b) Memory address M and the R and A registers after instruction execution:

M	∅	9	4	∅	2	3̄	5	7
R	∅	9	4	∅	2	3̄	5	7
A	∅	∅	∅	9	6	8	3	∅

4. 1. 7 DIVIDE LONG.

Command Character    D

Instruction            DZYXMMMM

Options                Index, Indirect-Address, Absolute, Floating-Point,  
Normalize, Tagged Halt, Flag Return, PDPM, MASM

Mnemonic              LDIV, LDIVM

Divide the 16-character content of the combined A and R registers by the specified ZY field in the effective operand address M. The most significant half of the dividend is initially in the A register. The least significant half, including the sign, is supplied by the R register. The sign in the A register is ignored. The Z-character divisor must be greater in absolute magnitude than the Z most significant character positions of the A register to prevent the Arithmetic Overflow Error Interrupt toggle from being set ON.

The ZY field is padded automatically with 8-Z least significant decimal zeros to provide an 8-character divisor. The 8-character quotient, with proper algebraic sign, then remains in the A register. The 8-character remainder, with the sign of the quotient, is retained in the R register.

EXAMPLE:

(1) (a) M and the R and A registers before execution of instruction D51XMMMM:

			Z = 5					Y=1
M	3	4	6	5	8	9	2̄	7
R	1	9	6	2	4	3	2	5
A	∅	7	2	3	4	1	6	8

Section IV  
 Paragraphs 4. 1. 8 to 4. 1. 8. 2

(b) M and the R and A registers after instruction execution:

M	3	4	6	5	8	9	2̄	7
R	1	6	2	3	2	3	2	5̄
A	1	0	9	7	8	8	2	6

4. 1. 8 DIVIDE SHORT.

Command Character /

Instruction /ZYXMMMM

Options Index, Indirect-Address, Absolute, Tagged Halt, Flag Return, PDPM, MASM

Mnemonic DIV, DIVM

Divide the 8-character content of the A register by the specified ZY field in the effective operand address M. The 8-character quotient, with proper algebraic sign, remains in the A register. The 8-character remainder, with the sign of the quotient, is retained in the R register.

4. 1. 8. 1  $Z > 4$ . When the value of  $Z > 4$ , the content of the A register is transferred to the R register. The ZY field is padded automatically with 8-Z more significant decimal zeros to provide an 8-character divisor. The A register is padded with 8 decimal zeros, and a divide operation similar to Divide Long then is executed.

4. 1. 8. 2  $Z \leq 4$ . When the value of  $Z \leq 4$ , the content of the A register is shifted right four places into the most significant half of the R register. The ZY field is padded automatically with 4-Z more significant decimal zeros and with 4 least significant decimal zeros to provide an 8-character divisor. The most significant half of the A register and the least significant half of the R register are padded with decimal zeros, and a divide operation similar to Divide Long then is executed.

EXAMPLES:

(1)(a)  $Z > 4$

M and the R and A registers before execution of instruction /51XMMMM:

			Z = 5					Y = 1
M	3	4	6	5	8	9	$\bar{2}$	7
R								
A	∅	7	2	3	4	1	6	8

(b) M and the R and A registers after instruction execution:

M	3	4	6	5	8	9	$\bar{2}$	7
R	∅	∅	∅	5	1	8	8	$\bar{0}$
A	∅	∅	∅	∅	∅	1	∅	$\bar{9}$

(2)(a)  $Z \leq 4$

M and the R and A registers before execution of instruction /32XMMMM:

			Z = 3			Y = 2		
M	9	8	6	6	1	5	3	4
R								
A	∅	7	1	3	4	5	6	8

(b) M and the R and A registers after instruction execution:

M	9	8	6	6	1	5	3	4
R	∅	∅	∅	∅	∅	5	6	8
A	∅	∅	∅	1	1	6	∅	∅

#### 4.2 DATA MOVE INSTRUCTIONS.

The ten data move instructions specify and control either the transfer of fields or words between Core Memory and the Central Processor registers, the transmission of data between registers, or the shift of characters within the A register and the combined A and R registers.

The Bring M to A instruction (B) clears the A register to decimal zero, and then loads the specified ZY field of the content of the effectively addressed operand word into corresponding character positions of the A register. The Combine M with A instruction (C) inserts the ZY specified field into the A register without modifying the other character positions. The Hold A to M instruction (H) inserts the ZY specified field of the A register into corresponding character positions of the designated memory word. The transfers between memory and the A register specified by the B, C and H instructions are via the R register.

The Jerk M to R instruction (J) loads the designated Core Memory word into the R register. The Keep R to M instruction (K) transfers the word in the R register to the specified Core Memory address. The J and K instructions provide full parallel word transfers. Together with the indexing capability, they permit efficient multi-word Core Memory to Core Memory transfer operations.

For the five instructions, B, C, H, J and K, the final content of the Register is identical to the final content of the selected word in Core Memory.

The Load M in ZY instruction ( $\oplus$ ) loads the content of the least significant character positions of the designated Core Memory word into the ZY specified register. The Save ZY in M instruction (\$) stores the content of the ZY specified register into the least significant character positions of the designated Core Memory word. The Copy Z to Y instruction (=) copies the content of the  $\emptyset Z$  specified register into the  $\emptyset Y$  selected register. Appendix B, ZY Addressable Registers, designates those registers that may be addressed by the  $\oplus$ , \$, and = instructions. The data transfers employ the Q and R registers, the R Repeater, and the X Counter.

The Left Shift instruction (L) shifts the content of the A register left by Y character positions if  $Z5=\emptyset$ . Decimal zeros are inserted into the least significant end of the A register, and the characters shifted out of the most significant position are dropped. When  $Z5=1$ , the content of the combined A and R registers is shifted left Y character positions. The characters that are shifted out of the most significant end of the R register enter the least significant end of the A register. The Right Shift instruction

(R) shifts the content of the A register right by Y character positions if Z5=0. If the numeric bits of Z also provide an odd decimal digit, the shift is circular or cyclic, and the characters that are shifted out of the least significant end of the A register enter the most significant end of the A register. If Z provides an even decimal digit, decimal zeros are inserted into the most significant end of the A register, and the characters shifted out of the least significant character position are dropped. When Z5=1, the content of the combined A and R registers is shifted right Y character positions. Decimal zeros are inserted into the most significant end of the A register, the characters that are shifted out of the least significant end of the A register enter the most significant end of the R register, and the characters shifted out of the least significant character position of the R register are dropped.

4. 2. 1 BRING M TO A.

Command Character    B  
 Instruction            BZYXMMMM  
 Options                Index, Indirect-Address, Tagged Halt, Flag Return, PDPM, MASM  
 Mnemonic              BRING, CLRA

Bring the specified ZY field of the word in the effective operand address M into the A register. Clear the remaining characters of the A register to decimal zero, and retain the content of memory address M in the R register.

EXAMPLE:

(1)(a) Memory address M and the R and A registers before execution of instruction B43XMMMM:

		Z = 4				Y = 3		
M	A	B	C	D	E	F	G	H
R	/	/	/	/	/	/	/	/
A	1	2	3	4	J	A	C	K

Section IV  
Paragraph 4. 2. 2

(b) Memory address M and the R and A registers after execution:

M	A	B	C	D	E	F	G	H
R	A	B	C	D	E	F	G	H
A	∅	B	C	D	E	∅	∅	∅

4. 2. 2 COMBINE M WITH A.

Command Character C

Instruction CZYXMMMM

Options Index, Indirect-Address, Tagged Halt, Flag Return, PDPM, MASM

Mnemonic COMB

Combine the specified ZY field of the word in the effective operand address M, with the remaining characters of the A register. Retain the content of memory address M in the R register.

EXAMPLE:

(1)(a) Memory address M and the R and A registers before execution of instruction C43XMMMM:

		Z = 4				Y = 3		
M	A	B	C	D	E	F	G	H
R								
A	1	2	3	4	5	6	7	8

(b) Memory address M and the R and A registers after instruction execution:

M	A	B	C	D	E	F	G	H
R	A	B	C	D	E	F	G	H
A	1	B	C	D	E	6	7	8

4. 2. 3 HOLD A TO M.

Command Character    H  
 Instruction            HZYXMMMM  
 Options                Index, Indirect-Address, Tagged Halt, Flag Return,  
                           PDPM, MASM  
 Mnemonic              HOLD

Hold the content of the A register, and insert the specified ZY field of the A register into the content of the effective operand address M. The A register is not modified, and the R register contains the final content of memory address M.

EXAMPLE:

(1)(a) Memory address M and the R and A registers before execution of instruction H43XMMMM:

M	A	B	C	D	E	F	G	H
R								
A	1	2	3	4	5	6	7	8
		Z = 4				Y = 3		

(b) Memory address M and the R and A registers after instruction execution:

M	A	2	3	4	5	F	G	H
R	A	2	3	4	5	F	G	H
A	1	2	3	4	5	6	7	8

4. 2. 4 JERK M TO R.

Command Character    J  
 Instruction            JZYXMMMM    ZY not functional (except Z6)  
 Options                Index, Indirect-Address, Tagged Halt, Flag Return,  
                           PDPM, MASM  
 Mnemonic              JERK



Section IV  
 Paragraph 4. 2. 5

Transfer the content of the effective operand address M to the R register. The A register is not modified.

EXAMPLE:

(1)(a) Memory address M and the R and A registers before execution of instruction J00XMMMM:

M	A	B	C	D	E	F	G	H
R								
A	5	2	4	3	6	7	9	8

(b) Memory address M and the R and A registers after instruction execution:

M	A	B	C	D	E	F	G	H
R	A	B	C	D	E	F	G	H
A	5	2	4	3	6	7	9	8

4. 2. 5 KEEP R TO M.

Command Character K

Instruction KZYXMMMM ZY not functional (except Z6)

Options Index, Indirect-Address, Tagged Halt, Flag Return, PDPM, MASM

Mnemonic KEEP

Transfer the content of the R register to the effective operand address M. The content of the R register and the A register remain unchanged.

EXAMPLE:

(1)(a) Memory address M and the R and A registers before execution of instruction K00XMMMM:

M								
R	A	B	C	D	E	F	G	H
A	5	6	3	2	1	P	E	N

(b) Memory address M and the R and A registers after instruction execution:

M	A	B	C	D	E	F	G	H
R	A	B	C	D	E	F	G	H
A	5	6	3	2	1	P	E	N

4. 2. 6 LOAD M IN ZY.

Command Character   ⊕

Instruction           ⊕ZYXMMMM

Options               Index, Indirect-Address, Tagged Halt, Flag Return, PDPM

Mnemonic             LDXR, LDRG

Transfer the content of the effective operand address M into the register specified by ZY. Refer to Appendix B for ZY addressable registers. The transmission begins at the least significant character positions, and halts on reaching the most significant end of the register. The R register retains the content of M if the ZY register is not in memory; otherwise, the R register retains the final content of the memory word containing the register. The A register is not modified unless it is the addressed ZY register. Use of an unassigned ZY address generates a Register Address Error Interrupt request.

EXAMPLE:

(1)(a) Memory address M, the R register, and address 0003 before execution of instruction ⊕03XMMMM. A ZY = 03 designates IR#3, which is contained in the least significant half of memory location 0003.

M	7	9	3	2	4	8	5	6
R								
0003	6	8	1	5	2	4	3	7
					IR#3			

Section IV  
 Paragraph 4. 2. 7

(b) Memory address M, the R register, and address 0003 after instruction execution:

M	7	9	3	2	4	8	5	6
R	6	8	1	5	4	8	5	6
0003	6	8	1	5	4	8	5	6

4. 2. 7 SAVE ZY IN M.

Command Character \$

Instruction \$ZYXMMMM

Options Index, Indirect-Address, Tagged Halt, Flag Return, PDPM

Mnemonic SVXR, SVRG

Transfer the contents of the register specified by ZY to the effective operand address M. The transmission begins with the least significant character positions, and halts on reaching the most significant end of the register. The R register retains the final content of memory address M, and the A register is not modified.

Refer to Appendix B for ZY addressable registers. Use of an unassigned ZY address generates a Register Address Error Interrupt request.

EXAMPLE:

(1)(a) Memory address M, the R register, and address 0001 before execution of instruction \$11XMMMM. A ZY = 11 designates the Return Address register, which is contained in the most significant half of memory location 0001.

M	9	7	4	1	6	3	2	5
R								
0001	0	1	5	0	7	8	3	4
	Return Address Register							

(b) Memory address M, the R register, and address 0001 after instruction execution:

M	9	7	4	1	0	1	5	0
R	9	7	4	1	0	1	5	0
0001	0	1	5	0	7	8	3	4

#### 4. 2. 8 COPY Z TO Y.

Command Character =

Instruction =ZYXMMMM X and MMMM not functional (except X6)

Options Tagged Halt, Flag Return

Mnemonic CPYRG

Transfer the content of the register specified by Z to the register designated by Y. The registers selected have address of 0Z and 0Y. The transmission begins with the least significant character positions, and halts on reaching the most significant end of the smaller register.

When the Y register is in memory, the R register retains the final content of the full memory word. When the Y register is not in memory, the R register has its least significant character positions modified by the content of the Z register. The A register is not modified unless it is the addressed 0Y register.

Refer to Appendix B for ZY addressable registers. Use of unassigned Z or Y addresses generates a Register Address Error Interrupt request.

#### EXAMPLES:

(1)(a) Memory address 0013, the R register, and address 0014 before execution of instruction =(JXMMMM. A Z = ( specifies the 0( address of the Other Processor Interrupt Entry register, which is contained in the most significant half of memory location 0013. A Y = ] provides the 0 ] address of Index register 14, stored in the least significant half of memory word 0014.

Section IV  
Paragraph 4.2.8

	Other Proces- sor Interrupt Ent. reg.							
0013	0	4	5	0	6	8	3	2
R								
0014	1	9	2	4	7	5	3	8

Index Reg. 14

(b) Memory address 0013, the R register, and address 0014 after instruction execution:

0013	0	4	5	0	6	8	3	2
R	1	9	2	4	0	4	5	0
0014	1	9	2	4	0	4	5	0

(2)(a) Other Processor Interrupt Entry Register Z = ( R register, and Index Register 16 (Y = +) before execution of instruction = (+XMMMM):

	Other Proces- sor Interrupt Ent. reg.							
0013	3	4	6	8	5	7	9	8
R	5	6	7	4	2	3	5	6
	Index Reg. 16				1	2	3	4

(b) Instruction Address register, R register, and Index register 16 after instruction execution:

0013	3	4	6	8	5	7	9	8
R	5	6	7	4	3	4	6	8
	Index Reg. 16				3	4	6	8

4. 2. 9 LEFT SHIFT.

Command Character    L  
 Instruction            LZYXMMMM    X (except X6) and MMMM not functional  
 Options                Tagged Halt, Flag Return  
 Mnemonic              LEFT, LLFT

4. 2. 9. 1 SHORT LEFT SHIFT (Z5 = 0) - LEFT. Shift the content of the A register left by Y character positions, where  $0 \leq Y \leq 8$ . The Y most significant characters of the A register are dropped, and decimal zeros remain in the Y least significant character positions. The R register is not modified.

NOTE

A Short Left Shift with Y = 8 is the fastest way to clear the A register.

4. 2. 9. 2 LONG LEFT SHIFT (Z5 = 1) - LLFT. Shift the content of the combined A and R registers left by Y character positions, where  $0 \leq Y \leq 8$ . The Y most significant characters of the R register shift into the least significant end of the A register, and the Y most significant characters of the A register are dropped. Decimal zeros remain in the Y least significant character positions of the R register.

EXAMPLES:

(1)(a) The A and R registers before execution of instruction L04000000, where Z5 = 0:

R	J	A	B	2	4	5	6	I
A	1	2	5	8	3	0	9	7

(b) The A and R registers after instruction execution:

R	J	A	B	2	4	5	6	I
A	3	0	9	7	0	0	0	0

Section IV  
 Paragraphs 4.2.10 to 4.2.10.2

(2)(a) The A and R registers before execution of instruction L+3000000, where Z5 = 1:

R	J	A	B	2	4	5	6	I
A	1	2	5	8	3	∅	9	7

(b) The A and R registers after instruction execution:

R	2	4	5	6	I	∅	∅	∅
A	8	3	∅	9	7	J	A	B

4.2.10 RIGHT SHIFT.

Command Character R

Instruction RZYXMMMM X and MMMM not functional (except X6)

Options Tagged Halt, Flag Return

Mnemonic RIGHT, LRGT, RYCL

4.2.10.1 SHORT RIGHT SHIFT (Z5 = ∅) - RIGHT, RYCL. Shift the content of the A register right by Y character positions, where  $0 \leq Y \leq 8$ . The R register is not modified.

a. If the numeric bits of Z provide an even decimal digit: The Y least significant characters of the A register are dropped and decimal zeros remain in the Y most significant character positions.

b. If the numeric bits of Z provide an odd decimal digit: The characters right shift out of the least significant end of the A register into the most significant end.

4.2.10.2 LONG RIGHT SHIFT (Z5 = 1) - LRGT. Shift the content of the combined A and R registers right by Y character positions, where  $0 \leq Y \leq 8$ . The Y least significant characters of the A register shift into the most significant end of the R register, and the Y least significant characters of the R register are dropped. Decimal zeros remain in the Y most significant character positions of the A register.

EXAMPLES:

(1)(a) The R and A registers before execution of instruction R24000000, where Z5 = 0 and Z<sub>1-4</sub> is even:

R	J	A	B	2	4	5	6	I
A	1	2	5	8	3	0	9	7

(b) The R and A registers after instruction execution:

R	J	A	B	2	4	5	6	I
A	0	0	0	0	1	2	5	8

(2)(a) The R and A registers before execution of instruction R14000000, where Z5 = 0 and Z<sub>1-4</sub> is odd:

R	J	A	B	2	4	5	6	I
A	1	2	5	8	3	0	9	7

(b) The R and A registers after instruction execution:

R	J	A	B	2	4	5	6	I
A	3	0	9	7	1	2	5	8

(3)(a) The R and A registers before execution of instruction R+3000000, where Z5 = 1:

R	J	A	B	2	4	5	6	1
A	1	2	5	8	3	0	9	7

(b) The R and A registers after instruction execution:

R	0	9	7	J	A	B	2	4
A	0	0	0	1	2	5	8	3



### 4.3 COMPARE INSTRUCTIONS.

The four Compare instructions compare the content of an Index register or a field in the A register with the corresponding character positions in a designated Core Memory location. All but the Index Register Compare instruction may operate also in the Search mode; a search of consecutive Core Memory word or character positions is specified when the Search Mode toggle (ZY=52) is ON. The Alphabetic Compare and the Numeric Compare instructions both search on the same designated field position in increasing memory word locations. The Field Compare instruction searches through consecutive character positions in consecutive words, without regard to word boundaries.

Three comparison toggles may be set, reset, and tested by program. These are the Compare HI (ZY=55), Compare LO (ZY=56), and Compare EQ (ZY=57) toggles, which indicate either the result of a comparison, or the conditions for search. The toggles are reset OFF automatically prior to any compare instruction execution in the Non-Search mode, except for the Field Compare instruction. The proper toggle is then set ON following instruction execution. One or more of the comparison toggles must be set ON by the program prior to initiating an Alphabetic Compare or a Numeric Compare instruction in the Search mode, or a Field Compare instruction in both the Non-Search mode and the Search mode. If not, the Program Instruction Error Interrupt toggle (ZY=X1) is set ON.

A search through increasing character or memory locations will halt on meeting the condition specified by any comparison toggle that is ON, or upon reaching the end of memory, or upon the occurrence of an Interrupt. The final operand address is stored in the X specified Index register, and the content of this address is retained in the R register. A halt due to a match on comparison will set ON a Compare toggle (ZY=3<) and the program continues with the next instruction in normal sequence. The Compare toggle may only be tested by program, and is reset OFF automatically by initiating an Alphabetic, Numeric, or Field Compare instruction in either the Non-Search mode or the Search mode. An end of memory halt sets ON the End of Memory Error Interrupt toggle instead of the Compare toggle. An interrupt halt ends the Search mode.

The Alphabetic Compare instruction (E) provides an alphanumeric comparison between the content of the specified ZY field in the A register and corresponding character positions of the designated memory word. The comparison, which includes the four numeric and two zone bits of each character, assumes an ascending sequence

for code combinations 11 0000  $\rightarrow$  10 1111, 01 0000  $\rightarrow$  01 1111, and 00 0000  $\rightarrow$  00 1111. The E instruction is useful in sorting applications, but should not normally be used for verifying the results of arithmetic operations.

The Numeric Compare instruction (Q) produces an algebraic comparison between the content of the specified ZY field in the A register and the corresponding character positions of the designated memory word. The comparison includes only the decimal value of the numeric bits and the sign bit of the specified fields. It indicates for example, that a +4 > +3, a +5 > -6, a -3 > -7, and a +0 > -0. The Q instruction is the only compare instruction that may specify absolute and/or floating-point operation. Floating-point Q instruction execution, however, will always indicate that a +0 fraction is equal to a -0 fraction.

The Field Compare instruction (: ) searches for the condition specified by the ON status of any of the three comparison toggles. In the Non-Search mode, this instruction tests between the Z most significant characters in the A register and Z consecutive characters in up to two sequential memory words. It starts with the Z most significant characters of the designated memory address, searches toward the least significant end of the word, and may continue with the most significant characters in the next higher memory address. The operation terminates when finding equality or completing eight comparisons. In the Search mode the Field Compare instruction also searches for the condition specified by any of the comparison toggles. It starts with the Z most significant characters in the designated memory address, and will continue until satisfying the compare requirement, or until reaching the end of memory or an Interrupt occurs. The addressable X Counter (ZY=15) will contain the number of comparisons completed (1 to 8, modulo 8) following Field Compare instruction execution.

The Index Register Compare instruction (Y) compares the content of the Index register specified by the Z character with the four least significant characters of the designated memory word. It provides a normal format address comparison, and considers only the decimal value in the three least significant character positions and the binary value of all six bits in the most significant character position.

#### 4. 3. 1 ALPHABETIC COMPARE.

Command Code	E
Instruction	EZYXMMMM
Options	Index, Indirect-Address, Tagged Halt, Flag Return, MASM
Mnemonic	COMPA

The Alphabetic Compare instruction provides a full character-by-character comparison. It assumes the following collation sequence, listed in increasing order:

11 0000 → 11 1111, 10 0000 → 10 1111, 01 0000 → 01 1111, 00 0000 → 00 1111.

A +0 is therefore greater than a -0.

4. 3. 1. 1 NON-SEARCH MODE. The Alphabetic Compare instruction operates in the Non-Search mode if the Search Mode toggle (ZY=52) is OFF. The three comparison toggles (HI, LO and EQ) and the Compare toggle are then reset OFF automatically prior to instruction execution.

Compare the specified ZY field in the A register with the corresponding characters in the effective operand address M. The Compare HI toggle (ZY=55) is set ON if the field in A is greater, the Compare LO toggle (ZY=56) is set ON if the field in A is less, and the Compare EQ toggle (ZY=57) is set ON if the field in A is equal to that in M. The content of M is retained in the R register, and the A register is not modified.

4. 3. 1. 2 SEARCH MODE. The Alphabetic Compare instruction operates in the Search mode if the Search Mode toggle is ON. One or more of the three comparison toggles (HI, LO, EQ) must be set ON by the program prior to initiating the Compare instruction, or the Program Instruction Error Interrupt toggle (ZY=X1) is set ON. The Compare toggle is turned OFF automatically prior to command execution.

Compare the specified ZY field in the A register with the corresponding characters in consecutive and increasing memory locations, beginning with the effective operand address M. Instruction execution is completed when a compare meets the requirements of any comparison toggle that is ON, which sets ON the Compare toggle. The final operand address is stored in Index register X, and the content of this address is retained in the R register. The A register and the status of the comparison toggles are not modified. The search also terminates on reaching the end of memory and sets ON the End of Memory Error Interrupt toggle (ZY=X3). An interrupt ends the Search mode.

#### 4.3.2 NUMERIC COMPARE.

Command Character	Q
Instruction	QZYXMMMM
Options	Index, Indirect-Address, Absolute, Floating-Point, Tagged Halt, Flag Return, MASM
Mnemonics	COMP <sub>N</sub> , COMP <sub>NM</sub>

The Numeric Compare instruction considers binary values the numeric bits and the sign bit of the specified fields. It provides an algebraic comparison with, for example:  $a + 4 > +3$ ,  $a + 5 > -6$ ,  $a - 3 > -7$ , and  $a + \emptyset > -\emptyset$ .

4.3.2.1 NON-SEARCH MODE. The Numeric Compare instruction operates in the Non-Search mode if the Search Mode toggle is OFF. The three comparison toggles (HI, LO, EQ) and the Compare toggle are then reset OFF automatically prior to instruction execution. Compare the specified ZY field in the A register with the corresponding characters in the effective operand address M. The Compare HI toggle is set ON if the field in A is algebraically greater, the Compare LO toggle is set ON if the field in A is algebraically less, and the Compare EQ toggle is set ON if the field in A is equal to that in M. The content of M is retained in the R register, and the A register is not modified.

4.3.2.2 SEARCH MODE. The Numeric Compare instruction operates in the Search mode if the Search Mode toggle is ON. One or more of the three comparison toggles must be set ON by the program prior to initiating the Compare instruction, or the Program Instruction Error Interrupt toggle is set ON. The Compare toggle is reset OFF automatically prior to instruction execution.

Compare the specified ZY field in the A register with the corresponding characters in consecutive and increasing memory locations, beginning with operand address M. Instruction execution is completed when a compare meets the requirement of any comparison toggle that is ON, which turns ON the Compare toggle. The final operand address is stored in Index register X, and the content of this address is retained in the R register. The A register and the status of the Comparison toggle are not modified. The search also terminates on reaching the end of memory, which sets ON the End of Memory Error Interrupt toggle. An interrupt ends the Search mode.

#### 4. 3. 3 FIELD COMPARE, AUTOMATIC SHIFT.

Command Character :  
Instruction :ZYXMMMM Y not functional  
Options Index, Indirect-Address, Tagged Halt, Flag Return, MASM  
Mnemonic FCOMP

The Field Compare instruction provides full character-by-character comparison, using the same collation sequence as the Alphabetic Compare instruction.

4. 3. 3. 1 NON-SEARCH MODE. The Field Compare instruction operates in the Non-Search mode if the Search Mode toggle is OFF. One or more of the three comparison toggles (HI, LO, EQ) must be set ON by the program prior to initiating the Compare instruction, or the Program Instruction Error Interrupt toggle (ZY = X1) is set ON. The Compare toggle is reset OFF automatically prior to instruction execution.

The content of the effective operand address, M is transferred to the R register, and the content of M + 1 is placed in the Q register. Compare for a match between the Z ( $1 \leq Z \leq 8$ ) most-significant characters in the A register and the corresponding characters in the R register. The Compare toggle is set ON if the compare meets the requirement of any comparison toggle that is ON. If not, the Z most significant characters of the A register are compared with the next lower order field of the R and Q registers, where the Q register is treated as a lower order extension of the R register. This repetitive operation continues until either the Compare toggle is set ON, or eight comparisons have been executed. The number of comparisons completed (maximum of 8) is contained in the addressable X counter (ZY=15), and the A register is not modified.

4. 3. 3. 2 SEARCH MODE. The Field Compare instruction operates in the Search mode if the Search Mode toggle is ON. If one or more of the three comparison toggles (HI, LO, EQ) is not set ON by the program prior to initiating the Compare instruction, the Program Instruction Error Interrupt toggle is set ON. The Compare toggle is reset OFF automatically prior to instruction execution.

The content of the effective operand address, M, is transferred to the R register, and the content of M + 1 is placed in the Q register. Compare the Z ( $1 \leq Z \leq 8$ ) most significant characters in the A register with the corresponding characters in the R register. The operation is completed and the Compare toggle set ON if the compare

meets the requirement of any comparison toggle that is ON. If not, the Z most significant characters of the A register are compared with the next lower order field of the R and Q registers, where the Q register is treated as a lower order extension of the R register. This operation continues with M+2, M+3, etc., replacing each full word as it is read out of the Q register into the R register. Execution is complete when a compare sets ON the Compare toggle. The operand address that supplied the most significant character of the matching field is stored in the X Index register, and the X counter contains the number of required comparisons, 1 to 8, modulo 8. The search also terminates on reaching the end of memory by setting On the End of Memory Error Interrupt toggle. An Interrupt terminates the search. The instruction operates in Non-Search mode until a successful comparison occurs or the X Counter reaches 8, terminating the instruction operation. The operand address that supplied the most significant character of the last field compared is stored in the X specified Index Register.

#### 4. 3. 4 INDEX REGISTER COMPARE.

Command Character	Y
Instruction	YZYXMMMM Y not functional
Options	Index, Indirect-Address, Tagged Halt, Flag Return.
Mnemonic	CXM

The Index Register Compare instruction considers the binary value of the numeric bits in character positions 0, 1, and 2, and the binary value of the numeric and zone bits of character position 3. All other bit positions, including sign, are ignored.

The three comparison toggles (HI, LO, EQ) are reset OFF automatically prior to instruction execution. The content of Index register Z is compared with the four least significant characters in the effective operand address M. The Compare HI toggle is set ON if the content of Index register Z is greater, the Compare LO toggle is turned ON if the content of Index register Z is less, and the Compare EQ toggle is turned ON if the content of Index register Z is equal to that in M. The A register is not modified, and the R register retains the content of M.

#### 4. 4 SET-CONVERT INSTRUCTIONS.

The five set-convert instructions specify the set-reset of a designated toggle, bit, or combination of bits, and the binary-decimal conversion of the most significant memory address character.

Section IV

Paragraphs 4. 4. 1 to 4, 4. 1. 2

The Set-Reset toggle instruction (V) specifies that the ZY selected toggle be set ON or reset OFF. The Set-Reset Bit instruction (#) specifies that the ZY selected bit in the A register be set to 1 or to 0. The Logical Or instruction (v) produces a 1 bit in the Y selected character position of the A register whenever there is a 1 in the corresponding bit position of either the Z or Y specified character position of the A register. The Logical And instruction (^) produces a 1 bit in the Y selected character position of the A register whenever there is a 1 in the corresponding bit positions of both the Z and Y specified character positions of the A register. The V, #, v, and ^ instructions facilitate bit manipulation to establish sign and flag conditions.

The Convert instruction (P) will either convert the binary value of character C3 in the A register to a two digit decimal equivalent in C4 and C3, or convert the two digit decimal number in character C3 and character C4 to a single binary character in C3. The P instruction is useful in converting the most significant character of a memory address, whose value is in excess of 9999, to its decimal equivalent prior to subjecting it to an arithmetic operation. The result may then be converted back to the binary-decimal address format used in the instruction word.

4. 4. 1 SET-RESET TOGGLE.

Command Character	V
Instruction	VZYXMMMM X (except X6) and MMMM not functional
Options	Tagged Halt, Flag Return
Mnemonic	SET, RESET

4. 4. 1. 1 RESET TOGGLE (Y6 = 0) -RESET. Reset OFF toggle ZY. Continue with the next instruction in normal sequence.

4. 4. 1. 2 SET TOGGLE (Y6 = 1) -SET. Set ON toggle ZY. Continue with the next instruction in normal sequence.

The list of ZY addressable toggles and lines is given in Appendix C. Addressing an unassigned ZY designation gives an undefined result.

#### 4. 4. 2 SET-RESET BIT.

Command Character	#
Instruction	#ZYXMMMM X (except X6) and MMMM not functional
Options	Tagged Halt, Flag Return
Mnemonic	RBA, SBA, SSP, SSM

The decimal value of the numeric bits in the Z character of the instruction word specifies a character position in the A register ( $0 \leq Z \leq 7$ ). The decimal value of the numeric bits in the Y character of the instruction word selects a bit within that character ( $1 \leq Y \leq 6$ ). ( $Z > 7$  or  $Y > 6$  gives no operation).

4. 4. 2. 1 RESET BIT ( $Z5 = 0$ )—RBA, SSP. Reset bit ZY in the A register to 0. The parity bit is automatically corrected, and no other bit position is modified. Continue with the next instruction in normal sequence.

4. 4. 2. 2 SET BIT ( $Z5 = 1$ )—SBA, SSM. Set bit ZY in the A register to 1. The parity bit is automatically corrected, and no other bit position is modified. Continue with the next instruction in normal sequence.

#### 4. 4. 3 LOGICAL OR.

Command Character	∨
Instruction	∨ZYXMMMM X (except X6) and MMMM not functional
Options	Tagged Halt, Flag Return
Mnemonic	OR

The decimal value of the numeric bits in both the Z character and the Y character of the instruction word each specify a character position in the A register;  $0 \leq (Z \text{ and } Y) \leq 7$ . ( $Z$  or  $Y \geq 8$  leads to no operation.)

Logically OR each numeric and zone bit contained in the Z character position of the A register with the corresponding bit in the Y character position of the A register. The OR operation produces a 1 bit if there is a 1 in either the Z or Y specified corresponding bit positions, and provides a 0 if there is not. Place the result bit-by-bit in the Y character position. The parity bit is automatically corrected, and no other bit position is modified. Continue with the next instruction in normal sequence.



Section IV  
Paragraphs 4. 4. 4 to 4. 4. 5. 2

4. 4. 4 LOGICAL AND.

Command Character	^
Instruction	^ ZYXMMMM X (except X6) and MMMM not functional
Options	Tagged Halt, Flag Return
Mnemonic	AND

The decimal value of the numeric bits in both the Z character and the Y character of the instruction word each specify a character position in the A register;  $0 \leq (Z \text{ and } Y) \leq 7$ . (Z or  $Y \geq 8$  leads to no operation.)

Logically AND each numeric and zone bit contained in the Z character position with the corresponding bit in the Y character position of the A register. The AND operation produces a 1 bit if there is a 1 in both the Z and Y specified corresponding bit positions, and provides a 0 if there is not. Place the result bit-by-bit in the Y character position. The parity bit is automatically corrected, and no other bit position is modified. Continue with the next instruction in normal sequence.

4. 4. 5 CONVERT.

Command Character	P
Instruction	PZYXMMMM Z, Y, X, (except Z6, Y6, X6) and MMMM not functional
Options	Tagged Halt, Flag Return
Mnemonics	CBD, CDB

4. 4. 5. 1 BINARY TO DECIMAL ( $Y_6 = 1$ ) - CBD. Convert the binary value of bits 1-6 of character C3 in the A register to a two digit decimal equivalent. Store the most significant digit in the numeric bits of character C4, retain the least significant digit in character C3, and reset the zone bits of both character positions to 0. No other character position in the A register is modified, and the content of the R register is unchanged.

4. 4. 5. 2 DECIMAL TO BINARY ( $Y_6 = 1$ ) - CDB. Convert the two digit decimal number retained in the numeric bits of character C4 (most significant) and character C3 (least significant) in the A register to an equivalent binary number. Store the result in bits 1-6 of character C3, and reset bits 1-6 of character C4 to 0. No other character position in the A register is modified, and the content of the R register is

unchanged. A converted binary value greater than 63 will set ON the Convert Overflow Error Interrupt toggle (ZY = W4).

#### 4.5 PROGRAM CONTROL INSTRUCTIONS.

The seven program control instructions specify conditional or unconditional program skip, jump, and halt operations. Unless otherwise specified, the program will continue with the new instruction sequence after a jump in program control.

The No Operation instruction (N) advances the program to the next instruction in normal sequence, without executing any operation. It can be useful in deleting an instruction used only during program debugging. The Unconditional Transfer instruction (U) jumps the program to the designated memory address, and unconditionally initiates a new instruction sequence. The N and U instructions also may be utilized in conjunction as a simple means of providing switching points within the program. The branches are set to one of two positions by inserting either the N or U order code into the command character position of the instruction word.

The Test instruction (T) checks for the specified True or False status of the ZY designated line, switch, or toggle. If the stated condition is met, the program jumps to the effective memory address. The program testable toggles and lines are listed in Appendix C.

The Bit Compare instruction (\*) tests for the specified 1 or 0 status of the ZY selected bit in the A register, and jumps the program to the effective operand address if the stated condition is met. Also, it may compare for equality or complete non-equality between the Y selected bits of the Z designated character in the A register and the corresponding bits in the effective operand address. The Bit Compare instruction facilitates the checking of individual or combinations of flags and switches.

The Execute instruction (X) provides execution of a single designated instruction, with unconditional or optional return to the initial program sequence. This capability is useful in program debugging, by permitting the step-by-step execution and monitoring of individual instructions.

The Modify Index Register instruction ( $\square$ ) will algebraically modify the absolute content of the Z selected Index register by Y, where  $-9 \leq Y \leq +9$ . This instruction automatically provides proper binary-decimal address modification, and the zone bits in the three least significant character positions of the Index register are ignored. The program continues with the next instruction in normal sequence if index modification exceeds core memory addressing capacity (0-63,999); otherwise

Section IV  
Paragraphs 4. 5. 1 to 4. 5. 3. 1

the program jumps to the designated memory address. The Modify Index Register instruction, in conjunction with the indexing option, facilitates the repetitive use of program instructions and routines.

The Halt instruction (Z) provides either an unconditional program halt, or a conditional halt dependent on the ZY addressed Breakpoint button on the Control Console being in the ON position. Depressing the Start or Step button on the Control Console, following a halt, jumps the program to the effective memory address. The Z instruction increases debugging flexibility by permitting optional halts and checks at specific points in the program.

#### 4. 5. 1 NO OPERATION.

Command Character	N
Instruction	NZYXMMMM Z, Y, X (except Z6 and X6) and MMMM not functional
Options	Tagged Halt, Flag Return
Mnemonic	NOP

No operation; continue with the next instruction in normal sequence.

#### 4. 5. 2 UNCONDITIONAL TRANSFER.

Command Character	U
Instruction	UZYXMMMM Z (except Z6) and Y not functional
Options	Index, Indirect-Address, Tagged Halt, Flag Return
Mnemonic	JUMP

Unconditionally access the next instruction at the effective operand address M.

#### 4. 5. 3 TEST.

Command Character	T
Instruction	TZYXMMMM
Options	Index, Indirect-Address, Tagged Halt, Flag Return
Mnemonic	TIOF, TION

4. 5. 3. 1 TRANSFER IF FALSE (Y6 = 0) — TIOF. Access the next instruction at the effective memory address if the ZY addressed toggle or line is false; otherwise continue with the next instruction in normal sequence.

4. 5. 3. 2 TRANSFER IF TRUE ( $Y6 = 1$ ) – TION. Access the next instruction at the effective memory address if the ZY addressed toggle or line is true; otherwise continue with the next instruction in normal sequence.

Refer to paragraph 6. 9 for the use of the Test instruction in the I/O Interface.

Refer to paragraph 7. 8. 3 for the use of the Test instruction in the Uni-Record Interface.

The list of ZY addressable toggles and lines is given in Appendix C.

Use of an unassigned ZY address gives an undefined result.

#### 4. 5. 4 BIT COMPARE.

Command Character	*
Instruction	*ZYXMMMM
Options	Index, Indirect-Address, Tagged Halt, Flag Return
Mnemonic	TBAT, TCHM

4. 5. 4. 1 TEST BIT ( $Z5 = \emptyset$ ) – TBAT. The decimal value of the numeric bits in the Z character of the instruction word specifies a character position in the A register ( $\emptyset \leq Z \leq 7$ ). The decimal value of the numeric bits in the Y character of the instruction word selects a bit within that character ( $1 \leq Y \leq 6$ ). (Other values of Z and Y lead to undefined results.)

Test bit ZY in the A register. Access the next instruction at the effective operand address M if the bit is equal to 1; otherwise continue with the next instruction in normal sequence.

4. 5. 4. 2 TEST (COMPARE) CHARACTER ( $Z5 = 1$ ) – TCHM. The decimal value of the numeric bits in the Z character of the instruction word specifies a character position in both the A register and in M ( $\emptyset \leq Z \leq 7$ ). The numeric and zone bits in the Y character of the instruction word control the bit-by-bit comparison between the content of the Z selected character positions. The comparison toggles Equal and the Equal to Complement toggle ( $ZY = 58$ ) are reset OFF automatically prior to instruction execution.

Each numeric and zone bit contained in the Z character position of the A register is compared with the corresponding bit position in memory address M when the equivalent bit position of the Y character in the instruction word is a 1. Those bit positions where the Y character contains a  $\emptyset$  are ignored. The Compare EQ toggle is set ON

Section IV  
Paragraphs 4. 5. 5 to 4. 5. 6

if all the  $Y = 1$  specified bits in the Z character position of the A register and of M are equal. The Equal to Complement toggle is set ON if all the specified bits are unequal. Any other condition will not modify the status of any toggle unless all bits of the Y character are  $\emptyset$ 's in which case both the Equal and Equal to Complement toggles are set ON.

The program continues with the next instruction in normal sequence.

4. 5. 5 EXECUTE.

Command Character	X
Instruction	XZYXMMMM
Options	Index, Indirect-Address, Tagged Halt, Flag Return, PDPM
Mnemonic	XECR, XECA

4. 5. 5. 1 UNCONDITIONAL RETURN ( $Z5 = \emptyset$ ) – XECR. Execute the instruction located at the effective operand address M. Return to the instruction following the Execute instruction.

4. 5. 5. 2 CONDITIONAL RETURN ( $Z5 = 1$ ) – XECA. Execute the instruction located at the effective operand address M. Return to the instruction following the Execute instruction unless the instruction at M produces a program jump (conditionally or unconditionally). If such a jump is specified, transfer program control to the jump address and continue with the new instruction sequence.

4. 5. 5. 3 REMARKS. An Executed instruction does not have the MASM Instruction Modify (bits 15 and 16) and Flag Return (Z6) options. After fetch of the executed instruction, the sequence of operations leading to instruction execution is re-entered at point ② of paragraph 3. 3. 3.

4. 5. 6 MODIFY INDEX REGISTER.

Command Character	<input type="checkbox"/>
Instruction	<input type="checkbox"/> ZYXMMMM
Options	Index, Indirect-Address, Tagged Halt, Flag Return
Mnemonic	TXD, TXI

The numeric bits and the sign bit of the Y character in the instruction word specify an algebraic decimal value of  $-9 \leq Y \leq 9$ . A  $Y6 = 1$  designates a minus sign and a  $Y6 = \emptyset$  provides a plus sign. Bits 1-5 of the Z character in the instruction word

provide a binary value to select index registers XR 1 through XR 17. (A value of Y > 9 in magnitude is treated as a zero.)

The absolute content of Index register Z is modified by the algebraic value of Y. The program continues with the next instruction in normal sequence if the result exceeds Core Memory capacity (less than 0000 or greater than 63,999); otherwise, the program accesses the next instruction at the effective operand address. If no Index register is selected by the Z character (binary value not in the range 1 to 17), the □ instruction terminates after completing all specified program control options.

#### 4. 5. 7 HALT.

Command Character	Z
Instruction	ZZYXMMMM
Options	Index, Indirect-Address, Tagged Halt, Flag Return
Mnemonic	BPHTR, HTR

4. 5. 7. 1 UNCONDITIONAL HALT – HTR. An unconditional halt is specified if the ZY characters in the instruction word are 5□. Depressing the Start or Step button on the Control Console, after a Halt, causes the next instruction to be accessed at the effective operand address.

4. 5. 7. 2 CONDITIONAL HALT – BPHTR. There are 9 Breakpoint buttons in the Control Console. They are labelled 1, 2, 3, 4, 5, 6, 7, 8, 9 and have ZY address of 5/, 5S, 5T, 5U, 5V, 5W, 5X, 5Y, 5Z, respectively. A Halt is specified if the Breakpoint button addressed by the ZY characters of the instruction word has been depressed and is in the ON position. Depressing the Start or Step button on the Control Console, after a Halt, causes the next instruction to be accessed at the effective operand address M.

The Central Processor continues with the next instruction in normal sequence if no Halt is specified. Other ZY addresses give no operation.

#### 4. 6 FLOATING-POINT INSTRUCTIONS.

Floating-point operation is specified by the Add (A), Subtract (S), Add A to M and place in R (+), Subtract A from M and Place in R (-), Multiply Long (M), Divide Long (D), and Numeric Compare (Q) instructions if the Y6 bit in the instruction word is 1. Instruction execution then processes each operand as a floating-point

Section IV  
Paragraph 4.6

number, contained as a signed decimal fraction times a signed power of ten. For floating-point number format refer to paragraph 2.3.5.

Any of the floating-point instructions may designate indexing, indirect addressing, and absolute execution. All but the Numeric Compare instruction (Q) also may specify unnormalized or normalized operation, and the operands compared in Q instruction execution will always be normalized. The absolute and normalize options are selected by zone bits in the ZY characters of the instruction word. The numeric bits of ZY, however, are not functional for the floating-point instructions.

Floating-point A, S, +, and - instruction execution first tests the algebraic difference between the exponents of the two operands. If within specified limits (difference  $\leq 8$ ), the fraction having the algebraically smaller exponent is shifted right until both exponents are equal. The digit 5 is added to the last digit shifted out of the register to provide add-carry rounding.

Each of the A, S, +, and - instructions can specify either an absolute addition or an absolute subtraction operation, depending on the sign of the fraction in the A register. An addition may overflow the most significant character position of the result in the A or R register (without error). For this overflow, the content of the register is automatically shifted right one place, without add-carry rounding, and a 1 is inserted into the most significant fraction position. A subtraction provides an underflow past the most significant character position when subtracting a larger fraction from a smaller fraction. An automatic complement cycle then reverses the sign of the fraction initially in the result register, and subtracts the difference of the fractions from zero to produce the true algebraic magnitude and sign.

The floating-point M instruction provides the 16-digit algebraic product of the fractions in the combined A and R registers. The algebraic sum of the exponents is the exponent in the A register and the sum minus 8 is in the R register. The floating-point D instruction always normalizes the divisor first. Then, this instruction will shift the dividend in the combined A and R registers right one character position, if necessary, to provide a divisor whose fraction has a greater absolute value than the most significant half of the dividend. There is no add-carry rounding. The algebraic difference of the exponent in the A register minus the exponent of the divisor is stored as the exponent in the A register and the difference minus 8 in the R register. The algebraic quotient of the fractions is in the A register. The remainder, with the sign of the quotient, is retained in the R register.

The automatic right shift of the fraction that may occur during execution of the A, S, +, -, or D instructions also algebraically increments the exponent. If the exponent attempts to overflow past +99, the exponent is set to -99 and the Floating-Point Overflow Error Interrupt toggle (ZY = W2) is set ON. The automatic left shift of the fraction that occurs when normalizing an operand and/or the result during execution of any floating-point instruction also algebraically decrements the exponent. If the exponent value attempts to underflow past -99, the exponent is set to +99 and the Floating-Point Overflow Error Interrupt toggle is set ON.

Normalized floating-point execution of the A, S, +, and - instructions will change operands and results that have an all zero fraction to  $+.000000000 \times 10^{-99}$ . Any  $-00$  exponents are changed to  $+00$  during A, S, +, and - instruction execution, if the fraction is not  $\pm 0$ .

Bits 55, 56, 65, 66, 75, and 76 are not altered in the A register during A, S, +, or - instructions, or in the word from memory as it appears in the R register.

#### 4. 6. 1 FLOATING-POINT ADD.

Command Character	A
Instruction	AZYXMMMM Y6 = 1
Options	Index, Indirect-Address, Absolute, Normalize, Tagged Halt, Flag Return, PDPM, MASM
Mnemonics	FAD, UFAD, FADM, UFADM

4. 6. 1. 1 UNNORMALIZED OPERATION (Y5 = 0) — UFAD, UFADM. Add the floating-point number in the effective operand address M to the floating-point number in the A register. The algebraic floating-point sum remains in the A register. The R register initially receives the word in M, but the R register content may be shifted right during floating-point execution.

The word in memory address M is transferred to the R register. The exponents of the floating-point numbers in the A register and the R register are then compared algebraically.

- a. When the exponents differ by more than 8, the greater exponent and its fraction appear as the sum in the A register.
- b. When the exponents differ by less than or equal to 8, the fraction with the smaller exponent is shifted right until the exponents are equal. The digit 5 is added to the last character shifted out of the register to provide add-carry rounding.



Section IV  
Paragraph 4. 6. 1. 2

If the signs of the fractions are the same, the specified addition produces the sum in the A register. Addition of the fractions may overflow the register word. Then, the content of the A register is shifted right one place without rounding, a 1 is inserted into the most significant character position, and the exponent is incremented by 1 algebraically.

If the signs of the fractions are different, the fraction in the R register is subtracted from the fraction in the A register. The difference, with the sign of the fraction initially in the A register, develops in the A register. Underflow will reverse the sign bit of the fraction in the A register, and provide a complement cycle.

4. 6. 1. 2 NORMALIZED OPERATION ( $Y_5 = 1$ ) - FAD, FADM. Add the floating-point number in the effective operand address M to the floating-point number in the A register. The normalized algebraic floating-point sum is in the A register. The R register, which initially receives the word in M, may have its content shifted during floating-point execution.

The word in memory address M is transferred to the R register. Then the exponents of the floating-point numbers in the A register and the R register are compared algebraically.

a. When the exponents differ by more than 15, the greater exponent and its fraction become the sum in the A register. Then the content of the A register is normalized.

b. When the exponents differ by less than or equal to 8, the arithmetic operation proceeds as in paragraph 4. 6. 1. 1 b. Then the sum or difference in the A register is normalized.

c. When the exponents differ by more than 8 but less than or equal to 15, the fraction with the greater exponent is first normalized. If the exponents still differ by more than 8, the greater exponent and its fraction become the normalized sum in the A register. If the exponents differ by less than or equal to 8, the arithmetic operation proceeds as in 4. 6. 1. 1. b. Then the sum or difference in the A register is normalized.

EXAMPLES:

(1) Unnormalized Operation.

(a) Memory address M and the R and A registers before execution of instruction AZYXMMMM ( $Y_6 = 1$  and  $Y_5 = \emptyset$ ):

M	2	3	6	$\frac{+}{5}$	1	7	9	2	$\times 10^{-03}$
R									
A	$\emptyset$	6	7	$\frac{+}{6}$	3	5	2	1	$\times 10^{+02}$

(b) Memory address M and the R and A registers after instruction execution:

M	2	3	6	$\frac{+}{5}$	1	7	9	2	$\times 10^{-03}$
R	$\emptyset$	$\emptyset$	$\emptyset$	$\frac{+}{\emptyset}$	$\emptyset$	2	3	7	$\times 10^{+02}$
A	$\emptyset$	6	7	$\frac{+}{6}$	3	7	5	8	$\times 10^{+02}$

(2) Normalized Operation.

(a) Memory address M and the R and A registers before execution of instruction AZYXMMMM (Y6 = 1 and Y5 = 1):

M	9	2	5	$\frac{+}{1}$	7	3	8	4	$\times 10^{-09}$
R									
A	$\emptyset$	$\emptyset$	3	$\frac{-}{8}$	4	2	6	1	$\times 10^{-06}$

(b) Memory address M and the R and A registers after instruction execution:

M	9	2	5	$\frac{+}{1}$	7	3	8	4	$\times 10^{-09}$
R	$\emptyset$	$\emptyset$	$\emptyset$	$\frac{+}{9}$	2	5	1	7	$\times 10^{-06}$
A	2	9	1	$\frac{-}{7}$	4	4	$\emptyset$	$\emptyset$	$\times 10^{-08}$

4.6.2 FLOATING-POINT SUBTRACT.

Command Character S

Instruction SZYXMMMM Y6 = 1

Section IV  
Paragraphs 4.6.2.1 to 4.6.2.2

Options                      Index, Indirect-Address, Absolute, Normalize, Tagged  
                                Halt, Flag Return, PDPM, MASM

Mnemonics                    FSB, UFSB, FSBM, UFSBM

4.6.2.1 UNNORMALIZED OPERATION ( $Y5 = \emptyset$ ) – UFSB, UFSBM. Subtract the floating-point number in the effective operand address M from the floating-point number in the A register. The algebraic floating-point difference remains in the A register. After the R register initially receives the word in M, the R register content may be right shifted during floating-point execution.

The word in memory address M is transferred to the R register. Then the exponents of the floating-point numbers in the A register and the R register are compared algebraically.

a. When the exponents differ by more than 8, the greater exponent and its fraction with appropriate sign become the difference in the A register.

b. When the exponents differ by less than or equal to 8, the fraction with the smaller exponent is shifted right until the exponents are equal. The digit 5 is added to the last character shifted out of the register, to provide add-carry rounding.

If the signs of the fractions are the same, the fraction in the R register is subtracted from the fraction in the A register. The difference, with the sign of the fraction initially in the A register, appears in the A register. Underflow will reverse the sign bit of the fraction in the A register, and provide a complement cycle.

If the signs of the fractions are different, the specified addition develops the sum in the A register. Addition of the fractions may overflow the register word. When overflow occurs, the content of the A register is shifted right one place without rounding, a 1 is inserted into the most significant character position, and the exponent is incremented by 1 algebraically.

4.6.2.2 NORMALIZED OPERATION ( $Y5 = 1$ ) – FSB, FSBM. Subtract the floating-point number in the effective operand address M from the floating-point number in the A register. The normalized algebraic floating-point difference remains in the A register. After the R register initially receives the word in M, the R register content may be shifted during floating-point execution.

The word in memory address M is transferred to the R register. Then the exponents of the floating-point numbers in the A register and the R register are compared algebraically.

a. When the exponents differ by more than 15, the greater exponent and its fraction become the difference in the A register with the appropriate sign. Then the content of the A register is normalized.

b. When the exponents differ by less than or equal to 8, the arithmetic operation proceeds as in paragraph 4.6.2.1 b. Then the result in the A register is normalized.

c. When the exponents differ by more than 8 but less than or equal to 15, the fraction with the greater exponent is first normalized. If the exponents still differ by more than 8, the greater exponent and its fraction with appropriate sign become the difference in the A register. If the exponents differ by less than or equal to 8, the arithmetic operation proceeds as in paragraph 4.6.2.1 b. Then the result in the A register is normalized.

EXAMPLES:

(1) Unnormalized Operation.

(a) Memory address M and the R and A registers before execution of instruction SZYXMMMM (Y6 = 1 and Y5 = 0):

M	2	3	6	$\frac{+}{5}$	1	7	9	2	$\times 10^{-03}$
R									
A	0	6	7	$\frac{+}{6}$	3	5	2	1	$\times 10^{+02}$

(b) Memory address M and the R and A registers after instruction execution:

M	2	3	6	$\frac{+}{5}$	1	7	9	2	$\times 10^{-03}$
R	0	0	0	$\frac{+}{0}$	0	2	3	7	$\times 10^{+02}$
A	0	6	7	$\frac{+}{6}$	3	2	8	4	$\times 10^{+02}$

(2) Normalized Operation, Absolute.

(a) Memory address M and the R and A registers before execution of instruction SZYXMMMM (Y6 = 1, Y5 = 1, and Z5 = 1):

Section IV  
 Paragraphs 4. 6. 3 to 4. 6. 3. 1

M	∅	∅	7	$\bar{6}$	5	4	3	2	$\times 10^{-06}$
R									
A	∅	∅	∅	$\bar{\emptyset}$	∅	3	2	1	$\times 10^{+04}$

(b) Memory address M and the R and A register after instruction execution:

M	∅	∅	7	$\bar{6}$	5	4	3	2	$\times 10^{-06}$
R	∅	∅	∅	∅	∅	∅	∅	8	$\times 10^{-01}$
A	3	2	1	$\bar{\emptyset}$	∅	∅	∅	8	$\times 10^{-01}$

4. 6. 3 FLOATING-POINT ADD A TO M AND PLACE IN R.

Command Character +

Instruction +ZYXMMMM Y6 = 1

Options Index, Indirect-Address, Absolute, Normalize, Tagged Halt, Flag Return, PDPM, MASM

Mnemonics FRAD, UFRAD, FRADM, UFRADM

4. 6. 3. 1 UNNORMALIZED OPERATION (Y5 = ∅) - UFRAD, UFRADM. Add the floating-point number in the A register to the floating-point number in the effective operand address M. The algebraic floating-point sum remains in the R register. The initial content of the A register may be right shifted during floating-point execution.

The word in memory address M is transferred to the R register. The exponents of the floating-point numbers in the A register and the R register are then compared algebraically.

a. When the exponents differ by more than 8, the greater exponent and its fraction become the sum in the R register.

b. When the exponents differ by less than or equal to 8, the fraction with the smaller exponent is shifted right until the exponents are equal. The digit 5 is added to the last character shifted out of the register to provide add-carry rounding.

If the signs of the fraction are the same, the specified addition places the sum in the R register. Addition of the fractions may overflow the register word. When an overflow occurs, the content of the R register is shifted right one place, a 1 is inserted into the most significant character position, and the exponent is incremented by 1 algebraically.

If the signs of the fraction are different, the fraction in the A register is subtracted from the fraction in the R register. The difference, with the sign of the fraction initially in the R register, develops in the R register. Underflow will reverse the sign bit of the fraction in the R register and provide a complement cycle.

4. 6. 3. 2 NORMALIZED OPERATION (Y5 = 1) - FRAD, FRADM. Add the floating-point number in the effective operand address M. The normalized algebraic floating-point sum develops in the R register. The initial content of the A register may be shifted during floating-point execution.

The word in memory address M is transferred to the R register. The exponents of the floating-point numbers in the A register and the R register then are compared algebraically.

- a. When the exponents differ by more than 15, the greater exponent and its fraction become the sum in the R register. The content of the R register then is normalized.
- b. When the exponents differ by less than or equal to 8, the arithmetic operation proceeds as in paragraph 4. 6. 3. 1b. The result in the R register then is normalized.
- c. When the exponents differ by more than 8 but less than or equal to 15, the fraction with the greater exponent is first normalized. If the exponents still differ by more than 8, the greater exponent and its fraction become the sum in the R register. If the exponents differ by less than or equal to 8, the arithmetic operation proceeds as in paragraph 4. 6. 3. 1b. The result in the R register is then normalized.

EXAMPLES:

(1) Unnormalized Operation.

- (a) Memory address M and the R and A registers before execution of instruction +ZYXMMMM (Y6 = 1 and Y5 =  $\emptyset$ ):

Section IV  
Paragraph 4.6.4

M	0	6	7	$\bar{6}$	3	7	5	8	$\times 10^{+02}$
R									
A	2	3	6	$\bar{5}$	1	7	9	2	$\times 10^{-03}$

(b) Memory address M and the R and A registers after instruction execution:

M	0	6	7	$\bar{6}$	3	7	5	8	$\times 10^{+02}$
R	0	6	7	$\bar{6}$	3	9	9	5	$\times 10^{+02}$
A	0	0	0	$\bar{0}$	0	2	3	7	$\times 10^{+02}$

(2) Normalized Operation.

(a) Memory address M and the R and A registers before execution of instruction +ZYXMMMM (Y6 = 1 and Y5 = 1):

M	0	0	0	$\bar{1}$	2	3	4	5	$\times 10^{+01}$
R									
A	8	7	6	$\bar{5}$	4	3	2	1	$\times 10^{-03}$

(b) Memory address M and the R and A registers after instruction execution:

M	0	0	0	$\bar{1}$	2	3	4	5	$\times 10^{+01}$
R	2	1	1	$\bar{1}$	0	0	0	0	$\times 10^{-02}$
A	0	0	0	$\bar{0}$	8	7	6	5	$\times 10^{+01}$

4.6.4 FLOATING-POINT SUBTRACT A FROM M AND PLACE IN R.

Command Character -

Instruction -ZYXMMMM Y6 = 1

Options Index, Indirect-Address, Absolute, Normalize, Tagged  
Halt, Flag Return, PDPM, MASM

Mnemonics FMIN, UFMIN, FMINM, UFMINM

4. 6. 4. 1 UNNORMALIZED OPERATION ( $Y5 = \emptyset$ ) - UFMIN, UFMINM. Subtract the floating-point number in the A register from the floating-point number in the effective operand address M. The algebraic floating-point difference develops in the R register. The initial content of the A register may be right shifted during floating-point execution.

The word in memory address M is transferred to the R register. The exponents of the floating-point numbers in the A register and the R register then are compared algebraically.

a. When the exponents differ by more than 8, the greater exponent and its fraction with the appropriate sign become the difference in the R register.

b. When the exponents differ by less than or equal to 8, the fraction with the smaller exponent is shifted right until the exponents are equal. The digit 5 is added to the last character shifted out of the register to provide add-carry rounding.

If the signs of the fraction are the same, the fraction in the A register is subtracted from the fraction in the R register. The difference, with the sign of the fraction initially in the R register, develops in the R register. Underflow will reverse the sign bit of the fraction in the R register, and provide a complement cycle.

If the signs of the fractions are different, the sum of the specified addition develops in the R register. Addition of the fractions may overflow the register word. When overflow occurs, the content of the R register is shifted right one place without rounding, a 1 is inserted into the most significant character position, and the exponent is incremented by 1 algebraically.

4. 6. 4. 2 NORMALIZED OPERATION ( $Y5 = 1$ ) - FMIN, FMINM. Subtract the floating-point number in the A register from the floating-point number in the effective operand address M. The normalized algebraic floating-point difference develops in the R register. The initial content of the A register may be shifted during floating-point execution.

The word in memory address M is transferred to the R register. The exponents of the floating-point numbers in the A register and the R register then are compared algebraically.



Section IV  
Paragraph 4.6.4.2

a. When the exponents differ by more than 15, the greater exponent and its fraction with the appropriate sign become the difference in the R register. The content of the R register then is normalized.

b. When the exponents differ by less than or equal to 8, the arithmetic operation proceeds as in paragraph 4.6.4.1b. The result in the R register then is normalized.

c. When the exponents differ by more than 8 but less than or equal to 15, the fraction with the greater exponent is first normalized. If the exponents still differ by more than 8, the greater exponent and its fraction with appropriate sign become the difference in the R register. If the exponents differ by less than or equal to 8, the arithmetic operation proceeds as in paragraph 4.6.4.1b. The result in the R register then is normalized.

EXAMPLES:

(1) Unnormalized Operation.

(a) Memory address M and the R and A registers before execution of instruction -ZYXMMMM (Y6 = 1 and Y5 = 0):

M	9	5	8	+	2	6	3	7	9	$\times 10^{+06}$
R										
A	8	7	2	-	5	3	6	1	4	$\times 10^{+05}$

(b) Memory address M and the R and A registers after instruction execution:

M	9	5	8	+	2	6	3	7	9	$\times 10^{+06}$
R	1	0	4	+	5	5	1	7	4	$\times 10^{+07}$
A	0	8	7	-	2	5	3	6	1	$\times 10^{+06}$

(2) Normalized Operation.

(a) Memory address M and the R and A registers before execution of instruction -ZYXMMMM (Y6 = 1 and Y5 = 1):

M	6	5	8	$\bar{2}$	6	3	7	9	$\times 10^{-06}$
R									
A	$\emptyset$	7	2	$\bar{5}$	3	6	1	4	$\times 10^{-04}$

(b) Memory address M and the R and A registers after instruction execution (required complement cycle):

M	6	5	8	$\bar{2}$	6	3	7	9	$\times 10^{-06}$
R	6	5	9	$\bar{5}$	3	5	$\emptyset$	$\emptyset$	$\times 10^{-05}$
A	$\emptyset$	7	2	$\bar{5}$	3	6	1	4	$\times 10^{-04}$

4. 6. 5 FLOATING-POINT MULTIPLY LONG.

Command Character      M

Instruction              MZYXMMMM    Y6 = 1

Options                  Index, Indirect-Address, Absolute, Normalize, Tagged  
Halt, Flag Return, PDPM, MASM

Mnemonics              FMLT, UFMLT, FMLTM, UFMLTM

4. 6. 5. 1 UNNORMALIZED OPERATION (Y5 =  $\emptyset$ ) – UFMLT, UFMLTM. Multiply the floating-point number in the A register by the floating-point number in the effective operand address M. The algebraic sum of the exponents becomes the exponent in the A register and the sum minus 8 in the R register. The 16-character algebraic product of the fractions develops in the combined A and R registers. The sign and most significant half of the product is contained in the A register, and the same sign and least significant half of the product is retained in the R register.

4. 6. 5. 2 NORMALIZED OPERATION (Y5 = 1) – FMLT, FMLTM. The multiply operation proceeds as above, but the final result in the combined A and R registers then is normalized. The A and R registers will contain the same fraction sign, and exponents as above possibly modified by the normalization.

EXAMPLES:

(1) Unnormalized Operation.

(a) Memory address M and the R and A registers before execution of instruction MZYXMMMM (Y6 = 1 and Y5 =  $\emptyset$ ):

Section IV  
Paragraph 4.6.5.2

M	∅	∅	5	$\bar{9}$	6	3	7	4	$\times 10^{+02}$
R									
A	∅	8	9	$\bar{1}$	4	2	6	1	$\times 10^{-03}$

(b) Memory address M and the R and A registers after instruction execution:

M	∅	∅	5	$\bar{9}$	6	3	7	4	$\times 10^{+02}$
R	3	3	4	$\bar{8}$	9	6	1	4	$\times 10^{-09}$
A	∅	∅	∅	$\bar{5}$	3	1	6	2	$\times 10^{-01}$

(2) Normalized Operation.

(a) Memory address M and the R and A registers before execution of instruction MZYXMMMM. (Y6 = 1 and Y5 = 1):

M	∅	∅	5	9	6	3	7	4	$\times 10^{+02}$
R									
A	∅	8	9	$\bar{1}$	4	2	6	1	$\times 10^{-03}$

(b) Memory address M and the R and A registers after instruction execution:

M	∅	∅	5	$\bar{9}$	6	3	7	4	$\times 10^{+02}$
R	8	9	6	$\bar{1}$	4	∅	∅	∅	$\times 10^{-12}$
A	5	3	1	$\bar{6}$	2	3	3	4	$\times 10^{-04}$

#### 4.6.6 FLOATING-POINT DIVIDE LONG.

Command Character	D
Instruction	DZYXMMMM Y6 = 1
Options	Index, Indirect-Address, Absolute, Normalize, Tagged Halt, Flag Return, PDPM, MASM
Mnemonics	FDIV, UFDIV, FDIVM, UFDIVM

4.6.6.1 UNNORMALIZED OPERATION ( $Y5 = \emptyset$ ) – UFDIV, UFDIVM. Divide the 16-character floating-point number in the combined A and R registers by the floating-point number in the effective operand address M. The most significant half of the dividend, including the sign of the fraction and the signed exponent, is initially in the A register. The least significant half of the fraction is initially in the R register, but its sign and exponent positions are ignored.

The divisor first is normalized, and the absolute value of its fraction then is tested to determine if it is greater than the absolute value of the fraction in the A register. The content of the combined A and R registers will be shifted right one position, if necessary, to meet this requirement. The exponent of the divisor is subtracted from the exponent of the dividend, and the algebraic difference as the signed exponent is placed in the A register and the difference minus 8 is placed in the R register. The 8-character quotient of the fractions, with proper algebraic sign, develops in the A register. The 8-character remainder, with the sign of the quotient, is retained in the R register.

4.6.6.2 NORMALIZED OPERATION ( $Y5 = 1$ ) – FDIV, FDIVM. The divide operation proceeds as above, except that the content of the combined A and R registers is normalized prior to the test of the absolute value of the fractions. The content of the A and R registers then is shifted right one position, if necessary. These actions produce a normalized quotient in the A register with a remainder in the R register that may or may not be normalized.

#### EXAMPLES:

##### (1) Unnormalized Operation.

(a) Memory address M and the R and A registers before execution of instruction DZYXMMMM ( $Y6 = 1$  and  $Y5 = \emptyset$ ):

Section IV  
Paragraph 4. 6. 7

M	∅	∅	∅	$\bar{7}$	8	2	4	1	$\times 10^{+02}$
R	4	3	5	7	4	1	6	7	
A	∅	∅	∅	$\bar{2}$	7	6	8	9	$\times 10^{-04}$

(b) Memory address M and the R and A registers after instruction execution:

M	∅	∅	∅	$\bar{7}$	8	2	4	1	$\times 10^{+02}$
R	7	2	7	$\bar{8}$	5	1	6	7	$\times 10^{-11}$
A	∅	∅	∅	$\bar{3}$	5	3	8	9	$\times 10^{-03}$

(2) Normalized Operation.

(a) Memory address M and the R and A registers before execution of instruction DZYXMMMM (Y6 = 1 and Y5 = 1):

M	∅	∅	4	$\bar{3}$	6	2	1	8	$\times 10^{-02}$
R	4	7	3	1	6	2	5	7	
A	∅	∅	∅	$\bar{6}$	1	2	5	9	$\times 10^{+07}$

(b) Memory address M and the R and A registers after instruction execution:

M	∅	∅	4	$\bar{3}$	6	2	1	8	$\times 10^{-02}$
R	∅	9	7	$\bar{3}$	6	9	∅	∅	$\times 10^{-07}$
A	1	4	0	$\bar{4}$	3	3	1	6	$\times 10^{+01}$

4. 6. 7 FLOATING-POINT NUMERIC COMPARE.

Command Character Q

Instruction QZYXMMMM Y6 = 1

Options Index, Indirect-Address, Absolute, Tagged Halt, Flag Return

Mnemonics COMPF, COMPFM

The Numeric Compare instruction provides an algebraic comparison of the specified floating-point numbers. As examples,  $a + .4 \times 10^{+03} > +.5 \times 10^{+02}$ ,  $a + .5 \times 10^{-03} > -.7 \times 10^{+06}$ ,  $a + .0 \times 10^{+25} = -.0 \times 10^{-30}$ , and  $a + .5 \times 10^{+00} = +.5 \times 10^{-00}$ .

4. 6. 7. 1 NON-SEARCH MODE. The Numeric Compare instruction operates in the Non-Search mode if the Search Mode toggle (ZY = 52) is OFF. The three comparison toggles (HI, LO, EQ) and the Comparison toggle then are set OFF automatically prior to instruction execution.

The floating-point number in the effective operand address M is transferred to the R register and the content of both the A register and of the R register is normalized. Compare the normalized floating-point number in the A register with the normalized floating-point number in the R register. The Compare HI toggle is set ON if the number in A is algebraically greater, the Compare LO toggle is set ON if the number in A is algebraically less, and the Compare EQ toggle is set ON if the number in A is equal to that in R.

4. 6. 7. 2 SEARCH MODE. The Numeric Compare instruction operates in the Search mode if the Search Mode toggle is ON. If one or more of the three comparison toggles is not set ON by the program prior to initiating the compare instruction, the Program Instruction Error Interrupt toggle is set ON. The Compare toggle is set OFF automatically prior to instruction execution.

The floating-point number in the effective operand address M is transferred to the R register, and the content of both the A register and of the R register is normalized. Compare the normalized floating-point number in the A register with that in the R register. The operation is completed by setting ON the Compare toggle if the compare meets the requirement of any comparison toggle that is ON. If not, the content of M+1, M+2, etc., is transferred in sequence to the R register, normalized, and compared with the content of the A register. Instruction execution is completed when a comparison satisfies a comparison toggle, turning on the Compare toggle. The final operand address is stored in Index register X, and the normalized content of this address is retained in the R register. When the Search reaches the end of memory, the operation is terminated and the End of Memory Error Interrupt toggle is set ON. Also, any Interrupt terminates the Search operation and the address of the last word compared is stored in Index register X.



## SECTION V

### I/O INTERFACE

The I/O Interface executes Input (I) and Output (O) instructions addressed to real-time devices. Real-time devices are the L-119 Buffer Processors, Section VI, and the Display Consoles, Section XV. Input and Output instructions are processed first in the Central Processor Instruction register for any memory address modification that may be specified. They are then transferred to the I/O Interface on the basis of the ZY selection address. The I/O Interface executes the instructions independently of the Central Processor program execution which continues after the instruction is transferred to the Interface. Figure E-1 shows the I/O Interface in block diagram form.

#### 5. 1 I/O BUFFER REGISTER.

The I/O Buffer register is a full-word static register that communicates with the Core Memory in full-word parallel. For input, this register assembles a full word for transfer to storage. It is loaded bit-serially and a partial word will be padded with blank characters.

#### 5. 2 CHARACTER HOLDER.

The Character Holder is a two-character static register that communicates character-serially with the I/O Buffer register and bit-serially with selected real-time devices. It has sufficient capacity to continue transmission at the maximum rate while the I/O Buffer register is transferring a word to or from Core Memory.

#### 5. 3 SCANNER.

The Scanner is a two level counter that selects a device that has a message available. This Scanner looks at all three L-119 Module positions and one of the Display Console positions during one complete cycle. On each succeeding cycle, the Scanner advances one Display Console position until a maximum of 25 Display Console positions have been scanned.

The Scanner runs at Central Processor clock rate until it stops at a device with a Message Available signal. Stopping of the Scanner at a Display Console position, however, is inhibited by the set condition of the Ignore Display Console Message



Section V  
Paragraphs 5.4 thru 5.6

Available toggle (ZY = 70). The Scanner may be advanced one position by a Set command with ZY = 38, after which it may run again. The Scanner is not active during an Input or Output operation, or when an I/O Interface Error Interrupt signal exists that is not ignored at I/O Interface Error subclass level.

To determine the address of the device selected, the Scanner content may be addressed by a Save instruction (ZY = 1w), and stored in Core Memory. It will appear as the device ZY address in the ZY characters of a word otherwise cleared to zeros.

Refer to paragraph 5.7 for further details on the operation of the Scanner.

#### 5.4 I/O INSTRUCTION REGISTER.

The I/O Instruction register holds the Input or Output instruction while it is being executed (and until replaced by another instruction). It has a ZY Holder for device selection. It has a Command Character Holder. It has an Operand Address Holder that counts its way through memory addresses. During or after instruction execution the content of the I/O Instruction register may be stored in Core Memory by a Save instruction. When this is done, an instruction word is regenerated containing an I or O command character, the ZY device address and the next MMMM memory address to be used. As parity is not held in the I/O Instruction register, it is regenerated in the R Repeater.

#### 5.5 I/O INSTRUCTION LOCATION REGISTER.

The I/O Instruction Location register is a four-character static register that is loaded with the address of each Input or Output instruction as the instruction is sent to the Interface. During or after instruction execution the content of the I/O Instruction Location register may be stored in Core Memory by a Save instruction.

#### 5.6 OUTPUT OPERATION.

An Output operation is initiated when an Output instruction is transferred to the I/O Interface from the Central Processor Instruction register. When the I/O Interface is Busy, the Central Processor will wait. A memory cycle is initiated to get the first word into the I/O Buffer register. When an Output Priority signal is sent, the ZY selected device must respond with a Device Available signal. Then the first bit of the first character is put on the Data line and the device is sent an Output Operation signal. The device sends its own clock pulses to the Central Processor when the device is ready to receive successive data bits.

An EOM (⌵) character recognized by the device terminates the transmission. In the L-119 program, the EOM character can be ignored within a data block, and then recognized in a control field. When the device sends a signal indicating reception of a valid EOM character, the transmission terminates. Transmission and termination are under control of the device clock. Two device clock pulses are required after the EOM signal to shut down transmission and render the I/O Interface Not Busy.

An I/O Interface Not Busy Interrupt request is generated when an Output operation terminates. Immediately any Message Available Interrupt request from the Scanner is generated, paragraph 5. 7.

Transmitted data is checked in the Core Memory and at the device for correct character parity. Detection of incorrect parity is signaled by termination of transmission without an EOM Recognition signal. Appropriate error interrupt requests are generated at the time the Interface goes Not Busy, paragraph 5. 9.

#### 5. 7 INPUT OPERATION.

A Message Available input request signal appears from the Input L-119 program when a block of data has been received. The Display Console operator generates Message Available input requests when he pushes any of several buttons. These input requests appear to the I/O Interface as Message Available signals and are received by the Scanner.

If the Interface is Not Busy and any error conditions have been removed or ignored, the Scanner will step until it selects a Message Available signal. When this happens, an Input Interrupt request is generated for either a Display Console or an L-119. The Scanner is not operative and is held stationary while the Interface is Busy; therefore, another Message Available signal can not initiate selection until after the Interface goes Not Busy.

The Scanner position can be noted by the Central Processor program after an Input interrupt by storing it in Core Memory with a Save instruction. It is not necessary to address the available device with the Input instruction. Rather, any device address, and the Input Any address, will be replaced by the Scanner position in the I/O Instruction register when the Input instruction is transferred to the I/O Interface. Subsequent recall of the instruction will show the ZY address of the actual device selected.

The Input operation is initiated when an Input instruction is transferred to the I/O Interface from the Central Processor Instruction register. When the I/O Interface

is Busy, the Central Processor will wait. After an Input Operation signal is sent to the device, and when it can, the device sends clock pulses and data bit-serially. Bits are assembled into characters in the Character Holder; Assembled characters are shifted into the I/O Buffer register; and the words are transferred to Core Memory, starting at the effective operand address in the Input instruction. This address may have been modified by any of several options while it was in the Central Processor Instruction register.

When an EOM character is recognized in the Character Holder, transmission ends. If the Scanner has selected an L-119, the EOM character will not be recognized in the first 10 words (80 characters), but only in a subsequent control field, making a full 64 data character code available. After the Interface sends a signal indicating reception of a valid EOM character, the transmission terminates. The last word, if not complete, is padded to the LSD with blank characters, and is stored. Two device clock pulses are required after the EOM signal to shut down transmission, and the Interface goes Not Busy as soon as the last Core Memory cycle is complete.

An I/O Interface Not Busy Interrupt request is generated when an Input operation terminates.

Transmitted data is checked by the device as it is sent and in the Core Memory. Detection of incorrect character parity is signaled by termination of transmission without an EOM recognition signal. Appropriate error interrupt requests are generated at the time the Interface goes Not Busy.

#### 5.8 ERROR CONSIDERATION.

An I/O Interface Error Interrupt request will terminate the operation immediately unless the interrupt request is ignored at the I/O Interface Error level. After an operation is terminated because of error (or without error), the I/O Instruction register contains the address where the next word would have been located. If bad parity occurs in the last character of a word during input, the word with bad parity may already be in Core Memory before the Interface stops the operation.

Words are checked for character parity as they are transferred to and from Core Memory. When bad parity is detected, an M Register Error Interrupt request is generated. If an error occurs as an Input or Output instruction is read from Core Memory, or is index modified, it generates a Computer Error Interrupt signal. A subsequent error during data transfer generates an Interface Error Interrupt signal.

Should the I/O Instruction register count up to a non-existent Core Memory address, the operation is terminated just as with a parity error, and an I/O Interface Operand Address Error Interrupt request is generated.

Should, for any reason, an operation fail to terminate normally or by detected error, a timed disconnect will occur that causes an I/O Interface Hang-Up Error Interrupt request to be generated. A hang-up error will also occur if an Output instruction addresses a device that is not available, (busy, turned off, or not connected), and when the Output instruction addresses a Display Console that is waiting with a message available.

5.9 STATUS INDICATORS AND DEVICE ADDRESSES.

The I/O Interface can be tested Busy or Not Busy by a Test instruction (ZY = 3°). Whenever the Interface goes Not Busy, the I/O Interface Not Busy Interrupt toggle is set ON (ZY = 39). (Refer to paragraph 11.7 for other interrupts.)

The readiness of an Output L-119 to receive an output message can be determined by addressing it with a Test instruction (ZY = 63, 64 or 65). The Test instruction will test the Device Available signal from the addressed L-119.

The status of the duplexing switching and the pairing of Input with Output L-119's is indicated in the Manual Switch Status word, paragraph 12.3.5.

An Ignore Display Console Message Available toggle (ZY = 70) inhibits the generation of a Display Console Message Available Interrupt signal, even though the Detail Interrupt toggle (ZY = 2,) is turned ON. When the Ignore toggle is ON, the Scanner will recognize, but not stop on, Display Console Message Available positions.

Table 5-1. I/O Interface Device ZY Addresses

DEVICE	ADDRESSES
Input L-119	60, 61, 62
Output L-119	63, 64, 65
Input Any	70
Display Consoles	71, 72, 73, 74, 75, 76, 77, 78, 79, 7=, 7', 7>, 7<, 7], 7~

Section V  
Paragraphs 5. 10 to 5. 10. 2

5. 10 I/O INTERFACE INSTRUCTION FORMAT.

5. 10. 1 INPUT.

Command Character	I
Instruction	IZYXMMMM (Y6 not functional)
Options	Index, Indirect Address, Tagged Halt, Flag Return
Mnemonic	RIO

Input from the Scanner selected real-time device. The execution of the Input instruction in the I/O Interface is described in paragraph 5. 7.

5. 10. 2 OUTPUT.

Command Character	O
Instruction	OZYXMMMM (Y6 not functional)
Options	Index, Indirect Address, Tagged Halt, Flag Return
Mnemonic	WIO

Output to the ZY selected real-time device. The execution of the Output instruction in the I/O Interface is described in paragraph 5. 6.

## SECTION VI

### BUFFER PROCESSOR

The L-119 Buffer Processor is the communications buffer between the Autodin digital data links and the Central Processor. It is a stored program input/output buffering device. The L-119 has sixteen basic commands and uses a magnetic disc as a memory element. A single L-119 is used to perform the send or receive function for a digital data link. Two L-119's are paired to form a send/receive set for a digital data link, as shown in figure 6-1.

#### 6. 1 BUFFER PROCESSOR CONFIGURATION.

Four L-119 Modules are packaged in the Buffer Processor Console. Two of the units perform receiving functions and two of the units perform sending functions. The sending and receiving L-119 Modules are identical except for their internal programs and the interface logic which performs a send only or receive only function.

In addition to the L-119 Modules there is a Maintenance Module which contains switches and indicators that enable the operator to enter programs and monitor L-119 operation.

Data is transmitted to and received from the Data Links bit-serially at a 1200 bit per second rate. Data is transferred between the Central Processor and the L-119 bit-serially at a 800 kc rate.

#### 6. 2 INPUT MESSAGE TRANSFER.

##### 6. 2. 1 DATA LINK TO RECEIVING L-119.

Data is received by the Receiving L-119 bit-serially at a 1200 bit per sec rate from the Data Terminal Bay in Autodin format and code. The Autodin format is shown in figure 6-2. Data is transmitted and received in 80 character blocks framed by four control characters. The Receiving L-119, under program control, performs the following tasks:

- a. Format Check - It checks the format of the received data to insure that it is in Autodin format.

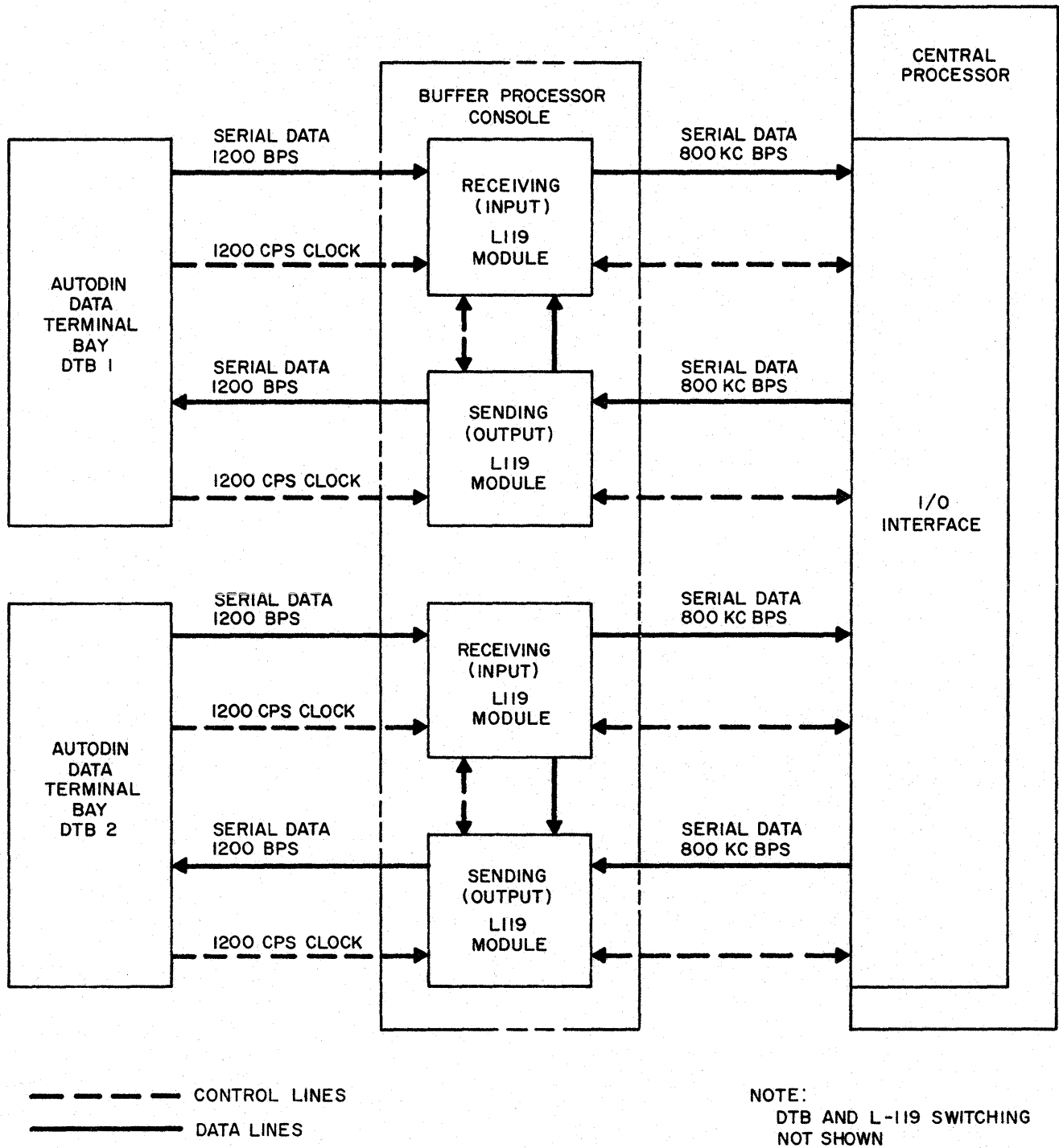
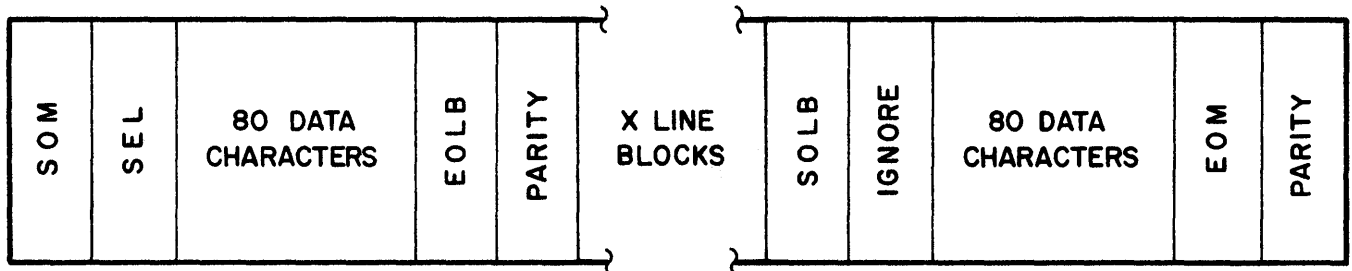


Figure 6-1. Buffer Processor Console Function Configuration



SOM - START OF MESSAGE CHARACTER  
 SEL - SELECT CHARACTER  
 EOLB - END OF LINE BLOCK CHARACTER  
 PARITY - HORIZONTAL CHECK SUM CHARACTER  
 SOLB - START OF LINE BLOCK CHARACTER  
 IGNORE - IGNORE CHARACTER ( FIELDATA i )  
 EOM - END OF MESSAGE CHARACTER

Figure 6-2. C or T Select Message Format



Section VI  
Paragraph 6.2.2

b. Translation - It translates the received data from Autodin (Fielddata) code to Central Processor code.

c. Error Checks - It checks all data for vertical and longitudinal parity. It also signals the sender if an error is detected.

d. Message Acknowledgment - It acknowledges the reception of all messages to the sender by signaling the Sending L-119 to transmit acknowledgment characters.

e. Operator Intervention - It signals the operator of Data Link malfunctions and stoppage of data transmission.

f. Formatting - It formats received data into the proper format for transmission to the Central Processor.

#### 6.2.2 RECEIVING L-119 TO CENTRAL PROCESSOR.

The format for input messages from the Receiving L-119 to the Central Processor is shown in figures 6-3 and 6-4. Each data transmission from the Receiving L-119 to the Central Processor consists of 80 data characters followed by a control field. The control field is 5 characters in length. The control field carries the following indications:

a. The last line block of a message.

b. The message being received has been discarded by the sender, and the Central Processor should ignore any part of the message it has received.

c. The message presently being transmitted by the Sending L-119 on the Data Link has been rejected by the remote station.

d. The data block received by the Central Processor is valid data.

e. The end of the present transmission from the Receiving L-119

Five characters always constitute the control field and a specific character indicates the absence of a control character.

An end of message character (EOM) is located in the first character of the control field. When present, this character identifies the last line block of a received message. The EOM character is the letter E in Central Processor code.

A discard message code (DM) is located in the second character position of the control field. When present, this character indicates that all data blocks received

## A. SINGLE LINE BLOCK MESSAGE: L-119 TO CENTRAL PROCESSOR

80 DATA CHARACTERS	EOM	BLANK	BLANK	BLANK	ET			
--------------------	-----	-------	-------	-------	----	--	--	--

## B. SINGLE LINE BLOCK MESSAGE: L-119 TO CENTRAL PROCESSOR - INDICATING SENDER IS DISCARDING THE LAST UNCOMPLETED MESSAGE

80 DATA CHARACTERS	BLANK	DM	BLANK	BLANK	ET			
--------------------	-------	----	-------	-------	----	--	--	--

## C. SINGLE LINE BLOCK MESSAGE: L-119 TO CENTRAL PROCESSOR - INDICATING THE REMOTE STATION HAS REJECTED THE MESSAGE BEING TRANSMITTED BY THE OUTPUT L119 AND THAT THIS IS A DUMMY MESSAGE.

80 DATA CHARACTERS	BLANK	BLANK	RM	DUD	ET			
--------------------	-------	-------	----	-----	----	--	--	--

Figure 6-3. Single Block Input Message Format

A. FIRST LINE BLOCK

80 DATA CHARACTERS	BLANK	BLANK	PRIORITY	SELECT	ET			
--------------------	-------	-------	----------	--------	----	--	--	--

B. X NUMBER LINE BLOCK

80 DATA CHARACTERS	BLANK	BLANK	BLANK	BLANK	ET			
--------------------	-------	-------	-------	-------	----	--	--	--

C. LAST LINE BLOCK

80 DATA CHARACTERS	EOM	BLANK	BLANK	BLANK	ET			
--------------------	-----	-------	-------	-------	----	--	--	--

Figure 6-4. Multi-Block Input Message Format

by the Central Processor since the last complete message (including the data block with the DM code) are being voided by the sender. This character is the letter M in Central Processor code.

A reject message code is located in the third character position of the control field. When present, this character indicates to the Central Processor that the message presently being transmitted, by the Sending L-119 connected to the Data Link, is being rejected by the remote station. This character is the letter R in Central Processor code.

A dud message code (DUD) is located in the fourth character position of the control field. When present, this character indicates that the data block associated with the control field is a dummy block and it should be ignored. This code is used when signaling the Central Processor that a reject message code has been received when there is no input message being received by the L-119. This character is the letter D in Central Processor code.

An end of message transmission character is located in the fifth character position of the control field. It indicates to the Central Processor the termination of transmission from the Receiving L-119. This is an end of message character (—) in Central Processor code.

A blank character (BLANK) in Central Processor code is inserted in the control field in place of a control code if the control function is not used.

### 6.3 OUTPUT MESSAGE TRANSFER.

#### 6.3.1 CENTRAL PROCESSOR TO SENDING L-119.

The format for output messages from the Central Processor to the L-119 is shown in figures 6-5 and 6-6. Each data transmission from the Central Processor consists of 80 data characters followed by a control field. The control field is five characters in length. In addition to providing end of message indication to the Sending L-119, the control field also indicates to the Sending L-119 the various control functions the Central Processor program desires transmitted on the Data Link, as well as the priority of the outgoing message and the format of the output message.

The control fields, for output messages from the Central Processor to the Sending L-119, are shown in figure 6-6. There are seven control characters and one character to indicate the absence of a control character.

**A. SINGLE LINE BLOCK MESSAGE: CENTRAL PROCESSOR TO L-119**

80 DATA CHARACTERS	EOM	BLANK	PRIORITY	SELECT	ET			
--------------------	-----	-------	----------	--------	----	--	--	--

**B. SINGLE LINE BLOCK MESSAGE: DISCARDING THE PREVIOUS UNCOMPLETED MESSAGE FROM THE CENTRAL PROCESSOR TO THE L-119**

80 DATA CHARACTERS	BLANK	DM	BLANK	BLANK	ET			
--------------------	-------	----	-------	-------	----	--	--	--

Figure 6-5. Single Block Output Message Format

A. FIRST LINE BLOCK

80 DATA CHARACTERS	BLANK	BLANK	BLANK	BLANK	ET			
--------------------	-------	-------	-------	-------	----	--	--	--

B. X NUMBER LINE BLOCK

80 DATA CHARACTERS	BLANK	BLANK	RM NOTE 1	BLANK	ET			
--------------------	-------	-------	--------------	-------	----	--	--	--

C. LAST LINE BLOCK

80 DATA CHARACTERS	EOM	BLANK	BLANK	BLANK	ET			
--------------------	-----	-------	-------	-------	----	--	--	--

Figure 6-6. Multi-Block Output Message Format

Section VI  
Paragraph 6. 3. 2

An end of message character (EOM) is located in the first character position of the control field. When present, this character indicates the last line block of an outgoing Data Link message. It is the letter E in Central Processor code.

A discard message code (DM) is located in the second character position of the control field. When present, this character indicates to the Sending L-119 that the Central Processor desires to discard the message presently being transmitted and that the Sending L-119 should execute the discard message procedure on the Data Link. This character is the letter M in Central Processor code.

A priority character (PR) is located in the third character position of the control field. This character indicates the priority of the message being transmitted. This character is the letter L in Central Processor code for a low priority message, and the letter H for a high priority message.

A select character (SEL) is located in the fourth character position of the control field. This character indicates whether the output message is in card or teletype format. This character is the letter C in Central Processor code for a punched card message, and the letter T for a teletype format message.

An end of transmission character is located in the fifth character position of the control field. It indicates to the Sending L-119 the termination of transmission from the Central Processor. This character is an end of message character (  $\neg$  ) in Central Processor code.

A blank character (BLANK), in Central Processor code, is inserted in the control field in place of a control code if the control function is not used.

### 6. 3. 2 SENDING L-119 TO DATA LINK.

Upon receipt of a block of data from the Central Processor, the Sending L-119, under program control, performs the following tasks:

- a. Formatting - It formats the data received from the Central Processor into Autodin format.
- b. Translation - It translates the data from Central Processor code to Autodin code.
- c. Error Checks - It generates the horizontal block parity character required in the Autodin format.

d. Message Acknowledgment - It checks for correctness of acknowledgment on all transmissions.

e. Operator Intervention - It indicates when the Data Link has malfunctioned and operator intervention is required. (See Section 12.)

f. Transmission of Acknowledgment Codes - It transmits acknowledgment codes requested by the Receiving L-119.

#### 6.4 MESSAGE STORAGE.

As the L-119 Modules handle input and output data on a message by message basis, storage is provided for 336 eight-bit message characters, with control codes used to indicate partial message reception if the message exceeds 336 characters.

Messages and message fragments of up to 336 characters can be transferred to the Central Processor at the disc clock serial bit rate of 800 kc through the use of addressing logic built into the Message track. This allows the use of the Message track as a multi-slot storage element.

Message characters, of eight bits each, are stored and treated as individual words by the L-119 Module programs.

#### 6.5 L-119 MODULE DESCRIPTION.

A block diagram of the L-119 Module is shown in figure 6-7. The L-119 Module operates under control of its internal program.

Programs can be entered manually into the L-119 Modules through the maintenance panel, or by the use of a manually inserted bootstrap program through one of the standard input devices. A program entry device (not part of Buffer Processor) allows more efficient use of the program storage in the machine and also faster entry of a program.

##### 6.5.1 INPUT/OUTPUT INTERFACE.

All data lines between a Data Link and the L-119 Modules are terminated at the Input/Output Interface in each L-119 Module. The Interface consists principally of timing and level standardizing circuits which ensure electrical compatibility between the Data Link and each L-119 Module.

The L-119 Modules communicate with the Central Processor through the I/O Interface in the Central Processor, Section V. This Interface selects a particular L-119 Module and designates the operation to be performed.



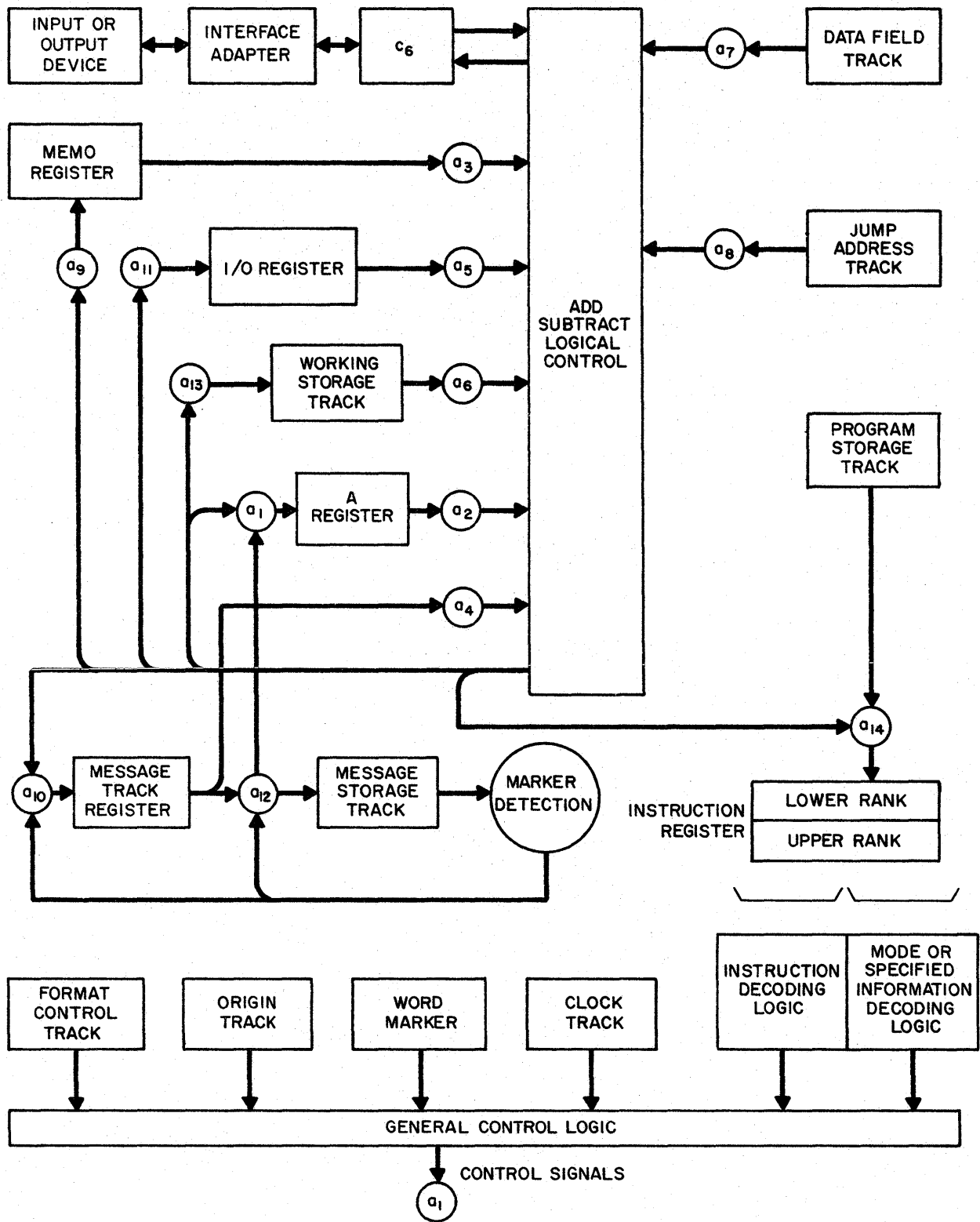


Figure 6-7. L-119 Module Block Diagram.

### 6. 5. 2 DISC MEMORY.

The Disc Memory consists of a 7-inch diameter disc mounted on the shaft of an 8000 rpm motor. The disc is coated with a nickel-cobalt alloy which possesses optimum magnetic and wear characteristics. The top plate of the disc housing is the mounting surface for 10 read/write heads and 6 write heads which run in contact with the memory disc. These heads record or read from the 10 memory tracks. The three recirculating register tracks each use one write head and one read/write head, as do the permanent storage tracks. The remaining four timing tracks each use one read/write head.

For program and data storage, the Disc Memory has the following memory tracks:

- a. Message Track
- b. Program Track
- c. Working Storage Track
- d. Jump Address Track
- e. Data Field Track
- f. Buffer Processor Word Marker Track
- g. Central Processor Word Address Track
- h. Clock Track
- i. Origin Pulse Track
- j. Index Marker Track

The Message track serves as intermediate storage for input or output message data. The Message track is a 336-word circulating register. It occupies one half of a disc track and is circulated twice per disc revolution.

The Program track holds two complete copies of the program, each occupying one half of the track. Each program consists of up to 336 L-119 Module words. Allowing for a space bit between L-119 Module words, the total storage around the disc is 6,048 bits per track.

The Working Storage track is a fast access scratch pad. It uses one head for reading and one head for writing. The heads, which are placed eight words apart, circulate the contents of the working storage, thus permitting access to each word once every eight word times.

Section VI  
Paragraph 6. 5. 3

The Jump Address track specifies a jump address for the Table Look Up and recognition type instructions. It has one read/write head. This track is the same length as the Program track.

The Data Field track is a 336-word permanent storage track which is accessible to the A register (read only). Each word from the Data Field track has nine bits, including parity.

The Buffer Processor Word Marker track provides an indication of the end of each L-119 Module word. It has one read/write head. This track has a 1 recorded on it for every nine L-119 Module clock pulses. The primary use of the word marker pulse is to signal instruction termination and to gate the next instruction from the lower rank to the upper rank of the Instruction register.

The Central Processor Word Address track has recorded on it an address for each Central Processor word (one address per eight L-119 Module word times). The addresses on this track are used when transferring data between the L-119 Modules and the Central Processor. All data transferred between the L-119 Modules and the Central Processor is stored on the Message track. The Message track is 336 words in length (42 Central Processor words). The Central Processor Word Address track contains binary-coded addresses from 0 to 41 for the 42 Central Processor words of Message track storage. This track has one read/write head.

The Clock track contains the L-119 Module clock. The clock frequency of the L-119 Modules is 800 kc. The Clock track has one read/write head.

The Origin Pulse track provides an indication of the start of each of the recorded programs on the disc. A pulse recorded on this track coincides with the start of each of the programs on the Program track. One read/write head is associated with this track.

The Index Marker track is a 336-word circulating register. This track contains an indexing bit which controls program execution timing both in the Normal mode and the One-Step mode. This bit is movable under program control.

### 6. 5. 3 MAJOR LOGICAL ELEMENTS.

In addition to the Disc Memory and Input/Output Interface, each L-119 Module consists of the following major logical sections:

- a. A register - The A register is a one-word (8 bit) accumulator for all arithmetic instructions, as well as being the central information exchange location

within the L-119 Module. Parity is checked on all transfers to the A register, and odd parity is generated in the ninth bit position on all transfers from the A register, except for transfers to or from the memo registers.

b. B register - The B register is a one word (8 bit) register that is used as a counter in the Compare instruction for branching to a specified location. Also, the B register is a holder for a successful table look up operation.

c. Instruction register - The Instruction register is a one word (8 bit), double ranked register. The upper rank holds the instruction presently being executed while the lower rank is being loaded with the next instruction. During spacer bit time, the next instruction is gated from the lower to the upper rank of the Instruction register.

d. Input/Output register - The Input/Output register is a one word (8 bit) register that serves as a buffer between the input or output device and the A register.

e. Memo registers - Memo 1 and 2 registers monitor the state of both the Data Link and Central Processor Interface control functions. Both are 8 bit registers with no parity and are accessible only to the A register. Memo 2 register also has the function of communication with another L-119 Module.

f. Message Track register - The Message Track register (MTR) is used primarily as a read-write buffer between the A register and the Message track. Also, the MTR is an address holder for message track transfer and for marker bit control. Since read-write timing is dependent on the location of the Message Track marker, the Message Track register operates automatically when the proper instruction is given. The Message Track register holds nine bits, including parity, but will write only the first eight bits of a character onto the Message track.

g. Add-Subtract Logical Control Section - The Add-Subtract Logical Control Section contains the logic necessary to perform the required algebraic functions of the instructions.

h. Marker Detection Logic - The marker detection logic is used with the Message track. A marker indicates the correct position for writing into or reading from the Message track. The marker is located in the ninth bit position (i. e., the spacer bit position) of a word.

When a word is to be read from the Message track to the MTR, the Message track is scanned for the marker. Upon detecting the marker (i. e., ninth bit set ON), the

Section VI  
 Paragraphs 6. 5. 4 to 6. 5. 5

marker control logic gates the following word from the Message track into the MTR and advances the marker one word position.

When a word is to be written onto the Message track from the MTR, the Message track is scanned for the marker. Upon detecting the marker, the marker control logic gates the data from the MTR through the write amplifier into the following word position on the Message track and advances the marker by one (8 bit) word.

6. 5. 4 BUFFER PROCESSOR WORDS.

The Buffer Processor (L-119 Module) word contains eight bits plus an odd parity bit. The instruction word format is as follows:

PARITY	COMMAND				SOURCE OR DESTINATION			
9	8	7	6	5	4	3	2	1

Bits 1 through 4 designate the source or destination of information being operated upon or the mode of the command. A Central Processor character is contained in one Buffer Processor word. A Central Processor word is contained in eight Buffer Processor words.

6. 5. 5 MEMORY ADDRESSES.

The two basic types of L-119 memory addresses are: L-119 word addresses and message addresses. In addition, an address field is used that is called a mode. This mode is a command modifier occupying the address portion of the instruction word.

The content of an addressed word consists of 8 bits plus a parity bit, except for the Memo 1 and 2 registers which do not carry parity. Register content is accessible from the A register which is the central transfer location within the L-119 Module. Table 6-1 shows the Buffer Processor word address assignments.

Table 6-1. L-119 Module Word Addresses

0	0000	Memo 1 Register
1	0001	Working Storage Track
2	0010	Message Track Register
3	0011	Data Field Track

Table 6-1. L-119 Module Word Addresses (Cont)

4	0100	A Register
5	0101	Input/Output Register
6	0110	Memo 2 Register
7	0111	B Register

Message addresses refer to the content of the various permanent storage tracks or to the I/O Interface in the Central Processor and may include up to 336 Buffer Processor words of information. Table 6-2 shows the message address assignments.

Table 6-2. Message Addresses

0	0000	Central Processor
1	0001	Program Track
2	0010	Jump Address Track
3	0011	Data Field Track

In normal programming, no address but that of the Central Processor would be used. The other addresses are used only for original entry of program and data. All address locations in table 6-2 are accessible from the Message track.

Address locations 1 through 3 all have nine bits per word including parity. The Message track word is eight bits with no parity bit since the ninth bit position is used for the Message Track marker. When data is transferred from the Message track to these locations, a parity bit is generated and added in the ninth bit position.

#### 6. 5. 6 MEMO ASSIGNMENTS.

Memo bit assignments are as shown in table 6-3.

Table 6-3. Memo Register Assignments

BIT	MEMO 1 REGISTER	MEMO 2 REGISTER
1	Character Present	Message Track Busy
2	Device Ready	
3	Message Available	

Table 6-3. Memo Register Assignments (Cont)

BIT	MEMO 1 REGISTER	MEMO 2 REGISTER
4	Transfer Operation	Acknowledge Transfer Ready
5	Internal Error	Acknowledge Transfer
6	Interface Error	Acknowledge Transfer
7		Acknowledge Transfer
8		Acknowledge Transfer

Individual memo bits are testable, by bringing in a bit from a memo register into the A register, through use of a Logical Product instruction with the appropriate bit. The single bit (toggle) in the A register can then be compared with a word in the Data Field track. Individual memos can be set by a Store instruction from the A register through an appropriate mask in the Data Field track.

The Character Present Memo toggle indicates that an input character is present in the Input Holder and is accessible to the Input/Output register. It is automatically set but must be reset by program.

The Device Ready Memo toggle indicates that a character can be output to the Output Holder from the Input/Output register. It must be set by program but is automatically reset when the Output Holder is cleared.

The Transfer Operation Memo toggle is a signal on a line indicating that the Central Processor has accessed the L-119 Module. This line is not settable by the L-119 Module.

The Message Available Memo toggle is program settable and indicates to the Central Processor that a message is available on input or that a message slot is available on output. It is automatically reset by the Central Processor when the message transfer has been completed.

The Internal Error Memo toggle is automatically set upon detection of an instruction word parity error, an input parity error to A, or a message transfer parity error from the Central Processor. It can also be set under program control, if any program detected inconsistency occurs.

The Interface Error Memo toggle is set automatically upon detection of a parity error in the Input Holder. It is also settable under program control.

The Message Track Busy Memo toggle indicates that a Message Track instruction should not be given. It is set and reset automatically.

The Acknowledge Transfer Ready Memo toggle is set by a Receiving L-119 to indicate to a Sending L-119 that an acknowledge code is ready to be transferred from the Receiving L-119. The four Acknowledge Transfer Bit Memo toggles can transfer up to 16 codes from a Receiving L-119 to a Sending L-119.

The Memo toggles labelled Open can be used in any way desired by the programmer.

### 6. 6 BUFFER PROCESSOR INSTRUCTIONS.

The Buffer Processor has an instruction repertoire of 15 commands, plus command variations. These commands are as follows:

a. 0000 No Operation.

b. 0001 Logical Sum to A. Transfer the logical sum of the designated information and the content of the A register into the A register.

EXAMPLE:

A register	01001011
Designated Information	<u>11001100</u>
Sum	10000111

c. 0010 Logical Product to A. Transfer the logical product of the contents of the data field and the designated information into the A register.

EXAMPLE:

Data Field	01001011
Designated Information	<u>11001100</u>
Product (A Register)	01001000

d. 0011 Add to A. Add algebraically the designated information to the contents of the A register and insert the sum in the A register. Overflow is ignored.

e. 0100 Subtract from A. Subtract algebraically the designated information from the content of the A register. Overflow is ignored. This operation is not masked by the contents of the data field.

f. 0101 Compare. Compare the designated information with the contents of the A register. Correspondence, bit by bit, causes the program to branch to the



Section VI  
Paragraph 6.6 (Cont.)

location specified by the corresponding word on the Jump Address track. Non-correspondence causes the next instruction in sequence to be executed. This instruction destroys the contents of the B register.

g. 0110 Table Look Up. Compare the contents of the A register with the designated information. Correspondence, bit by bit, causes the loading of the corresponding word of the Jump Address track to the B register.

h. 0111 Transfer to Message Track. Transfer to Message track conditions the MTR and associated logic to accept input information from the designated source, starting at the word address specified by the MTR. This operation occurs simultaneously with L-119 Module program execution. Therefore, enough time must be provided by the program to permit completion of the operation.

i. 1000 Transfer from Message Track. Transfer the content of the Message track to the designated destination, starting at the word address specified in the MTR. This instruction conditions the Message track logic to transfer the contents of the Message track serially to the Central Processor or to another disc track, beginning at the specified address and continuing until an end of message character is detected or the origin is detected. At that time, the MTR and associated logic return to the normal state. This operation is performed simultaneously with L-119 Module program execution. Therefore, sufficient time must be provided by the program to permit completion of the operation.

j. 1001 Marker Control. Move the marker as designated:

(1) Mode 0000 Advance Marker. Advance the Message Track marker until correspondence is reached between an address on the Central Processor Word Address track and the MTR.

(2) Mode 0001 Set Index Mark. Advance the program Index mark until correspondence is reached between an address on the Central Processor Word Address track and the MTR.

k. 1010 Word Transfer. Write words on the Message track or read one word from the Message track:

(1) Mode 0000 Transfer Word to Message Track. Transfer the contents of the MTR onto the Message track at the Message Track marker. The marker is advanced one position by this operation, which occurs simultaneously with L-119 Module instruction execution.

(2) Mode 0010 Transfer Words to Message Track. Transfer the contents of the MTR onto the Message track in successive word positions after the Message Track marker until the next Central Processor Word marker is sensed. The marker is advanced to the first Central Processor Word marker by this operation.

(3) Mode 0001 Transfer Word from Message Track. Condition the MTR to detect the next marker and transfer the content of the corresponding word from the Message track into the MTR. Execution of this instruction causes the marker to be advanced one word position. This instruction is executed simultaneously with the L-119 Module program. Therefore, sufficient time must be allowed by the program to permit completion of the operation.

1. 1011 Store Masked A. Transfer the content of the A register masked by the Data Field track to the designated location. The unmasked portion of the designated location remains unchanged.

m. 1100 Halt.

(1) Mode 0000 Unconditional Halt.

(2) Mode 0001 Conditional Halt. Halt when Control Panel switch is ON.

n. 1101 Input. Transfer the content of the designated input/output device to the Input/Output Holder.

o. 1110 Output. Transfer the content of Input/Output Holder to the designated input/output device.

## 6.7 INSTRUCTION SEQUENCING.

### 6.7.1 INSTRUCTION RETRIEVAL.

When the L-119 Module is in the non-halted state, it will be loading instructions into the lower rank of the Instruction register. Upon entering the instruction into the upper rank at the end of the L-119 Module word time, the instruction will be executed.

When the L-119 Module is halted, it will always start program retrieval at the instruction location after the Index mark. The last instruction retrieved after a programmed halt is the instruction which is first executed after the Start switch is depressed, unless another instruction has been manually entered into the lower rank.

### 6.7.2 NORMAL INSTRUCTION TIMING.

All instructions using L-119 Module word addresses, except the Compare and Table Look Up instructions, are executed in the word time after instruction retrieval.

Section VI  
Paragraphs 6. 7. 3 to 6. 7. 4

Therefore, each word recorded on the disc must be available the word time after its corresponding instruction.

In the Compare operation, the actual comparison takes place the word time following instruction retrieval. During this word time, the B register is loaded from the Jump Address track and is tested immediately for all zeros each word time thereafter as long as the comparison is true, until it counts itself down to all zeros. During the time the B register is counted down, normal instruction execution will be halted and will remain halted until the count down is complete. Up to 256 instructions can be skipped in this manner. Instruction retrieval into the lower rank will not be interrupted.

In the Table Look Up operation, comparison takes place the word time after instruction retrieval. If the comparison is true, the following word from the Jump Address track is loaded into the B register.

#### 6. 7. 3 MESSAGE TRACK INSTRUCTION TIMING.

All instructions pertaining to the Message track or the Index Mark track are off-line type operations which may take up to two disc revolutions to complete. When any such instruction is executed, a Message Track Busy Memo toggle is set and remains set until the operation is complete. Normal instruction retrieval and execution will not be interrupted. The burden of timing Message Track instructions rests on the programmer.

#### 6. 7. 4 INPUT MESSAGE SEQUENCE.

When a message is transmitted from the Data Link, via the Receiving L-119 Module, to the Central Processor, it goes through the following sequence.

As the first character of the message is loaded into the Input/Output register, a Character Present toggle is set automatically and is tested under program control. Normally, the L-119 Module program tests the Character Present Memo toggle once per program cycle time, unless the input device can be inhibited under L-119 Module program control. If the Character Present line is found to be true, the character in the Input Holder is brought into the A register, and the Character Present Memo toggle is reset. The program then goes through the input processing routine, paragraph 6. 2. After processing, the input character then is stored on the Message track.

When the end of message or end of block character is detected, or when the Message track is filled, a Message Available Memo toggle is set, which indicates a request for transmission to the Central Processor. As long as input continues, the message is transmitted to the Central Processor in segments while the input processing routine continues.

The setting of the Message Available Memo toggle indicates to the Central Processor, over the Message Available line, that a processed input message exists. The Central Processor tests the Message Available line from the L-119 Module and, upon finding it true, selects the L-119 Module by setting the Input Operation line true for that L-119 Module. The Input Operation line is tested, under control of the L-119 Module program, and, upon detecting that the L-119 Module has been selected by the Central Processor, a Message Transfer instruction is executed. The message is gated out bit-serially to the Central Processor, starting with the first word of the message on the Message track and stopping when the end of message character is detected. Clock signals synchronize the Central Processor and Buffer Processor operation during input or output message transfer.

When the Central Processor detects the end of message character in the input message, the Operation Complete line to the L-119 Module is set true. This resets the Message Available Memo toggle which, in turn, sets the Message Available line to the Central Processor false and indicates to the Data Link that the L-119 Module Interface is not busy.

Should the Central Processor, for any reason, set the Input Operation line false, the Message Available line to the Central Processor remains true and the message is retransmitted to the Central Processor the next time that L-119 Module is selected. Access time for input of information to the Central Processor is dependent upon the priority assigned the L-119 Module in the Central Processor program.

#### 6. 7. 5 OUTPUT MESSAGE SEQUENCE.

When a message is transmitted from the Central Processor through the Sending L-119 Module to the Data Link, it goes through the following sequence.

The L-119 Module, under program control, sets the Device Available Memo toggle true if the previous message has been completely transmitted. If the Central Processor has an output message to transmit to the L-119 Module, it tests the Device Available line. If the Device Available line is true, the Central Processor sets the Output Operation line to the selected L-119 Module true. The L-119 Module, under

program control, tests the Output Operation line and, upon finding it true, executes a Message Transfer instruction. The Message Transfer instruction gates the L-119 Module clock to the Central Processor which outputs data bit-serially at the L-119 Module clock rate to the Message track.

When an end of message character is detected in the input message by the L-119 Module, and its parity is found to be correct, the Device Operation Complete Memo toggle is set, indicating to the Central Processor that the output operation is terminated. Device Operation Complete Memo toggle, in turn, resets the Device Available line to the Central Processor. If the Central Processor, for any reason, sets the Output Operation line false, the Device Available line remains true and the message can be retransmitted by the Central Processor.

When the Device Available line is found to be false, the L-119 Module program initiates the output message processing routine. The output message processing routine, paragraph 6. 3, processes the message character by character under program control and transmits the message to the Data Link. After the L-119 Module has transmitted the last character of the output message, the Device Available Memo toggle is set ON to indicate to the Central Processor that the Buffer Processor is ready for another output message.

## 6. 8 BUFFER PROCESSOR OPERATION.

### 6. 8. 1 CONTROLS AND INDICATORS.

The controls and indicators necessary to operate the Buffer Processor are listed below. The Maintenance Module Control Panel contains the controls for switching to any of the four L-119 Modules.

6. 8. 1. 1 L-119 MODULE CONTROLS. The following controls are located on the individual L-119 Modules:

a. Power On - General Clear. This is a momentary switch which turns on power to the power supplies and to the disc motor of the module. The 400-cycle power must be on for the disc motor to be started. A time delay relay is activated to enable the disc to get up to speed, halt the L-119 Module program, blank all registers, and reset all marker bits to the origin. The lighting of the Power On - General Clear switch is split. While the time delay relay is on, the General Clear half of the switch is ON. When the time delay relay goes off and the L-119 Module becomes operational, the Power On half of the switch is ON.

b. Power Off. This is a momentary activate switch which releases the latching relay controlling the main power.

6. 8. 1. 2 MAINTENANCE MODULE CONTROLS AND INDICATORS. The following switches are located on the Maintenance Module Control Panel, figure 6-8:

a. Code Entry. These eight switches comprise one latching switch per bit for generation of any eight-bit code as data or instruction with the ninth bit inserted automatically as the parity bit. Each Code Entry switch lights when depressed.

b. Program Write. This is a latching switch that applies a write voltage to the Program track. This switch lights when depressed.

c. Jump Write. This is a latching switch that applies a write voltage to the Jump Address track. This switch lights when depressed.

d. Data Write. This is a latching switch that applies a write voltage to the Jump Address track. This switch lights when depressed.

e. Insert Instruction. This is a momentary make switch that inserts the code generated by the Code Entry switches into the lower rank of the Instruction register of the selected L-119 Module. This switch lights when depressed.

f. Insert Data. This is a momentary make switch that inserts the code generated by the Code Entry switches into the MTR of the selected L-119 Module. This switch lights when depressed.

g. Index Mode. This is a latching switch which causes the selected L-119 Module to execute one instruction at a time and to retrieve the next instruction at the location specified by the program Index marker, and causes the selected L-119 Module to automatically advance the Index marker each time the Start switch is depressed. This switch lights when depressed.

h. Repeat Mode. This is a latching switch which causes the selected L-119 Module to execute one instruction at a time, under control of the Start switch at the Index marker, and inhibits instruction retrieval. This switch lights when depressed.

i. Start Switch. The Start switch (one for each L-119 Module) is a momentary make switch. When depressed, it causes the L-119 Module to initiate program execution at the location specified by the Index marker. The switch lights whenever the L-119 Module is executing a program.

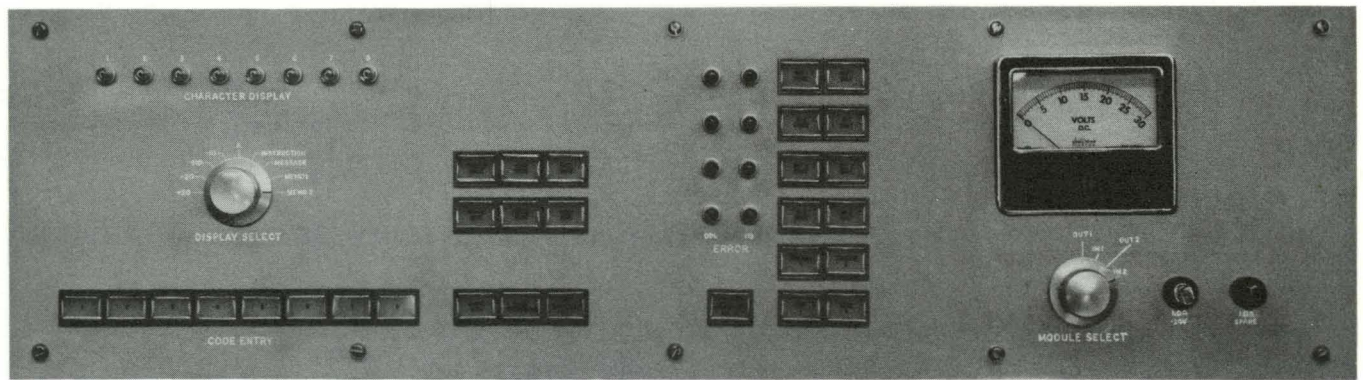


Figure 6-8. Buffer Processor Maintenance Panel

j. Halt Switch. The Halt switch (one for each L-119 Module) is a latching switch. It is testable by program and enables a programmed halt. The switch lights whenever the L-119 Module is halted.

k. On-Off. This is a latching switch which controls 20V power to the Maintenance Module Control Panel.

l. Module Select. This is a rotary switch which transfers the following control panel functions to the selected L-119 Module: (1) Insert Data, (2) Insert Instruction, (3) Program Write, (4) Data Write, (5) Repeat Mode, (6) Index Mode, (7) Display Select.

m. Display Select. This is a rotary switch used to select one of four voltages or one of five registers from the selected module to be displayed on the Maintenance Module Control Panel.

The following indicators are located on the Maintenance Module Control Panel:

a. Normal Mode. This lamp indicates that the selected module is in the normal operating mode; i. e., not in the Repeat or Index mode.

b. Character Display. The character display consists of eight indicator lamps which display the eight data bits of a selected register. The least significant bit is labeled 1; the remaining bits are labeled in order of significance up to 8.

c. Error. There is one Device Error lamp and one Internal Error lamp per L-119 Module. These indicators display the state of the two Error Memo toggles.

## 6. 8. 2 BUFFER PROCESSOR OPERATING PROCEDURES.

The Maintenance Module Control Panel is used for entering of program and constants and for checkout and program debugging. In normal on-line operation, panel switches will be used to start the program and to intervene in case of error. The power-control switching is self-explanatory. When an L-119 Module is first turned ON; the logic is halted, all circulating registers are cleared, and the program Index and the Message Track marker are reset to the origin.

6. 8. 2. 1 CODE ENTRY. The Code Entry switches are used when an L-119 Module is in a Halt state. As an example of the use of the Code Entry switches, data can be entered onto the Message track by the following process:

a. Depress the Repeat Mode button to inhibit normal program retrieval.



Section VI  
Paragraphs 6. 8. 2. 2 to 6. 8. 2. 3

b. Insert a Message Track Control (write) instruction into the Instruction register by generating the instruction code on the Code Entry switches and depressing the Insert Instruction switch.

c. Insert the desired data code into the MTR by generating the code on the Code Entry switches and depressing the Insert Data switch.

d. Depress the Start switch to write the desired data code on the Message track. Each depression of the Start switch writes one code on the Message track.

e. Because the L-119 Module is in the Repeat mode, the Write instruction will be held in the Instruction register. Load additional data characters into the MTR from the Code Entry switches, and write onto the Message track by depressing the Start switch.

6. 8. 2. 2 MODE SELECTION. Two Mode Selection switches which control the instruction retrieval method are available for checkout or program debugging. If neither is depressed, the L-119 Module will be in the Normal mode. In the Normal mode of operation, depressing the Start switch causes the instruction in the lower rank of the Instruction register to be executed and starts instruction retrieval at the Index mark, after which program execution proceeds automatically.

In the Repeat mode, depressing the Start switch causes the instruction in the lower rank of the Instruction register to be executed at the Index mark, but no new instruction will be retrieved and the L-119 Module will halt after one operation.

In the Index mode, depressing the Start switch causes the present instruction to be executed and the next instruction to be retrieved at the Index mark. The Index mark is advanced one place and the L-119 Module will halt after one operation.

The Index mark can be moved by executing a Message Track Marker Control instruction to move the Index marker to any word mark time. To move the Index marker one character time, depress the Index Mode switch and execute a No Operation instruction.

6. 8. 2. 3 PROGRAM ENTRY. Using the Maintenance Module Control Panel, complete programs may be entered into on L-119 Module permanent storage tracks; i. e., the Program track, Data Field track, Jump Address track. Basically, program information would be loaded onto the Message track by the operator through the Maintenance Module Control Panel and transferred half a track at a time to permanent storage.

To insert information onto the Data Field track, information from the program code sheets for the Data Field track is entered character by character by the operator onto the Message track. Then, with the L-119 Module still in the Repeat mode, a Transfer from Message Track to the Data Field Track instruction is entered into the Instruction register and the proper starting address is entered into the MTR. The Data Write is depressed to execute the transfer. If the same information is to be written on both halves of the Data Field track, the starting address must be changed in the MTR and the Transfer instruction executed again. After the Data Field track is recorded, the Data Write must be unlatched to safeguard the Data Field information.

The three permanent storage tracks are entered half a track at a time in this manner. The parity of the stored track may be checked by recalling the stored track to the Message track. If a parity error exists, the L-119 Parity Error lamp for the given module lights. Since it is impossible to deliberately write bad parity onto the permanent storage tracks, such an error indication is evidence of a hardware error.

If an instruction or data character is found to be wrong while program debugging, the faulty character or characters can be replaced in the following manner. The incorrect memory location must be recalled to the Message track. After positioning the Message Track marker with the Message Track Marker instruction, the correct character code is recorded over the incorrect code. When the corrections are completed on the Message track, the content of the Message track may be recorded over the incorrect memory location.

It should be noted that the read and write heads are one-half disc revolution apart on the permanent storage tracks. To retrieve data recorded at a specified address on the track, the address of the other half of the disc must be given.



## SECTION VII

### UNI-RECORD INTERFACE

The Uni-Record Interface executes Input (I), Output (O), and Test (T) instructions addressed to any Uni-Record device, which may be a Magnetic Tape Console, a Line Printer or a Card Reader/Punch. Also some specialized Tape Transport control instructions are executed by the Uni-Record Interface. For a description of the Uni-Record Consoles and their operation, refer to Section VIII.

All Interface instructions are processed first in the Central Processor Instruction register where operand address modification may be specified. Then Interface instructions are transferred to the Uni-Record Interface, as determined by the ZY address. The Uni-Record Interface executes each instruction independently of the Central Processor program, which continues after the instruction has been transferred to the Interface. Figure E-1 is a block diagram that shows the Uni-Record Interface.

#### 7.1 UNI-RECORD BUFFER REGISTER.

The Uni-Record Buffer register is a one word (8 character) static register that communicates with the Core Memory in word parallel. On input this register assembles words for transfer to Core Memory. Data from the Trunkline is assembled into complete words (partial words are padded with zeros) before transfer to Core Memory.

#### 7.2 CHARACTER HOLDER.

The Character Holder is a one character static register that holds one character at a time for transfer between the Uni-Record Buffer register and the Uni-Record Trunkline.

#### 7.3 WORD COUNTER.

The Word Counter controls the number of words to be transferred with Input (Y6 ON) and Output instructions. This Counter is preloaded by a Load instruction (ZY =  $\emptyset\#$ ) with a number from  $\emptyset\emptyset\emptyset$  to 999, which will be 1 less than the number of

words transferred. As words are transferred through the Uni-Record Interface, the counter counts down to 000 unless an error terminates the transfer, in which case the Word Counter indicates the point of termination that can be stored in Core Memory by a Save instruction (ZY = 0#).

The Word Counter can be stored or loaded while a data transfer through the Uni-Record Interface is in progress.

#### 7.4 UNI-RECORD INSTRUCTION REGISTER.

The Uni-Record Instruction register holds the instruction while it is being executed by the Uni-Record Interface and after it has been executed until it is replaced by a new instruction. This register consists of a ZY Holder for device selection, an Operation Code Holder, and an Operand Address Holder that specifies the next Core Memory address location to be used for data transfer. During or after instruction execution the content of the Uni-Record Instruction register may be stored in Core Memory by a Save instruction (ZY = 1/). When this is done, an instruction word is regenerated that contains the command character, the device ZY address and the first unused Core Memory address.

#### 7.5 UNI-RECORD INSTRUCTION LOCATION REGISTER

The Uni-Record Instruction Location register is loaded with the address of each Uni-Record Interface instruction as the instruction is sent to the Interface. During or after instruction execution the content of the Uni-Record Instruction Location register may be stored in Core Memory by a Save instruction (ZY = IU).

#### 7.6 STATUS AND ERROR TOGGLES.

The Uni-Record Interface can be tested by a Test instruction (ZY = 3;) for Busy status.

The Status toggles DSG1 and DSG2 indicate special data transfer conditions after completion of an Input operation with Y6 ON, as follows:

a. DSG1. Information transfer stopped within a tape record or before the end of a card (ZY = 4').

b. DSG2. Information transfer stopped because a file mark, or end of tape was encountered, or because the card read hopper was depleted (ZY = 4>).

These toggles are set ON or reset OFF every time the Uni-Record Interface selects a Trunkline Device and executes an operation, they are meaningful only after completion of an Input operation with Y6 ON, or when a subsequent Test operation addresses that same Trunkline Device.

The following seven Status toggles reflect the status of a device only when a Test instruction addresses that device:

- a. DSF1 - The addressed device is operative. (ZY = 45).
- b. DSF2 - The addressed device is on-line. (ZY = 47).
- c. DSF3 - The addressed device is busy; that is, the last operation has not been completed (ZY = 46).
- d. DSF4 - The write inhibit is on for the addressed Tape Transport or the addressed Line Printer is in Automatic Format mode (ZY = 4=).
- e. DSF5 - The addressed device has been set to the Even Parity mode (ZY = 3>).
- f. DSF10 - The addressed Tape Transport is at the start of tape (ZY = 48).
- g. DSF11 - The addressed Tape Transport is at the end of tape (ZY = 49).

The Test instruction itself responds to the DSF1, Device Operative, signal. The Device Operative signal will be generated even if the addressed device is performing an operation in overlap. Although the Test Instruction has no effect on the Status signals within the addressed device, it does gate the Status signals into the Status toggles.

Two Detail Error toggles, DSF6 and DSF7, reflect errors that occur in a data transfer operation after the Uni-Record Interface has become Not Busy for that operation. These two toggles have the following significance:

- a. DSF6 - The addressed device has detected a data transfer error off-line (ZY = T8).
- b. DSF7 - The addressed device has detected a mechanical error which disrupted the operation being performed (ZY = T9).

Each of these toggles is set by any Interface instruction addressing the device in error. If a Test instruction addresses a device, the Status toggles are set and a Uni-Record Interface Error Interrupt request is generated. Any instruction, other than a Test instruction, is terminated immediately before execution, and a

Uni-Record Interface Error Interrupt request is generated. When an Input or Output instruction addresses the device, an on-line error indication will appear also. The error indications will continue until another instruction is addressed to the device generating the error signal.

Nine Detail Error toggles are set by errors detected while the Uni-Record Interface is Busy with an operation. When such an error occurs, the operation is terminated, the Interface goes Not Busy, and a Uni-Record Interface Error Interrupt request is generated. The first seven error toggles are set by signals from the addressed device (if all are meaningful); the last two error toggles are set by signals to the Uni-Record Interface from Core Memory. These error toggles have the following significance:

- a. DSE1 - A parity error has been detected by the addressed device (ZY = T1).
- b. DSE2 - A mechanical error disrupted the data transfer being performed (ZY = T3).
- c. DSE3 - The addressed Tape Transport is unable to perform the current instruction because it is interlocked at the load or end of tape (ZY = T>).
- d. DSE4 - The addressed Tape Transport detected an eavesdrop parity error while performing an Output operation (ZY = T2).
- e. DSE5 - The addressed Tape Transport detected a horizontal check character error while performing an Input operation (ZY = T5).
- f. DSE6 - The addressed device is overloaded by having its data handling capacity exceeded by an Output operation (ZY = T7).
- g. DSE7 - The addressed device does not recognize the current instruction code (ZY = T6).
- h. Uni-Record Interface M register Parity Error - A data word containing incorrect parity was transferred through the Core Memory M register on an Input or Output operation (ZY = T4).
- i. Uni-Record Interface Operand Address Error - An attempt was made to access a nonexistent memory location for data. (ZY = T<).

7.7 UNI-RECORD DEVICE ADDRESSING.

Each Uni-Record Interface instruction carries a device address in its ZY characters. Bits Z4 and Z5 must be 1's to signify the Uni-Record Interface.

Device address assignment is by means of a three-octal character switch on the Uni-Record Device Control Panel, figure 7-1. Bits Z4 and Z5 must be 1's for Disc Memory Interface operation. The least significant octal character  $O_1$  corresponds to bits Y1, Y2, and Y3. The second octal character  $O_2$  corresponds to bits Y4, Y5, and Z1. Because the most significant octal character  $O_3$  corresponds to bits Z2 and Z3, it can have only the values of  $\emptyset$ , 1, 2, and 3. The total number of possible Uni-Record device addresses is 256.

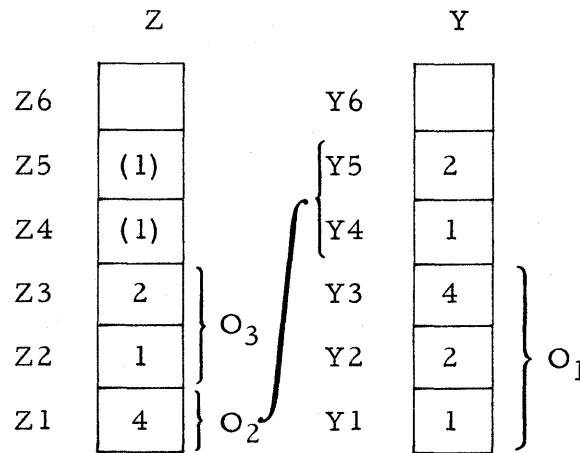


Figure 7-1 Uni-Record Trunkline ZY Addressing

7.8 UNI-RECORD INTERFACE INSTRUCTIONS.

The three Uni-Record Interface instructions are the Input, Output and Test instructions. Input instructions select a Tape Transport or a Card Reader by the ZY address. The Y6 bit option selects between termination of input by end of message control or by a preset number of words. Output instructions select. Tape Transport, Card Punch or Line Printer by the ZY address. Output is terminated after a preset number of words. The Output instruction also can send control messages to devices. The Test instruction tests operative status of the addressed device and makes available other Status and Error signals from the device.



## Section VII

### Paragraphs 7.8.1 to 7.8.1.2

The detection of errors will terminate data transfer unless the Ignore Uni-Record Interface Error toggle is set ON.

A Uni-Record Interface instruction encountered while the Uni-Record Interface is Busy will hold up the Central Processor program execution until the Interface goes Not Busy. An instruction selecting a device that is Busy in overlap operation waits in the Interface until the device goes Not Busy, and then is executed.

#### 7.8.1 INPUT.

Command Character I

Instruction IZYXMMMM

Options Index, Indirect Address, Tagged Halt, Flag Return  
Variable or Fixed Number of Words

7.8.1.1 VARIABLE NUMBER OF WORDS - Y6 OFF. Data will be transferred from the device selected by the ZY characters to the Central Processor and stored in Core Memory starting at the effective operand address location. The selected device will terminate the Input operation with an End of Message signal.

If the selected device is a Tape Transport, an End of Message (EOM) signal is generated when a record gap is encountered. Reading a file mark causes a one-character message transfer consisting of the EOM character.

If the selected device is a Card Reader, the Input operation is terminated by the depletion of the stack of cards being read. Also, the input operation is terminated by detection of an error unless the Ignore Uni-Record Interface Error toggle is set ON.

7.8.1.2 FIXED NUMBER OF WORDS - Y6 ON. Data will be transferred from the device selected by the ZY characters to the Central Processor and stored in Core Memory starting at the effective operand address location. The number of words to be transferred is preset into the Word Counter, paragraph 7.3.

The Word Counter counts down to 000 during the transfer. If the transfer is completed before reaching the end of a tape record gap, or the end of a card, the remaining portion of the tape record or card is passed over and the data is not transferred into the Central Processor. Several tape records or cards may be read by a single Input instruction.

The detection of an error will terminate data transfer, unless the Ignore Uni-Record Interface Error toggle is ON.

## 7.8.2 OUTPUT.

Command Character O

Instruction OZYXMMMM

Options Index, Indirect Address, Tagged Halt, Flag Return, Data or Control Message

7.8.2.1 DATA - Y6 OFF. Data will be transferred to the device selected by the ZY characters from Core Memory starting at the effective operand address location. The number of words to be transferred must be preset into the Word Counter, paragraph 7.3.

If the selected device is a Tape Transport, a record gap is generated after the last word is transferred.

If the selected device is a Card Punch, an incomplete last card will be padded with blank characters.

If the selected device is a Line Printer, printing may be automatic or under control of a header character, as determined by a Mode toggle in the printer. The Mode toggle is set or reset by output control messages. If the Controlled Format mode is used, only one line of 132 characters (after the header character) is printed per Output instruction. The header character indicates the channel of the vertical format paper tape to be selected before printing. Any data characters over 132 will be lost, but no error indication will be generated.

In Automatic mode, vertical format paper tape channel 8 is always used. Two lines are printed for each 33 words transferred, with the last four characters of each seventeenth word forming the first four characters of the second line.

7.8.2.2 CONTROL MESSAGE - Y6 ON. A single word constituting a Control Message is sent to the device selected by the ZY characters from the effective operand address location in Core Memory. The first character of the word, which is the Control Message, may have values  $\emptyset$ , 1, or 2 (any other character is interpreted as value  $\emptyset$ ). The rest of the word is ignored. Control Messages are interpreted as follows:

- a.  $\emptyset$  Character - Render device inoperative
- b. 1 Character - Tape Transport: Set on Write Inhibit  
Line Printer: Select Automatic Format mode

Section VII  
Paragraphs 7.8.3 to 7.9.2

- c. 2 Character - Tape Transport: Complement Parity Control toggle  
Line Printer: Select Header Character Format mode

### 7.8.3 TEST.

Command Character T

Instruction TZYXMMMM

Options Index, Indirect Address, Tagged Halt, Flat Return

This is the same Test instruction described in paragraph 4.5.3; because it addresses a Uni-Record device, this Test instruction is executed in the Uni-Record Interface.

The Test instruction is used to ascertain the status of the device selected by the ZY characters. The Test instruction itself responds (with a conditional jump to its effective operand address) to the condition of the Device Operative (DSF1) line. Additionally, the device status signals are set into the Device Status toggles, and Detail Error toggles. If either of the Detail Error toggles is set ON in response to the Test instruction, it will be followed by an Interface Error Interrupt request. Refer to paragraph 7.6 for a description of the Status and Error toggles.

## 7.9 TAPE TRANSPORT CONTROL INSTRUCTIONS.

The following instructions are control instructions that are applicable only to Tape Transports. The Uni-Record Interface is Busy only briefly for a Control instruction, about 30 microseconds, or long enough for the Magnetic Tape Console to clock and acknowledge the control code. A Tape Transport executing a Control instruction, if tested, will indicate operative, on-line, busy, and not interlocked. If the Tape Transport cannot perform the control operation because it is at the load point or the end of tape, an Interlocked Error Interrupt request will be generated.

### 7.9.1 BACKSPACE TAPE - (.

The Backspace Tape instruction, coded ( (left parentheses), instructs the Tape Transport selected by the ZY characters to backspace. With Y6 ON, the Tape Transport will backspace to the last previous file mark or load point. With Y6 OFF, the Tape Transport will backspace to the last previous record gap.

### 7.9.2 FORWARD-SPACE TAPE - ).

The Forward-Space Tape instruction, coded ) (right parentheses), instructs the Tape Transport selected by the ZY characters to forward space. With Y6 ON, the

Tape Transport will forward space to the next file mark or to the end of tape. With Y6 OFF, the Tape Transport will forward space to the next record gap.

#### 7.9.3 REWIND TAPE - [.

The Rewind Tape instruction, coded [ (left bracket) instructs the Tape Transport selected by the ZY characters to rewind to the load point.

#### 7.9.4 WRITE FILE MARK - ].

The Write File Mark, coded ] (EOM character), instructs the Tape Transport selected by the ZY characters to write a file mark 3 inches from the end of the last record.



## SECTION VIII

## UNI-RECORD CONSOLES

The Uni-Record Trunkline is the means of Central Processor operation with associated Tape Transports, Card Reader/Punch, and Line Printer. A Uni-Record Trunkline can accommodate 256 peripheral devices, a device being defined as a tape transport, a Card Reader/Punch or Line Printer. Each of these devices is controlled through the Magnetic Tape Console, which contains Tape Transport, Card Reader/Punch, and Line Printer Control Sections.

The Magnetic Tape Console contains logic, control, and data circuits for each Tape Transport and peripheral device necessary to perform on-line and off-line functions: on-line functions under Central Processor program control and off-line under manual control. Major functions of the Magnetic Tape Console are:

- a. Transferring of programs and data from magnetic tape or cards into Core Memory (on-line).
- b. Transferring of data and programs between magnetic tape and cards (off-line).
- c. Controlling of data transfer between cards and line printer (off-line).
- d. Transferring of data from magnetic tape to line printer (off-line).
- e. Transferring of data from Core Memory to magnetic tape or cards under program control (on-line).
- f. Transferring of data from core memory to line printer (on-line).

For on-line operation, the Central Processor program may select either one of two Tape Transports to send or receive data from the Core Memory, via the Uni-Record Trunkline; however, only one Tape Transport may be operated on-line at a given time. While one Tape Transport is operating on-line, the other Tape Transport may be operated simultaneously to perform off-line functions. The Card Reader/Punch or Line Printer may, at program option, be selected to receive data from the Core Memory; or the Card Reader/Punch may be selected to transmit data to Core Memory. Each Tape Transport, Card Reader/Punch, or Line Printer requires a specific address for on-line operation. All addressing and data transfers between the Magnetic Tape Console and the Central Processor are via the Uni-Record Trunkline.

Section VIII  
Paragraph 8. 1

All off-line operations are under operator control. Off-line operations include data transfer between the Tape Transport and Card Reader/Punch, data transfer from the Tape Transport to the Line Printer, or data transfer from the Card Reader/Punch to the Line Printer. Tape Transport to Tape Transport operation is not possible. Off-line data transfer is not possible between Magnetic Tape Consoles.

8. 1 TAPE TRANSPORTS.

Each L-3555 Magnetic Tape Console contains two Tape Transports as well as the control sections for a Card Reader/Punch and Line Printer. L-3556 Magnetic Tape Consoles do not contain control sections for a Card Reader/Punch or Line Printer.

On each Tape Transport, data is recorded at a density of 556 characters to the inch, operating at the rate of 90 inches per second. Records are separated by 3/4 inch inter-record gaps.

The end of file indication is a 3 inch gap followed by an end of file character. The 3/4 inch inter-record gap follows the end of file character. Tape can be read and recorded with either odd or even parity. Maximum tape start plus stop time is 7 milliseconds; minimum is 5 milliseconds.

Some degree of simultaneous operation is provided, in that, a data transfer operation can be in process on one Tape Transport, overlapping non-data transfer operations on other Tape Transports; i. e., Rewind, Backspace Record, Backspace File, Skip Record, Skip File, Write File Mark.

Tape may be moved forward or backward; however, operations requiring data transfer can take place only while tape is moving forward. Forward is from left to right, over a split head. This split head has a write station and a read station. During recording the read station checks the parity of each character 3 milliseconds after it has been recorded.

After the tape start time of 3. 5 milliseconds has passed, data is transferred at the rate of 50, 000 characters per second. Maximum tape stop time is 3. 5 milliseconds also.

Tape normally moves at the rate of 90 inches per second. There are two rewind speeds: a hi-speed 270 inches per second, and a low-speed of 90 inches per second, paragraph 8. 1. 4.

### 8. 1. 1 TAPE TRANSPORT ON-LINE READ.

For program testing of the On-Line, Off-Line status of a Tape Transport, refer to Section VII.

On-line Read operation is accomplished in one of two ways: Normal Input and Controlled Input. In both cases, the data is read into Core Memory starting with the location specified by the effective operand address of the Input instruction.

8. 1. 1. 1 DATA FLOW. Data is read from tape a character at a time into the Magnetic Tape Console Control Section. Then the data is transmitted to the Uni-Record Interface in the Central Processor. Having collected a full word, the Interface transmits that word through the M register into Core Memory. The Operand portion of the Interface Instruction register, and/or the Interface Word Counter are modified immediately after each word is stored in Core Memory.

8. 1. 1. 2 END OF TRANSFER. Data transfer can be terminated by any of the following conditions: end of operation, end of file, error. These conditions and their programming implications are discussed in the description of the Normal and Controlled Input instructions in paragraph 8. 1. 1. 4.

If transfer is terminated short of a complete record, the tape continues to move until an inter-record gap is reached. The unread portion of the record is passed over and the Interface is notified by an Incomplete Record signal.

Where the Controlled Input instruction is used, and the number of words specified by the Word Counter is greater than the number of words in the record, the tape will move over the inter-record gap and read words from the next record. If the specified word count is greater than the number of words available when a file mark is read, the Interface is notified by an Overload signal.

8. 1. 1. 3 ERROR CHECKING. Vertical parity is checked as the tape is read into the Tape Console Control Section. At the end of each record, the horizontal check character is checked. Parity is checked again at the M register.

If a parity error is detected during data transfer, the last correct word is stored. An incorrect word is not stored unless either the Uni-Record Interface Error Interrupt toggle is ignored, or the error is an M Register Parity Error, specifically the least significant character. In either case, the Tape Transport will continue moving tape until the next inter-record gap is reached.



8. 1. 1. 4 TAPE READ OPERATION.

Command Character	I
Instruction	IZYXMMMM
Options	Index, Indirect Address, Tagged Halt, Flag Return
Mnemonics	RDT, RDTXR

a. Controlled Input (Y6 = 1) – Read N Words – RDTXR. An Input instruction addressing a Tape Transport will cause data to be read into the Core Memory. The data words transferred will be stored in consecutive memory locations, beginning with the location specified by the effective operand address of the Input instruction. The number of words to be transferred is specified by the Uni-Record Interface Word Counter which must be preloaded, paragraph 7. 3. When the Word Counter has been decremented to 000, one more word is transferred, and the operation terminated.

The Interface remains busy throughout the operation. When the operation is terminated, the Interface requests an Interface Not Busy Interrupt.

The Interface is notified via an Incomplete Record signal if the number of words specified does not complete a record. Although data transfer is terminated, the tape will continue to move until an inter-record gap is reached.

When a file mark is sensed an End of File signal is sent to the Interface, the word,  $\neg 00000000$ , is transferred to the Central Processor, and the operation is terminated. However, if the Word Counter has not been counted to 000 when end of file is sensed, an Overload signal is transmitted.

When an end of tape marker is sensed, an End of Tape signal is sent to the Interface; however, the operation continues.

Should the tape have a record which ends in a partial word (from another system), the partial word will be padded with zeros.

When an error occurs, the Word Counter can be used to implement error recovery. The Word Counter content will have been decremented by the number of words transferred.

b. Normal Input (Y6 = 0) – Read One Record – RDT. An Input instruction with the Y6 bit OFF will cause data to be read into the Central Processor. The data words transferred will be stored in consecutive Core Memory locations, beginning

with the location specified by the effective operand address. Reading and data transfer will continue until an inter-record gap has been sensed. When all of the data has been transferred, the Interface requests an Interface Not Busy Interrupt, to complete the operation. The Interface remains Busy throughout the operation.

When a file mark is sensed, an end of file indication is sent to the Interface, the word,  $\neg 00000000$ , is transferred into Core Memory, and the operation is terminated.

When an end of tape marker is sensed, an End of Tape signal is sent to the Interface, however, the operation continues.

The Uni-Record Instruction register in the Central Processor is of significant value in error recovery since it will reflect the point of termination.

#### 8. 1. 2 TAPE TRANSPORT ON-LINE WRITE.

8. 1. 2. 1 DATA FLOW. The data to be written is transferred word parallel from Core Memory to the Uni-Record Interface, and then is transferred character serial to the Magnetic Tape Console for transfer onto tape. Both the Operand portion of the Interface Instruction Register, and the Interface Word Counter are modified immediately after each word has been read out of Core Memory. Unless interrupted, this flow continues until the operation is complete, as specified by the Interface Word Counter, paragraph 7. 3.

As soon as the last character is received by the Magnetic Tape Console, the Uni-Record Interface requests an Interface Not Busy Interrupt.

8. 1. 2. 2 END OF TRANSFER. Data transfer can be terminated by any of the following conditions: end of operation, mechanical error, parity error.

8. 1. 2. 3 ERROR CHECKING. Data is parity checked first as it is read out of the Core Memory M register, then checked again upon receipt at the Magnetic Tape Console. Three milliseconds after a character has been recorded, it is parity checked by the read side of the split head. This is called the Eavesdrop Check.

Also a horizontal check character is recorded on tape after each block and file mark on tape. This check character is examined by the Eavesdrop Check, and again while reading the tape.

Section VIII  
Paragraphs 8. 1. 2. 4 to 8. 1. 3. 1

8. 1. 2. 4 TAPE WRITE OPERATION.

Command Character	O
Instruction	OZYXMMMM Y6 = 0
Options	Index, Indirect Address, Tagged Halt, Flag Return
Mnemonic	WRT

An Output instruction addressing a Tape Transport with the Y6 bit OFF will cause consecutive words to be transferred from Core Memory to tape beginning with the location specified by effective operand address of the Output instruction. The number of words to be transferred is specified by the Uni-Record Interface Word Counter, paragraph 7. 3, which must be preloaded. When the Word Counter has been decremented to 000, one more word is transferred before the operation is terminated.

When the last character is received by the Magnetic Tape Console, the operation is completed and the Interface requests an Interface Not Busy Interrupt.

When the Interface goes Not Busy, the moving tape will take 5. 8 to 6. 3 milliseconds to finish the read check and stop.

The Interface is notified via a Capacity Exceeded signal when an end of reel reflective marker is sensed; however, the operation proceeds uninterrupted.

The Word Counter reflects the point of termination with respect to error recovery.

8. 1. 3 TAPE TRANSPORT MODE CHANGE OPERATION.

Command Character	O
Instruction	OZYXMMMM Y6 = 1
Options	Index, Indirect Address, Tagged Halt, Flag Return
Mnemonic	WRTCTL

8. 1. 3. 1 RENDER INOPERATIVE. A Tape Mode Change instruction will render the selected Tape Transport inoperative, if its effective operand address references a location containing a first character other than 1 or 2. Nothing will be written on tape.

Restoring the Tape Transport to an Operative state may be done manually only.

8.1.3.2 WRITE INHIBIT. The programmed Write Inhibit mode will be set ON, if the effective operand address of the Tape Mode Change instruction references a word beginning with 1. The state of the Write Inhibit mode is a status signal that can be tested. Restoring the unit to Non-Write Inhibit mode may be done manually only.

8. 1. 3. 3 COMPLEMENT MODE OF PARITY. It is possible to write or read on a tape with either odd parity or even parity. The current Parity mode will be reversed by executing a Tape Mode Change instruction and reading a word beginning with 2. The current state of the Parity mode is a status signal that can be tested.

8. 1. 3. 4 MODE CHANGE CONDITIONS. All of the applicable data transfer timing and error considerations apply. The Interface Word Counter need not be loaded and will not be affected. The Operand portion of the Interface Instruction register will be incremented by 1. None of the Tape Mode Change instructions move or write on the tape. The characters of the referenced word beyond the first character (C7) are ignored.

#### 8. 1. 4 TAPE TRANSPORT CONTROL OPERATIONS.

Command Character	( , [ , 7, or )
Instruction	(See below) X, MMMM not functional
Options	Tagged Halt and Flag Return

a. ( ZYXMMMM (Y6 OFF) – Mnemonic: BKSR. The Backspace Record operation causes the Tape Transport to backspace to the previous gap. Backspace Record is an Overlap operation; i. e. , the Uni-Record Interface goes Not Busy immediately (within 40 microseconds) and can execute another instruction. The Tape Transport will test Operative and Busy.

b. ( ZYXMMMM (Y6 ON) – Mnemonic: BKSF. The Backspace File operation causes the Tape Transport to backspace to the previous file. Backspace File is an Overlap operation.

c. ( ZY000000 (Y6 Ignored) – Mnemonic: REW. The Rewind operation causes the Tape Transport to rewind to the load point. Rewind is an Overlap operation. The tape will rewind at 270 inches per second until it comes within 50 feet of the start of tape. At this point it slows down to 90 inches per second. A 2400 foot tape will rewind in approximately 2 minutes.

Section VIII  
Paragraph 8. 1. 5

d. ⌋ZY000000 (Y6 ON) – Mnemonic: WEF. The Write File Mark operation causes the Tape Transport to record a file gap and a file mark. Write File Mark is an Overlap operation.

e. )ZY000000 (Y6 OFF) – Mnemonic: ADVR. The Forward Space Record operation causes the Tape Transport to move to the next record gap. Forward Space Record is an Overlap operation.

f. )ZY000000 (Y6 ON) – Mnemonic: ADVF. The Forward Space File operation causes the Tape Transport to move past the next file mark. Forward Space File is an Overlap operation.

During the Overlap operation, i. e., until the operation is complete, but after the Interface goes Not Busy, the Tape Transport will test Busy, and a new instruction addressed to it will be stored in the Uni-Record Interface. When the unit goes Not Busy, the new instruction will be executed.

#### 8. 1. 5 TAPE TRANSPORT OFF-LINE OPERATIONS.

When off-line, the Tape Transports can be used to List Tape on the Line Printer or Card Reader/Punch, List File on the Line Printer or Card Reader/Punch, or to record Tape from the Card Reader/Punch.

The List Tape operation causes the tape to be listed a record at a time on either the Line Printer (printed out) or the Card Reader/Punch (punched cards). The record can be a maximum of 132 characters for Line Printer listing or a maximum of 80 characters for Card Reader/Punch punching. The listing process continues until an end of tape is encountered.

The List File operation causes the tape to be listed a record at a time on either the Line Printer or the Card Reader/Punch. The record length is as indicated above. The listing process continues until a file mark or end of tape is encountered, whichever occurs first. Should end of tape be encountered first, depressing the List File button will continue the operation until a file mark is reached.

The Record Tape operation causes the tape to be recorded. Each record consists of the contents of one punched card as it is read and transmitted from the Card Reader/Punch. The recording process continues until all the cards in the Card Reader/Punch have been recorded. No manual intervention is required to record the last few cards.

8. 1. 6 TAPE TRANSPORT OPERATION.

8. 1. 6. 1 TAPE CONSOLE CONTROLS. The Magnetic Tape Console Control Panel has the following controls, figure 8-1:

- a. A rotary switch with the following four positions:

OFF

ON-LINE

OFF-LINE CARD UNIT

OFF-LINE PRINTER

- b. A three decade switch specifying the octal address of the tape unit.

- c. A rotary switch with the following five positions:

FAST FORWARD

FORWARD

STOP

REVERSE

FAST REVERSE

- d. A Logic Reset Button

- e. An On Button

- f. An Alarm Indicator

- g. An Auto Mode Button

- h. A Write Inhibit Button

- i. A Load Button

- j. A Backspace Record Button

- k. A Backspace File Button

- l. A Forward Space Button

- m. A Parity Mode Button (Odd-Even)

8. 1. 6. 2 TAPE CONSOLE INDICATORS. The Magnetic Tape Console Control Panel has the following indicators, figure 8-1. These Indicators correspond to the Status

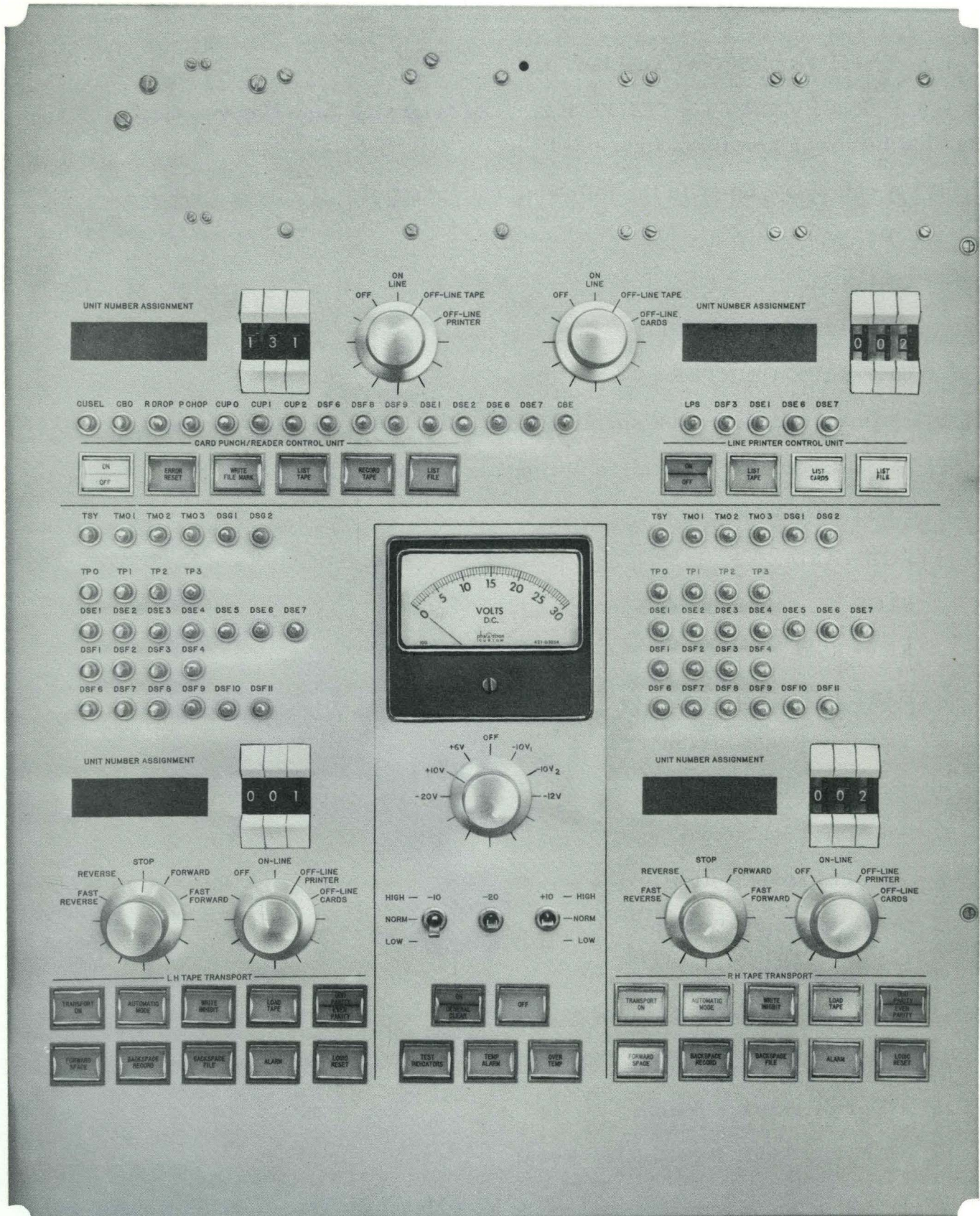


Figure 8-1. Magnetic Tape Console Control Panel

and Error toggles in the Uni-Record Interface which will obtain signals from the last addressed Uni-Record device. These indicators have the following significance:

- a. DSE1 - A parity error has been detected by the addressed device.
- b. DSE2 - A mechanical error disrupted the Input or Output instruction that had been performed.
- c. DSE3 - The addressed Uni-Record device was unable to perform the current instruction because it is interlocked at the load point.
- d. DSE4 - The addressed Uni-Record device detected an eavesdrop parity error while performing an Output instruction.
- e. DSE5 - The addressed Uni-Record device detected a check character error while performing an Input instruction.
- f. DSE6 - The data handling capacity of the Uni-Record device has been exceeded by the output message transfer being performed.
- g. DSE7 - The Uni-Record device did not recognize the instruction code.
- h. DSF1 - The Uni-Record device is operative. The operative signal will be generated even if the addressed device is performing an off-line operation; i. e. , the operative signal will be used to indicate that this device is capable of performing on-line operations.
- i. DSF2 - The Uni-Record device is on-line.
- j. DSF3 - The Uni-Record device is Busy, i. e. , the device is off-line, or the last performed on-line operation is being sequenced off.
- k. DSF4 - The write inhibit is ON for the addressed Tape Transport.
- l. DSF6 - The Uni-Record device detected a data error after releasing the Central Processor for the last previous data transfer.
- m. DSF7 - The Uni-Record device detected a mechanical error which disrupted the operation being performed after releasing the Central Processor for the last previous data transfer.
- n. DSF10 - The addressed Tape Transport is at the start of tape.
- o. DSF11 - The addressed Tape Transport is at the end of tape.
- p. TSY - Will be ON when the tape unit is selected under on-line conditions.



Section VIII  
 Paragraphs 8. 1. 7 to 8. 1. 7. 1

- q. TMO 1 - When ON, indicates a forward motion of tape.  
 When OFF, indicates reverse motion of tape.
- r. TMO 2 - When ON, indicates a Write operation.  
 When OFF, indicates a Read operation.
- s. TMO 3 - When ON, indicates a Record type operation.  
 "OFF" indicates a File type operation.

Table 8-1 indicates the significance of TMO 1-TMO 3. The Write File Mark operation appears as a Write operation (1 1 1) to the Tape Transport until tape motion is begun, it then changes to the configuration shown in table 8-1.

- t. TP0 - Indicator lights when the Tape Transport is in the Standby condition.
- u. TP1 - Indicator lights (momentarily) at the start of an operation.
- v. TP2 - Indicator lights as long as the Tape Transport is performing a transfer operation (Read or Write).
- w. TP3 - Indicator lights during the time the check character is being verified.

Table 8-1. TMO 1 - TMO 3 Operations

TMO 1	TMO 2	TMO 3	OPERATION
0	0	0	Rewind
0	1	0	Backspace File Mark
0	1	1	Backspace Gap
1	0	0	Read
1	1	1	Write
1	1	0	Forward Space - File Mark
1	0	1	Forward Space - Gap
0	0	1	Write File Mark

8. 1. 7 TAPE TRANSPORT OPERATING PROCEDURES.

8. 1. 7. 1 START OPERATION. To operate the Tape Transports, perform the following procedure:

- a. Depress the console On-General Clear button.

- b. Depress the Logic Reset button.
- c. Depress the Transport On button.

Now the Tape Transport is ready to operate in either a Manual or Automatic mode. The only control available in the Manual mode is derived from the rotary switch that determines the tape operating speed; i. e., Forward, Fast Forward, Stop, Reverse, Fast Reverse.

8.1.7.2 ON-LINE OPERATION. In addition to the general procedure in paragraph 8.1.7.1, the following operations are required to place the Magnetic Tape Console in an on-line mode:

- a. Select the desired address on the Address switches
- b. Depress the Auto Mode button
- c. Switch the rotary control switch associated with the Tape Transport to On-Line.

The Tape Transport is now ready to operate in an on-line mode under control of the Central Processor through the Uni-Record Interface.

8.1.7.3 OFF-LINE OPERATION. In addition to the general procedure in paragraph 8.1.7.1, the following operations are required in order to place the Magnetic Tape Console in an Off-Line mode:

- a. Under Line Printer Control:
  - (1) Depress the Auto Mode button.
  - (2) Turn the rotary control switch associated with the Tape Transport to OFF-LINE PRINTER.
  - (3) Turn the rotary control switch associated with the Line Printer to OFF-LINE TAPE.

The Magnetic Tape Console is now ready to operate off-line under control of the Line Printer. The off-line operation will be defined by the depression of the LIST TAPE or LIST FILE button associated with the Line Printer Control Section.

- b. Under Card Unit Control:
  - (1) Depress the Auto Mode button
  - (2) Turn the rotary switch associated with the Tape Transport to OFF-LINE CARDS position

(3) Turn the rotary switch associated with the Card Reader/Punch to the OFF-LINE TAPE position.

Now the Tape Transport is ready to operate off-line under control of the Card Reader/Punch. The off-line operation will be defined by the depression of the List Tape, List File, Record Tape or Write File Mark buttons associated with the Card Reader/Punch Control Section.

c. Under Manual Control - To perform a Backspace Record, Backspace File, Load Tape or Forward Space operation, perform the following:

(1) Turn the rotary switch associated with the Tape Transport to the OFF-LINE CARDS or OFF-LINE PRINTER position.

(2) Depress the Auto Mode button.

(3) Depress the appropriate manual function button.

8. 1. 7. 4 MAGNETIC TAPE CONSOLE OPERATING GUIDELINES. The following items are related to operation of the Magnetic Tape Console:

a. Power to the console logic is obtained by depressing the main power On-General Clear switch located in the center of the Console Control Panel.

b. Depressing the Logic Reset button will cause the Tape Transport to return to an initial condition state.

c. An Alarm indicator will light in case of any mechanical malfunction.

d. When the Transport On button is depressed, the Tape Transport may be placed in any one of five manually controlled positions: Reverse, Fast Reverse, Stop, Forward and Fast Forward. In the Manual mode, the logic is inoperative.

e. Depressing the Auto Mode button places the Tape Transport in the automatic mode and under control of the connected device. The List Tape, List File or Record Tape buttons may be depressed only when the Tape Transport is in the Auto mode.

f. When the Tape Transport is placed on-line, the List Tape, Record Tape and List File buttons associated with the Line Printer or Card Unit, will be ineffective. In order for on-line operation to commence, the Tape Transport must be placed in the Auto mode.

- g. When the Load Tape button is depressed, tape will be driven forward for a short period of time, will start a fast reverse, and then will stop when the start of tape marker (load point) is encountered. This button may be used to rewind the tape.
- h. When the Forward-Space button is depressed, the tape will be driven forward until a file mark has been read. The operation will be terminated with the read and write heads located in the inter-record gap between the file mark and the next record.
- i. When the Backspace Record button is depressed, the tape will be driven back into the next inter-record gap. The operation will be terminated with the read and write heads located in the inter-record gap. No distinction will be made between a file mark and a record.
- j. When the Backspace File button is depressed, the tape will be driven backward until a file mark has been read. If the first character read when the operation is initiated is a file mark, it will be ignored. Any subsequent file mark encountered will be recognized. The operation will be terminated with the read and write heads located in the file gap between the file mark and the last record of the previous file.
- k. When the Odd-Even button is depressed, the Tape Transport is placed in the indicated Parity mode. The tape may be either recorded in this mode or an even parity tape may be read and translated into an odd parity output.
- l. Turning the rotary switch to the OFF-LINE PRINTER position will make the List Tape and List File buttons associated with the Line Printer effective. (Only one may be depressed). The Card Reader/Punch may not be placed off-line. Placing the unit off-line to the Card Reader/Punch will allow transfers to the Card Reader/Punch and inhibit operation with the Line Printer. When the Tape Transport is placed off-line, the appropriate operation buttons must be depressed at the off-line console.
- m. When the Write Inhibit button is depressed, writing is inhibited on tapes. Write Inhibit is also possible by program or by removing the Write Inhibit ring from the tape reel.

## 8. 2 CARD READER/PUNCH.

### 8. 2. 1 CONSOLE DESCRIPTION.

The Card Reader/Punch, consists of a Card Reader and Card Punch mounted on the same frame. It functions under control of the Magnetic Tape Console, Uni-Record Interface and Central Processor. This unit can read cards at the rate of 800 cards per minute, and punch at the rate of 250 cards per minute.

All operations are double checked for both hardware and character validity errors in the buffer and in the unit. Both the read and punch feeds are equipped with card jam and misfeed detection devices. A card jam or misfeed in either the read or punch feed causes a control panel lamp to light and signals an error to the Uni-Record Interface, paragraph 7. 6.

There is no electrical or mechanical coupling between the Card Reader and Card Punch, although they use the same buffer. Therefore, for a reproducing operation, information must be read from the Card Reader into the Central Processor Core Memory and then read out of the Core Memory to the Card Punch. Off-line punching is from magnetic tape only.

The Card Reader/Punch has five radial-type stackers, with a capacity of 1000 cards each. Two stackers are assigned exclusively to the Card Reader, and three to the Card Punch. Cards can be removed from the stackers without stopping operation.

8. 2. 1. 1 CARD MOVEMENT. A card cycle is the time expended in advancing a card through one card station. The Card Reader has two stations and the Card Punch has three. Advancing a card from hopper to stacker, requires three cycles on the read side and four on the punch side. An initial cycle is required to move the card from the hopper to the first station.

The cards in the hoppers can be loaded into their appropriate stations, either manually or automatically. For example, if a card is not at its appropriate station when an Input or Output instruction is given, it will be advanced automatically to that station (read station or punch station) before the operation begins.

Card movement, and the associated reading, punching and checking functions, are terminated only at the end of a cycle at which time the Card Control Section and/or the Uni-Record Interface is notified.

8. 2. 1. 2 READING. An empty hopper will not terminate the reading process. That is, no manual intervention, or additional Input instruction is necessary to empty the

Card Reader. When the last card has been read, checked, and transferred, an end of message indication is sent to the Interface to terminate the operation.

8.2.1.3 PUNCHING. After the last card of a given output request has been punched, the cards are advanced for an extra cycle so that parity can be checked. Therefore, when the punch operation ends, the last card has been punched, passed through the punch check station, and stacked with the rest of the punched cards. However, the extra cycle has advanced a blank card beyond the punch station that is ready to be sent through the punch check station. When the next Output instruction is issued, the blank card will be directed automatically to the Start Reject (SR) stacker.

#### 8.2.2 CARD READER.

The card hopper on the Card Reader is equipped with a device for large capacity card loading. This File Feed device holds up to 3,600 cards.

The cards feed through the Card Reader 9-edge first, face down. The feed path is from right to left, passing two sets of 80 reading brushes. The first reading station reads 80 columns of the card and establishes a parity check. At the second read station, the parity check is proved and the data directed to the Core Buffer in the Magnetic Tape Console. At the end of the read transport path, two radial-type stackers are provided. The Normal Read (NR) stacker, the one closest to the read hopper, is used unless an error occurs, in which case all cards following (but not including) the error card are directed to the Read Error (RE) stacker, figure 8-2.

The Core Buffer in the Magnetic Tape Console has a capacity of 80 characters (10 words). It performs no formatting or padding functions, and although it retains the card data until replaced, that data is not available after the card has been read.

The Card Reader may be operated on-line or off-line depending on a manual switch setting at the Tape Console associated with that unit. The switch controls the whole unit; i. e., both the Card Reader and Card Punch.

8.2.2.1 CARD READER ON-LINE. The On-Line, Off-Line status of the Card Reader may be tested by the program, Section VII.

The rated and effective speed of the Card Reader is 800 cards per minute. There is a period of 74.8 milliseconds of computing time available between card cycles. The effective card reading speed is reduced only when the available computing time is exceeded, paragraph 8.2.2.1 b.

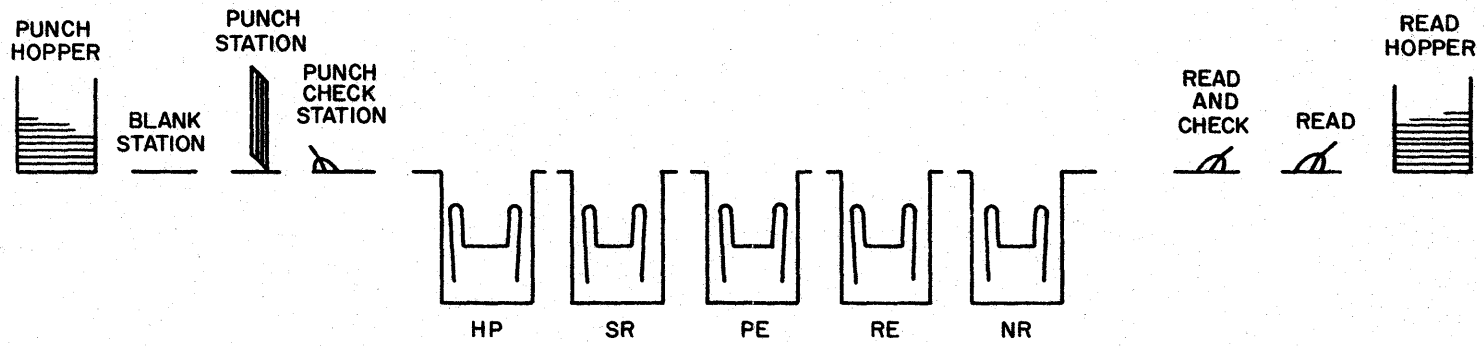


Figure 8-2. Card Flow Diagram

a. Data Transfer. On-line card reading is accomplished in one of two ways: Normal Input and Controlled Input. In both cases the data is read into the Core Buffer starting with the location specified by the effective operand address in the Input instruction.

(1) Data Flow. Data is read from the Card Reader, paragraph 8.2.1.1, into the Core Buffer until the buffer is full. The information is then transmitted to the Uni-Record Interface. Having collected a full word, the Interface transmits that word to the M register in the Central Processor, and then to Core Memory. The Operand portion of the Instruction register, and/or the Word Counter in the Uni-Record Interface are incremented or decremented (whichever is applicable) immediately after a word has been stored in Core Memory. Unless interrupted, this process continues until the Core Buffer is empty.

(2) End of Transfer. Data transfer can be terminated by any of the following conditions: End of Operation, Last Card, paragraph 8.2.1.1, or Error. These conditions and their programming implications are discussed in the description of the Normal and Controlled Input instructions, paragraph 8.2.2.1b.

If transfer is terminated short of a full card, the remaining portion of the card is passed over, the data is not transferred, and the Interface is notified via an Incomplete Record signal.

If the Controlled Input instruction is used, and the number of words requested by the Word Counter is greater than the number of words (cards) available, the Interface is notified via an Overload signal, paragraph 7.6.

(3) Error Checking. While reading and intermediate storing is taking place, the cards are subjected to both parity and character validity checks. Should either type of error occur, a parity indication is sent to the Uni-Record Interface.

As a card is read, it is moved past two read stations. A parity bit is produced and saved at the first read station for each card row. Another parity bit is produced for each card row at the second station. These two sets of parity bits are compared and if they do not agree, an error signal is sent to the Interface.

While data is being read and rows are being horizontally parity checked, character validity processing is taking place. The reading and checking process continues until the Core Buffer is full, at which time the data is transferred to the Interface. As each character is read out of the Core Buffer, the results of the character



## Section VIII

### Paragraph 8.2.2.1 (Cont.)

validity process is checked. If the Card Reader is on-line, parity is checked in the Central Processor M register.

When a parity error occurs during data transfer, the last correct word is stored; the incorrect word is not stored unless the Uni-Record Interface Error Interrupt is ignored, or only the least significant character of a word is in error, or the first indication of parity error is in the M register.

If a parity error occurs, all cards read after the error card are directed to the Read Error (RE) stacker. The error card will be the last card to be stacked in the Normal Read (NR) stacker. Issuing another Input instruction or, a Test instruction (addressing the reader) will restore the stacking process to the normal condition.

b. Card Reader Instruction Operation. For Instruction Format, refer to paragraph 8.1.1.4.

Mnemonic

RDCD, RDCDXR

(1) Controlled INPUT (Y6 ON) – RDCDXR. An Input instruction addressing the Card Reader with the Y6 bit ON, will cause cards to be read into the Central Processor. The data words transferred will be stored in consecutive Core Memory locations, beginning with the effective operand address. The number of words (and thus cards) to be transferred is specified by the Uni-Record Interface Word Counter which must be preloaded, paragraph 7.3. When the Word Counter has been decremented to 000, one more word is transferred before the operation is terminated.

When the operation is terminated, the Interface generates an Interface Not Busy Interrupt request. The Interface remains Busy throughout the operation, and can be tested.

The Interface is notified via an Overload signal if the number of words specified is greater than the number of words (cards) available, or if an Input instruction was given and the Card Reader was empty.

Data is always read from the card starting with word 1, column 1, thus words may not be selectively read, except to specify the point of termination.

(a) Timing (Y6 ON). The effective card reading speed is 800 cards per minute, allows a period of 74.8 milliseconds of computing between card cycles. To maintain the effective card reading speed when specifying 1 card at a time, another Input instruction must be initiated sometime during that 74.8 milliseconds.

Reading a card into the Central Processor from the Card Reader Cycle Start until End of Transfer requires 67.3 milliseconds. Therefore, the Interface will remain Busy for that length of time. If another Input instruction is issued during that period, the Central Processor will wait until the Interface goes Not Busy, thus, wasting the time between initiation of the instruction and sensing of the Interface Not Busy.

There is a period of 7.5 milliseconds of Interface Not Busy time between end of transfer (record in memory) and the beginning of the next card cycle. Therefore, if the next Input instruction is issued during that 7.5 milliseconds, the Central Processor will not wait and the maximum card reading speed will be maintained.

During the 7.5 milliseconds of Interface Not Busy time, Line Printer and/or Tape Transport operations may be performed without loss of card reading speed. This is possible because the Line Printer has its own Core Buffer and the Tape Transport has a fast transmission rate. However, the length of the tape record that may be written is limited by the 3.5 millisecond maximum tape start time, and by whatever other instructions are to be executed within the 7.5 milliseconds.

Since the same Core Buffer is shared by both Card Reader and Card Punch, alternate card reading and punching will slow the Card Reader down to the Card Punch rate.

(b) Errors (Y6 ON). The two types of possible errors are those that can occur at the Card Reader/Punch, and those that can occur during data transfer. See paragraph 7.6 for a list of applicable detail error and status toggles.

Until data transfer to Core Memory begins, the Uni-Record Interface Word Counter is not disturbed. Therefore, should an error occur at the Card Reader, the Word Counter may be used to locate the point of termination in terms of cards.

During data transfer the Uni-Record Interface Word Counter is decremented after each word is stored in Core Memory. Therefore, should an error occur during data transfer, the Word Counter may be used to locate the point of termination in terms of words. A bad parity word can be stored; if the first indication of parity is in the M register, or if the least significant character of the word is the only one in error, or if the Uni-Record Interface Error Ignore toggle is ON.

(2) Normal Input (Y6 OFF) - RDCD. An Input instruction with the Y6 bit OFF will cause cards to be read into the Central Processor. The data words transferred will be stored in consecutive Core Memory locations, starting with the effective

operand address. When the card hopper is empty, the Uni-Record Interface is notified via a Last Card signal. The Interface then signals a request for an Interface Not Busy Interrupt. The Uni-Record Interface remains Busy until the operation is complete; i. e., until all the cards in the hopper have been read.

(a) Timing (Y6 OFF). The effective card reading speed of 800 cards per minute is maintained throughout the execution of this operation. The reading rate is maintained since reading and Central Processor program execution take place in parallel.

A full card of information is available in Core Memory every 74.8 milliseconds. Apart from the use of the real-time facilities (Real-Time Clock and Interval Timer) the program can use the Uni-Record Instruction register to recognize end of card transfer and/or end of word transfer. There is no hardware end of card indication. The Uni-Record Interface stays Busy until the card hopper is empty and the last card has been transferred to Core Memory.

(b) Errors (Y6 OFF). The two types of errors are those that can occur at the Card Reader and those that can occur during data transfer. Refer to paragraph 7.6 for a list of applicable detail error and status toggles.

Until data transfer to Core Memory begins, the Uni-Record Instruction register is not disturbed. Therefore, should an error occur at the Card Reader, the Uni-Record Instruction register may be used to locate the point of termination in terms of cards, or 10 word blocks. Refer to paragraph 7.4 for detailed information on use of the Uni-Record Instruction register.

During data transfer the Uni-Record Instruction register is incremented after each word is stored in memory. Therefore, should an error occur during data transfer, the Uni-Record Instruction register may be used to locate the point of termination in terms of words. A bad parity word can be stored; if the first indication of parity is in the M register, or if the least significant character of a word is the only one in error, or if the Uni-Record Interface Error Ignore toggle is ON.

8.2.2.2 CARD READER OFF-LINE. Manual off-line controls are provided that allow an operator to place a combination of Uni-Record devices off-line. He may operate this equipment independently of the rest of the system.

The following off-line options are available with respect to the card reader; card to tape, card to printer. Most of the manual controls that provide the operator the means to select and operate the Card Reader off-line, are located on the Magnetic Tape Console Control Panel. A few others are on the Card Reader/Punch.

a. Tape Console Controls. When off-line, the Card Reader is controlled by the selected Tape Transport or the Line Printer. The following is a list and functional description of applicable manual controls necessary for off-line card reading operations, figure 8-1.

(1) Tape Transport Rotary Switch - Each Tape Transport, both left hand (LH) and right hand (RH) has a four position rotary switch associated with it. These switches have the following positions:

OFF

ON-LINE

OFF-LINE CARD UNIT - Allows Card Reader/Punch off-line data transfer

OFF-LINE PRINTER - Allows Line Printer off-line data transfer

(2) Record Tape Button - Depressing the Record Tape button, advances the card to the first station in the Card Reader and allows information to be transferred from punched cards to magnetic tape. Each card will be written on tape as one 80 character record with a horizontal check character. Card reading and recording will continue until the available cards have been exhausted, including those cards still in the Card Reader when the hopper has been emptied. The card reading speed is 800 cards per minute.

(3) List Cards Button - Depressing the List Cards button, allows data to be transferred from cards to the Line Printer. Each card will be printed as one line. The cycle is as follows: a card is read into the Card Buffer, the Line Printer is spaced (automatic format), the card content is printed. This process will continue at 800 cards per minute until the available cards have been exhausted, including those cards still in the Card Reader when the hopper has been emptied.

b. Card Reader Controls. The following controls are located on the Card Reader/Punch Control Panel, figure 8-3:

(1) Reader (Latching Switch) - When switch is in the ON position, the read cycle can be initiated.

(2) Read Non Process Runout - Clears Card Reader stations of cards. Hopper must be empty before switch is effective.

(3) Stop - This button will stop the Card Reader after it has completed its present cycle.

(4) Check Reset - This button must be depressed after a read non-process runout, a card jam, or a clutch failure has occurred. It allows normal operation to begin again.

c. Start-Stop Procedure.

(1) Start Procedure:

(a) Depress START button on the Card Reader/Punch Control Panel.

(b) Depress both the Punch ON and Reader ON buttons on the Card Reader/Punch Control Panel.

(2) Stop Procedure:

(a) Depress Stop button on Card Reader/Punch Control Panel. (Do not depress Punch Off or Reader Off buttons first.)

(b) Depress Punch OFF or Reader OFF button on Card Reader/Punch Control Panel.

### 8.2.3 CARD PUNCH.

The Card Punch has a rated speed of 250 cards per minute. The punch hopper has a capacity of 1,200 cards.

The cards feed 12-edge first, face down. The feed path is left to right, passing a blank station, a punch station and a read station. The punch station consists of 80 punches, for recording data. The 80 brush, punch/read station counts all the holes in all 80 columns of the card, for punch verifying.

At the end of the punch transport path, three stackers are available. Punched cards are directed to the Normal Punch (NP) stacker. Extra cycle blank cards are directed to the Start Reject (SR) stacker, and in the event of an error, all cards following (but not including) the card in error will be directed to the Punch Error (PE) stacker.

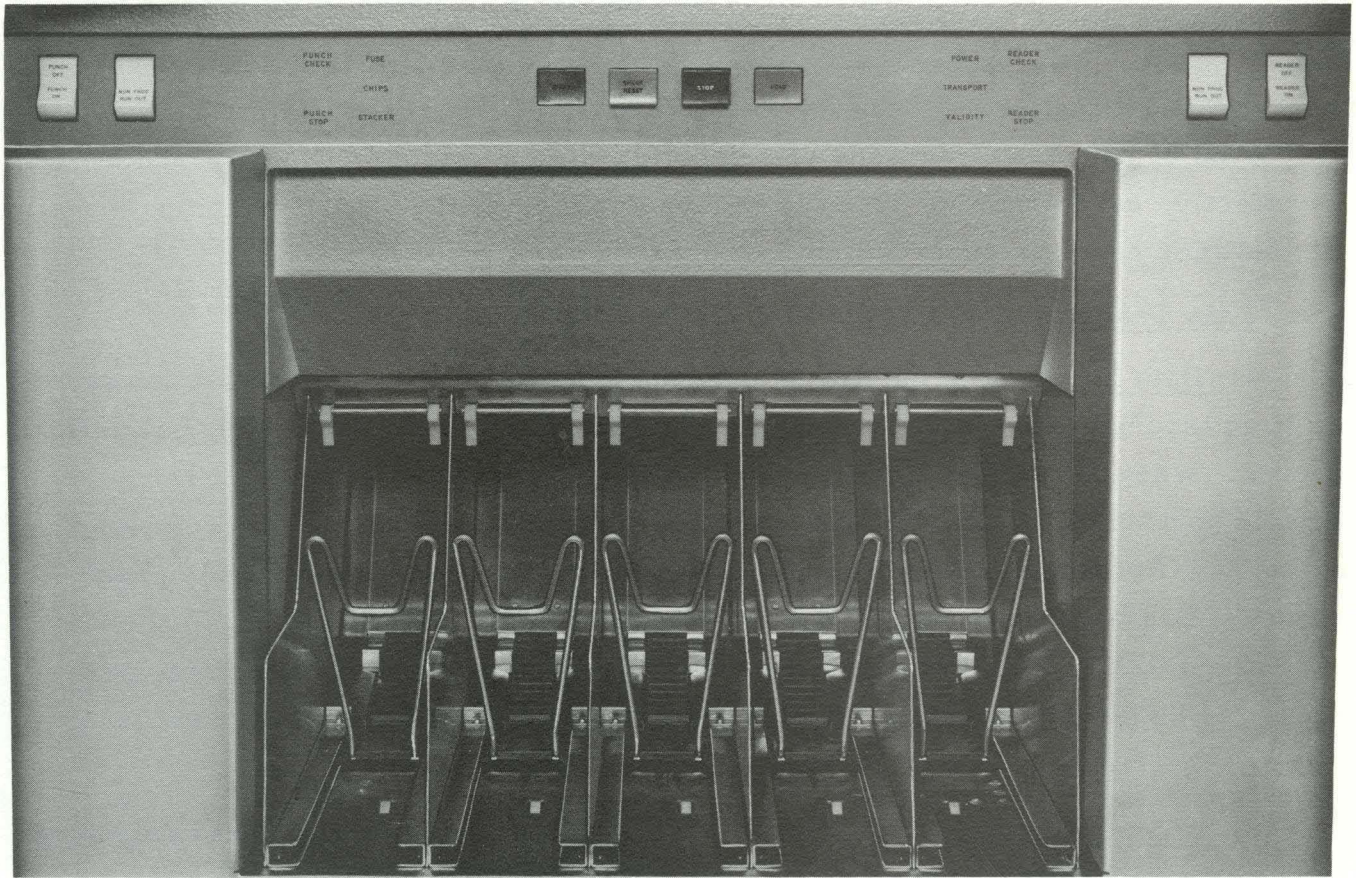


Figure 8-3. Card Reader/Punch Control Panel

Section VIII  
Paragraph 8.2.3.1

The Core Buffer in the Magnetic Tape Console can hold 80 characters (10 words). This Core Buffer is shared by the Card Reader and Card Punch. Data is not retained from a previous operation.

The Card Punch may be operated on-line or off-line depending on a manual switch setting at the Magnetic Tape Console. The On-Line Off-Line switch controls the whole Card Reader/Punch.

8.2.3.1 CARD PUNCH ON-LINE. The on-line off-line status of the Card Punch may be tested by the program, Section VII.

The rated and effective speed of the Card Punch is 250 cards per minute. There is a period of 239.8 milliseconds of computing time available between punch cycles. The effective card punching speed is reduced only when the available computing time between cards is exceeded, 8.2.3.1b(3).

a. Data Transfer.

(1) Data Flow. The data to be punched is read from Core Memory a word at a time through the Uni-Record Interface to the Core Buffer in the Magnetic Tape Console. When the Buffer is full, punching is initiated, paragraph 8.2.1.1. The Operand portion of the Uni-Record Interface Instruction register is incremented, and the Uni-Record Interface Word Counter is decremented, immediately after each word has been read out of Core Memory. Unless interrupted, this process continues until the operation is complete as specified by the Word Counter, paragraph 7.3. When the operation is complete, the Interface requests an Interface Not Busy Interrupt.

(2) End of Transfer. Data transfer can be terminated by any of the following conditions: end of operation, overload (empty hopper) error.

If transfer stops before filling a card, the balance of the card will be padded out with blanks.

(3) Error Checking. As a card is punched, it is moved past two stations. The first station is where punching takes place and the second is where the parity of the newly punched card is checked. Parity bits are generated and stored at the first station and compared with those generated at the second station. If they do not compare, a parity signal is sent to the Uni-Record Interface. If a parity error occurs, all cards punched after the error card will be directed to the Punch Error (PE) stacker. The error card will be the last card stacked in the Normal Punch

(NP) stacker. Issuing another Output instruction, or a Test instruction addressing the Card Punch, will restore the stacking process to its normal condition.

Error checking is performed in the M register, at the Core Buffer, and at the Card Punch.

b. Card Punch Instruction Operation. For instruction format, refer to paragraph 8.1.2.4. Mnemonics PNCH, PCHCTL

(1) Output (Y6 OFF) - PNCH. An Output instruction addressing the Card Punch while the Y6 bit is OFF will cause consecutive words to be transferred from Core Memory to the Card Punch beginning with the effective operand address. The number of word(s), and thus card(s), to be transferred, is specified by the Uni-Record Interface Word Counter, which must be preloaded, paragraph 7.3. When the Word Counter has been decremented to 000, one more word is transferred before the operation is terminated.

When the operation is terminated, the Uni-Record Interface generates a request for an Interface Not Busy Interrupt. The Interface remains busy throughout the operation. When the Interface goes Not Busy, the last card will just be beginning the punching process.

The Interface is notified via an Overload signal when the number of words specified is greater than the number of cards available, and when an Output instruction is given and the punch hopper is empty.

Data is always punched starting with word 1, column 1, thus words may not be selectively punched, except to specify the point of termination.

(2) Output (Y6 ON) - PCHCTL. An Output instruction with the Y6 bit ON, addressing the Card Punch or the Card Reader, will render the Card Reader or Card Punch inoperative. Restoring the unit to an operative state may be accomplished by manual means only.

All of the applicable timing and error considerations apply. The Uni-Record Interface Word Counter need not be loaded, and is ignored and undisturbed. The Interface Instruction Register is loaded automatically and the Operand portion is incremented by 1.

(3) Timing. The rated Card Punch speed is 250 cards per minute. The Uni-Record Interface stays Busy until the specified number of words have been transferred to the Core Buffer in the Magnetic Tape Console.



When punching of more than one card is specified, the program may need to know when a card of data is transferred out of the Central Processor. This may be accomplished through the use of the Interface Word Counter or the Interface Instruction register. The real-time facilities may be used if so desired. Ten words (one card) will be transferred every 240 milliseconds.

When punching of one card at a time is specified, the Uni-Record Interface remains Busy only while data is being transferred to the Core Buffer (less than 2 milliseconds). Therefore, the program is free to use other Uni-Record devices during the punch cycle. However, to maintain the rated card punching speed, the next Output instruction must be issued within 239.8 milliseconds.

(4) Errors. Refer to paragraph 7.6 for a list of applicable detail error and status toggles.

Parity errors can be recognized in the M register, at the Core Buffer and in the Card Punch. The only difference of concern to the programmer, is the condition of the Uni-Record Interface Word Counter and Uni-Record Interface Instruction register.

Should a parity error be detected in the M register, the Interface registers will not have been changed. However, any subsequent error detection will occur after the registers have been modified.

8.2.3.2 CARD PUNCH OFF-LINE. Manual off-line controls are provided that allow an operator to place the card punch off-line for operation with a Tape Transport which is also off-line. The controls enable the operator to list a tape onto punched cards at a rate of 250 cards per minute.

a. Tape Console Controls. When off-line, the Card Punch controls the selected Tape Transport. To punch cards the operator must perform the following:

(1) Rotate the Tape Transport Selection switch to the OFF-LINE CARD position. This selects the Tape Transport for the Card Reader/Punch.

(2) Rotate the Printer Unit Selection switch to the OFF-LINE TAPE position. This connects the Card Reader/Punch to the off-line Tape Transport.

(3) Depress the Logic Reset button located on the Tape Control section of the Magnetic Tape Console Control Panel. This clears the tape control and places it in an Idle state.

(4) Depress the Error Reset button located on the Card Control Section of the Magnetic Tape Console Control Panel. This clears the card control logic and places it in the Idle state.

(5) Position the tape using the manual Tape Position rotary switch.

(6) Place the Tape Transport in the Automatic mode by depressing the Automatic pushbutton located on the Tape Control section of the Magnetic Tape Console Control Panel. This places the tape in a Ready state to transmit data to the Card Punch.

(7) Depress the List Tape or List File button located on the Card Control Section of the Magnetic Tape Console Control Panel. Depressing either of these buttons causes the punching cycle to be initiated. Depressing the List File button causes the punching of cards until a file mark is detected. Depressing the List Tape button causes the punching of cards until the end of tape marker is detected.

b. Card Punch Controls. To turn on the Card Punch, the operator must depress the Punch On button on the Card Reader/Punch and the On/Off button on the Card Control Section of the Magnetic Tape Console Control Panel. Several indicators are located on the Card Reader/Punch to signal the operator of a malfunction when punching cards.

A Punch Check indicator signals that a card has been punched in error. A Punch Stop indicator signals the operator that a card did not feed into the punch, or a card was already in the punch when a punch cycle was initiated. When the latter condition occurs, depressing the Non-Process Run Out button on the Card Reader/Punch causes the card in the punch to be fed into the stacker.

A Stacker Indicator on the Card Reader/Punch indicates that the card stacker is filled with punched cards. A chips indicator indicates that the basket containing punched card chips has been filled. A fuse indicator on the Card Reader/Punch indicates a blown fuse in the unit.

### 8.3 LINE PRINTER.

The Line Printer is of the rotating drum and print hammer type. It permits copy on multiple carbon continuous form paper. It is capable of printing 132 characters per line at a rated speed of 1000 lines per minute. Each of the 132 print positions can print 64 characters: 26 alphabetic, 10 numbers, 27 special symbols, and a blank.

The Line Printer prints on-line from the Data Processor or off-line with a Tape Transport or Card Reader.

Vertical space, hammer penetration, and horizontal form size, are manually adjustable. These adjustments allow the satisfactory handling of paper from 4 to 19 inches in width. The vertical paper tension may be relieved or increased, to ensure high quality print, and to permit various weights of paper or card stock to be handled.

In addition, a manual control is provided that raises or lowers the form, so that printed lines may be precisely located.

The Line Printer does not perform editing functions.

Skipping and vertical formatting are accomplished through the use of a pre-punched, eight channel vertical format control tape.

#### 8.3.1 LINE PRINTER ON-LINE.

The on-line off-line status of the Line Printer may be tested by the program, refer to Section 7.

The rated speed of the Line Printer is 1000 lines per minute. The effective speed is governed by the line content, the computing time, and the between line skipping requirements, of the particular application.

##### 8.3.1.1 DATA TRANSFER.

a. Data Flow. The data to be printed is read, from Core Memory, a word at a time to the Uni-Record Interface, and on through the Magnetic Tape Console to the Core Buffer in the Line Printer. When the Core Buffer is full, and the previous line has been printed, the printer skips, and printing is initiated. Immediately after each word has been read out of Core Memory, the operand portion of the Interface Instruction register is incremented, and the Interface Word Counter is decremented. When the last word to be printed has been transferred to the Printer Core Buffer, the operation is complete, and the Interface requests a Uni-Record Interface Not Busy Interrupt.

b. Error Checking. Parity error checking is performed in the M register and at the Magnetic Tape Console. No parity checking is done in the Line Printer.

### 8. 3. 1. 2 LINE PRINTER INSTRUCTION OPERATION.

For instruction format, refer to paragraph 8. 2. 1. 4. Mnemonic PRNT, PRTCTL

a. Output (Y6 OFF) - PRNT. An Output instruction that addresses the Line Printer with the Y6 bit OFF, will cause up to 17\* consecutive words to be transferred from Core Memory to the Line Printer, beginning with the effective operand address. The number of word(s) transferred is specified by the Uni-Record Interface Word Counter, which must be preloaded. When fewer than 17 words are specified, the remainder of the line will be padded out with blanks.

b. Output (Y6 ON) - PRTCTL. An Output instruction with the Y6 bit ON that addresses the Line Printer is a control instruction. It will turn Vertical Format to Automatic mode if the word at the effective address begins with a 1. If the effective address references a word that begins with any character except 1 or 2, the Line Printer will be rendered inoperative. Restoring the unit to an operative state may be accomplished by manual means only. Since the Word Counter is not loaded for these operations, it is ignored and undisturbed.

Vertical Formatting is turned to Controlled mode by executing an Output instruction with Y6 ON, referencing a word beginning with a 2.

c. Vertical Format (Skipping). Vertical spacing is determined by program or automatic format control, utilizing a pre-punched, 8 channel, vertical format control tape. Vertical spacing is 6 lines to the inch, manually adjustable to 8.

(1) Controlled Vertical Format. In the Controlled Format mode, the three least significant bits of the most significant character in the first word transferred to the Line Printer (carriage control character) specify a channel. Channels are specified with the characters 0 through 7 for Channels 1 through 8, respectively. The carriage will advance the paper until a hole in the specified channel on the control tape is sensed, at which time the line will be printed. The second character transferred will be printed in the first hammer position. The carriage control character will not be printed, however, up to 132 consecutive characters will be printed, as specified by the Word Counter. For each line printed, the Word Counter must be loaded and an Output instruction issued.

---

\*If Controlled Vertical Format is used, no matter what number is loaded into the Interface Word Counter, no more than 133 characters will be transferred to the Line Printer. If Automatic Vertical Formatting is used, the Uni-Record Interface Word Counter may specify any number of words (lines).

(2) Automatic Vertical Format. The Automatic Format mode allows continuous printing without beginning each line with a format control character. Channel 8 on the vertical format tape is selected and the skipping will be dictated by the contents of that channel. Any number of consecutive words, and thus lines, may be specified by the Interface Word Counter. Sixteen words and four characters from the seventeenth word will be printed. The remaining four characters of the seventeenth word will be printed as the first four characters of the next line, which will then fill out with 16 more words. The Uni-Record Interface remains Busy until all specified words have been transferred.

Vertical format channels 1 and 2 are used with the manual Line Printer controls, paragraph 8.3.2.

d. Timing. The rated printing speed is 1000 lines per minute. The effective speed varies with the skipping, line content, and computing requirements.

For each line skipped, there is a 25% loss of speed. For example, double space printing will print at 800 lines per minute.

Printing successive lines containing characters from the last quadrant of the print drum will slow the printing speed down to 500 lines per minute. The characters on the Print Drum are divided into four quadrants, table 8-2.

In the case where more than one line is specified, the program may need to know when a line has been transferred out of the Central Processor. This may be accomplished through the use of the Interface Word Counter and/or the Interface Instruction register. Seventeen words (one line) will be transferred from the Central Processor at a maximum rate of one line every 59.8 milliseconds.

When one line at a time operation is specified, the Uni-Record Interface remains Busy only while data is being transferred to the Printer Buffer (less than 3 milliseconds). Therefore, the program is free to use other Uni-Record devices during the 56.8 milliseconds of print time. However, to maintain maximum effective printing speed, the next Output instruction must be issued within 59.8 milliseconds.

Table 8-2 Line Printer Characters

PRINT DRUM QUADRANTS

<u>Quadrant I</u>	<u>Quadrant II</u>	<u>Quadrant III</u>	<u>Quadrant IV (Note 1)</u>
∅	+	:	>
1	/	J	<
2	S	K	]
3	T	L	⌋ (EOM)
4	U	M	#
5	V	N	°(degree)
6	W	O	;
7	X	P	[
8	Y	Q	^
9	Z	R	\$
=	(	-	∇
(apostrophe)	(comma)	*	⊕
(blank)	D	H	□
A	E	I	(underline)
B	F	)	≠
C	G	(period)	?

Note 1: The inclusion of these characters in successive lines will slow the operation to 500 lines per minute.

e. Error Checking. Parity is checked as the data is being transferred through the Central Processor M Register, and at the Magnetic Tape Console. A blown fuse, paper jam, or ribbon jam will give a mechanical failure indication.

Any error will terminate the operation.

Error recovery procedures are governed largely by the application, however, the Interface Instruction register and the Interface Word Counter may be used to indicate the point of termination.

### 8.3.2 LINE PRINTER OFF-LINE.

Manual controls are provided that allow an operator to operate the Line Printer with the Card Reader and with a Tape Transport off-line. Using the manual controls, the operator can list card information from the Card Reader or list tape information.

8.3.2.1 OPERATION WITH A TAPE TRANSPORT. To list a tape on the Line Printer, the operator must perform the following, figure 8-4:

- a. Rotate the Tape Transport Selection switch to the OFF-LINE PRINTER position. This selects the Tape Transport for the Line Printer.
- b. Rotate the Printer Unit Selection switch to the OFF-LINE TAPE position. This selects the Line printer for the off-line Tape Transport.
- c. Depress the Logic Reset button located on the Tape Control Section of the Magnetic Tape Console Control Panel. This clears the tape control logic and places it in an Idle state.
- d. Depress the Logic Reset button located on the Line Printer Control Section of Magnetic Tape Console Control Panel. This clears the Line Printer control logic and places it in an Idle state.
- e. Position the tape using the manual Tape Position rotary switch.
- f. Place the Tape Transport in the Automatic mode by depressing the Auto Mode button located on the Tape Control section of the Magnetic Tape Console Control Panel. This makes the Tape Transport ready to transmit data to the Line Printer.
- g. Place the Line Printer on-line with the Tape Unit Transport, paragraph 8.3.2.3.
- h. Depress the List Tape or List File button located on the Line Printer Control Section of the Magnetic Tape Console Control Panel. Depressing either of these buttons causes the printing of data to be initiated. Depressing the List File button causes the printing of data from the tape until a file mark is detected. Depressing the List Tape button causes the printing of data from the tape until an end of tape marker is detected. When an end of tape marker is detected, the Tape Transport will rewind automatically.

8.3.2.2 OPERATION WITH THE CARD READER/PUNCH. To list cards on the Line Printer, perform the following:

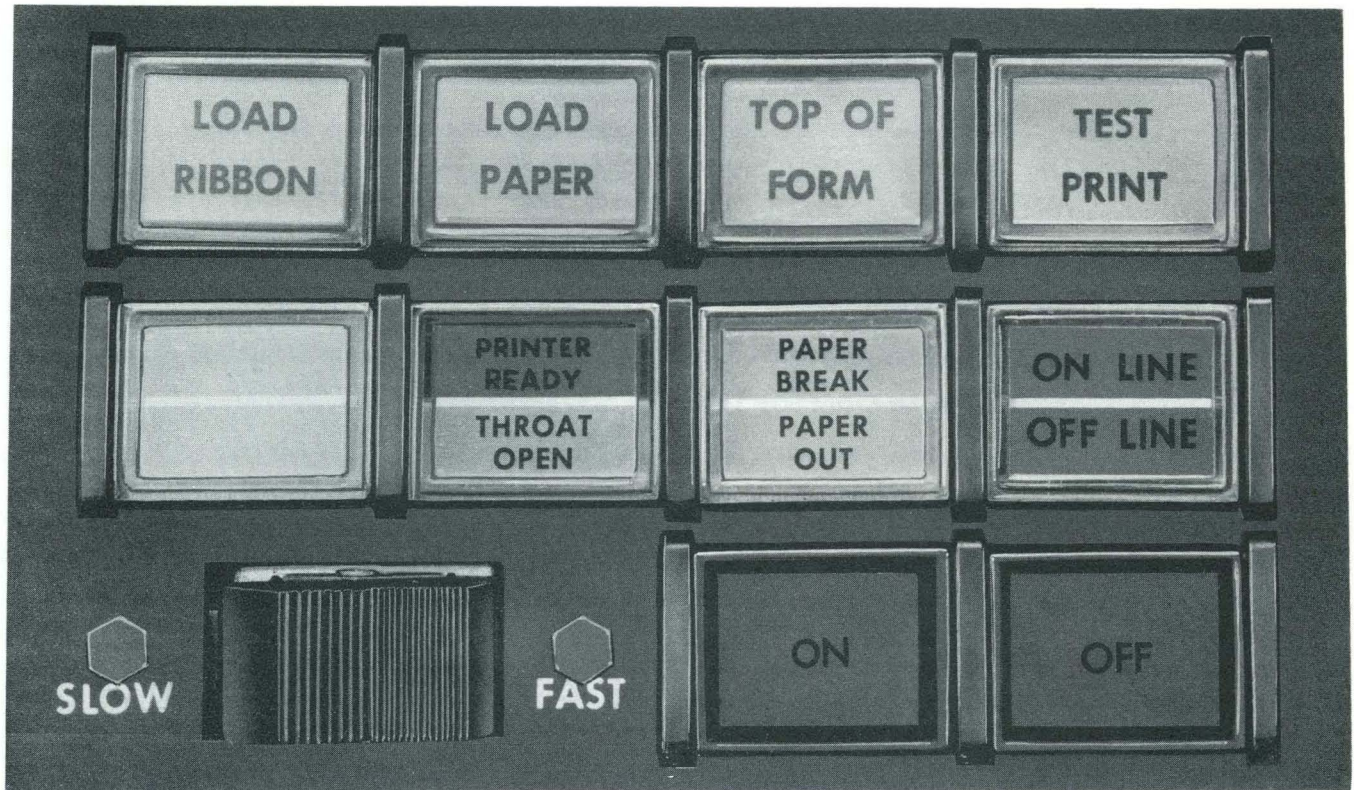


Figure 8-4. Line Printer Control Panel



Section VIII  
Paragraph 8.3.2.3

- a. Rotate the Card Unit Selection switch to the OFF-LINE PRINTER position. This selects the Card Reader/Punch for the Line Printer.
- b. Rotate the Line Printer Selection switch to the OFF-LINE CARDS position.
- c. Depress the Reader On buttons located on the Card Reader/Punch and the Card Control Section of the Magnetic Tape Console Control Panel.
- d. Depress the Line Printer On button located on the Line Printer and place the Line Printer on-line with the Tape Transport.
- e. Depress the Logic Reset button located on the Line Printer Control Section of the Magnetic Tape Console Control Panel. This action clears the Line Printer control logic.
- f. Depress the Error Reset button located on the Card Control Section of the Magnetic Tape Console Control Panel. This clears the card control logic.
- g. Depress the List Cards button located on the Line Printer Control Section of the Magnetic Tape Console Control Panel. This initiates the listing of cards that will continue until a card stacker is filled with cards, or all the cards have been printed, or a parity error has occurred. In a filled stacker condition, removing the cards will cause the printing of cards to be re-initiated. In an error condition, the operator must depress, in sequence, the Error Reset button for the card control logic, the Logic Reset button for the line printer control logic and the List Cards button to restart the printing of cards.

8.3.2.3 LINE PRINTER CONTROLS. To turn on the Line Printer, the operator must depress the On button on the Line Printer. In addition to this, the operator must depress the On-Line/Off-Line button on the Line Printer. This places the Line Printer on-line.

Several controls are located on the Line Printer which enable the operator to position the paper in the Line Printer correctly. A Top of Form button advances the paper in the Line Printer until a Top of Form punch (channel 2) is detected in the control tape. A Load Paper button advances the paper one line at a time for ease of positioning of the paper, responding to punches in channel 1 of the control tape.

A Test Print button on the Line Printer, when depressed, causes the Line Printer to print continuous rows of characters. This enables the operator to adjust the character phasing and penetration on the Line Printer. The character to be

printed is determined by the settings of the DS1-DS6 switches located on the back of the Line Printer. The character phasing and penetration controls are located respectively on the front and top of the Line Printer. A Form Position control on the front of the Line Printer allows the operator to position a printed line of information accurately on the printed lines of the paper being used.

Several indicators on the front of the Line Printer indicate a Line Printer hang-up condition.

A Paper Break indicator signals the operator that paper is not being fed through the Line Printer. A Paper Out indicator signals the operator that paper is not being fed into the Line Printer. A Throat Open indicator signals the operator that the sliding glass cover over the Line Printer is open. Both the Paper Break and Paper Out indicators, when activated, place the Line Printer off-line.

The Fast-Slow switch has no effect on Line Printer operation, except when printing with the Test Print button, in which case it selects test printing at 500 or 1000 lines per minute.



## SECTION IX

## DISC MEMORY

The Disc Memory provides bulk storage of 124 megabits of data and provides real-time storage of one million bits of data for the Data Processing System. The Disc Memory is connected to the Central Processor via a trunkline that has provisions to accommodate a total of seven Disc Memories. Each Disc Memory is in turn equipped with provisions to accommodate seven Auxiliary Disc Memory consoles. If required, the Disc Memory associated with a Central Processor may be expanded to seven Disc Memories and forty-nine Auxiliary Disc Memories to provide bulk storage of approximately seven billion bits of data.

Operationally, the Central Processor program, via the Disc Memory Interface in the Central Processor, addresses a Disc Memory and transmits prepare-file data to the Disc Memory. This operation may be in one of many modes, including storing and retrieving by fixed address, storing on available obsolete areas, retrieving by record content, obsoleting purged data or output to display. After being instructed for an operation, the Disc Memory operates simultaneously with the Central Processor, and data transfer to or from the Core Memory is on an interlace basis with the Central Processor program, until the transfer is complete. When the transfer is complete, the Central Processor is notified via a Unit Not Busy Interrupt. Disc Memory to Central Processor and Central Processor to Disc Memory transfers are buffered through a high speed (400 nanoseconds cycle time) Core Buffer that provides storage for the key, mask and control information used to set up the search criteria on content retrieval operations.

Bulk/auxiliary storage is arranged in 128 character block storage units. A block or multiples of a block can be key searched, transferred, obsoleted or counted. The next addressable unit of information is a data-band which contains 1350 blocks. Each of the two twins in a Disc Memory contains 60 data-bands. Each Disc Memory contains six discs, three discs per twin. These twins may be operated in redundant mode (write and read the same data in parallel) or in separate mode, at program option. Error checking is performed on all Disc Memory operations and data transfers to protect stored data and provide an error alarm.

Section IX  
Paragraphs 9.1 to 9.1.3

Real-time storage capacity of a Disc Memory is approximately one million bits of data. Data for display on the Display Consoles is transmitted by the Central Processor to the Disc Memory in the same manner as all other data. The data format is changed in the Disc Memory to conform to the requirements of the respective display interface.

The twin feature allows complete duplexing by allowing, at program option, the same data to be written into both Twins. Similarly, the data may be read in dual, and a detected error in one Twin or a difference between the Twins, will be indicated to the Central Processor. If dual operation is not desired, either Twin may be operated in a single mode and addressed as a single unit at program option.

9.1 ORGANIZATION OF DATA.

9.1.1 CHARACTER.

A character in the Disc Memory is returned as a Central Processor character, as defined in paragraph 2.3.2. Therefore, Disc Memory Read operations can transfer selectively, on a character by character basis and content searches can be specified on individual bits of characters (i. e. bit packing or binary coding within a character is possible).

9.1.2 BLOCK.

The block, consisting of 128 characters, is the unit of storage in the Disc Memory, and represents the least amount of information that can be transferred by a Write operation. The capacity of a Disc Memory is 162,000 blocks, each of which is equivalent to 16 Central Processor words.

9.1.3 DATA-BAND.

The next level of storage in Disc Memory is the data-band, containing 1350 blocks. The data-band is the maximum amount of storage that can be treated in a single disc revolution. Since operations can proceed from one data-band to the next without loss of time, the maximum sustained block examination rate is approximately 20,000 per second. (In other areas of documentation, the data-band is referred to as a file. The term, file, is avoided in this area to prevent confusion with the functional sense of file, which could refer to any number of blocks stored in Disc Memory.)

A portion of a data-band configuration is shown in figure 9.1, and as illustrated, the blocks are divided into two sections. The first 18 character section of each

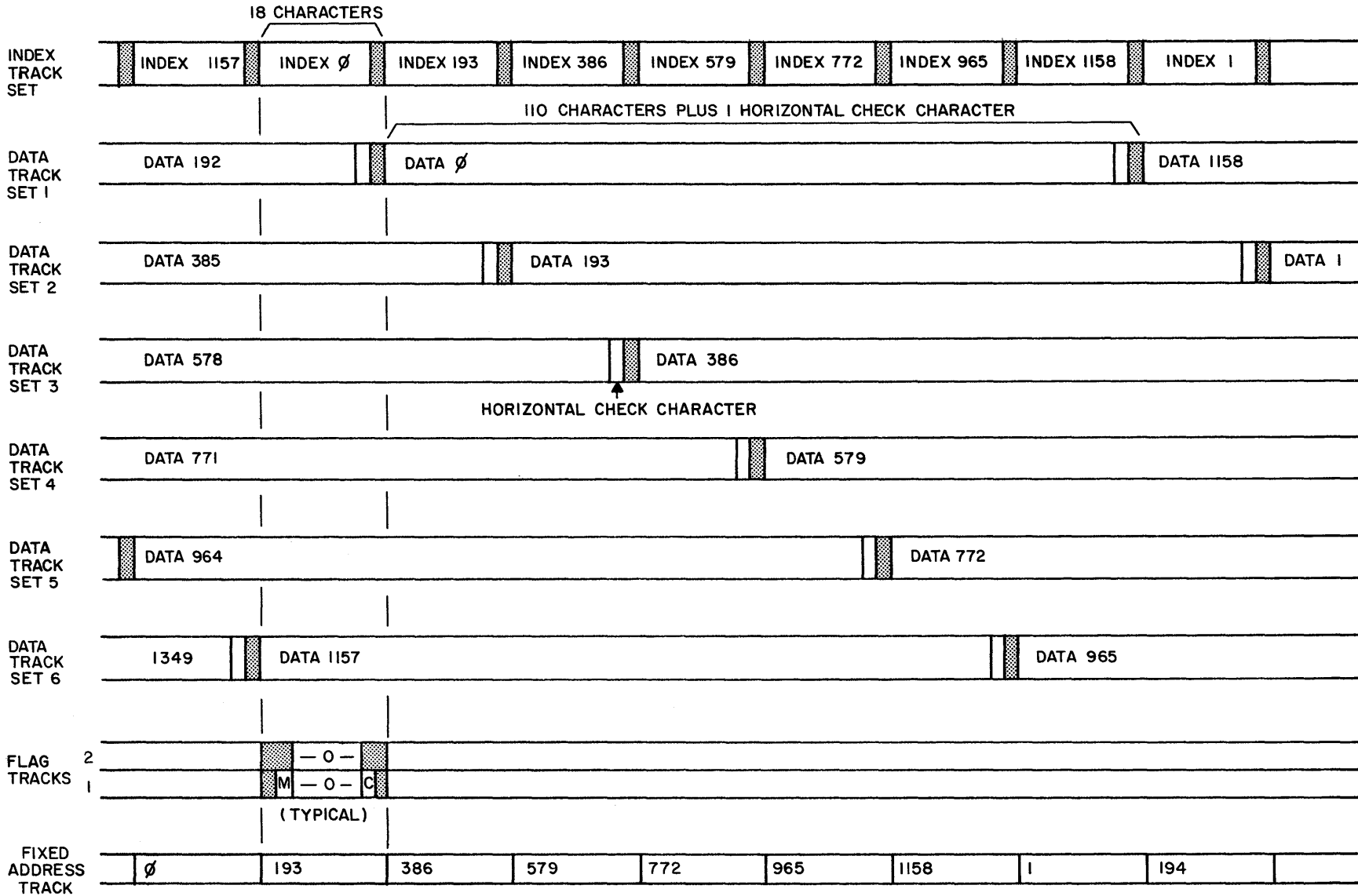


Figure 9-1. Index-Data-Fixed Address-Flag Format

Section IX  
Paragraphs 9. 1. 4 to 9. 1. 5

block is called the index, and is held in the Index Track Set. The remaining 110 characters are held in one of six additional track sets in such a manner as to follow the index immediately. (Each track set is comprised of 3 tracks carrying data in 3-bit parallel.)

The block numbers within a data-band are arranged with a modulo seven interlace pattern in the Fixed Address track. The arrangement was chosen to permit the writing by fixed address of successively numbered blocks as a continuous operation. Within a data-band, blocks are numbered decimally 0 to 1349.

#### 9. 1. 4 TWIN.

A twin is composed of 60 data-bands and is the maximum amount of storage that can be treated by a single Disc Memory operation. A Disc Memory is comprised of two twins designated Twin A and Twin B. Operations can be addressed to either twin (single mode), or to both twins (Twin mode). In Twin mode operation, writing is in duplicate, and reading involves comparison of the data and flags from the two twins.

#### 9.1.5 FLAG TRACKS.

Each Twin has a set of Flag tracks that carry three types of flags. The Flag tracks are shown in figure 9-1.

The first type is the Obsolete flag, of which there is one for each block of each data-band in each Twin. These are held in two tracks per twin. Obsolete flags for data-bands 00 to 29 are held in the first track and for 30 to 59 in the second track. The Obsolete flag indicates that a block has been obsoleted, i. e., does not have useful information, and may be written over. (There is no erase process as such.) The flag can be written by content search and by fixed address, and is removed when new information is written.

The second flag type is the Missed flag. Missed flags occur during a content search on the first revolution in a data-band, and identify those blocks that were not searched. Flags are necessary because comparison continues on the pertinent data track set after content agreement. As a result, as many as six indices may be missed. These indices are flagged with Missed flags, and compared on successive disc revolutions. There is a Missed flag for each block in one data-band in each Twin.

The third flag type is the Compare flag. During a content search, Compare flags identify those blocks within a data-band that meet the search criteria, but have not been transferred to the Central Processor. Special operations transfer such blocks and erase the Compare flags. Any operation that generates Compare flags is preceded by removal of all existing Compare flags. A Compare flag is provided for each block in one data-band in each Twin.

The Flag tracks are parity checked, and after a Flag Parity Error, paragraph 11.15.3, a Flag Error Character indicates the type of error, paragraph 10.6.1.

## 9.2 DISC MEMORY INSTRUCTION FORMAT.

A Disc Memory instruction is stored in normal Central Processor program sequence, but is composed of three consecutive words:

- a. The Interface Instruction word
- b. The Disc Memory Operation word
- c. The Key/Data Address and Mask/Control Address word

### 9.2.1 THE INTERFACE INSTRUCTION WORD.

The Interface Instruction word is recognized as the start of a Disc Memory instruction in the Central Processor Instruction register by the command character F. Operand address modification, indexing and indirect addressing, is specified and accomplished in the Central Processor Instruction register. The Interface Instruction register, where the Y character provides Disc Memory selection and the effective operand address specifies a Core Memory location for the start of data transfer.

The Interface Instruction word format, F Z Y X M M M M, is described as follows:

Command Character	F
Instruction	F Z Y X M M M M - In the Z character only Z6 is functional (Tagged Halt). The Y character provides Disc Memory selection.
Options	Index, Indirect Address, Tagged Halt, and Flag Return.
Mnemonic	D S K F I L



### 9.2.2 DISC MEMORY OPERATION WORD.

The Disc Memory Operation word has the function of specifying a Disc Memory operation, and selecting a zone, twin, and data-band for the start of the operation. Also specified are the number of data-bands over which the operation may occur, and the number of blocks that may be transferred by the operation.

Disc Memory Operation word format, O Z T A N N N N, is described as follows:

Operation Code	O - If bit O6 is true during Read operations, it specifies that a word containing a fixed address will be transferred prior to each block.
Zone and Twin Selection	Z - Bits Z1, Z2 and Z3 are reserved for zone selection in a future expanded Disc Memory. The present Disc Memory has only one zone and Z1, Z2, Z3 must be 1, 0, 0. Bits Z4 and Z5 select Twin A and Twin B, respectively. Bit Z6, if true, enables access to blocks that have been flagged obsolete for read operations.
Data-band Selection	TA - The numeric bits of the T and A characters specify a beginning data-band number from 00 to 59. Bits T6, T5, A6, and A5 form the 8, 4, 2, and 1 bits of a binary number from 0 to 15. A value from 1 to 15 specifies that many data-bands. A value of 0 specifies through data-band.
Maximum Blocks Transferred	NNNN - The numeric bits constitute a decimal number from 0000 to 9999. Zone bits are ignored. This specifies a limit on the number of blocks transferred between the Core Memory and Disc Memory.

### 9.2.3 KEY/DATA ADDRESS AND MASK-CONTROL ADDRESS WORD.

The Key/Data Address and Mask-Control Address word is formed by two four-character decimal addresses. These may be Core Memory addresses in which case they are in normal format, paragraph 3.2.1. The application of the first address depends on the type of Disc Memory operation. For fixed address operations it is a block address within a data-band. Otherwise, subject to indirect addressing, it is the location of the first word of key information in the Core Memory. The second address is the location, subject to indirect addressing, of the first word of Mask and Control information. Either address field may specify

indirect addressing by a 1 bit in the sixth bit position of the least significant character.

Key/Data Address and Mask-Control Address word format,  $M_2 M_2 M_2 M_2 M_3 M_3 M_3 M_3$ , is described as follows:

**First Address**       $M_2 M_2 M_2 M_2$  - In content search operations, is a Core Memory address for the first word of key information, and in fixed address operations, a block number within a data-band (subject to indirect addressing). When indirectly addressed, as indicated by a 1 in the sixth bit position and a  $\emptyset$  in the fifth bit position of the rightmost character, the four high order characters of the specified word in Core Memory become the  $M_2 M_2 M_2 M_2$  field. On a content search operation a Blank character in the rightmost character position of the address field specifies no change to the key.

**Second Address**       $M_3 M_3 M_3 M_3$  - In content search operations, is a Core Memory address for the first word of Mask and Control information (subject to indirect addressing). When indirectly addressed, as indicated by a 1 in the sixth bit position and a  $\emptyset$  in the fifth bit position of the rightmost character, the four low order characters of the specified word in Core Memory become the  $M_3 M_3 M_3 M_3$  field. On a content search operation a Blank character in the rightmost character position specifies no change to the Mask or Control.

#### 9.2.4 KEY FORMAT.

The Key information, starting in the effective  $M_2 M_2 M_2 M_2$  address, is specified by fields. Each field is headed by a field specifier word that gives the starting character position in the Key register and the number of characters in the field. The field specifier word must be all decimal characters, except when heading the last field, in which case the first character is EOM ( $\neg$ ).

The Key format is as follows:

$M_2$	$\emptyset X X X \emptyset N N N$	Field Specifier Word
$M_2 + 1$	$K_1 K_2 \dots \dots \dots )$	First Field of Key Characters
$M_2 + i$	$\dots \dots K_n \dots \dots )$	

Section IX  
Paragraph 9.2.5

$M_2 + i + 1$	$\emptyset X X X \emptyset N N N$	Occupies next available word
.	$K_1 K_2 \dots \dots \dots )$	Second Field of Key Characters
.	$\dots \dots K_n \dots \dots )$	
.	$\neg X X X \emptyset N N N$	
.	$K_1 K_2 \dots \dots \dots )$	Last Field of Key Characters
.	$\dots \dots K_n \dots \dots )$	
.		

The XXX characters give in decimal the Key register character location (from 1 to 128) of the first Key character. The NNN characters give in decimal the number of characters in the field.

A value of  $\emptyset\emptyset\emptyset$  for XXX specifies the use of the first Key character,  $K_1$ , in the following key field for the first NNN Key register character positions.

9.2.5 MASK AND CONTROL FORMAT.

The Mask information, starting in the effective  $M_3 M_3 M_3 M_3$  address, is specified by fields in a manner very similar to the Key, paragraph 9.2.4. A decimal  $\emptyset$  Mask character specifies no comparison.

The field specifier word has a Control character in the  $C_3$  position that specifies the Control register content to be loaded in all character positions in the field.

This character is interpreted as follows:

Bits 1 and 2 specify the type of search for this field.

- $\frac{2}{\emptyset} \frac{1}{\emptyset}$  specifies Equal to Key search
- $\emptyset \frac{1}{\emptyset}$  specifies Greater Than or Equal to Key search
- $\frac{1}{\emptyset} \emptyset$  specifies Less Than or Equal to Key search
- $\frac{1}{\emptyset} \frac{1}{\emptyset}$  specifies Bounded search, or Not Equal to Key search, see paragraph 9.5.8 for Content Access operation.

Bit 3 controls transfer of the fields to the Central Processor (1 indicates no transfer).

Bits 4 and 5 and 6 of the control character are disregarded, but the control character is required to be a decimal character.

Bit 4 in the Control register is a field designator set to a 1 at the beginning of the field and in the character just after the field.

The program cannot select a portion of a search field for transfer; the whole search field must be transferred, or none of it.

### 9.3 DISC MEMORY COMPONENTS.

#### 9.3.1 DISC STORAGE.

The disc assembly of the Disc Memory rotates at 900 rpm (less 5% slip), providing an average access time of 35 milliseconds. Each storage track has its own read/write head and thereby prevents additional access time for positioning heads.

Control circuits and search flag tracks are provided to deal with one data-band at a time within one twin, and operations involving several data-bands deal with each data-band in consecutive order. No time is lost in switching from one data-band to the next.

Information on the discs is non-volatile, and will be preserved through normal shut-down. If a power failure or data transfer error occurs during writing, the maximum loss of information will be confined to the block being written.

#### 9.3.2 CORE BUFFER AND SEARCH REGISTERS.

A special purpose high speed Core Memory is used as a buffer for data transfers and as Key, Mask and Control registers for content searching.

The buffer portion is wholly automatic and the speed is high enough to interlace loading and unloading, both at disc character rate or greater. The buffer holds one block of characters (128), each with its parity bit. The parity bits are stripped as blocks are recorded on the disc, and a block longitudinal parity (3-bit) character added. Character parity bits are reconstituted when reading from the disc to the buffer.

The Key, Mask and Control registers are used to specify the conditions of a content search. They hold a six-bit key character, a six-bit mask character, four bits of control and a parity bit for each of the 128 character positions in a block.

#### 9.3.3 DISC MEMORY OPERATION REGISTER.

The Disc Memory Operation register is part of the Disc Memory Control Section. It is loaded with the Disc Memory Operation word, OZTANNNN. It also has a

Section IX  
Paragraph 9.3.4

block address holder, BBBB, that is loaded with the  $M_2M_2M_2M_2$  address on fixed address operations.

The NNNN characters of the Disc Memory Operation register constitute a counter that limits the number of blocks transferred or obsoleted and counts the number of blocks on a Count operation.

The BBBB characters of the Disc Memory Operation register always carry the block address of any block being operated on, or in the case of error, the address where the error occurred is indicated.

The content of the Disc Memory Operation register can be recalled by a Return Disc Memory Operation ( $\geq$ ) instruction, paragraph 10.6.1.

#### 9.3.4 MC BUFFER STORE.

The disc assembly has a specialized real-time storage area with a capacity of 1000 blocks for buffer storage of messages enroute to Multi-color (MC) Displays. This storage area is loaded and unloaded cyclically under control of counters.

A special Disc Memory instruction selects an individual display (by the TA characters) and loads the number of blocks specified by NNNN into the MC Buffer Store. Insufficient space for the specified number of blocks causes Display Back-up Status which may be tested by program.

Blocks are picked up automatically from the MC Buffer Store and sent word by word to the MC Displays by Output Channels. There are 15 Output Channels which are capable of simultaneous transmission in response to MC demand. Each Output channel has a maximum transmission rate of 180 words per second.

As an Output Channel completes transmission of the last (the 16th) word of a block, it finds the block in the MC Buffer Store and marks it obsolete, and obtains another block for transmission. When all blocks addressed to a given MC are depleted, a Display Interrupt request is generated.

The Unload counter keeps track of the oldest block in MC Buffer Store that has not been transmitted and marked obsolete. This counter is used to prevent overwriting when loading the real-time store from the Central Processor.

If a message encounters transmission difficulties, the transmission will abort after three transmission attempts and all blocks of the message will be obsoleted. At the same time a Display Interrupt request is generated.

A special Disc Memory instruction recalls a status word indicating the condition of each Output Channel.

A message addressed to an Output Channel that is busy transmitting, or that does not have an active MC, will not load into MC Buffer Store, but a Display Interrupt request is generated immediately.

#### 9.3.5 CP BUFFER STORE.

The disc assembly has a specialized storage area with a capacity of 120 blocks for buffer storage of messages enroute to Console Printers (CP). The CP Buffer Store is organized into 15 sectors of 8 blocks each, and each sector is storage for an individual CP.

A special Disc Memory instruction selects an individual CP and loads into its sector the number of blocks, up to 8, specified by NNNN. An attempt to load more than 8 blocks produces an Instruction Error Interrupt request in the Disc Memory Interface.

After being loaded, sectors output characters in response to CP demand at a maximum rate of one character per disc revolution. All 15 sectors can output simultaneously.

When an EOM character is sent by a sector, or when all the blocks loaded are sent, transmission terminates, and a Display Interrupt request is generated. A special Disc Memory operation recalls a status word indicating the condition of each Output Channel.

A message addressed to a sector that is busy transmitting, or that does not have an active CP will not load into CP Buffer Store, but a Display Interrupt request is generated immediately.

#### 9.4 DISC MEMORY INTERFACE.

The Disc Memory Interface is a part of the Central Processor that executes the data transfer function of Disc Memory instructions. The data transfer between Core Memory and Disc Memory proceeds independently of Central Processor program execution.

The Disc Memory Interface, figure E-1, contains registers, and error and status toggles which are described in the following sections.

## Section IX

### Paragraphs 9.4.1 to 9.4.4

#### 9.4.1 INTERFACE INSTRUCTION REGISTER.

The Disc Memory Interface Instruction register is loaded with the Y and MMMM characters of the Interface Instruction word, and with the O character of the Disc Memory Operation word.

The Y character provides Disc Memory module selection.

The MMMM characters provide a Core Memory location for data transfer, and count through Core Memory addresses. The O character defines the Disc Memory operation, but is also used in the interface.

During or after a Disc Memory operation, the content of the Interface Instruction register can be recalled by a Save instruction.

#### 9.4.2 INTERFACE INSTRUCTION LOCATION REGISTER.

The Disc Memory Interface Instruction Location register is loaded with the location of the third word of each Disc Memory instruction. During or after a Disc Memory operation, its content can be recalled by a Save instruction.

#### 9.4.3 INTERFACE ADDRESS PROCESSING REGISTER.

The Disc Memory Interface has a register, inaccessible to the Central Processor program, in which the  $M_2 M_2 M_2 M_2$  and  $M_3 M_3 M_3 M_3$  addresses are processed, and then used to obtain Key and Mask-Control data from Core Memory.

#### 9.4.4 DISC MEMORY INTERFACE OPERATION.

The three word Disc Memory instruction is intended to be put in the program sequence as though it were three program steps. A normal instruction fetch memory cycle puts the first word, the Disc Memory Interface Instruction, in the Central Processor Instruction register. The address modifying options take place, as described in paragraph 3.3, and produce an effective operand address.

At the point of instruction execution, the instruction is transferred to the Disc Memory Interface. The Central Processor Instruction Address register is incremented twice, with Core Memory cycles initiated. The second and third words of the Disc Memory Instruction are transferred directly from the Core Memory M register to the Interface. Along with the third word, its location, which is the content of the Central Processor Instruction Address register, is transferred to the Disc Memory Interface. The Central Processor then proceeds with the fetch and execution of the next instruction in sequence.

As the three words of the Disc Memory instruction become available, the following portions are transferred to the Interface Instruction register:

- a. The Y character and the effective operand address.
- b. The Disc Memory operation code.
- c. The  $M_2 M_2 M_2 M_2$  address.

The Interface Instruction Location register is also loaded with the location of the third word for later recall of the  $M_3 M_3 M_3 M_3$  address. The second word, the Disc Memory operation, is transferred to the Disc Memory.

Any indirect address processing is done on the  $M_2$  address, and then it is transferred to the Disc Memory, or used to start transfer of the Key information to Disc Memory, as is appropriate to the type of operation.

When  $M_2$  is disposed of,  $M_3$  is recalled from Core Memory and processed for indirect addressing in place of  $M_2$ .  $M_3$  is then used to start transfer of Mask and Control information to the Disc Memory.

A Blank in the least significant character position of either  $M_2$  or  $M_3$  indicates that there is no transfer to take place, hence use of the previous Key, Mask or Control information.

Should the Disc Memory Interface be Busy when the interface instruction is ready for transfer, the Central Processor is held up until the Interface goes Not Busy and the transfer takes place.

An Execute (X) instruction can be substituted for the first word of the Disc Memory instruction, provided that the Execute instruction references a valid Disc Memory Interface instruction. The second and third words of the Disc Memory instruction will be those following the Execute instruction.

#### 9.5 DISC MEMORY OPERATION.

The Disc Memory goes Busy when it is selected by the Y character in the Interface Instruction register. The Disc Memory Operation is then transferred to the Disc Memory Operation register. This transfer is checked, and an error generates a Transmission Error Interrupt request.



Section IX  
Paragraphs 9.5.1 to 9.5.2

Depending on the type of instruction, the Interface then transfers fixed address or Key and Mask-Control information. When this is complete, the Disc Memory takes over and executes its operation.

Disc Memory operations may be classified as to the type of access and function, such as read, write, or count. Not all combinations of access and function are meaningful, so a definite and specific list of operations is provided, detailed in Section 10. Some recall operations are provided to enable the Central Processor program to find out what has happened in the Disc Memory.

#### 9.5.1 WRITE FUNCTION.

The Write function involves the transfer of words from successive locations in Core Memory and the writing of these words in blocks of 16 words each in Disc Memory. As blocks are written, the Obsolete Flag is set to indicate non-obsolete. Blocks may be written continuously in successive block locations, or intermittently.

At the completion of a Write operation, the Disc Memory Operation register will indicate how much of the operation was accomplished and where it finished. The N register will contain the number of blocks not written on the disc. The TA characters will indicate the last data-band addressed as well as the number of data-bands not written upon. The B register will contain the address of the last block written.

If the write operation is an 8 (write individual blocks by obsolete flag), then when more than one data-band is specified and no obsolete flags existed on the latter data-bands, the TA characters will not indicate the last data-band on which a block was written. The B register however, still contains the fixed address of the last block written. This case can only occur when the operation ends with an overload or because of an error.

#### 9.5.2 READ FUNCTION.

The Read function involves the transfer of data under control of the Control register from Disc Memory into successive words of Core Memory. Control information in the Control register designates, character by character, the characters from qualifying blocks to be transferred. By padding out the last word transferred from each block, blocks do not become intermixed in Central Processor words. The padding is with Blank characters. Bit O6 of the Disc Memory Operation word, if true, specifies that each block read will be preceded by a word containing the fixed address of the block and part of the Disc Memory Operation Word.

The Block Fixed Address word format, O Z TA BBBB , is described as follows:

- O      Disc Memory Operation (not changed).
- Z      Zone and Twin Selection (not changed).
- TA     Data-band from which the block was read. The data-band count, carried in bits T6, T5, A6 and A5, will be decremented according to the progress of the Read operation.
- BBBB    Block Number.

During a read operation, the B register is filled with the fixed address of each block read. The fixed address is also transferred to the B register when no block is being read (except on a fixed address read) so that at the completion of the operation, the fixed address of the last block transferred to the Core Memory is not necessarily that in the B register. The above also applies to a Count or Obsolete operation. When the operation is on consecutive blocks, it ends after the last block; so this address will be in the B register.

The TA characters (at the completion of the operation) will also indicate the data-band of the last block only on a Multi-block operation.

### 9.5.3 OBSOLETE FUNCTION.

The Obsolete function sets the Obsolete Flags of qualifying blocks to the Obsolete condition. This function can be employed by itself, or in conjunction with the Read function. A block is non-obsolete by the function of writing it.

### 9.5.4 COUNT FUNCTION.

The Count function is used to determine the number of blocks that qualify for a designated search criterion without the necessity of transferring any data into Core Memory. The count is accumulated in the NNNN part of the Disc Memory Operation register and can be retrieved by the Central Processor program.

### 9.5.5 FIXED ADDRESS ACCESS.

Fixed Address Access is the specification of a specific block within a specified data-band which is the first block to be operated on by a function. In a multi-block function, blocks are operated on in sequence of fixed addresses, including overlapping from one data-band to the next. As the number of data-bands over which the function is to operate is specified as well as the number of blocks, it is possible to get overload status, because the data-band limitation is the tighter one.

#### 9.5.6 OBSOLETE ACCESS.

Obsolete Access specifies, for write functions, the use of blocks flagged obsolete as they appear in the specified data-band. If allowed by the data-band count, the Write function can overlap into the next data-band when one is filled. Obsolete Access operates on a block by block basis, typically writing blocks here and there in a partially filled data-band.

#### 9.5.7 CONTENT ACCESS.

The use of Content Access implies a search operation, wherein Key, Mask and Control information is given. A search operation may be specified on any combination of bit, character, and field positions within a block. The different types of search criteria and their formats are discussed in paragraphs 9.2.4 and 9.2.5. An example is given below.

The Content Access process works on only one data-band at a time, although a region of several data-bands can be specified. The Content Access process is to determine the blocks that qualify the search criteria contained in the Key and Mask/Control registers. These blocks may be read, counted and/or obsoleted.

The location within the block of the mask information has a definite bearing on the speed of the search. If all blocks happen to disqualify, or if the mask only allows comparison, in the index part of the block (the first 18 characters), these can all be compared in one disc revolution for one data-band. The comparison of the indices is interrupted by either a comparison or a transfer of the rest of the block. Those indices passed by on the first revolution are labeled with Missed Flags, and are treated on subsequent revolutions. The reading of six blocks in a worst case condition takes seven revolutions, including a final one that ensures that there are no remaining Compare or Missed Flags.

Obsolete and Count functions, unlike Read functions, always go to completion over the specified data-bands. Compare flags remain pertaining only to the last data-band examined.

The Content Access operation is controlled by the information stored in the Key, Mask, and Control registers. In most cases the search is performed on a bit by bit correspondence of the Key and the data being read. The different types of searches as specified by the control bits and the functions performed by the Key and Mask registers are described below.

<u>Control Bits</u>		<u>Type of Search</u>	<u>Qualifying Requirements</u>
2	1		
0	0	Equal to	The bits of the data being read must be identical to the corresponding bits of the key where the mask bits are ones.
0	1	Greater than	The field of data read must be equal to or larger, in the collation sequence, than the key field in those bit positions where the mask bits are ones.
1	0	Less than	The field of data read must be equal to or smaller, in the collation sequence, than the key field in those bit positions where the mask bits are ones.
1	1	Bounded	The field of data read must be equal to or smaller, in the collation sequence, than the key field and equal to or larger than the data in the mask field. No bit masking is possible.
1	1	Negative	The field of data read must be not equal in at least one bit position to the key field, and the mask character(s) must be $\neg ( 0 0 1 1 1 1 )$ . No bit masking is possible.

The beginning of a field is specified by a 1 in the fourth bit position of the Control register. This bit is loaded by the transfer of a new mask field with a new control character to the registers. In addition, the fourth control bit in the next position after the last mask character of the field is also made a 1. Thus within a field, the type of search is always the same.

The third control bit, when a 1, is used to inhibit the transfer of that data character to the Central Processor Core Memory. A search field will have the same third control bit throughout; because when the mask field is loaded, the control character cannot change.

If a bounded search field is specified where the lower limit is larger than the upper limit, then no blocks will meet this criteria. However, no error signal will be turned on to indicate this condition.

If a negative search field is specified where one of the characters in the mask is not a  $\neg$  (EOM), then the instruction error will be turned on as soon as this field of a block is read, providing the read operation is by content. This means that a bounded search cannot be specified with a  $\neg$  as the most significant character of a field in the lower limit. If it is desired to do this, then an equal to search should

Section IX  
Paragraph 9. 5. 7 (Contd. )

be indicated on the most significant character(s) and a bounded search on the remainder of the field. The most significant character of the lower limit of a bounded search should therefore always be examined to make sure that it is not the largest character in the collation sequence ( $\neg$ ). However, this character can be used anywhere else in the lower limit of the bounded search field.

On the first revolution of a contents search on a data-band, for each block read into Core Memory there will be six blocks that cannot be read because of the inter-lace format of the blocks. When the block cannot be examined due to the reading of a previous block, the non-searched block on the first revolution is marked with a Missed flag. On succeeding revolutions, only those blocks with Missed flags are searched. Thus the maximum number of revolutions necessary to search a data-band is seven, and the last revolution may result in no data being found because the Missed flag blocks did not meet the search requirements.

When reading by content, it is only necessary to examine a block until the content does not satisfy the search criteria. Thus, if the search field is in the most significant portion of the block, fewer Missed flags will be written on the first revolution due to non-qualifying blocks. For example, if the search field(s) is entirely within the index of the block, Missed flags will only be written when a block meets the search criteria. This will result in fewer revolutions being necessary to examine the blocks with Missed flags, and hence, a higher speed of operation.

The read and/or obsolete multi-block operations ignore all remaining Missed flags once a matching block has been found. If no matching block is found on the first revolution, then searching continues on the present data-band until all blocks have been examined (no more Missed flags) or a matching block is found. (The Missed flag is erased as the block is examined).

The count by contents operations do not finish with a data-band until all blocks have been searched and no Missed flags remain. The Compare Flag Count operation leaves Compare flags on the last selected data-band only.

The Content Access read operations erase all Compare flags for qualifying blocks (if they are transferred to Central Processor Core Memory) as well as writing Missed flags for those blocks not examined. After N blocks have been transferred, the search continues and any blocks that qualify will be marked with a Compare flag as the Missed flag is erased. The Compare flagged blocks are not examined on

succeeding revolutions until more blocks are requested by the program. The search continues until all Missed flags are removed from the current data-band, but will not proceed to the next data-band until all Compare flags have been removed by transferring the blocks to the Core Memory.

Table 9-1 is an example of Key and Mask/Control information. The resulting register content and selected block content are shown in figure 9-2.

Fields 3 and 5 are the only ones transferred to the Central Processor in this example. The remaining two characters of the word are padded to the LSD with blank characters. A word such as this is transferred for each qualifying block, within the search restrictions.

Caution is necessary in overlaying fields because the field marks represent the fields as applied in the called out sequence.

Table 9-1 Example of Key and Mask/Control Information for Content Access

KEY INFORMATION		MASK/CONTROL INFORMATION		NOTE
-		Loc M3	00004128	Clears Mask register to 0's and sets 4's in the Control register (no compare, do not transfer). Character 1 gets a Start of Field bit (1 in bit 4 of Control register). No clearing of the Key register is done, and the previous content will remain where blanks are shown.
-		M3+1	0	
Loc M2	00020004	M3+2	00024004	Specifies an equal comparison with the four characters L, A, T, and E in the four characters starting in character 2, but no transfer of this field.
M2+1	LATE	etc	????	
-			00100004	Specifies no comparison of the four character field starting with character 10, but transfer into the Central Processor. No Key characters are loaded.
-			0000	
M2+2	00060001		00065001	Specifies a comparison in character 6 where the character must be less than 9 in the numeric bits, as masked by the 7 character. The zone bits are not compared, and the field will not be transferred.
M2+3	9		7	

Table 9-1 Example of Key and Mask/Control Information for Content Access (Cont'd)

KEY INFORMATION		MASK/CONTROL INFORMATION	NOTE
etc	01190002 4T	01192002 ??	Specifies a comparison in the two character field starting in character 119, where the 6-bit, alphanumeric characters, as masked by the ?'s, must be larger than 4T (in the character collation sequence, Appendix A). This field is transferred into the Central Processor.
	00080002 40	00087002 22	Specifies a bounded search where the two character field starting in character 8 must be greater than or equal to 22 and less than or equal to 40 (this is an alphanumeric comparison). The field is not transferred.
	01220003 RED	01227003 ⌈⌈⌈	Specifies a not-equal comparison in the three character field beginning in location 122. The alphanumeric content of the three character field must be anything except RED; for instance, REC qualifies the field.
	⌈1270001 2	⌈12740001 S	Specifies a bit masked comparison on character 127 where the 2's bit must be a 1 and the 5's bit must be a 0. This character is not transferred. As this is the last field of both the Key and Mask, the field specified words for each contain an ⌈(EOM) character.

9.5.8 COMPARE FLAG ACCESS.

A Compare Flag Access can be used for a Read function or a Read and Obsolete function following a Content Access that did not involve a Read function. A Compare Flag Access will operate over one data-band at most.

9.5.9 MULTI-BLOCK, FIRST QUALIFY ACCESS.

Access by Content, Obsolete Flag, or Compare Flag can operate in such a manner that the first block qualifying for the access criterion is the only basis for access,

CHARACTER POSITION

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

118 119 120 121 122 123 124 125 126 127 128

KEY REGISTER

	L	A	T	E	9		4	∅						
--	---	---	---	---	---	--	---	---	--	--	--	--	--	--

	4	T		R	E	D			2					
--	---	---	--	---	---	---	--	--	---	--	--	--	--	--

MASK REGISTER

∅	?	?	?	?	¬	∅	2	2	∅	∅	∅	∅	∅	∅
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

∅	?	?	∅	¬	¬	¬	∅	∅	S	∅				
---	---	---	---	---	---	---	---	---	---	---	--	--	--	--

CONTROL REGISTER (BY BIT)

4	1	1	∅	∅	∅	1	1	1	∅	1	∅	∅	∅	1	∅
3	1	1	1	1	1	1	1	1	∅	∅	∅	∅	1	1	
2	∅	∅	∅	∅	∅	∅	1	1	∅	∅	∅	∅	∅	∅	
1	∅	∅	∅	∅	∅	1	∅	1	1	∅	∅	∅	∅	∅	

	∅	1	∅	1	1	∅	∅	1	∅	1	1
	1	∅	∅	1	1	1	1	1	1	1	1
	∅	1	1	∅	1	1	1	∅	∅	∅	∅
	∅	∅	∅	∅	1	1	1	∅	∅	∅	∅

RESULTING BLOCK SELECTED

	L	A	T	E	≤ 9		≤ 4∅							
							≥ 22							

	≥ 4T		NOT "RED"				BIT 2=1							
							BIT 5=∅							

SEE NOTE

1		2		4	1	6		3		1		5	1	7		1	8	1
---	--	---	--	---	---	---	--	---	--	---	--	---	---	---	--	---	---	---

Figure 9-2. Content Access - Example



but the function operates on it and consecutively following blocks. The total number of blocks operated on is specified by the NNNN characters.

#### 9.6 TWIN MODE.

Twin mode is specified for data transfer or data access instruction by specifying both Twins A and B with bits Z4 and Z5 of the Disc Memory Operation word.

On Twin mode Write operations, blocks are written in corresponding locations in Twins A and B. During a Write by Obsolete Location operation, only corresponding obsolete locations are noted, affording some protection against inadvertent loss of data.

On Twin mode Read operations, blocks are transferred from Twin A and checked for parity, with a Horizontal Parity check. In addition, the blocks and their corresponding flags are checked bit by bit against Twin B, A discrepancy is indicated by the Twin Compare Status toggle in the Disc Memory Interface. This status toggle does not generate an error interrupt request but is reset at the start of each Disc Memory instruction. Therefore, the status toggle should be tested after each operation in Twin mode.

#### 9.7 STATUS TOGGLES.

The Disc Memory Interface has the following Status toggles that can be tested true or false by a T instruction, and generally set or reset by a V instruction. They are all reset at the start of a Disc Memory Instruction.

##### 9.7.1 OVERLOAD (ZY = //).

The Overload Status toggle is related to the counter which holds the NNNN characters of the Disc Memory Operation. Overload status is set if:

- a. On a Write operation, insufficient block spaces are available.
- b. On a Read operation, insufficient blocks qualify.
- c. On a Count operation, 9999 is exceeded.

##### 9.7.2 TWIN COMPARE (ZY = /S).

Twin Compare is set if there is a difference in data read from Twin A and Twin B in the Twin mode, or a difference in the Compare, Missed, or Obsolete flags.

### 9.7.3 OPERATIVE (ZY = /U).

The Operative Status toggle is set if the selected Disc Memory module is on-line and operative.

### 9.7.4 DISPLAY BACKUP (ZY = 34).

This status toggle indicates that an attempt was made to load the MC Buffer store past the unload point, or to transmit more than 8 blocks to a CP slot.

### 9.7.5 INTERFACE BUSY (ZY = 3#).

Disc Memory Interface Busy may be tested true or false by a Test command. A V instruction has no effect.

### 9.7.6 DISC MEMORY UNIT BUSY (ZY = 37).

Disc Memory Unit Busy applies to the selected Disc Memory unit and can be tested true or false by a Test command. A V instruction (reset) can be used to terminate any operation in progress, leaving flags as they are, with generation of a Disc Memory Not Busy Interrupt request. The Disc Memory Interface also goes Not Busy, if it was Busy, and generates an Interface Not Busy Interrupt request.

The Disc Memory Unit Not Busy interrupt occurs at different times with respect to the Interface Not Busy interrupt, depending on the operation.

On a count or obsolete operation, the Interface Not Busy occurs as soon as the Key and Mask data (if any) are transferred to the Disc Memory. The Unit Not Busy occurs later (a maximum of 490 ms times the number of specified data-bands).

The Unit Not Busy interrupt occurs before the Interface interrupt on all Read operations. The difference in time varies between 6 and 25 microseconds.

A Multi-block Write operation will finish with an Interface interrupt between 50 and 400 microseconds before the unit Not Busy interrupt when more than one block is recorded. If no blocks are written on a 4 operation, then the two interrupts are almost simultaneous. If only one block is to be written on a  $\emptyset$  or 4 operation, then the Interface interrupt occurs as soon as this block is transferred; but the unit interrupt occurs after the block is recorded. This may be as much as one revolution later on a  $\emptyset$  operation, or 15 revolutions on a 4 operation.

The Interface interrupt always occurs before the Unit Not Busy interrupt on a Write by Obsolete Flag operation when all the specified blocks are written. How much before depends on how long it takes to find another Obsolete flag after the last block is transferred from the Interface to the Core Buffer in the Disc Memory unit. If there are not enough obsolete locations available, then the two interrupts occur simultaneously and overload goes on.

With an MC or CP Write operation, the Interface interrupt occurs before the Unit Not Busy interrupt by as much as one block time and as little as one character time. An MC character time is 19.6 microseconds and block time is 2.509 ms. The CP character time is 4.2 microseconds and block time is 538 microseconds.

On the Return operations, the Interface Not Busy interrupt occurs between 6 and 25 microseconds after the Unit Not Busy interrupt; with the exception of return disc memory operation where the unit does not go busy at all.

The Not Busy interrupts occur almost simultaneously at the completion of the loading of the Key and/or Mask and Control information on a ^ operation.

Table 9-2 lists the Detail Error toggles that generate Disc Memory Interface Interrupt requests. These toggles are described in detail in paragraph 11.15.

Table 9-2 Disc Memory Interface Interrupt Detail Error Toggles

Detail Error	ZY
Vertical Parity Internal	/1
Horizontal Check Internal	/2
Flag Parity	/3
Fixed Address	/4
M Register	/5
Write	/6
Instruction	/7
Hang Up	/8
Transmission Parity	/9
Out of Temperature	/=
Slow Down	/'
Operand Address	/>
Count	36

9.8 WRITE INHIBIT SWITCHES.

Toggle switches are provided on the Disc Memory Maintenance Panel to inhibit writing in groups of 10 data-bands each. A total of twelve switches control the two twins separately. An attempted write operation in a write-inhibited data-band generates a Write Error Interrupt request.



## SECTION X

## DISC MEMORY OPERATIONS

In the detailed description of the Disc Memory operations, the operations have been grouped under the following six general categories:

10.1 Fixed Address Operations.

10.1.1	Fixed Address Read	(2)
10.1.2	Fixed Address Obsolete	(1)
10.1.3	Fixed Address Read and Obsolete	(3)
10.1.4	Fixed Address Write	(∅)

10.2 Non-Fixed Address Individual Block Operations.

10.2.1	Content Access Read	(=)
10.2.2	Content Access Read and Obsolete	(')
10.2.3	Content Access Obsolete and Count	(9)
10.2.4	Content Access Count	(<)
10.2.5	Compare Flag Read	(])
10.2.6	Compare Flag Read and Obsolete	(7)
10.2.7	Obsolete Flag Write	(8)

10.3 Non-Fixed Address Multi-Block Operations.

10.3.1	Content Access Read Multi-Block	(6)
10.3.2	Content Access Obsolete Multi-Block	(5)
10.3.3	Content Access Read and Obsolete Multi-Block	(7)
10.3.4	Compare Flag Read Multi-Block	(S)
10.3.5	Compare Flag Read and Obsolete Multi-Block	(T)
10.3.6	Obsolete Flag Write Multi-Block	(4)

10. 4 Display Buffer Write Operations.

10. 4. 1 MC Buffer Write (°)

10. 4. 2 CP Buffer Write (#)

10. 5 Prepare Operations.

10. 5. 1 Select Disc Memory Module ([)

10. 5. 2 Load Key, Mask and Control (^)

10. 6 Return Operations.

10. 6. 1 Return Disc Memory Operation (>)

10. 6. 2 Return Key, Mask and Control Information (+)

10. 6. 3 Return Display Buffer Status (;)

The description of each Disc Memory operation lists the fields and bit options of the Disc Memory Operation word, and the M2 and M3 addresses, with their functions or applications, if any.

The Disc Memory instruction format is presented in paragraph 9. 2. As the Disc Memory Interface instruction word does not vary in function, it is not included in the following discussion.

10. 1 FIXED ADDRESS OPERATIONS.

Access is made to the starting block by specifying the twin (or Twin mode), a data-band, and a block number. The function operates on that block and on successive blocks, and can continue without interruption from one data-band to the next. Limits are specified on the number of blocks and data-bands involved. On read functions, the control information specifies the character positions within the block that actually are transferred. Bit Z6 does not apply.

If the specified number of blocks cannot be transferred before the end of the specified number of data-bands, Overload status is set.

At the end of the operation, the TA and BBBB registers will contain the address of the last block on which the operation was performed.

10. 1. 1 FIXED ADDRESS READ.

2                    Operation Code

06                   Fixed Address Transfer Option

Z4Z5	Twin Select
Z6	Not Functional
TA	Starting Data-band and Number of Data-bands
NNNN	Number of Blocks
M2M2M2M2	Starting Block Number
M3M3M3M3	Control Information Location
Mnemonic	PICK

Read NNNN consecutive blocks, starting at the specified fixed address. If bit 06 is a one, each block is preceded by its fixed address word.

#### 10.1.2 FIXED ADDRESS OBSOLETE.

1	Operation Code
06	Not Functional
Z4Z5	Twin Select
Z6	Not Functional
TA	Starting Data-band and number of Data-bands
NNNN	Number of Blocks
M2M2M2M2	Starting Block Number
M3M3M3M3	Mask and Control Information Location
Mnemonic	OBS

Flag as obsolete NNNN consecutive blocks, starting with the specified fixed address.

An effective M3 address will load the Mask and Control registers without affecting the operation.

#### 10.1.3 FIXED ADDRESS READ AND OBSOLETE.

3	Operation Code
06	Fixed Address Transfer Option
Z4Z5	Twin Select
Z6	Not Functional
TA	Starting Data-band and number of Data-bands
NNNN	Number of Blocks
M2M2M2M2	Starting Block Number
M3M3M3M3	Control Information Location
Mnemonic	PURGE

Read and flag as obsolete NNNN consecutive blocks, starting with the specified fixed address. If bit 06 is true, each block is preceded by its fixed address word.



Section X  
Paragraphs 10. 1. 4 to 10. 2. 1

10. 1. 4 FIXED ADDRESS WRITE.

Ø	Operation Code
06	Not Functional
Z4Z5	Twin Select
Z6	Not Functional
TA	Starting Data-band and Number of Data-bands
NNNN	Number of Blocks
M2M2M2M2	Starting Block Number
M3M3M3M3	Mask-Control Information Location
Mnemonic	WRITE

Write, and flag as Non-Obsolete, NNNN consecutive blocks, starting with the specified fixed address.

10. 2 NON-FIXED ADDRESS INDIVIDUAL BLOCK OPERATIONS.

Access is made by content, compare flag, or obsolete flag where each block or block location must qualify. The twin (or Twin mode) and starting data-band are specified. Limits are specified on the number of data-bands considered and the number of blocks transferred.

10. 2. 1 CONTENT ACCESS READ.

=	Operation Code
06	Fixed Address Transfer Option
Z4Z5	Twin Select
Z6	Ignore Obsolete Flags Option
TA	Starting Data-band and Number of Data-bands
NNNN	Number of Blocks Read
M2M2M2M2	Key Information Location
M3M3M3M3	Mask-Control Information Location
Mnemonic	PKBK

Read NNNN blocks that all qualify by content, starting in the specified data-band. If more than NNNN blocks qualify, the Disc Memory will continue searching and set Compare Flags for remaining blocks in the data-band in which reading stopped. The Disc Memory will remain Busy during and after the search. While the Disc Memory is in this condition, another Content Access Read operation must be instituted; any other operation results in an Instruction Error Interrupt request.

If bit 06 is a 1, each block transferred will be preceded by its fixed address word.

### 10.2.2 CONTENT ACCESS READ AND OBSOLETE.

'	Operation Code
06	Fixed Address Transfer Option
Z4Z5	Twin Select
Z6	Ignore Obsolete Flags Option
TA	Starting Data-band and Number of Data-bands
NNNN	Number of Blocks Read
M2M2M2M2	Key Information Location
M3M3M3M3	Mask-Control Information Location
Mnemonic	PGBK

Read and flag as obsolete NNNN blocks that all qualify by content, starting in the specified data-band. If more than NNNN blocks qualify, the Disc Memory will continue searching and set Compare Flags for remaining blocks in the data-band in which reading stopped. The Disc Memory will remain Busy during and after the search. While the Disc Memory is in this condition, another Content Access Read and Obsolete operation must be instituted; any other operation results in an Instruction Error Interrupt request.

If bit 06 is a 1, each block transferred will be preceded by its fixed address word.

### 10.2.3 CONTENT ACCESS OBSOLETE AND COUNT.

9	Operation Code
06	Not Functional
Z4Z5	Twin Select
Z6	Ignore Obsolete Flags Option
TA	Starting Data-band and Number of Data-bands
NNNN	Must be 0000
M2M2M2M2	Key Information Location
M3M3M3M3	Mask-Control Information Location
Mnemonic	OBBK

Flag as obsolete and count all blocks that qualify by content in the specified data-bands. The Disc Memory remains Busy during the counting, and the final count appears in the NNNN register. Overload status is set if the count goes past 9999.

Section X  
Paragraphs 10. 2. 4 to 10. 2. 5

10. 2. 4 CONTENT ACCESS COUNT.

<	Operation Code
06	Not Functional
Z4Z5	Twin Select
Z6	Ignore Obsolete Flags Option
TA	Starting Data-band and Number of Data-bands
NNNN	Must be 0000
M2M2M2M2	Key Information Location
M3M3M3M3	Mask-Control Information Location
Mnemonic	FLAG

Count the number of blocks that all qualify by content in the specified data-bands. The Disc Memory remains Busy during the counting, and the final count appears in the NNNN register. Overload status is set if the count goes past 9999.

Compare Flags are left on the qualifying blocks in the last data-band only. These Compare Flags are not discarded at the beginning of an operation unless it is a Compare Flag operation (<, =, or ').

10. 2. 5 COMPARE FLAG READ.

]	Operation Code
06	Fixed Address Transfer Option
Z4Z5	Twin Select
Z6	Ignore Obsolete Flag Option
TA	Data-Band Select (only one data-band is effective)
NNNN	Number of Blocks
M2M2M2M2	Key Information Location
M3M3M3M3	Control Information Location
Mnemonic	PKBF

Read NNNN blocks that have Compare Flags set from the specified data-band. Compare Flags may be set for only one data-band at a time, so even if more than one is specified, only one will supply blocks. Compare Flags are removed as blocks are read.

If bit 06 is a 1, each block transferred will be preceded by its fixed address word.

### 10. 2. 6 COMPARE FLAG READ AND OBSOLETE.

7	Operation Code
06	Fixed Address Transfer Option
Z4Z5	Twin Select
Z6	Ignore Obsolete Flag Option
TA	Data-band Select (only one data-band is effective)
NNNN	Number of Blocks
M2M2M2M2	Key Information location
M3M3M3M3	Control Information Location
Mnemonic	PGBF

Read and flag as obsolete NNNN blocks that have Compare Flags set from the specified data-band. Compare Flags are held for only one data-band at a time, so that even if more than one is specified, only one will supply blocks. Compare Flags are removed as blocks are read.

If bit 06 is 1, each block transferred will be preceded by its fixed address word.

An effective M2 address will load the Key register without affecting the operation.

### 10. 2. 7 OBSOLETE FLAG WRITE.

8	Operation Code
06	Not Functional
Z4Z5	Twin Select
Z6	Not Functional
TA	Data-band Select and Number of Data-bands
NNNN	Number of Blocks
M2M2M2M2	Key Information Location
M3M3M3M3	Mask-Control Information Location
Mnemonic	WROBS

Write and flag as non-obsolete NNNN blocks in locations flagged as obsolete in the specified data-bands. One band will be filled before going on to the next.

Effective M2 and M3 addresses will load the respective registers, without affecting the operation.

### 10.3 NON-FIXED ADDRESS MULTI-BLOCK OPERATIONS.

The multi-block operations are based on locating a block that qualifies for the access criterion, and then operating on it and consecutively following blocks treated as a single record. The number of blocks operated on is specified by NNNN, so this is, in effect, the length of the record.

The Ignore Obsolete Flag option (bit Z6) applies only to the qualifying block. Similarly, the obsolete function leaves an Obsolete Flag only for the qualifying block with operation code 7.

#### 10.3.1 CONTENT ACCESS READ MULTI-BLOCK.

6	Operation Code
06	Fixed Address Transfer Option
Z4Z5	Twin Select
Z6	Ignore Obsolete Flags Option
TA	Starting Data-band and Number of Data-bands
NNNN	Number of Consecutive Blocks
M2M2M2M2	Key Information Location
M3M3M3M3	Mask-Control Information Location
Mnemonic	PKRK

Read NNNN consecutive blocks where the first qualifies by content. If bit 06 is a 1, each block will be preceded by its fixed address word. The Ignore Obsolete Flag option applies to the qualifying block only.

#### 10.3.2 CONTENT ACCESS OBSOLETE MULTI-BLOCK.

5	Operation Code
06	Not Functional
Z4Z5	Twin Select
Z6	Ignore Obsolete Flags Option
TA	Starting Data-band and Number of Data-bands
NNNN	Number of Consecutive Blocks
M2M2M2M2	Key Information Location
M3M3M3M3	Mask-Control Information Location
Mnemonic	OBRK

Flag as obsolete NNNN consecutive blocks where the first qualifies by content.

### 10. 3. 3 CONTENT ACCESS READ AND OBSOLETE MULTI-BLOCK.

7	Operation Code
06	Fixed Address Transfer Option
Z4Z5	Twin Select
Z6	Ignore Obsolete Flags Option
TA	Starting Data-band and Number of Data-bands
NNNN	Number of Consecutive Blocks
M2M2M2M2	Key Information Location
M3M3M3M3	Mask-Control Information Location
Mnemonic	PGRK

Read NNNN consecutive blocks and flag the first block as obsolete, where the first block qualifies by content.

### 10. 3. 4 COMPARE FLAG READ MULTI-BLOCK.

S	Operation Code
06	Fixed Address Transfer Option
Z4Z5	Twin Select
Z6	Ignore Obsolete Flags Option
TA	Data-band Select (only one data-band is effective)
NNNN	Number of Consecutive Blocks
M2M2M2M2	Key Information Location
M3M3M3M3	Control Information Location
Mnemonic	PKRF

Read NNNN consecutive blocks where the first qualifies by having a Compare Flag set. Compare Flags are held for only one data-band at a time so that even if more than one is specified, only one will supply the qualifying block. The Compare Flag is removed from all blocks read.

A new key may be loaded by an effective M2 address without affecting the operation.

### 10. 3. 5 COMPARE FLAG READ AND OBSOLETE MULTI-BLOCK.

T	Operation Code
06	Fixed Address Option
Z4Z5	Twin Select
Z6	Ignore Obsolete Flags Option
TA	Data-band Select (only one data-band is effective)

Section X  
Paragraphs 10.3.6 to 10.4.1

NNNN	Number of Consecutive Blocks
M2M2M2M2	Key Information Location
M3M3M3M3	Control Information Location
Mnemonic	PGRF

Read NNNN consecutive blocks where the first qualifies by having a Compare Flag set. The Compare Flag is removed from all blocks read and the first block is flagged obsolete.

10.3.6 OBSOLETE FLAG WRITE MULTI-BLOCK.

4	Operation Code
06	Not Functional
Z4Z5	Twin Select
Z6	Not Functional
TA	Starting Data-band and Number of Data-bands
NNNN	Number of Consecutive Blocks
M2M2M2M2	Key Information Location
M3M3M3M3	Mask-Control Information Location
Mnemonic	WROB

Write and flag as non-obsolete NNNN consecutive blocks where the first block is written in a location flagged as obsolete.

10.4 DISPLAY BUFFER WRITE OPERATIONS.

The Display Buffer Write operations are the means for loading messages enroute to the Display Consoles.

10.4.1 MC BUFFER WRITE.

°	Operation Code
06	Not Functional
Z4Z5	Not Functional
Z6	Not Functional
TA	MC Select
NNNN	Number of Blocks
M2M2M2M2	Not Functional
M3M3M3M3	Not Functional
Mnemonic	DISOUT

Write NNNN blocks into the MC Buffer, addressed to the selected MC Display.

The MC select addresses are given in paragraph 10. 6. 3.

#### 10. 4. 2 CP BUFFER WRITE.

#	Operation Code
06	Not Functional
Z4Z5	Not Functional
Z6	Not Functional
TA	CP Select
NNNN	Number of Blocks (limited 1 to 8)
M2M2M2M2	Not Functional
M3M3M3M3	Not Functional
Mnemonic	CPOUT

Write NNNN blocks into the sector of the CP Buffer for the selected CP.

The CP select addresses are given in paragraph 10. 6. 3.

#### 10. 5 PREPARE OPERATIONS.

The Prepare operations consist of a Select operation and a Load Key, Mask and Control operation.

The Select operation is effectively a no operation for the Disc Memory, but is useful, because the Status and Error signals reach the Disc Memory Interface only from the last Disc Memory selected. The selection is by the Y character of the Disc Memory Interface instruction word.

#### 10. 5. 1 SELECT DISC MEMORY MODULE.

[	Operation Code
06	Not Functional
Z4Z5	Not Functional
Z6	Not Functional
TA	Not Functional
NNNN	Not Functional
M2M2M2M2	Not Functional
M3M3M3M3	Not Functional
Mnemonic	FNOP



Section X  
Paragraphs 10. 5. 2 to 10. 6. 1

Select the Disc Memory module specified by the Y character of the Disc Memory Interface instruction word. Status and error signals will appear in the Disc Memory Interface from the selected Disc Memory.

10. 5. 2 LOAD KEY, MASK AND CONTROL.

^	Operation Code
06	Must be a one
Z4Z5	Not Functional
Z6	Not Functional
TA	Not Functional
NNNN	Not Functional
M2M2M2M2	Key Information Location
M3M3M3M3	Mask-Control Information Location
Mnemonic	LDKM

Load the Key, Mask and Control registers with the specified information.

10. 6 RETURN OPERATIONS.

The three Return operations are: Returning the Disc Memory Operation; the Key, Mask, and Control; and the Display Buffer Status.

The information sent by the Disc Memory on Return operations is stored in Core Memory, starting at the effective operand address in the Disc Memory Interface instruction word.

10. 6. 1 RETURN DISC MEMORY OPERATION.

>	Operation Code
06	Must be a zero
Z4Z5	Not Functional
Z6	Not Functional
TA	Not Functional
NNNN	Not Functional
M2M2M2M2	Not Functional
M3M3M3M3	Not Functional
Mnemonic	RCLI

Read into the specified location in Core Memory two words which are the content of the Disc Memory Operation register. The format is as follows:

1st word	OZTABBBB
2nd word	E000NNNN

O, Z	as specified by the previous Disc Memory operation.
TA	Indicates the last data-band operated on and the remaining number of specified data-bands.
BBBB	Indicates the block number of the last block operated on.
E	Flag Error Character
000	Fixed
NNNN	Block count.

The Disc Memory does not go Busy for this operation.

#### 10. 6. 2 RETURN KEY, MASK AND CONTROL INFORMATION.

+	Operation Code
06	Not Functional
Z4Z5	Not Functional
Z6	Not Functional
TA	Not Functional
NNNN	Not Functional
M2M2M2M2	Key Information Location
M3M3M3M3	Mask-Control Information Location
Mnemonic	RCLC

Read the content of the Key, Mask and Control registers. Core Memory will be loaded with 16 words of Key, 16 words of Mask, and 16 words of Control information. Although the Control register has only four-bit characters, the content will be transferred as Central Processor six-bit characters, with zeros in the zone bits.

If the M2 and M3 addresses are effective, Key and Mask-Control information will be loaded into the respective registers, and then will be returned to the Central Processor.

#### 10. 6. 3 RETURN DISPLAY BUFFER STATUS.

;	Operation Code
06	Not Functional
Z4Z5	Not Functional
Z6	Not Functional

Section X  
Paragraph 10.6.3 (Cont.)

TA	Not Functional
NNNN	Not Functional
M2M2M2M2	Not Functional
M3M3M3M3	Not Functional
Mnemonic	RECAL

Read one word which contains the Display Buffer status. The bits have the following code:

MC STATUS

<u>Bit 1</u>	<u>Bit 2</u>	
0	0	Transmission Terminated, Ready
0	1	Transmission Terminated by Error
1	0	Transmission in Progress
1	1	Display not Available

CP STATUS

<u>Bit 3</u>	
0	Transmission in Progress, or CP not Available
1	Transmission Terminated, Ready

COMPUTER BIT	STATUS BIT	C7	C6	C5	C4	C3	C2	C1	C0	COMPUTER CHARACTER
6	CP 3		7(	7W	7S	7J	7=	7C	7°	
5	MC { 2 1		7(	7W	7S	7J	7=	76 OR 77	72 OR 73	
4										
3	CP 3	6C	7Y	7U	7+	7>	78	7;	7#	
2	MC { 2 1	6C	7Y	7U	7+	7>	78	74 OR 75	70 OR 71	
1										

**TA ADDRESS ASSIGNMENT FOR DISPLAY CONSOLES :**

**PAIRED MC (LP) CHANNELS:**

70 AND 71  
72 AND 73  
74 AND 75  
76 AND 77

**CP ONLY CHANNELS:**

7; , 7# , 7C AND 7

**ALL OTHER ADDRESSES REFER TO COMBINED  
MC AND CP CHANNELS:**

Figure 10-1. Display Buffer Status Word



## SECTION XI

### PROGRAM INTERRUPTS

The Data Processing System is responsive to Real-Time operation with communication channels and operator controlled displays through a set of Program interrupts. Since the Central Processor has interfaces, such as the Disc Memory Interface, that are capable of carrying out data transfers independently of program execution, Program interrupts are provided to signal that further action should be taken. This is extended to include buffered data transfers, such as Disc Memory to Display Console, signaling the depletion of data in the buffer storage.

A major portion of the interrupts are responsive to errors. Parity checks in data transfers are noted as to location. Other malfunctions which only threaten to interfere with operation, such as overheating, are noted. The emphasis is on error location and correction.

Paragraphs 11.1 through 11.6 describe the structure and use of Program Interrupts, while paragraphs 11.7 through 11.17 describe Program Interrupt request conditions.

#### 11.1 INTERRUPT HIERARCHY.

The interrupts are arranged in a hierarchy with classes, subclasses, and detail interrupt signals. Detail interrupt signals are derived from toggles that are set on by the interrupt requests, which are momentary or are treated as being momentary. The detail interrupt toggles may be tested ON or OFF and set ON or OFF by program.

When detail interrupt signals appear, the appropriate class and subclass signals also appear. These signals are program testable as an aid to the program in locating the detail interrupt. When addressed with a reset, the class and subclass lines have the effect of resetting all their constituent detail interrupt toggles.

At the top of the hierarchy is the Master interrupt. Whenever the Master interrupt line has a signal, the Central Processor responds by entering Interrupt mode.

## 11.2 IGNORE INTERRUPT TOGGLES.

Program set, reset, and testable Ignore Interrupt toggles are provided for the Master interrupt, class, and subclass lines. When an Ignore Interrupt toggle is set ON, all constituent interrupt signals below its level in the hierarchy, are ignored to the extent that they cannot affect a higher level. However, the interrupt signals are not lost, as they can always be tested both at detail and class levels.

Figure 11.1 illustrates the interrupt hierarchy. Detail interrupts are represented as boxes with an inset in the lower left corner. The inset indicates a signal toggle with its ZY address for reset and testing. Interrupt classes and subclasses are represented as boxes with an inset in the lower right corner. The inset indicates the Ignore toggle and its ZY address. The Test or Detail Reset address is shown in the lower left corner.

Detail interrupt signals travel up the hierarchy, giving an indication at each class level until the Master interrupt is reached or until the signal is blocked by an Ignore toggle. Class reset signals travel only one step down the hierarchy; that is, only apply at the first class level above the detail interrupt toggles.

## 11.3 INTERRUPT OPERATION.

When a Program interrupt occurs in response to a Master Interrupt signal, it suspends computation when an instruction is completely executed and another is about to be read. At this time, the following actions occur:

- a. The Master Interrupt Ignore toggle is set ON to prevent response to any further interrupts.
- b. The R register content is stored in memory location 00000.
- c. The address of the unexecuted instruction is transferred from the Instruction Address counter into memory location 00014 (most significant half, not Index register 14).
- d. Control is transferred to the address specified in one of five memory locations, the choice being on the basis of the class of interrupt.

An exception occurs in the case of a Computer Error interrupt. For Program subclass interrupts, instruction execution is suspended immediately (within one clock time), as, presumably, the current instruction is in error and there is no reason to execute it. For other subclass interrupts, instruction execution is suspended after the error is detected at a suitable point for recovery but not

generally at the end of instruction execution. Generally, the return address will be that of the partially completed instruction.

A detected error in the transfer of a Disc Memory instruction of three words to the Disc Memory Interface is regarded as a Computer error, and not as a Disc Memory Interface error.

If a Compare instruction is being executed in Search mode and an interrupt occurs, the search is stopped incomplete. In this case, care must be exercised in the use of the return address. The Search Mode toggle is not affected by Program interrupt and may be tested by the interrupt program.

#### 11.4 INTERRUPT PROGRAM SELECTION.

According to the class of the interrupt, an Interrupt Program Start Location is obtained from one of the six memory locations, 00008 through 00013. The start location is a four character address, MMMM, contained in the four high order character positions. The low order characters form some of the Index registers. The Program Start Locations and related Classes of Interrupt are as follows:

<u>Program Start Location in</u>	<u>Class of Interrupt</u>
00008	Uni-Record Interface
00009	Disc Memory Interface
00010	Error
00011	I/O Interface
00012	Real-Time
00013	Other Processor

If interrupt signals in two or more classes occur simultaneously and cause a Program interrupt, an interrupt class priority determines the choice of the program start location. The interrupt class priority is as follows:

- a. Error (highest)
- b. I/O Interface
- c. Disc Memory Interface
- d. Real-Time
- e. Uni-Record Interface
- f. Other Processor (lowest)



### 11.5 INTERRUPT PROGRAM EXIT.

The Interrupt program, preparatory to exit to the return address, should restore all registers and toggles to their entry status. The procedure then is to reset the Master Interrupt Ignore and jump back to the normal program.

If another Non-Ignored Interrupt signal is present, it will not effect an immediate Program interrupt upon Master Interrupt Ignore Reset. This is because a Program interrupt, other than Computer Error, is prevented from taking effect immediately following a Set-Reset Toggle (V) instruction. Some other instruction, presumably the return jump to the normal program, must be executed before the second interrupt is effective. This prevents loss of the normal program return address, which would happen were an interrupt program instruction location substituted as the return address.

A Program Error subclass signal will cause an immediate halt during a V instruction that resets the Master Interrupt Ignore toggle. Other Computer Error subclass signals will cause interrupt immediately after such a V instruction.

### 11.6 FLAGGED RETURN ADDRESSES.

It is possible to make a return to the normal program from an interrupt program to a desired point where some special condition exists. These points are instructions flagged with a 1 bit in position X6. As the flagged instructions are processed preparatory to execution, their locations are stored in special memory locations, namely, the four most significant characters of locations 00001 through 00004. The type of instruction determines which locations, as follows:

<u>Type of Instruction</u>	<u>Flagged Return Address in</u>
Central Processor	00001
Disc Memory Interface	00002
Uni-Record Interface	00003
I/O Interface	00004

### 11.7 INTERRUPT CLASS: I/O INTERFACE (ZY = 42).

Ignore toggle ZY = 22

#### 11.7.1 DETAIL: INTERFACE NOT BUSY (ZY = 39).

Indicates termination of an input or output data transfer operation through the I/O Interface. The Interface is in standby condition and the Central Processor can initiate an input or output operation.

11.7.2 DETAIL: L-119 MESSAGE AVAILABLE (ZY = 2,).

Indicates that a Receiving L-119 which has been selected by the Scanner, is waiting with a block of an input message for the Central Processor to execute an Input instruction. This appears only when the I/O Interface is Not Busy.

11.7.3 DETAIL: DISPLAY CONSOLE MESSAGE AVAILABLE (ZY = 2 #).

Ignore toggle ZY = 70

Indicates that a Display Console which has been selected by the Scanner, is waiting with a message for the Central Processor to execute an Input instruction. This appears only when the I/O Interface is Not Busy.

11.7.4 DETAIL: SENDING L-119 AVAILABLE (ZY = 2 ( ).

Indicates that a block of an output message has been transmitted and acknowledged, and the sending L-119 is waiting for the Central Processor to execute an Output instruction. This appears whether I/O Interface is Busy or Not Busy.

11.8 INTERRUPT CLASS: REAL-TIME (ZY = 43).

Ignore toggle ZY = 23

11.8.1 DETAIL: REAL-TIME CLOCK (ZY = 3 ( ).

Indicates that the 24 hour Real-Time Clock has passed through full scale and reset to zero.

11.8.2 DETAIL: INTERVAL TIMER (ZY = 3,).

Indicates that the time interval set on the Timer has elapsed and the Timer has reached zero.

11.9 INTERRUPT CLASS (AND DETAIL): OTHER PROCESSOR (ZY = 44).

Ignore toggle ZY = 24

Indicates that the other Central Processor in a duplex system has by program requested setting the Detail toggle ON. The other Central Processor does this by a Set Toggle instruction with ZY = 3'.

11.10 INTERRUPT CLASS: DISC MEMORY INTERFACE (ZY = 3[ ).

Ignore toggle ZY = 25

11.10.1 DETAIL: INTERFACE NOT BUSY (ZY = 2Y).

Indicates termination of a data transfer operation through the Disc Memory Interface. The Interface is in Standby and the Central Processor can initiate a Disc Memory operation.

11.10.2 DETAIL: DISC MEMORY UNIT NOT BUSY (ZY = 2Z).

Indicates termination of a Disc Memory operation. During a Read operation that does not involve content search, the Disc Memory Not Busy interrupt will be concurrent with the Interface Not Busy interrupt after the last block is transferred into Core Memory. Otherwise, the interrupts are generally not concurrent, paragraph 9.7.

11.10.3 DETAIL: DISPLAY (ZY = 35).

Indicates termination of a message transmission from the buffer storage to either the Multi-color Display or the Console Printer in any Display Console. It will appear only when the Disc Memory is or goes Not Busy.

If a Disc Memory instruction attempts to load a message into buffer storage for a Busy or unavailable display, the message transfer from Core Memory will not take place, but a Display Interrupt will occur immediately.

11.11 INTERRUPT CLASS: UNI-RECORD INTERFACE (ZY = 3W).

Ignore toggle ZY = 3X

11.11.1 DETAIL: INTERFACE NOT BUSY (ZY = 3Y).

Indicates termination of a data transfer operation through the Uni-Record Interface.

11.11.2 DETAIL: OPERATOR (ZY = 3Z).

Indicates that the Operator Interrupt button on the Control Console has been depressed.

11.12 INTERRUPT CLASS: ERROR (ZY = 41) - (Computer, Data Transfer).

Ignore toggle ZY = 21

Subclass: Computer Error (ZY = U $\emptyset$ )

Ignore toggle ZY = 2U

Subclass: Data Transfer (ZY = V $\emptyset$ )

Ignore toggle ZY = 2V

11.12.1 DETAIL: M REGISTER PARITY ERROR: (ZY = V1).

Indicates that a character parity error has been detected in the M register in a data transfer with the Central Processor, excluding any interface data transfer. An M Register Parity Error is considered in the Computer Error Subclass if it occurs in a memory cycle requested by the Program Control section. (See M Register Parity Errors under the subclass Interface Error.)

11.12.2 DETAIL: R REGISTER (ZY = V2).

Indicates that a character parity error exists in data being transferred into the R register.

11.12.3 DETAIL: A REGISTER (ZY = V3).

Indicates that a character parity error exists in data being transferred into the A register.

11.12.4 DETAIL: Q REGISTER (ZY = V4).

Indicates that a character parity error exists in data being transferred into the Q register.

11.12.5 DETAIL: ARITHMETIC (ZY = V5).

Indicates that a character parity error exists in data being transferred through the Adder, or that the Adder has detected an internal malfunction.

11.12.6 DETAIL: R REPEATER (ZY = V6).

Indicates that a character parity error exists in data being transferred through the R Repeater.

11.13 INTERRUPT CLASS: ERROR (ZY = 41) - (Computer, Arithmetic).

Ignore toggle ZY = 21

Subclass: Computer Error (ZY = U0)

Ignore toggle ZY = 2U

Subclass: Arithmetic (ZY = W0)

Ignore toggle ZY = 2W

Section XI  
Paragraphs 11.13.1 to 11.14.1

11.13.1 DETAIL: OVERFLOW (ZY = W1).

Indicates an overflow of the result for an A, S, + or - instruction in fixed point beyond the most significant character position in the result register, or that for a Divide Long operation the magnitude of the dividend was not smaller than the magnitude of the divisor.

11.13.2 DETAIL: F. P. OVERFLOW (ZY = W2).

Indicates that an exponent in the processing of a Floating-Point command has exceeded the range -99 to +99. This may occur in the result or in the intermediate processing, such as the pre-normalization of an operand.

11.13.3 DETAIL: NUMERICAL (ZY = W3).

Indicates that an arithmetic process has been attempted on a character with a non-decimal value in the numeric bits.

11.13.4 DETAIL: CONVERT OVERFLOW (ZY = W4).

Indicates that a Convert (P) instruction with Y6ON has been attempted on a decimal value greater than 63.

11.14 INTERRUPT CLASS: ERROR (ZY = 41) - (Computer, Program).

Ignore toggle ZY = 21

Subclass: Computer Error (ZY = U0)

Ignore toggle ZY = 2U)

Subclass: Program (ZY = X0)

Ignore toggle ZY = 2X)

11.14.1 DETAIL: INSTRUCTION (ZY = X1).

This detail indicates:

- a. An unassigned command character has been read in an instruction.
- b. A parity error has been detected in an instruction as it came from Core Memory.
- c. An unassigned Core Memory location has been encountered in the fetching of an instruction.
- d. A MASM instruction in MASM mode has a 1 in bits 15 and 16, paragraph 3.5.

e. No Comparison toggles are ON when a Field Compare instruction is encountered, or, in Search Mode only, an Alphabetic Compare or Numeric Compare instruction is encountered, paragraph 4.3.

11.14.2 DETAIL: ILLEGAL ADDRESS (ZY = X2).

Indicates that an unassigned Core Memory location has been encountered in the fetching, processing, or execution of an instruction.

11.14.3 DETAIL: END OF MEMORY (ZY = X3).

Indicates that an instruction with a multiple word operand has counted past location 63,999.

11.14.4 DETAIL: ILLEGAL REGISTER ADDRESS (ZY = X4).

Indicates that an unassigned register address has been encountered in a Save, Load, or Copy instruction.

11.15 INTERRUPT CLASS: ERROR (ZY = 41) - (Interface, Disc Memory).

Ignore toggle ZY = 21

Subclass: Interface (ZY = + $\emptyset$ )

Ignore toggle ZY = 2+

Subclass: Disc Memory Interface (ZY = / $\emptyset$ )

Ignore toggle ZY = 2/

11.15.1 DETAIL: VERTICAL PARITY INTERNAL (ZY = /1).

Indicates a parity error has been detected in a transfer of data, Key, Mask or Control characters out of the Disc Memory Core Buffer. This cannot occur on a >, ;, [, or ^ operation. If it occurs on a count operation, then the Key, Mask or Control data has gone bad, and the entire operation (including reloading Key, Mask and Control) must be repeated. The error occurs as a character read from the disc is being compared with the Key through the Mask and Control. The B register will contain the fixed address of the block being read.

The above condition also applies to a Read operation, and in addition, the error could occur as the data character is being transferred to the Central Processor. In this case, a Transmission error signal will also be generated. The B register will contain the fixed address of either the block being transferred to the Central Processor or the next block read from the disc, depending on when the error

Section XI

Paragraphs 11.15.2 to 11.15.3

occurred. The N register will be one less than the number of blocks not transferred correctly. The Disc Memory instruction should be re-executed, but it is not necessary to reload Key, Mask and Control.

The only cause for a Vertical Parity Internal error on a Write or Load CP or MC operation is the writing of character with incorrect Parity from the Core Buffer to the disc. The B register will contain the fixed address of the bad block, so only the remainder of the operation (including the bad block) need be repeated in the case of a Write operation. The N register will be the number of blocks that have not been written correctly. If the error occurs on a + operation, the Key, Mask, or Control characters have bad parity. Data is not checked.

11.15.2 DETAIL: HORIZONTAL CHECK INTERNAL (ZY = /2).

Indicates an error in the transfer of data from the disc on a read or count operation. This cannot occur on a  $\emptyset$ , 4, 8, >, ^, +, #, °, ;, [ operation.

The error occurs after the last character of a block is stored in the Core Buffer and the B register will always contain the address of the block read incorrectly. The N register will have been changed due to reading the bad block and transferring it to the Central Processor. The operation can be repeated or the block in error can be re-read alone.

11.15.3 DETAIL: FLAG PARITY (ZY = /3).

Indicates that an error has been detected in the reading or writing on a flag track on the disc.

Five causes of flag errors are checked. They can be distinguished by an examination of the most significant character (C7) of the second word returned to the Central Processor with a > operation. The bits of the C7 character have the following meanings:

Bit 1 ON	Missed Flag Error
Bit 2 ON	Compare Flag Error
Bit 3 ON	Obsolete Flag Error
Bit 4 ON	Short Delay Flag Error

If a flag error exists without one of the above bits ON, then the cause was a write error on one of the selected flags.

a. MISSED FLAG ERROR (BIT ONE). The Missed Flag error can only occur for a read or count operation, 1, 2, 3, 5, 6, 7, 9, =, ', <, ],  $\neg$ , S, or T. The B register will contain the fixed address of the block with the bad flag, and the N register will indicate the number of blocks read or remaining to be read from the disc (depending on the operation). The block with the bad flag is read if it meets the search criteria specified by the operation.

The error is always corrected at the time it is detected, so it cannot be detected a second time unless it is re-created. Thus, the normal procedure is to repeat the entire read operation when consecutive blocks are not being read.

b. COMPARE FLAG ERROR (BIT TWO). The Compare Flag error can only occur for a read or count operation, 1, 2, 3, 5, 6, 7, 9, =, ', <, ],  $\neg$ , S, or T. When the Disc Memory goes Not Busy, the B register will contain the fixed address of the block with the bad flag. The N register will indicate the number of blocks read, either directly or indirectly. Once the error occurs, it will remain until the Compare flag is removed or written with another operation (=, ', <, ],  $\neg$ , S, or T with Ignore Disc Memory Error Interrupt ON.

c. OBSOLETE FLAG ERROR (BIT THREE). The Obsolete Flag error can occur for any operation using the stored data section of the disc. When the error does occur, the B register will contain the fixed address and the N register can be used to indicate the number of blocks operated on. The error can only be removed by re-writing or erasing the Obsolete flag with Ignore Interrupt ON. Only that portion of the operation which is incomplete need be repeated. If the error occurs on a Read operation, the block with the bad flag may or may not be read depending on the search criteria for the operation. With a write by obsolete (8) or write consecutive by obsolete on the first (4), the block with the bad flag will not be written on and its fixed address will appear in the B register. A write by fixed address ( $\emptyset$ ) will not correct the flag error (unless Ignore Disc Memory Error Interrupt is ON); but the block with the bad Obsolete flag will be written upon and this fixed address will be in the Disc Memory Operation register.

d. SHORT DELAY FLAG ERROR (BIT FOUR). This error indicates a malfunction in the read circuits of a short delay line of the flag tracks. When this occurs, the probabilities are that no flags will be damaged, but if any are, they will be the flags for the fixed address before the one in the B register. In order to prevent the loss of more flags and still try the short delay line a second time, a Read operation



Section XI  
Paragraphs 11.15.4 to 11.15.6

that does not alter Obsolete or Compare flags should be tried. If the error persists, repairs must be made before the damaged flags (or any flags) can be corrected. The fixed address of possible incorrect flags must be preserved since the Obsolete flags for the fixed address of all data-bands below number 30 (if the selected data-band was below 30) may be lost; or above 30 if the selected data-band was in that group.

The error can occur for any operation using the stored data section of the disc; but if no flags are being changed by this operation, then the only flag that can be damaged is the missed flag.

e. FLAG WRITE ERROR. The Write error can occur for any operation using the stored data section of the Disc Memory. In most cases no permanent change to a flag will result when this error occurs. However, if a flag is damaged, it will be a Missed flag, Compare flag or Obsolete flag for the block with an address one less than the B register content. When the Write error occurs, it may damage all the Obsolete flags stored in the short delay line mentioned above. Thus, if a data-band under 30 was selected, all the blocks with the proper fixed address in the data-bands under 30 must be checked for errors.

#### 11.15.4 DETAIL: FIXED ADDRESS (ZY = /4).

The fixed Address error indicates a parity error detected in reading the fixed address track. When the error occurs, the address contained in the B register is the address of the last block operated on, or the first block to be operated on in the case of a fixed address operation, and has no relation to the fixed address in error. Since no data has been damaged (the error will occur before data is transferred for that block), the operation need only be repeated for the incomplete portion, or it may be entirely repeated, if possible. If the error persists, it must be repaired, but Disc Memory Interface Error Interrupt Ignore should not be used, especially for a Write by Fixed Address operation.

#### 11.15.5 DETAIL: M REGISTER (ZY = /5).

Indicates that a Character Parity error has been detected in the Central Processor M register during a data transfer to or from the Disc Memory.

#### 11.15.6 DETAIL: WRITE (ZY = /6).

The Write error can occur for a Write operation ( $\emptyset$ , 4, 8, #, °) or an internal writing malfunction in the MC or CP Buffer Store. If an inhibited data-band is selected, the error will occur when the N register is being loaded from the

Central Processor. Otherwise, the error indicates a failure in a write amplifier for either the data or Display Buffer section of the disc. The B register will contain the address of the block where the Write error occurred in the case of the data section of the disc.

If a Write error occurs in the MC or CP Buffer Store section of the disc, the N register will indicate the number of blocks not yet transferred (excluding the one with the error) as was the case with the data section. When the Write error occurs in the Display Interface section, the signal does not appear until the Disc Memory has completed its current operation, which may or may not be a Write Display Buffer operation. If the operation just ended was a Write operation, then the error can be either in a data block or in the MC or CP Buffer Store. The N register, if not zero, indicates that the last data block is in error and the B register will contain its address. Even if the N register is zero the last block should be re-written in order to insure there will not be an error.

#### 11.15.7 DETAIL: INSTRUCTION (ZY = /7).

An instruction error is used to indicate incorrect format of data sent to the Disc Memory. Causes of instruction error are as follows:

- a. Unassigned O or Z character
- b. TA characters of non-permissible value for stored data section of Disc Memory.
- c. The fifth and sixth bits of TA are not a binary one when the operation stops at the end of data-band 59.
- d. Write or Obsolete operation specified to end of zone.
- e. Number of blocks specified as  $\emptyset$  or non-numerically on a data transfer operation, or not specified as  $\emptyset$  on a count operation.
- f. Fixed address specified non-numerically or larger than 1349 on a fixed address operation.
- g. Key or mask specified or attempted to load beyond 128th character or specified non-numerically (the location and number of characters).
- h. More than 8 blocks specified for a CP buffer slot.
- i. Read by content instruction (= or ') not followed by another read by content when Disc Memory is still busy.

Section XI  
Paragraph 11.15.8

- j. Non-operative Disc Memory selected.
- k. Disc Memory address of  $\emptyset$  for a non-select operation ( [ ).
- l. Incorrectly specified bounded search field.

11.15.8 DETAIL: HANG UP (ZY = /8).

Indicates termination of a Disc Memory operation by a timer (set to 30 seconds) because of failure to terminate otherwise, or immediate termination of an operation specified between an existent Disc Memory and a non-existent Auxiliary Disc Memory.

11.15.9 DETAIL: TRANSMISSION PARITY (ZY = /9).

Indicates an error in the transmission of information between Central Processor and Disc Memory or Disc Memory and Auxiliary Disc Memory. Possible causes of this error are:

- a. Transmission of data to the Disc Memory from the Core Memory. If the bad data is to be recorded on the disc, the block being written will be finished before the operation stops. The operation should be repeated starting with the block indicated by the B register. If writing is not taking place but the data is to be written on the disc, then the incompleting portion of the operation is all that need be repeated. This is indicated by the value in the N register. The general rule is to perform the operation again starting with the block indicated by the B and/or N register.
- b. Transmission of data to the Core Memory from the Disc Memory. If the bad data is from the disc, the contents of the B register may indicate the next block read. Therefore, if the operation was not reading consecutive blocks, the entire operation must be repeated.
- c. The address lines from Central Processor to Disc Memory have bad parity. Refer to a and b above for recovery procedure.
- d. The transfer control lines that indicate the prepare data being transferred have bad parity. Refer to a and b above for recovery procedure.
- e. The data read from the disc and transferred from Auxiliary Disc Memory to Disc Memory has incorrect parity. The B register will always indicate the block where the error took place.

f. Flag control data, address selection, data to be written, index track selection data, or data track selection characters have wrong parity when received by the Auxiliary Disc Memory from the Disc Memory. The B register will always contain the fixed address of the block where the error occurs, if the operation on the disc has started.

11.15.10 DETAIL: OUT OF TEMPERATURE (ZY = /=).

Indicates that a temperature sensor detects overheating which is a threat to continued operation.

11.15.11 DETAIL: SLOW DOWN (ZY = /').

Indicates that a Disc Memory rotating element has dropped its speed below a pre-set limit.

11.15.12 DETAIL: OPERAND ADDRESS (ZY = />).

Indicates that an attempt has been made to access an unassigned Core Memory address.

11.15.13 DETAIL: COUNT (ZY = 36).

Indicates an error in the data-band counter or character counter for a block being read or written on the disc. If the error is due to a data-band count, the blocks operated on in the last revolution may be incorrect. A character count error will only damage the current block; but since there is no method of distinguishing between the two, it may be important to reconstruct the operation (s) that took place in the last 70 milliseconds.

11.16 INTERRUPT CLASS: ERROR (ZY = 41) - (Interface, I/O).

Ignore toggle ZY = 21

Subclass: Interface (ZY = +∅)

Ignore toggle ZY = 2+

Subclass: I/O Interface (ZY = S∅)

Ignore toggle ZY = 2S

11.16.1 DETAIL: PARITY (ZY = S1).

Indicates that a character parity error has been detected by an L-119 or a Display Console in a message being transferred through the I/O Interface.

## Section XI

### Paragraphs 11.16.2 to 11.17.2

#### 11.16.2 DETAIL: M REGISTER (ZY = S2).

Indicates that a character parity error has been detected by the Central Processor M register in a word of a message being transferred through the I/O Interface. The erroneous word will load into Core Memory on a Receiving operation, but will not be transmitted to the device on a Sending operation.

#### 11.16.3 DETAIL: HANG UP (ZY = S3).

Indicates that a message transmission through the I/O Interface has not terminated normally or by detected error, but instead has been terminated by a timer. Also, if the Central Processor attempts to initiate an output message transfer to a Display Console which is unavailable (or has indicated that it is waiting with an input message), the message transfer will not start, but a Hang Up Error interrupt will occur.

#### 11.16.4 DETAIL: OPERAND ADDRESS (ZY = S4).

Indicates that the I/O Interface has erroneously generated a non-numeric Core Memory address, has counted into a non-assigned address, or has counted past the end of memory.

#### 11.17 INTERRUPT CLASS: ERROR (ZY = 41) - (Interface, Uni-Record).

Ignore toggle ZY = 21

Subclass: Interface (ZY = + $\emptyset$ )

Ignore toggle ZY = 2+

Subclass: Uni-Record Interface (ZY = T $\emptyset$ )

Ignore toggle ZY = 2T

#### 11.17.1 DETAIL: ON LINE DATA (ZY = T1).

Indicates that a parity error has been detected by the addressed device while the current data transfer was in progress and the Interface was Busy. This also indicates failure of the card punch read check, card reader read check or an unallowed card punch combination in a card being read.

#### 11.17.2 DETAIL: ON LINE EAVESDROP (ZY = T2).

Indicates that the addressed Tape Transport detected a parity error while the Interface was Busy in the read check of data being recorded.

11.17.3 DETAIL: ON LINE MECHANICAL (ZY = T3).

Indicates that a mechanical difficulty has interrupted the current data transfer while the Interface was Busy.

11.17.4 DETAIL: M REGISTER PARITY ERROR (ZY = T4).

Indicates that a character parity error was detected in a word of data being transferred to or from Core Memory through the M register.

11.17.5 DETAIL: HORIZONTAL CHECK (ZY = T5).

Indicates that the addressed Tape Transport or Card Reader-Punch detected a horizontal check error while the Interface was Busy.

11.17.6 DETAIL: INSTRUCTION (ZY = T6).

Indicates that the addressed device is Operative and On-Line but does not recognize the current instruction.

11.17.7 DETAIL: OVERLOAD (ZY = T7).

Indicates that the data handling capacity of the addressed device has been exceeded during a data transfer while the Interface was Busy.

11.17.8 DETAIL: REAL WORLD PARITY (ZY = T8).

Indicates that the error conditions listed as On-Line Data and On-Line Eavesdrop Errors occurred after the Interface went Not Busy, but before the device went Not Busy. The Detail Error toggle will be turned ON by the next instruction addressing the same device.

11.17.9 DETAIL: REAL WORLD MECHANICAL (ZY = T9).

Indicates that error conditions listed as On-Line Mechanical Error occurred after the Interface went Not Busy, but before the device went Not Busy. The Detail Error toggle will be turned ON by the next instruction addressing the same device.

11.17.10 DETAIL: DEVICE UNAVAILABLE (ZY = T>).

Indicates that the addressed device was inoperative or Off-Line, or that a Tape Transport was requested to move past Start or End of tape, or to write with the write inhibit ON.

Section XI  
Paragraph 11.17.11

11.17.11 DETAIL: OPERAND ADDRESS (ZY = T<).

Indicates that an attempt was made to access an unassigned Core Memory address.

Note: There are two unassigned Uni-Record Interface Detail Error toggles with (ZY = T=) and (ZY = T').

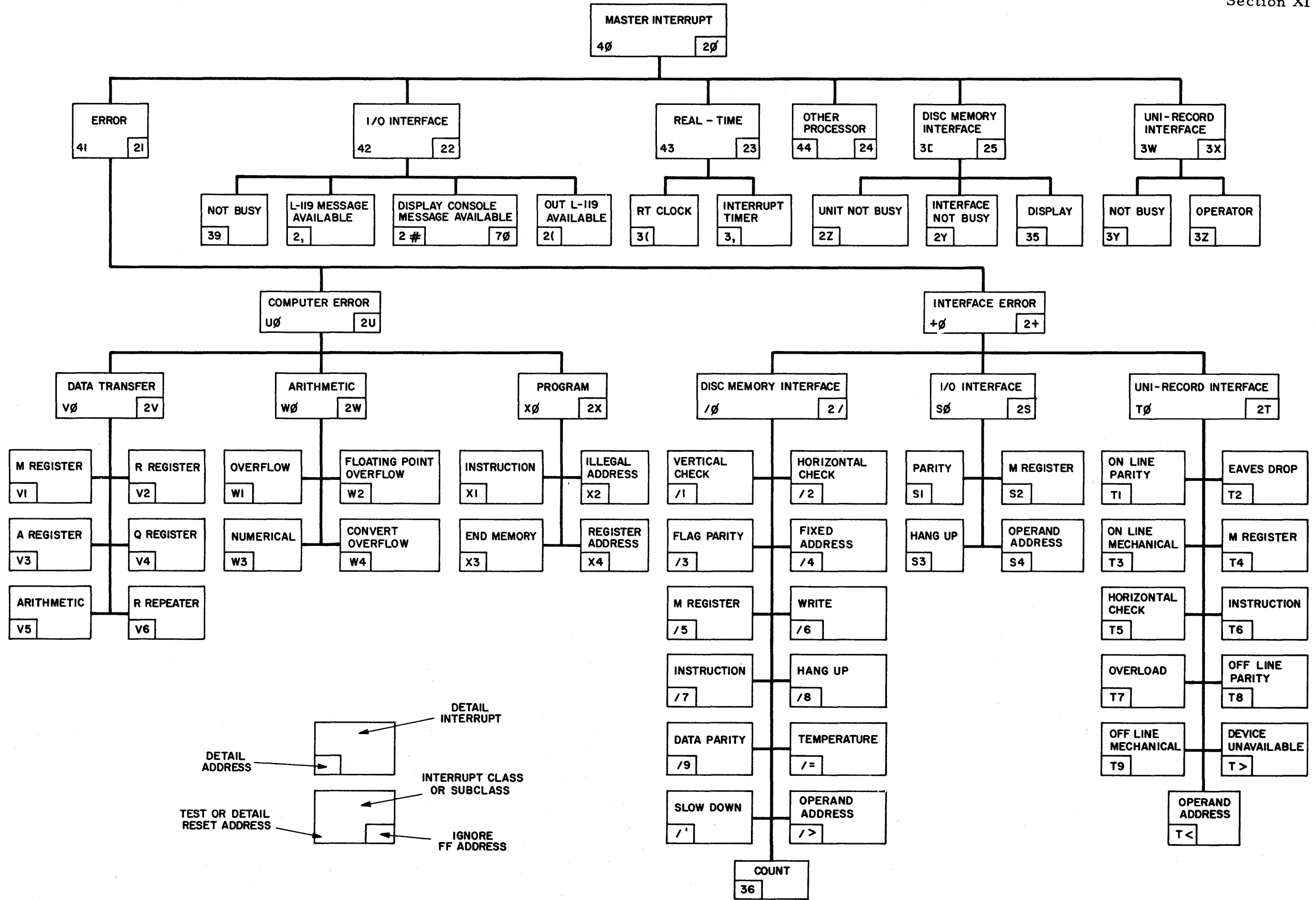


Figure 11-1. Display Buffer Status Word



## SECTION XII

### DUPLEXING

Data Processing System duplexing is accomplished by manual controls on the Central Switching Console. This Central Switching Console has switching control connections with the Data Processors, Buffer Processor, Disc Memories, and the Data Terminal Bay, figures 12-1 and 1-2.

Central Switching Console controls can duplex Console Printers, Large Panel Displays, and Multi-Color Displays between two Disc Memories, can duplex L-119 Modules and Electronic Typewriters between two Central Processors, and can synchronize and control system communication with the Autodin Data Terminal Bay.

#### 12. 1 BUFFER PROCESSOR SWITCHING.

The Central Switching Console contains control switches and indicators which enable monitoring and partial control of each L-119 Module in the Buffer Processor Console. Using the control switches, an operator is able to:

- a. Initiate or halt program execution in each L-119 Module.
- b. Select which L-119 Modules shall operate as an input/output pair with the Data Terminal Bay.
- c. Connect a pair of L-119 Modules in a Self-Test mode.
- d. Initiate connection of either Sending L-119 Module to the Central Processor.
- e. Connect a pair of L-119 Modules to either of two Central Processors.

#### 12. 1. 1 RECEIVING L-119 MODULE CONTROLS.

The following control switches are provided for each receiving L-119 Module, figure 12-2:

- a. Program Start Button - This control is a momentary make button switch. When depressed, it causes the L-119 Module to initiate program execution.
- b. Program Halt Button - This control is a latching button switch. This switch is testable by the L-119 program and enables a programmed halt.

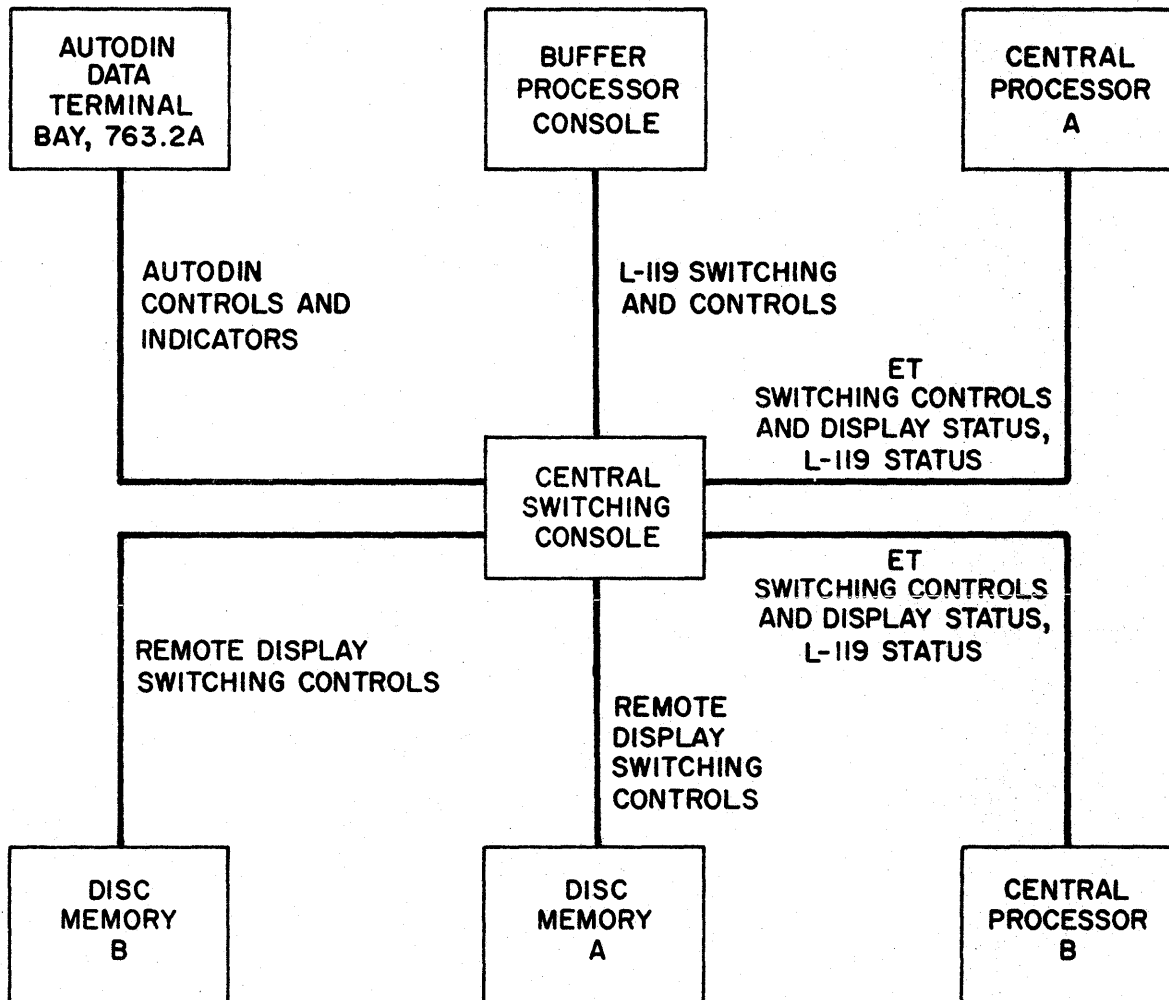


Figure 12-1. Switching Configuration

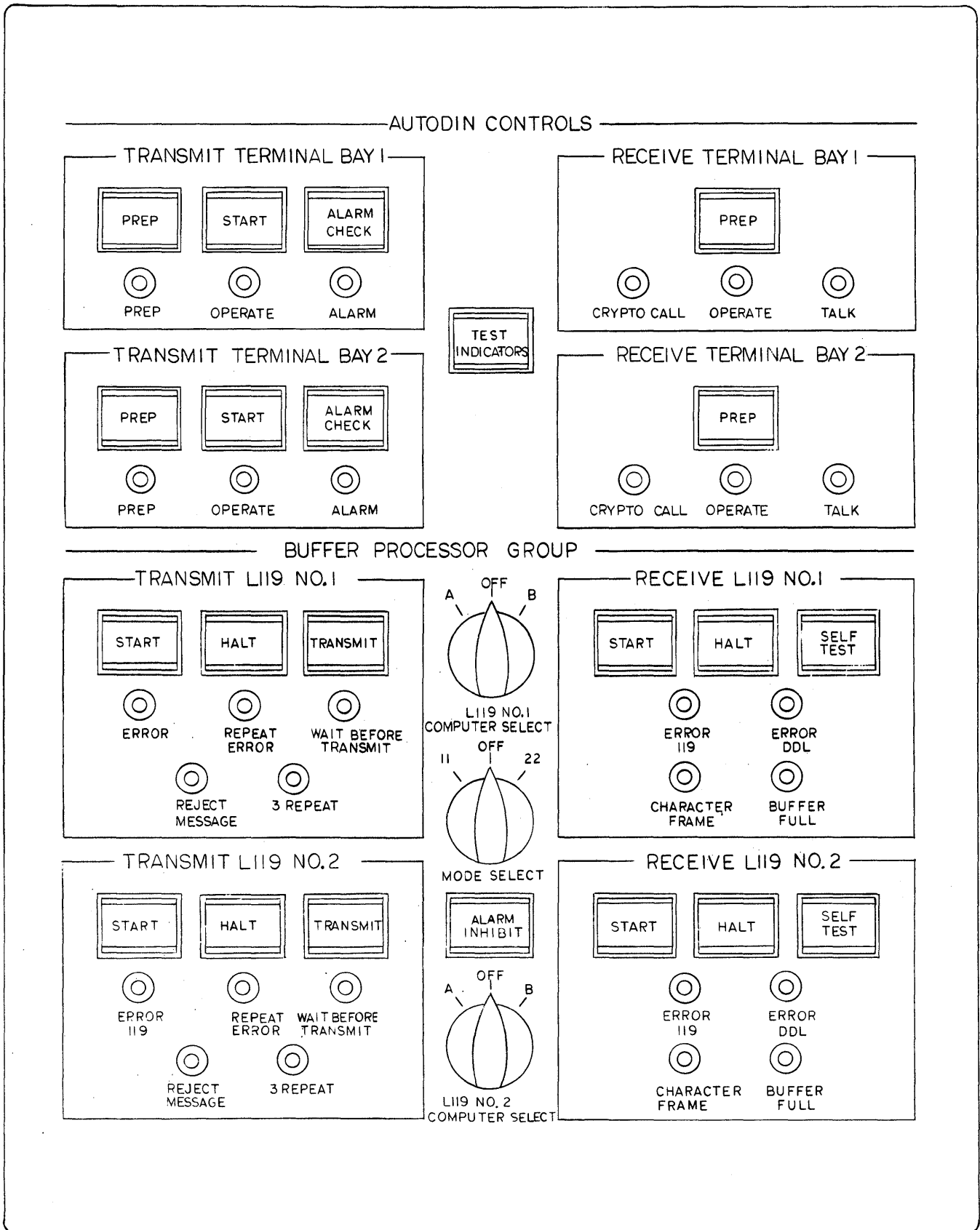


Figure 12-2. Central Switching Console Control Panel, Center Section

## Section XII

Paragraphs 12. 1. 2 to 12. 1. 3

### 12. 1. 2 SENDING L-119 MODULE CONTROLS.

The following control switches are provided for each Sending L-119 Module:

- a. Program Start Switch - See Paragraph 12. 1. 1
- b. Program Halt Switch - See Paragraph 12. 1. 1
- c. Transmit Button - This control is a momentary make button switch. When depressed it causes a Sending L-119 Module to be made available to the Central Processor by setting the L-119 available line to the true state. This control is used when bringing a Sending L-119 into an active state with a Data Terminal Bay.

### 12. 1. 3 L-119 MODULE PAIR CONTROLS.

The following control switches are provided for an input/output pair of L-119 units.

- a. Self Test Button - This latching switch, when depressed, enables the Sending L-119 Module to transmit data to the Receiving L-119 Module with which it is paired. This control enables the Central Processor to transmit data to a Sending L-119 Module and to receive the same data on a Receiving L-119 Module channel without going through the communications equipment. This enables the testing of the transmitting, receiving, checking and control circuits of the communications functions of the system using normal operating procedures.
- b. Module Selection Switch - This control is a rotary switch. Its function is to pair either Sending L-119 Module with either Receiving L-119 Module in the Buffer Processor Console to form a send-receive set. In the normal position (11), it causes the number one Receiving L-119 Module to function with the number one Sending L-119 Module, and the number two Receiving L-119 Module to function with the number two Sending L-119 Module. When switched to the 21 position, it switches the number one Receiving L-119 Module to function with the number two Sending L-119 Module, and the number two Receiving L-119 Module to function with the number one Sending L-119 Module. The setting of this switch is testable by the Central Processor.
- c. L-119 Central Processor Selection Switch - This control is a rotary switch. Its function is to duplex a pair (input/output) of L-119 Modules between two Central Processors. There are two such switches, one for each L-119 Module pair, and they are numbered 1 and 2 referring to Receiving L-119 Module number 1 and

Receiving L-119 Module number 2 respectively. In the A position, the switch selects the Receiving L-119 Module and its associated Sending L-119 Module, as determined by the Module Selection switch for Central Processor A. In the B position, the switch selects the Receiving L-119 Module and its associated Sending L-119 Module for Central Processor B. These switches are testable by the Central Processors.

#### 12. 1. 4 RECEIVING L-119 MODULE INDICATORS.

The following indicator lights are provided on the Central Switching Console for each Receiving L-119 Module:

- a. Compute Indicator - This light which is located behind the Start button is ON whenever the Receiving L-119 Module is actively executing a program.
- b. Halt Indicator - This light is located behind the Halt pushbutton. It comes ON when the L-119 Module is in a halted state (not executing program). When the indicator is ON, the Halt switch is illuminated red.
- c. L-119 Error Indicator - This indicator is red in color. It comes ON to indicate that a parity error has been detected in the internal information transfer of the L-119 Module. The error indication remains ON until the L-119 Module has been cleared by the operator.
- d. Digital Data Link Error Indicator - This indicator is red in color. It comes ON to indicate an error detected by the L-119 Module on the digital data link. It remains ON until the error condition is corrected by the remote station and normal data transmission is restored.
- e. L-119 Filled Indicator - This indicator is white in color. When ON, the light indicates that the complete data block received from the Switching Center by the Receiving L-119 Module is waiting for transmission to the Central Processor. A continuous ON condition of this indicator indicates that transmission of data to Central Processor has halted and manual intervention is required.
- f. Character Frame Indicator - This indicator is white in color. When ON, the light indicates that the Receiving L-119 Module is in character frame with the received data. The indicator must be ON for normal data reception.

### 12. 1. 5 SENDING L-119 MODULE INDICATORS.

The following indicator lights are provided on the Central Switching Console for each Sending L-119 Module:

- a. Compute Indicator - See paragraph 3. 4. 1 a.
- b. Halt Indicator - See paragraph 3. 4. 1 b.
- c. L-119 Error Indicator - See paragraph 3. 4. 1 c.
- d. L-119 Available Indicator - This indicator is white in color and is located behind the Transmit button. When ON, the light indicates that the Sending L-119 Module is available to the Central Processor and is requesting data blocks for transmission. When a data block has been correctly received from the Central Processor, the indicator is OFF during the period that it is processing and transmitting the data to the switching center.

After the transmission has been completed and acknowledged, the indicator again is set ON by the L-119. A continuous ON condition of the indicator indicates that data transmission from the Central Processor to the Sending L-119 Module is halted.

- e. Three Reply Indicator - The Three Reply indicator is red in color. When ON, the light indicates a reply code has been sent by the Sending L-119 Module three successive times without receiving an answer code; and manual intervention is required to restore data link transmission. This indicator is set ON by the Sending L-119 Module and is set OFF when data link transmission is restored.

- f. Reject Message Indicator - This indicator is white in color. The indicator is turned ON by the Sending L-119 Module when a Reject Message Code has been received from the switching center indicating that a message sent by the Sending L-119 Module has been rejected. The indicator is turned OFF by the Sending L-119 Module when a Discard Message Code is received from the Central Processor.

- g. Wait Before Transmit Indicator - This indicator is white in color. The indicator is turned ON by the Sending L-119 Module to indicate that a Wait Before Transmit code (WBT) has been received by the Sending L-119 Module and that the switching center is delaying continued transmission. This indicator is turned OFF by the Sending L-119 Module when normal data transmission is restored on the digital data link.

h. Error Reply Indicator - This indicator is red in color. The indicator is turned ON by the Sending L-119 Module to indicate that an Error Reply code (ER) has been received three successive times in answer to a transmitted line block. The indicator remains ON until the line block is answered correctly, or manual intervention is taken (i. e. , the operator clears the L-119 Module).

#### 12. 1. 6 TESTABLE L-119 MODULE CONTROL SWITCHES.

The positions of the L-119 Module Selection, Self Test, and Computer Selection switches are testable by the Central Processors. The testing of these switches is accomplished through use of the switch status word. The assignments of the switches in the switch status word are shown in figure 12-5. They occupy bits 1 through 5 of the first character of the word and bits 5 and 6 of the fourth character of the word. Bit one of the first character of the word is the Module Select switch. The Module Select switch selects which pair of L-119 Modules is to operate as a send/receive set.

In the 11 position, bit one of the status word is false. In the 21 position, bit one of the status word is true. If the bit is false (OFF), the pairing of the L-119 Modules is such that Receiving L-119 Module number 1 (address 60) is operating with Sending L-119 Module number 2 (address 63), and Receiving L-119 Module number 2 (address 61) is operating with Sending L-119 Module number 2 (address 64). If bit one of the status word is ON, then the above condition is reversed. Receiving L-119 Module number 1 is paired with Sending L-119 Module number 2 and Receiving L-119 Module number 2 is paired with Sending L-119 Module number 1.

Bit two of the first character of the status word is the number one Receiving L-119 Module/Central Processor Selection switch. With the Selection switch in the A position, bit two in Central Processor A is ON and bit two in Central Processor B is OFF. Bit two in the ON state indicates that the Receiving L-119 Module is selected to the Central Processor. In the B position, bit two in Central Processor A is OFF, and bit two in Central Processor B is ON. In the OFF position, the Receiving L-119 Module is disconnected from both Central Processors.

Bit three of the first character of the status word is the number two Receiving L-119 Module/Central Processor Selection switch. This bit operates in the same manner

with respect to the number two Receiving L-119 Module as bit one does with respect to the number one Receiving L-119 Module.

Bits four and five of the first character of the status word are the Sending L-119 Module number 1 and Sending L-119 Module number 2 selection bits. These bits are controlled by both the Receiving L-119 Central Processor Selection switches and the Module Selection switch. In switching Receiving L-119 Modules to a Central Processor through use of the Central Processor Selection switch, the Sending L-119 Module, which is paired with a given Receiving L-119 Module by the setting of the Module Selection switch, is also automatically selected. Hence bits four and five are a function of both switches. Either of the bits in the ON condition indicates that the assigned output Sending L-119 Module is selected to the Central Processor.

Bits 5 and 6 of the fourth character of the status word indicate the status of the Self Test switch on the Central Switching Console. Bit 5 in the ON state indicates that the Self Test switch associated with Receiving L-119 Module number 1 has been depressed and the Module is in the Self Test mode. The Self Test switch, when depressed, connects the Autodin side of a Receiving L-119 Module to the output side of a Sending L-119 Module. The Sending L-119 Module is determined by the setting of the Module Selection switch. This enables communication between a Receiving L-119 Module and a Sending L-119 Module without going through communications equipment.

Bit 6 in the ON condition indicates that the Self Test switch associated with Receiving L-119 Module number two has been depressed and the module is in the Self Test mode.

#### 12. 1. 7 L-119 MODULE OPERATION FROM THE CENTRAL SWITCHING CONSOLE.

After crypto synchronization has been established as described in paragraph 12. 2. 4, and with power on in the Buffer Processor console, the operator then can initiate transmission on the data link. This is accomplished by depressing the Start button on the Receiving L-119 Module. The Halt light goes OFF and the start light comes ON. The character frame light should also come on, indicating the L-119 Module is in frame with the received data. Next the operator should depress the Start button on the Sending L-119 Module. The Halt light goes OFF and the Start light comes ON, indicating the L-119 Module is executing a program. The operator should then depress the Transmit button and the Transmit light should come on, indicating that



the L-119 Module is available to the Central Processor. As data is being transmitted on the data link, the Transmit indicator on the Sending L-119 Module should flicker ON and OFF and the Buffer Full indicator on the Receiving L-119 Module should flicker ON and OFF. A steady ON state of either of these lights indicates transmission has stopped on the sending side or reception of data has stopped on the receiving side.

If at any time the operator desires to halt the L-119 Module, this can be accomplished by depressing the Halt switch. The L-119 Modules should be in a Halt state when switching of the modules (i. e. , between Central Processors, Self Test, etc, ) is initiated.

In the normal mode of operation, only the Start indicator on both L-119 Modules should be ON continuously. Should the L-119 Module detect an internal parity error, the L-119 Parity indicator will come ON and the module will halt.

The Sending L-119 Module has several additional indicators to signal the operator of the following conditions:

- a. Reply Error indicator ON and Start indicator ON - The L-119 Module has transmitted a line block three successive times without receiving an answer code.
- b. Reject Message indicator ON and Start indicator ON - The L-119 Module has received a Reject Message code from the Switching Center indicating a message sent by the Sending L-119 Module has been rejected.
- c. Wait Before Transmit indicator ON and Start indicator ON - The L-119 Module has received a Wait Before Transmit code from the Switching Center delaying transmission.
- d. Error Reply indicator ON and Start indicator ON - The L-119 Module has received an error Answer Acknowledgment three successive times for a transmitted line block.

#### 12. 1. 8 AUDIBLE ALARM.

The Central Switching Console contains an audible alarm system. The alarm system consists of a time delay relay, a loudspeaker and a 400 cycle source. The 400 cps

## Section XII

### Paragraphs 12. 1. 9 to 12. 2. 1

source is obtained from the Buffer Processor Console. When the alarm is activated, the time delay relay closes, causing the 400 cycle source to be connected through volume control to the loudspeaker located in the Central Switching Console. A time delay relay is used because some of the signals are of a pulse nature in the normal state, and a constant signal in the error state. The volume control for the audible alarm system is located behind the Center Section on the Central Switching Console Control Panel. An Alarm Inhibit button, which inhibits the audible alarm, is located on the Center Section of the Central Switching Console Control Panel. An indicator behind the button comes ON when the audible alarm system is inhibited. The audible alarm is activated when any of the following indicators are ON in the Buffer Processor Console:

- a. L-119 Error
- b. Three Reply
- c. Three Reply Error
- d. Buffer Full

#### 12. 1. 9 TEST INDICATORS BUTTON.

A Test Indicators button, when depressed, causes all indicators to come ON in the Central Switching Console. This control is included as a maintenance aid.

### 12. 2 AUTODIN REMOTE CONTROLS.

The Central Switching Console contains control switches and indicators to enable an operator to establish crypto synchronization of the Autodin equipment. There are three momentary buttons and three indicators associated with the Sending Crypto of each data link, and one momentary make button and three indicators associated with the Receiving Crypto of each data link. The buttons are not illuminated other than with normal room lighting. Remote controls and indicators for two Data Terminal Bays are implemented on the Central Switching Console, figure 12. 2.

#### 12. 2. 1 SENDING CRYPTO CONTROLS.

The remote control switches and indicators for the Sending Crypto are as follows:

- a. Momentary Make Button Switches:
  - (1) Alarm Check switch

(2) Prep. switch

(3) Start switch

b. Incandescent Indicators:

(1) Alarm indicator

(2) Operate indicator

(3) Prep indicator

### 12. 2. 2 RECEIVING CRYPTO CONTROLS.

The remote control switch and indicators for the Receiving Crypto are as follows:

a. Switch - Momentary Make Button Switch:

(1) Prep switch

b. Incandescent Indicators:

(1) Crypto Call indicator

(2) Operate indicator

(3) Talk indicator

### 12. 2. 3 CONTROL SWITCH AND INDICATOR POWER.

The power required to drive the incandescent indicators and any control functions operated by the remote buttons is supplied by the Autodin Data terminal Bay. A six volt incandescent indicator is used in the Central Switching Console.

### 12. 2. 4 OPERATION OF AUTODIN CONTROLS.

Synchronization of the Crypto is initiated by the operator by momentarily depressing the Alarm Check Key. This action causes the following sequence associated with the Sending Crypto to be performed automatically:

a. The Alarm Check switch on the Crypto is locked in the ALARM CHECK position. This initiates an Alarm Check of the Crypto.

b. The Alarm indicator on the Central Switching Console comes ON.

c. The Prep Key on the Crypto is momentarily depressed, which initiates a Prep of the Crypto.

## Section XII

- d. The Prep indicator on the Central Switching Console comes ON.
- e. The Operate and Alarm indicator on the Central Switching Console goes OFF.
- f. The Start Key on the Crypto is momentarily depressed.
- g. The Prep indicator light goes OFF on the Central Switching Console.
- h. The Alarm and Operate indicators on the Central Switching Console come ON as an acknowledgment of the completion of the alarm check function.
- i. The Alarm Check switch on the Crypto is released which initiates the transmission of a Crypto call sequence to the Switching Center.
- j. The Prep Key on the Crypto is momentarily depressed, which Re-Preps the Crypto.
- k. The Prep indicator on the Central Switching Console comes ON.
- l. The Operate and Alarm indicators on the Central Switching Console go OFF. (These 12 steps occur in rapid order - the operator should see the console Prep indicator comes ON and the console Operate and Alarm indicators go OFF at the completion of this automatic procedure.)
- m. The transmission of the Crypto call sequence to the Switching Center is completed.

The remote operator upon noticing the Operate indicator goes OFF that is associated with the Receiving Crypto and the lighting of the Crypto Call Lamp (the Crypto Call Detector at the Switching Center causes the Crypto Call indicator to come ON and Preps the Receiving Crypto when a Crypto call sequence is detected), depresses his Alarm Check Key, which should automatically put the Switching Center Sending Crypto through the thirteen steps, under control of the Switching Center Automatic Alarm Check unit.

The operator Crypto Call Detector causes the Crypto Call indicator to come ON and Preps the Receiving Crypto (which causes the console Operate indicator to go OFF that is associated with the Receiving Crypto) upon receipt of the Crypto call sequence from the remote station. This is an indication that the remote end (Switching Center) has received the operators Crypto call sequence and has put the Sending Crypto through the thirteen step sequence. The operator then momentarily depresses the

Start Key on the Central Switching Console, which should extinguish the console Prep Lamp and transmit a crypto synchronizing pattern to the Switching Center. The Console Sending Crypto Operate indicator comes ON shortly thereafter.

The reception of the synchronizing pattern by the Receiving Crypto at the Switching Center causes the Crypto Call indicator to go OFF and the console Receiving Crypto Operate indicator to come ON. An IL indicator comes ON shortly thereafter, which indicates that bit synchronization for that side of the communication channel has been established. The Center operator then depresses the Start Key which transmits a Crypto synchronizing pattern to the L-119 Module.

### 12. 3 INTEGRATED CONSOLE SWITCHING.

The Central Switching Console contains, in addition to the switches and controls specified in paragraphs 12. 1 and 12. 2, switches to enable the switching of the Electronic Typewriter, Control and Key sections of each of 15 ET consoles to either of two Central Processors. It also contains switches for the switching of Multi-Color Displays, Large Panel Displays, and Console Printers to either of two Disc Memories.

#### 12. 3. 1 ELECTRONIC TYPEWRITER SWITCHING

The control switches for switching of the Electronic Typewriter Control and Key sections are implemented on the Central Switching Console, as shown in figure 12. 2. The switches are three position rotary switches with the center position the OFF position (i. e. , the Electronic Typewriter is not connected to either Central Processor). The other two positions of the switch are A and B, indicating the connection of the ET to Central Processor A or B.

There are 15 switches to implement the switching of the ET, Control and Key sections of each of 15 Integrated Consoles to either of two Central Processors. The switches are arranged in a three row, five column matrix. Each switch is labeled to indicate which ET it is controlling. The labeling is ET1, to designate Electronic Typewriter one, ET2, to designate Electronic Typewriter two, etc.

The status of any of the 15 Electronic Typewriters may be determined by the Central Processor by examining the switch status word. The 15 Electronic Typewriter Selection switches have been assigned bit positions in the switch status word, as shown in figure 12-5. A bit in the ON (true) condition indicates that the Electronic

Section XII  
Paragraph 12. 3. 2

Typewriter is selected for the Central Processor. The Selection switch in the A position causes the status bit in computer A to be ON and in the B position causes the status bit in computer B to be ON. In the OFF position, both status bits are false. Thus, the Central Processor can determine if an Electronic Typewriter station is connected to its Interface, and by communication with the other Central Processor, it can determine if the Electronic Typewriter is connected to the other Central Processor or is disconnected from both Central Processors. Table 12-1 shows the relationship between the Electronic Typewriter Selection switches on the Central Switching Console and the ZY address used in the Central Processor.

Table 12-1. Electronic Typewriter Addresses

SWITCH DESIGNATION	ZY ADDRESS
ET1	71
ET2	72
ET3	73
ET4	74
ET5	75
ET6	76
ET7	77
ET8	78
ET9	79
ET10	7=
ET11	7'
ET12	7>
ET13	7<
ET14	7]
ET15	7⌋

12.3.2 DISPLAY AND CONSOLE PRINTER SWITCHING.

The control switches are on the Central Switching Console, figure 12-4, for switching of the Large Panel Displays (LPD), the Multi-Color Displays (MC), and the Console Printers (CP) to either of two Disc Memories. The switches are three position rotary switches with the center position the OFF position (i. e., the LPD, MC, CP, or MC/CP are disconnected from both Disc Memories). The other two

positions of the switch are A and B, indicating the connection of the device to Disc Memory A or Disc Memory B.

12. 3. 2. 1 LARGE PANEL DISPLAY SWITCHING. Eight switches implement the switching of the Large Panel Displays to either of two Disc Memories. Each LPD is switchable independently. The eight switches are arranged in a row on the Central Switching Console and labeled LPD1 through LPD8.

12. 3. 2. 2 MULTI-COLOR DISPLAY/CONSOLE PRINTER SWITCHING. Ten switches implement the switching of ten Multi-Color Display and Console Printer pairs to either of two Disc Memories. The switches are in two rows of five switches. The switches are located under the row of eight LPD switches and labeled MC/CP 1 through MC/CP 10. Each MC/CP pair is switchable individually.

12. 3. 2. 3 CONSOLE PRINTER SWITCHES. Five switches implement the switching of five Console Printers to either of two Disc Memories. The switches are in one row of five switches. The switches are located beneath the MC/CP switches and labeled CP1 through CP5. Each CP is switchable independently.

12. 3. 2. 4 MULTI-COLOR DISPLAY SWITCHES. A switch is provided for switching one Multi-Color Display to either of two Disc Memories. The switch is labeled MC1.

12. 3. 2. 5 TESTING DISPLAY AND CONSOLE PRINTER STATUS. The status of the Displays and Console Printers (i. e. , connected or disconnected to a Disc Memory) may be determined by the Central Processors by examining the switch status word. Bit positions have been assigned in the word for eight Large Panel Displays, ten Console Printer/Multi-Color Display combinations, five Console Printers, and one Multi-Color Display. In the straight connected system configuration Disc Memory A will operate with Central Processor A and Disc Memory B with Central Processor B. By examining the switch status word, Central Processor A can determine if a particular Display or Console Printer is selected for its Disc Memory. By interrogating Central Processor B, it can determine if it is selected for Disc Memory B or disconnected from the system.

A bit in the status word in the ON (true) state indicates a Display or Console Printer is selected to the Disc Memory. Table 12-2 shows the relationship between the Display and Console Printer Switches and their ZY address. Figure 12-5 shows the switching status word.

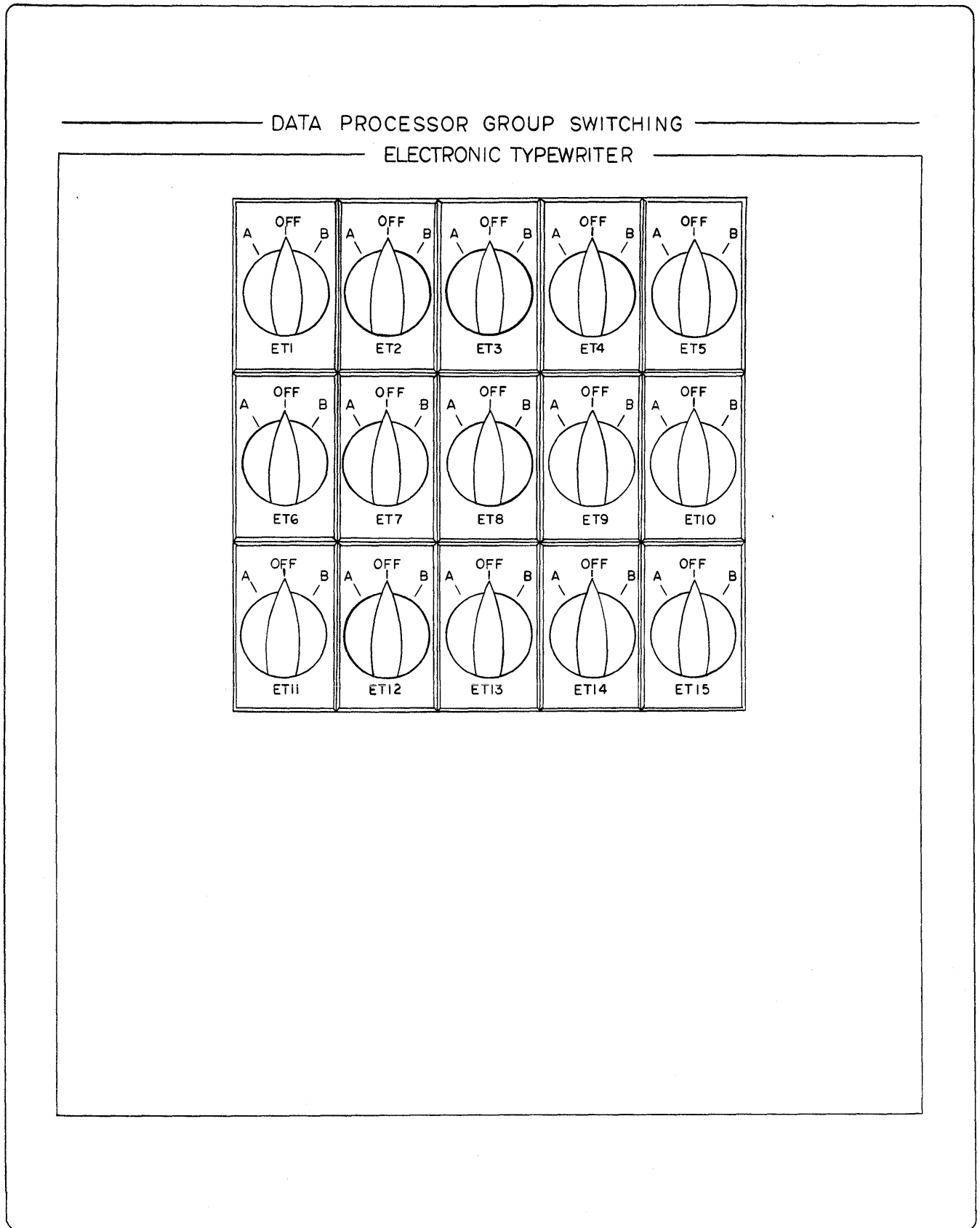


Figure 12-3. Central Switching Console Control Panel, Right Section



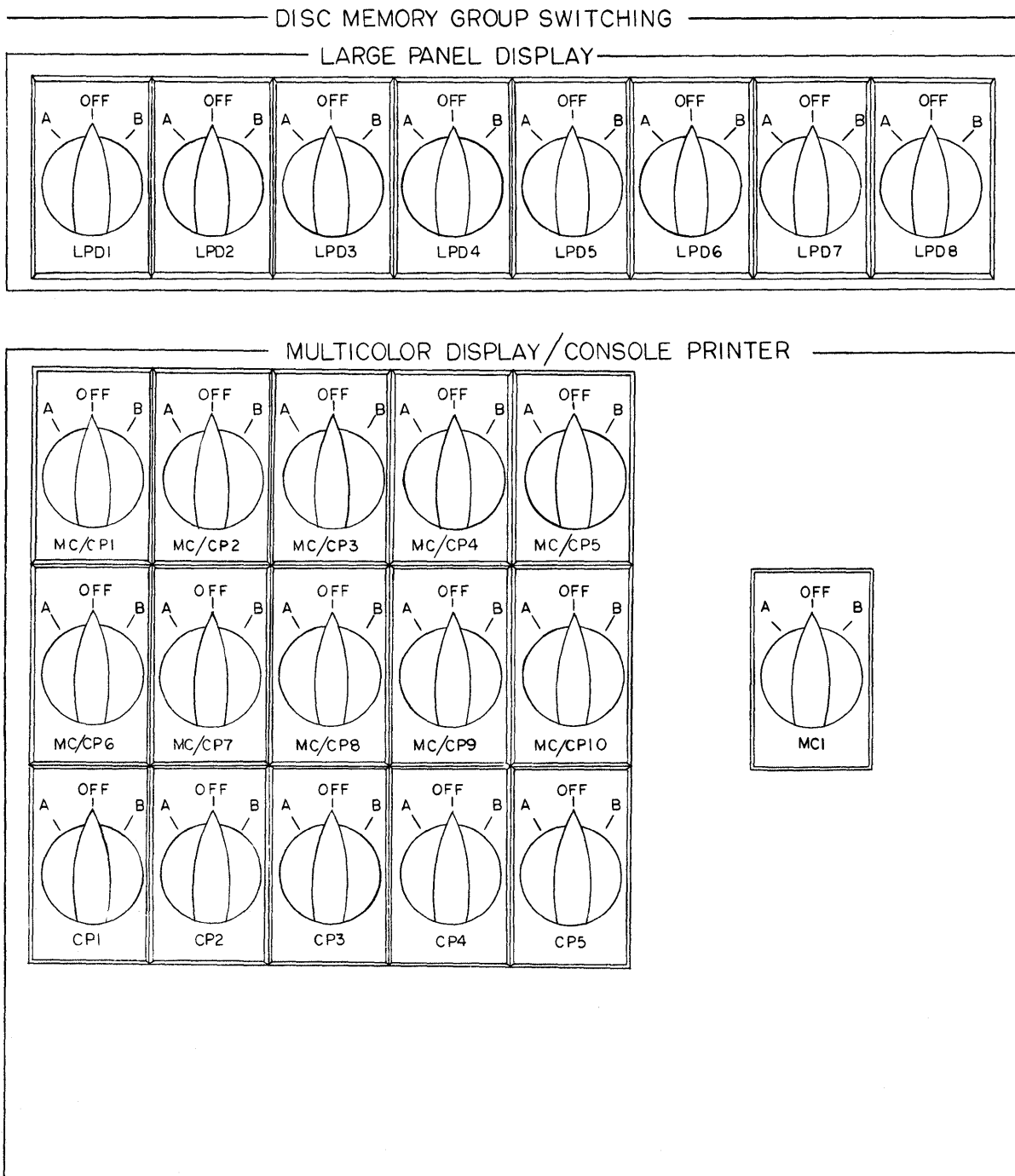


Figure 12-4. Central Switching Console Control Panel, Left Section

Table 12-2. Disc Memory Display Identification

SWITCH ON CENTRAL SWITCHING CONSOLE	STATUS WORD POSITION	ADDRESS	DISPLAY TO DISC MEMORY CONNECTION
LPD1	46	70	3J11
LPD2	44	72	3J13
LPD3	42	74	3J15
LPD4	51	76	3J17
LPD5	45	71	3J12
LPD6	43	73	3J14
LPD7	41	75	3J16
LPD8	52	77	3J18
MC/CP1	53	78	3J19
MC/CP2	54	7=	3J20
MC/CP3	55	7>	3J21
MC/CP4	56	7]	3J22
MC/CP5	61	7+	3J23
MC/CP6	62	7S	3J24
MC/CP7	63	7U	3J25
MC/CP8	64	7W	3J26
MC/CP9	65	7Y	3J27
MC/CP10	66	7(	3J28
MC1	71	6[	3J29
CP1	72	7[	3J34
CP2	73	7;	3J33
CP3	74	7°	3J32
CP4	75	7#	3J31
CP5	76	6]	3J30

7	6	5	4	3	2	1	0
← PARITY →							
6C MC	7C MC/CP	7J MC/CP	70 LPD	SELF TEST INPUT L-119-1	7> ET	76 ET	∅
6L CP	7Y MC/CP	7> MC/CP	7I LPD	SELF TEST INPUT L-119-2	7' ET	75 ET	64 OUTPUT L-119 2
7# CP	7W MC/CP	7= MC/CP	72 LPD	∅	7= ET	74 ET	63 OUTPUT L-119 1
7° CP	7U MC/CP	78 MC/CP	73 LPD	7~ ET	79 ET	73 ET	6I INPUT L-119 2
7; CP	7S MC/CP	77 LPD	74 LPD	7J ET	78 ET	72 ET	6∅ INPUT L-119 1
7E CP	7+ MC/CP	76 LPD	75 LPD	7< ET	77 ET	7I ET	L-119 PAIRING

ET = ELECTRONIC TYPEWRITER  
 CP = CONSOLE PRINTER  
 MC = MULTI-COLOR DISPLAY  
 LPD = LARGE PANEL DISPLAY

Figure 12-5. Switch Status Word (ZY17)

#### 12. 4 DISC MEMORY TRUNKLINE.

Each Central Processor has a Disc Memory Interface and a corresponding Disc Memory Trunkline. The purpose of the duplexing function is to permit either Central Processor to use either Disc Memory Trunkline and the associated Disc Memory modules, under control of certain program operated toggles.

Two basic modes of operation are possible: each Central Processor can communicate with its own Disc Memory Trunkline; or each Central Processor can communicate with the other Central Processor Trunkline. A further feature is that shut down of a Central Processor or of a Disc Memory should not interfere with operation of the remaining consoles.

The system configuration is under program control involving a number of status toggles interconnected between the two Central Processors.

##### 12. 4. 1 CONTROL STATUS TOGGLE.

One Central Processor at a time has Control Status, as indicated by its own Control Status toggle (ZY = 59). Control Status also affects the use of the Shared Core Memory Trunkline, paragraph 2. 2. 3. Only one processor may have the Control Status toggle ON at one time.

##### 12. 4. 2 DISC MEMORY TRUNKLINE COUPLING MODE TOGGLE.

Either Central Processor can test the state of its own Disc Memory Coupling Mode toggle (ZY = 5 =) with a true (ON) state designating straight coupled operation. The Central Processor with Control Status can set and reset the Disc Memory Coupling Mode toggles in the two Central Processors, both toggles at the same time.

##### 12. 4. 3 OTHER DISC MEMORY BUSY.

Either Central Processor can test the Other Disc Memory Busy (ZY = 5'), as well as its own Disc Memory Busy (ZY = 37). The Central Processor with Control Status can terminate a Disc Memory operation in either Disc Memory by resetting the appropriate Disc Memory Busy toggle. When the Disc Memories are cross coupled, the meaning is reversed.

##### 12. 4. 4 OTHER DISC MEMORY LOCKOUT.

The Central Processor that has Control Status can inhibit initiation of any Disc Memory operation by the Central Processor without Control Status. This is done by

setting the Other Disc Memory Lockout toggle (ZY = 5). If a Disc Memory operation is in progress, the operation will continue to completion before the Lockout is effective. If the Central Processor without Control Status ignores Lockout and attempts to execute a Disc Memory instruction, a Disc Memory Interface Hang Up Interrupt Error request is generated.



## SECTION XIII

### CONTROL CONSOLE AND CENTRAL PROCESSOR CONTROL PANEL

The Control Console consists of a Control Panel with a number of indicating push-button switches, for Central Processor manual control, and a Control Typewriter, which communicate directly with Central Processor registers. The Control Console Control Panel is shown in figure 13-1.

The Central Processor Control Panel consists of three sections which contain Interface indicators, figure 13-2, Program indicators, figure 13-3, and Maintenance controls, figure 13-4.

#### 13.1 CONTROL CONSOLE CONTROL PANEL.

The Control Console Control Panel, figure 13-1, contains both locking and non-locking pushbutton type switches that have integrally mounted lights. Depressing a locking type button locks the button and causes the light to come ON. Upon depressing the button again, the lock releases and the light goes OFF. Non-locking buttons are momentary make type controls. Internal lamps for these buttons generally reflect an internal condition; thus, they may light or go out in response to depressing the button. The Control Panel contains the following indicating push-button controls:

- a. Operator Interrupt Button — This momentary make indicator switch generates an Operator Interrupt request. The Operator Interrupt button sets ON the Operator Interrupt Detail toggle which remains ON until the toggle is reset OFF by program, paragraph 11.11.2.
- b. Breakpoint 1 - 9 Buttons — These locking indicator switches are nine program testable switches. By depressing a Breakpoint button, a corresponding switch is closed which can be individually tested by program.
- c. Tag Button — This locking indicator switch causes the Central Processor to halt after execution of a tagged instruction. By depressing the Tag button, the Tag switch is closed. It is possible to test this switch by program.
- d. Start Button — This momentary make indicator switch starts execution of the Central Processor program. The Start button is effective only if depressed while

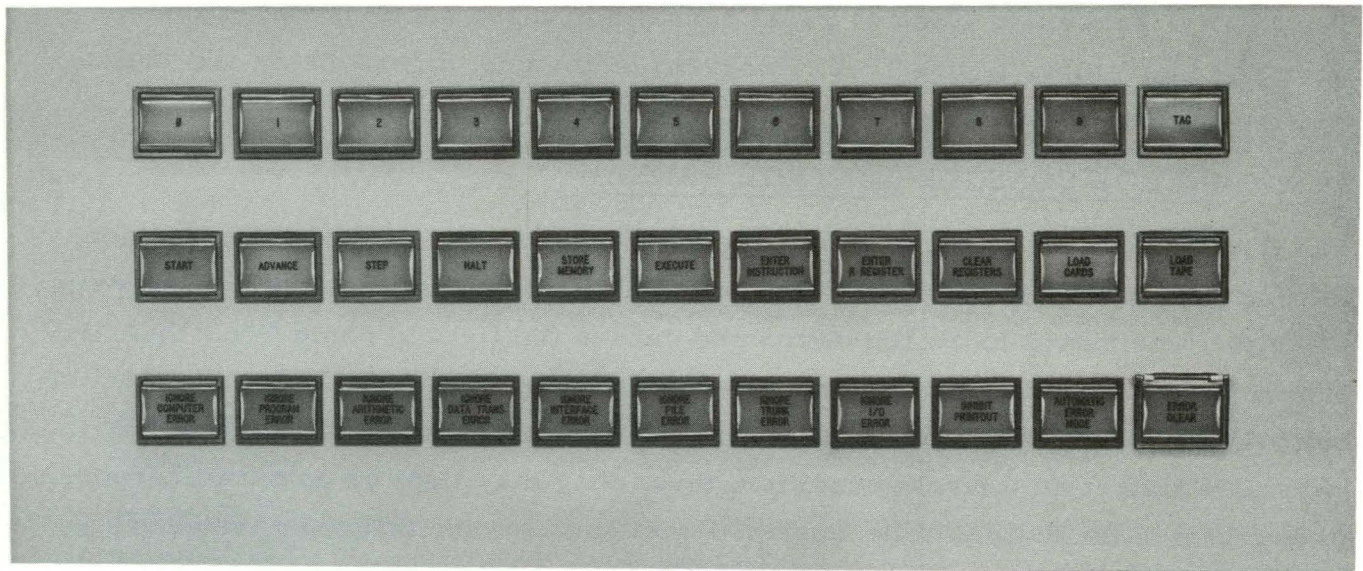


Figure 13-1. Control Console Control Panel



in the Halt mode. The Start button light is ON when program execution begins and remains ON until the Halt mode is reentered.

e. Advance Button — This locking indicator switch causes program execution at the rate of approximately one instruction per second. The Advance button is effective anytime.

f. Step Button — This momentary make indicator switch executes one instruction and halts each time the button is depressed. The light is ON while the button is depressed. The Step button is effective only in the Halt mode. The Central Processor accesses and executes the instruction specified by the Instruction Address register, advances the content of this register, and returns to Halt mode.

g. Halt Button — This momentary make indicator switch stops program execution and places the Central Processor in the Halt mode. The Halt light is ON as long as the Central Processor is halted. Depressing the Halt button causes entry into the Halt mode after execution of the current Central Processor instruction. Any Interface data transfers will proceed to completion.

h. Store Memory Button — This momentary make indicator switch stores the content of the R register in a specified memory location under manual control. The Store Memory button is effective in the Halt mode only. The content of the R register is stored in the Core Memory location specified by the Operand Address register, and the Operand Address register counts up by one. The indicator light is ON when the button is depressed.

i. Execute Button — This momentary make indicator switch initiates execution of the instruction contained in the Instruction register. The Execute button is effective only in Halt mode. When depressed, the Central Processor executes the instruction in the Instruction register, as though the instruction had just been read from memory (includes any specified address modification).

j. Enter Instruction Button — This momentary make indicator switch permits entry of information from the Control Typewriter into the Central Processor Instruction register when the Central Processor is in Halt mode. The Enter Instruction button, when depressed, lights, and stays ON until the Enter R Register button is depressed. Entry is effective only in Halt mode. When effective, eight characters of a Central Processor instruction may be typed directly into the Instruction register. If a ninth character is typed, the first eight will be lost.

k. Enter R Register Button — This momentary make indicator switch permits entry of information from the Control Typewriter into the R register anytime the Central Processor is in Halt mode. The Enter R Register button, when depressed, lights, and stays ON until the Enter Instruction button is depressed. Entry is effective only in the Halt mode. When effective, eight, or fewer, characters of a Central Processor word may be typed directly into the R register. If a ninth character is typed, the first eight will be lost.

l. Clear Registers Button — This momentary make indicator switch clears the Central Processor A and R registers, and Phase Control. When the Clear Registers button is depressed while in Halt mode, the A and R registers are cleared. When depressed while not in Halt mode, the Phase Control of instruction execution immediately reverts to the Halt state, but the A and R registers are not cleared. The indicator light is ON while the button is depressed.

m. Load Cards Button — This momentary make indicator switch selects a Card Reader and initiates an Input operation in the Uni-Record Interface. The indicator light is ON when the button is depressed. The Load Cards button is effective only in the Halt mode. To be selected, the Card Reader must have its Trunkline Address set to 002; ZY = 0. Core Memory will be loaded, starting at location 20. Loading will continue until the Card Reader hopper is empty. After the last card is read, program execution begins at location 20. The response to this button is disabled if the Computer Error or Uni-Record Interface Error signal is present (ignored or not).

n. Load Tape Button — This momentary make indicator switch selects a Tape Transport on a Magnetic Tape Console and initiates an Input operation in the Uni-Record Interface. The indicator light is ON during an Input operation. The Load Tape button is effective only in the Halt mode. To be selected, the Tape Transport must have its Trunkline Address set to 000, ZY = 0Y. The Core Memory will be loaded, starting at location 20. Loading will continue until an end of record gap is reached, and then program execution begins at location 20. The response to this button is disabled if the Computer Error or Uni-Record Interface Error signal is present (ignored or not).

o.. Ignore Error Buttons — These momentary make indicator switches complement the individual Ignore Error toggles. The indicator light is ON when the associated toggle is ON. It is possible to depress these buttons at any time. The

associated toggle may be set or reset either by pressing the button or by program control. When an Ignore Error toggle is set, the occurrence of errors continues to activate the proper error toggles and visual indicators, but the Central Processor does not receive a Program Interrupt. The Ignore Error buttons are as follows:

- (1) Computer Error
- (2) Program Error
- (3) Arithmetic Error
- (4) Data Transfer Error
- (5) Interface Error
- (6) File Error - (Disc Memory Interface Error)
- (7) Trunkline Error - (Uni-Record Interface Error)
- (8) Input-Output Error - (I/O Interface Error)

p. Inhibit Printout Button — This locking indicator switch inhibits output typing by the Print instruction, (<). When depressed, Print (<) instructions are treated as No Operation (N) instructions. The indicator light is ON when printing is inhibited.

q. Auto Error Mode Button — This momentary make switch complements the Automatic/Manual Error Mode toggle. The indicator light is ON when in the Automatic Error mode. It is possible to depress this button at any time. When in Manual Error mode, errors halt the Central Processor. When in the Automatic Error mode, errors cause Error Interrupt requests which will cause program interrupts unless an appropriate Ignore Error toggle is ON. If an Instruction Error Interrupt request (ZY = X1) is generated while the Master Ignore Interrupt toggle is ON, the Central Processor halts regardless of the state of the Automatic/Manual toggle.

r. Error Clear Button — This momentary make switch, with safety cover, resets all Detail Error toggles. The indicator light is ON when the button is depressed.

## 13.2 CONTROL TYPEWRITER.

The Control Typewriter is a 12-character per second electric typewriter. It has a 64-character code, obtained from the keyboard by the use of some upper-case characters. An On-Off switch under the keyboard controls power to the typewriter motor.

### 13.2.1 MANUAL INPUT.

Three buttons on the Control Console are associated with the typewriter in the Manual mode and are active only when the Central Processor is halted. The Enter Instruction button, paragraph 13.1j., allows typing into the Instruction register. The Enter R Register button, paragraph 13.1k., allows typing directly into the R register. Both registers have a capacity of eight characters. If a ninth character is typed, the first eight are lost. The Store Memory button, paragraph 13.1h., is provided to store the content of the R register into Core Memory at the location specified by the Operand Address portion of the Instruction register. The Control Typewriter keyboard is locked unless one of the Enter buttons is active, or program control unlocks it.

### 13.2.2 TYPEWRITER INSTRUCTIONS.

The Control Typewriter responds to two Central Processor instructions: Input from Typewriter and Print.

#### 13.2.2.1 INPUT FROM TYPEWRITER.

Command Character >

Instruction >ZYXMMMM

Options Index, Indirect Address, Tagged Halt, Flag Return

Mnemonic RTYP, RTYPC

The Input from Typewriter instruction causes a number of manually typed words to enter Core Memory starting at the effective operand address location. A fixed number of words may be specified by the ZY characters of the instruction. The numeric parts of Z and Y specify the tens and units, respectively, and the Y5 and Y6 bits specify 100 and 200. The maximum number that can be specified is 399. If the ZY characters are V $\emptyset$ , words will be entered until the Stop Code button on the Typewriter is depressed. If the last word typed is incomplete when the Stop

Code key is depressed, the remaining characters will be padded with zeros before the word is stored in memory. The Stop Code is not itself a character; therefore, it does not enter the R register or memory, but the key does print a Y symbol.

#### 13.2.2.2 PRINT.

Command Character <

Instruction <ZYXMMMM

Options Index, Indirect Address, Tagged Halt, Flag Return

Mnemonic WTYP

The Print instruction causes the specified number of words to be typed on the Control Typewriter sequentially, starting with the word specified by the effective operand address. The number of words is specified by the ZY field in the same way as for the Input from Typewriter instruction, paragraph 13.2.2.1, up to 399 words.

Words printed may be separated or not under program control by setting or re-setting the Tab Inhibit toggle (ZY = 3=). When tabs are allowed, the typewriter carriage skips to the next tab stop after printing each word. Carriage returns occur at the beginning of printing, automatically upon reaching the right hand margin, and at the end of printing.

The Print Inhibit button on the Control Console inhibits all output typing, by treating Print Instructions as No Operation instructions.

A ZY field of 00 will cause the Typewriter to carriage return and line feed twice with no printing.

### 13.3 CENTRAL PROCESSOR CONTROL PANEL.

The Central Processor Control Panel consists of three sections. The Left and Center Sections contain indicators. The Right Section contains controls and indicators.

#### 13.3.1 LEFT SECTION INDICATORS.

The Left Section of the Central Processor Control Panel, figure 13-2, contains Interface indicators.

13.3.1.1 REAL TIME INDICATORS.

a. (Digital Clock) — Indicates the hours, minutes and seconds of the time held in the Real-Time Clock.

13.3.1.2 SYSTEM STATUS INDICATORS.

a. Switch Control — The Switch Control indicator light is ON when the Central Processor has Control status.

b. Coupling Mode — The Disc Memory Coupling Mode indicator lights when the Central Processor is connected to its assigned Disc Memory Trunkline.

c. Other File Busy — The Other File Busy indicator light is ON when the other Central Processor is engaged in a Disc Memory operation.

d. Other File Lockout — The Other File Lockout indicator light is ON when the Central Processor with Control status inhibits initiation of any Disc Memory operation by the Central Processor without Control status.

e. General Purpose — The General Purpose indicator light is ON when the General Purpose Status toggle is ON.

13.3.1.3 PERIPHERAL DEVICES INDICATORS.

a. Master Error — The Master Error indicator light is ON if an Interface Error signal exists.

b. Ignore Error — The Ignore Error indicator light is ON when the Ignore Interface Error Toggle is set ON.

c. Waiting:

(1) File — The File indicator light is ON when the Disc Memory Interface is Not Busy.

(2) Uni-Record — The Uni-Record indicator light is ON when the Uni-Record Interface is Not Busy.

(3) I-O — The I-O indicator light is ON when the I/O Interface is Not Busy.

d. Busy:

(1) File — The File indicator light is ON when the Disc Memory Interface is Busy.

(2) Uni-Record — The Uni-Record indicator light is ON when the Uni-Record Interface is Busy.

(3) I-O — The I-O indicator light is ON when the I/O Interface is Busy.

e. Buff Proc Channel 1-6 — Each Buff Proc Channel indicator light is ON when a Device Available or Message Available signal exists in the respective Buffer Processor channel.

#### 13.3.1.4 I-O PROCESSOR INDICATORS.

a. Master Error — The Master Error indicator light is ON when an I/O Interface Error Interrupt request is present.

b. Ignore Error — The Ignore Error indicator light is ON when the Ignore I/O Interface Error Interrupt Toggle is ON.

c. Errors:

(1) Par — The Par (parity) indicator light is ON when a parity error is detected by a Buffer Processor or a Display Console during a data transfer through the I/O Interface.

(2) M Reg — The M Reg indicator light is ON when a parity error is detected in the M register during a data transfer through the I/O Interface.

(3) Hang Up — The Hang Up indicator light is ON when the following conditions exist: an L-119 Module or Display Console does not terminate an operation within a specified time for any reason, or the selected device is unavailable.

(4) Oper Add — The Oper Add (operand address) indicator light is ON for the following conditions in the I/O Interface: a non-valid address, or counting into a non-assigned Core Memory address.

#### 13.3.1.5 FILE STATUS AND ERROR INDICATORS.

a. Status:

(1) O'Load — The O'Load (overload) indicator light is ON for any one of the following conditions: insufficient block spaces are made available to NNNN Counter, or the NNNN Counter exceeds 9999.

(2) Twin Cmpr — The Twin Cmpr (twin compare) indicator light is ON for any one of the following conditions: a difference in data read from Twin A and

Twin B during the Twin mode, or a difference in the Compare, Missed, or Obsolete flags.

(3) File Inop — The File Inop (file inoperative) indicator light is ON when the selected Disc Memory is on-line and operating.

(4) Disp Inop — The Disp Inop (display inoperative) indicator light is ON when an attempt is made to load the Multi-Color (MC) Buffer Storage past the unload point.

b. Errors:

(1) Vert Check — The Vert Check (vertical check) indicator light is ON when a vertical parity error is detected in a transfer of data, key, mask or control characters out of the Disc Memory Core Buffer.

(2) Horiz Check — The Horiz Check (horizontal check) indicator light is ON when an error exists in the data transfer from the Disc Memory during a Read operation.

(3) Flag Par — The Flat Par (flag parity) indicator light is ON when an error parity is encountered while reading or writing data on a Disc Memory Flag track.

(4) Fix Add — The Fix Add (fixed address) indicator light is ON when a parity error is detected while reading a fixed address from a disc.

(5) M Reg — The M Reg (M register) indicator light is ON when a parity error is detected in the Core Memory M register during a data transfer to and from the Disc Memory.

(6) Write — The Write indicator light is ON when an error is detected in a disc write amplifier or an attempt is made to write in an inhibited disc location.

(7) Instr — The Instr (instruction) indicator light is ON when a format error exists in the second or third word of a Disc Memory instruction.

(8) Hang Up — The Hang Up indicator light is ON when a Disc Memory operation is not executed in 30 seconds, or termination of a memory operation due to addressing an existent Disc Memory and a non-existent Auxiliary Disc Memory.

(9) Data Par — The Data Par (an internal vertical parity error) indicator light is ON when data containing incorrect parity is transmitted between the Central Processor and Disc Memory.



(10) Temp — The Temp indicator light is ON when the air temperature in the Disc Memory cabinet exceeds a safe operation level.

(11) Slow Down — The Slow Down indicator light is ON when the Disc Motor speed drops below a preset limit.

(12) Oper Add — The Oper Add (operand address) indicator light is ON when an attempt is made by the Disc Memory Interface to address an unassigned Core Memory location.

(13) Count — The Count indicator light is ON when an error is detected in the operation of the Character Counter or the Data-Band Counter in the addressed Disc Memory.

c. Master Error — The Master Error indicator light is ON when a Disc Memory Interface Error signal exists.

d. Ignore Error — The Ignore Error indicator light is ON when the Disc Memory Ignore Interface Error toggle is set ON.

#### 13. 3. 1. 6 UNI-RECORD EQUIPMENT STATUS AND ERROR INDICATORS.

a. Status:

(1) Oper — The Oper (operative) indicator light is ON when the addressed Uni-Record device is operative.

(2) Busy — The Busy indicator light is ON when the addressed Uni-Record device is off-line, or when the last on-line operation is being sequenced off.

(3) On-Line — The On-Line indicator light is ON when the addressed Uni-Record device is operating with the Central Processor (on-line).

(4) Start Tape — The Start Tape (start of tape) indicator light is ON when the addressed magnetic tape is at the beginning of the reel.

(5) End Tape — The End Tape (end of tape) indicator light is ON when the addressed magnetic tape is at the end of the reel.

(6) Wr Inh — The Wr Inh (write inhibit) indicator light is ON when the Tape Unit Write Inhibit toggle is set ON in the addressed magnetic tape.

(7) Inc Xfer — The Inc Xfer (incomplete transfer) indicator light is ON when an incomplete tape record or partial card content is transferred to the Central Processor under control of the Input/Output Counter.

(8) File Mark — The File Mark indicator light is ON when the addressed magnetic tape has stopped at a file mark.

b. Errors:

(1) On-L Par — The On-L Par (on-line parity) indicator light is ON when a parity error is detected by the addressed Uni-Record device (Tape Transport, Card Reader/Punch and Line Printer).

(2) Eavesdrop — The Eavesdrop indicator light is ON when the addressed Tape Transport detects an eavesdrop parity error in the data being recorded after it was written.

(3) On-L Mech — The On-L Mech (on-line mechanical) indicator light is ON when the addressed Uni-Record Device encounters a mechanical error that disrupts the Input or Output operation.

(4) M Reg — The M Reg (M register) indicator light is ON when a parity error is detected in a word of data being transferred to or from Core Memory through the M register.

(5) Inst — The Inst (instruction) indicator light is ON when the addressed Uni-Record Device does not recognize the instruction code (invalid), or when a Tape Transport operation specifies drive in conflict with start or end of tape indications.

(6) O'Load — The O'Load (overload) indicator light is ON when the data handling capacity of the addressed Uni-Record Device is exceeded by the output message transfer being performed.

(7) Oper Add — The Oper Add (operand address) indicator light is ON when an attempt is made to access an unavailable Core Memory location.

(8) Of-L Par — The Of-L Par (off-line parity) indicator light is ON when the Tape Transport or Card Reader detects a data error after releasing the Central Processor from the previous data transfer.

(9) Horiz Check — The Horiz Check (horizontal check) indicator light is ON when the Tape Transport detects an invalid horizontal check character during execution of an Input instruction.

(10) Of-L Mech — The Of-L Mech (off-line mechanical) indicator light is ON when the addressed Tape Transport or Card Reader detects a mechanical error after releasing the Central Processor from the previous data transfer.

(11) Device Unavail — The Device Unavail (device unavailable) indicator light is ON if an addressed device is off-line or inoperative.

c. Master Error — The Master Error indicator light is ON when a Uni-Record Interface Error signal exists.

d. Ignore Error — The Ignore Error indicator light is ON when the Uni-Record Ignore Interface Error toggle is set ON.

#### 13.3.1.7 UNI-RECORD INTERFACE INDICATORS.

a. Master Interrupt — The Master Interrupt indicator light is ON when either the Not Busy or Operator interrupt conditions exist.

b. Ignore Interrupt — The Ignore Interrupt indicator light is ON when the corresponding toggle is ON, preventing the Master Interrupt condition from causing a program interrupt.

c. Detailed Interrupts:

(1) Not Busy — The Not Busy Interrupt indicator light is ON when the Uni-Record Interface has completed an operation and is ready for another.

(2) Operator — The Operator Interrupt indicator light is ON when an operator has depressed the Operator Interrupt button located on the Control Console.

#### 13.3.2 CENTER SECTION INDICATORS.

The Center Section of the Central Processor Control Panel, figure 13-3, consists of the following indicators.

##### 13.3.2.1 INTERRUPTS INDICATORS.

a. Master:

(1) Any — The Any indicator light is ON when a Master Interrupt signal exists in the Central Processor.

(2) Error — The Error indicator light is ON when either an Interface or Computer Error Interrupt indicator is ON.

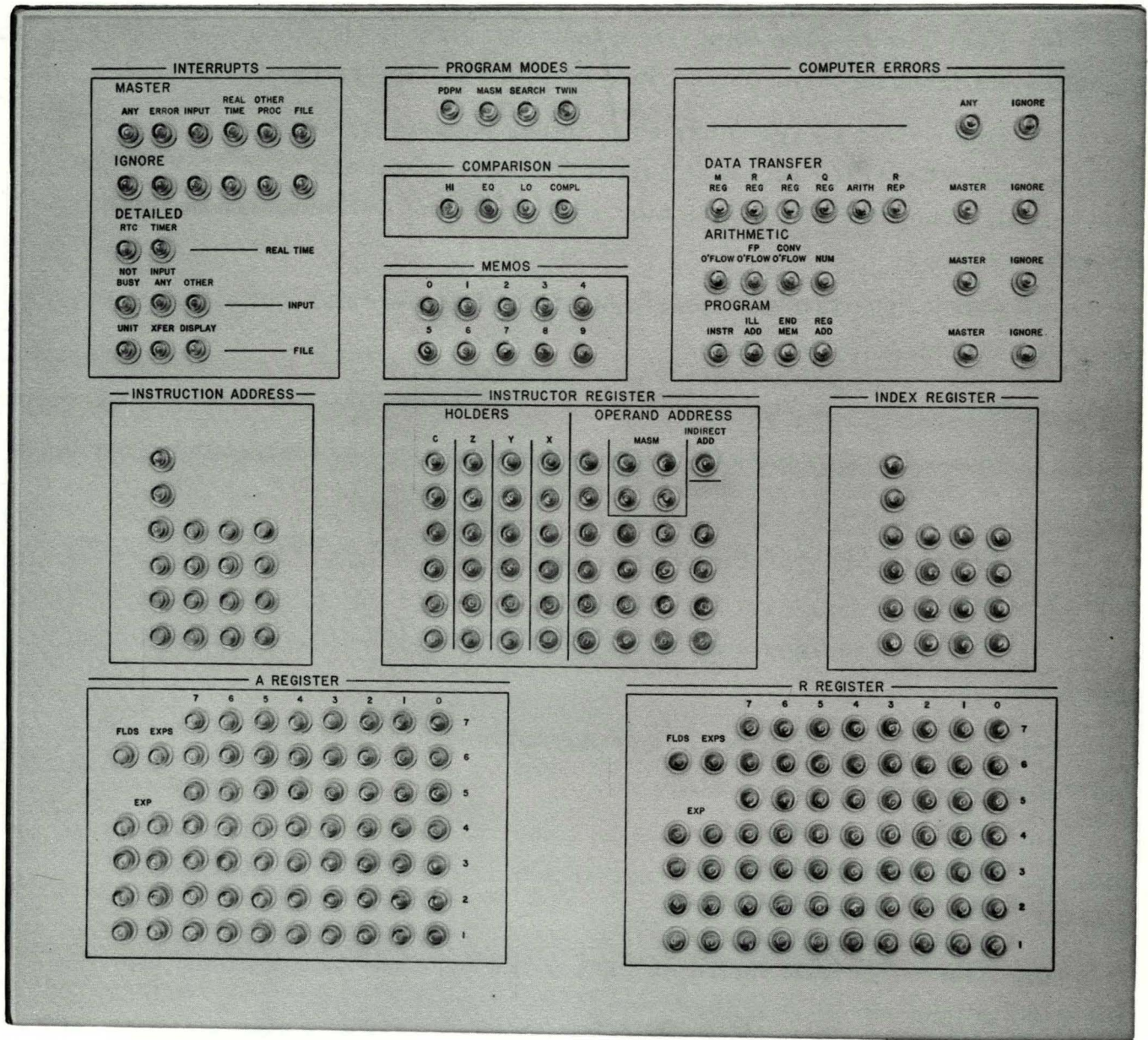


Figure 13-3. Central Processor Control Panel, Center Section

Section XIII  
Paragraph 13.3.2.1 (Contd.)

(3) Input — The Input indicator light is ON when an Interrupt signal exists in the I/O Interface.

(4) Real Time — The Real-Time indicator light is ON when an Interrupt signal exists from the Real-Time Clock or Interval Timer.

(5) Other Proc — The Other Proc (other processor) indicator light is ON when the other Central Processor sets it ON by program and generates an Interrupt signal. The Other Proc indicator also can be set ON by either Central Processor.

(6) File — The File indicator light is ON when an Interrupt signal is present from the Disc Memory Interface.

b. Ignore:

(1) Any — The Any indicator light is ON when the Any Master Interrupt is ignored.

(2) Error — The Error indicator light is ON when the Error Master Interrupt is ignored.

(3) Input — The Input indicator light is ON when the Master Input/Output Interrupt is ignored.

(4) Real Time — The Real Time indicator light is ON when the Master Real-Time Interrupt is ignored.

(5) Other Proc — The Other Proc (other processor) indicator light is ON when the Master Other Processor Interrupt in either Central Processor is ignored.

(6) File — The File indicator light is ON when the Disc Memory Interface Interrupt is ignored.

c. Detailed:

(1) Real Time:

(a) RTC — The RTC (real-time clock) indicator light is ON when the 24-hour Real-Time Clock passes through full scale and is reset to zero (00:00:00.000).

(b) Timer — The Interval Timer indicator light is ON when the Timer has counted down to zero.

(2) Input:

(a) Not Busy — The Not Busy indicator light is ON for any one of the following conditions: when an output from a Central Processor to a Sending L-119 Module or an Integrated Console Electronic Typewriter is terminated, or when an input from an L-119 Module or an Electronic Typewriter to a Central Processor is terminated.

(b) Input Any — The Input Any indicator light is ON for any one of the following conditions: when the I/O Interface Scanner detects a message available from a Receiving L-119 Module, or when the I/O Interface Scanner detects a message available from an Integrated Console device.

(c) Other — The Other indicator light is ON when a Sending L-119 Module has completed transmission of a message block to the Data Terminal Bay (DTB) and received acknowledgement.

(3) File:

(a) Unit — The Unit indicator light is ON when a Disc Memory operation is completed.

(b) Xfer — The Xfer (transfer) indicator light is ON when the data transfer through the Disc Memory Interface is completed.

(c) Display — The Display indicator light is ON when an output display message to a Console Printer (CP), Multi-Color (MC) Display, or a Large Panel (LP) Display is completed or aborted.

13.3.2.2 PROGRAM MODES INDICATORS.

a. PDPM — The PDPM (push-down, pop-up memory) indicator light is ON when the Central Processor is in Push-Down, Pop-Up Memory mode.

b. MASM — The MASM (multi-address stacking mode) indicator light is ON when the Central Processor is in Multi-Address Stacking mode.

c. Search — The Search indicator light is ON when the Central Processor is in the Search mode.

d. Twin — The Twin indicator light is ON when the Central Processor is in Twin mode.

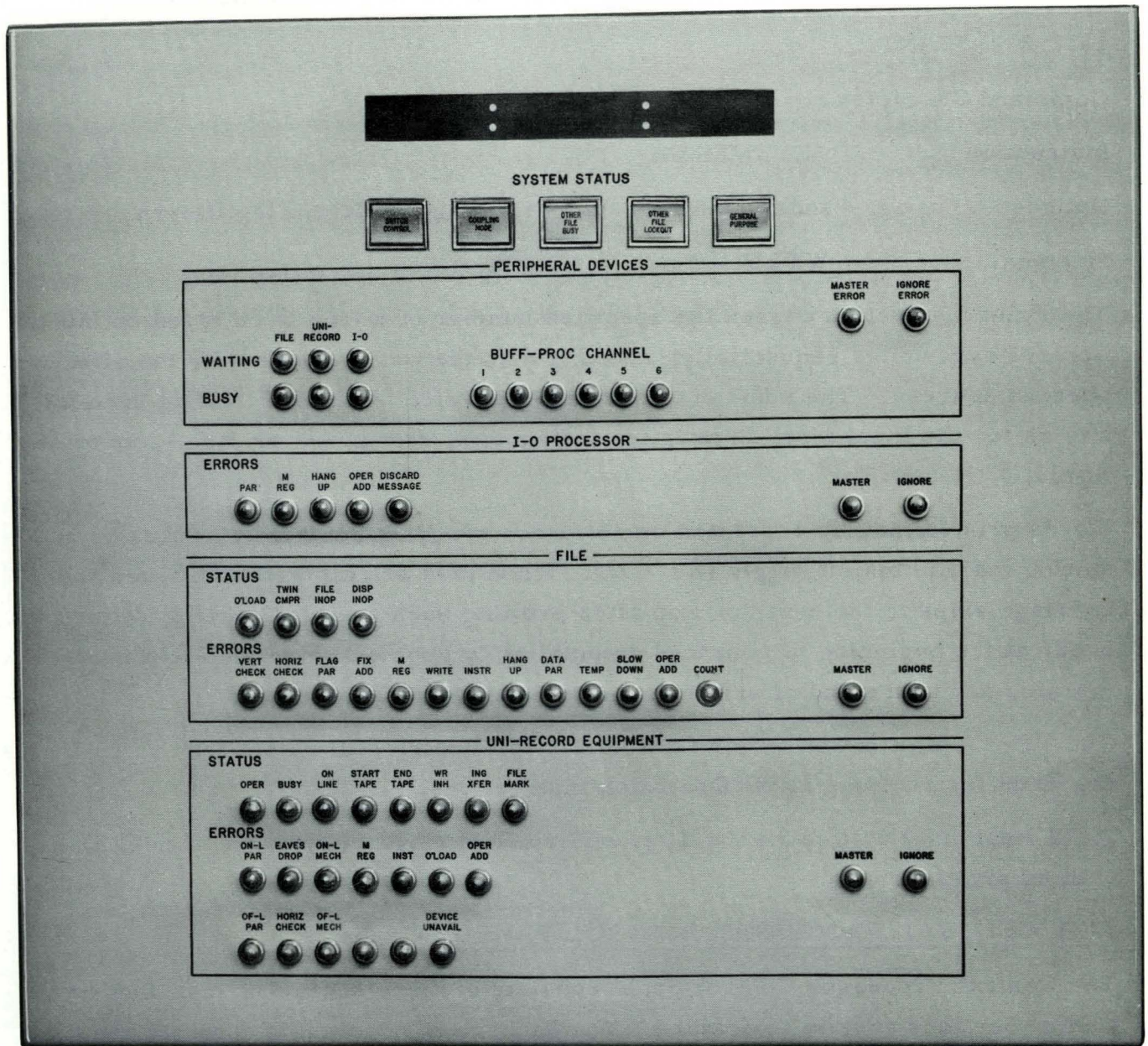


Figure 13-2. Central Processor Control Panel, Left Section

### 13.3.2.3 COMPARISON INDICATORS.

a. Hi — The Hi (high) indicator light is ON when the value of the Index register or A register content is greater than the content of the addressed Core Memory location.

b. Eq — The Eq (equal) indicator light is ON when the value of the Index register or A register is equal to the content of the addressed Core Memory location.

c. Lo — The Lo (low) indicator light is ON when the value of the Index register or A register content is less than the content of the addressed Core Memory location.

d. Compl — The Compl (complement) indicator light is ON when the value of the Index register or A register is equal to the complement of the content of the addressed Core Memory location.

13.3.2.4 MEMOS INDICATORS. The Memos indicators display the content of the Memo toggles. They are numbered zero through nine, corresponding to the associated toggle. An indicator light is ON or OFF indicating whether or not the toggle is ON or OFF.

### 13.3.2.5 COMPUTER ERRORS INDICATORS.

a. Any — The Any indicator light is ON when an error signal exists in the Data Transfer, Arithmetic, or Program subclass.

b. Ignore — The Ignore indicator light is ON when the Ignore Any Computer Error toggle is ON.

c. Data Transfer:

(1) Master — The Master indicator light is ON when any of the six detail data transfer error indicators is ON.

(2) Ignore — The Ignore indicator light is ON when the Master Data Transfer Error Interrupt is inhibited.

(3) M Reg — The M Reg (M register) indicator light is ON when the M register content contains incorrect parity.

(4) R Reg — The R Reg (R register) indicator light is ON when the R register detects incorrect parity in a data transfer.



(5) A Reg — The A Reg (A register) indicator light is ON when the A register detects an incorrect parity in a data transfer.

(6) Q Reg — The Q Reg (Q register) indicator light is ON when the Q register detects incorrect parity in a data transfer.

(7) Arith — The Arith (arithmetic) indicator light is ON when the Adder-Subtractor detects incorrect parity.

(8) R Rep — The R Rep (R repeater) indicator light is ON when the R repeater contains incorrect parity.

d. Arithmetic:

(1) Master — The Master indicator light is ON when any one of four detail arithmetic error toggles is ON.

(2) Ignore — The Ignore indicator light is ON when the Ignore Master Arithmetic Error Interrupt toggle is ON.

(3) O'Flow — The O'Flow (overflow) indicator light is ON when a register overflow exists due to a division or an addition.

(4) FP O'Flow — The FP O'Flow (floating-point overflow) indicator light is ON when the exponent in a floating-point operation exceeds  $\pm 99$ .

(5) Conv O'Flow — The Conv O'Flow (convert overflow) indicator light is ON when an attempt is made to convert (P instruction) a number from decimal to the binary, which has a decimal value greater than 64.

(6) Num — The Num (non-numeric) indicator light is ON when an attempt is made to perform an arithmetic operation on non-numeric data (when the binary coded decimal (BCD) value of a character exceeds 9).

e. Program:

(1) Master — The Master indicator light is ON when any one of four detail program error indicators is ON.

(2) Ignore — The Ignore indicator light is ON when the Ignore Program Error Interrupt toggle is ON.

(3) Instr — The Instr (instruction) indicator light is ON when a bad parity in the instruction, unrecognized order code, or conflicting MASM modifier bits is detected.

(4) Ill Add — The Ill Add (illegal address) indicator light is ON when an attempt is made to access Core Memory beyond the available range of addresses.

(5) End Mem — The End Mem (end of memory) indicator light is ON when an attempt is made to sequentially transfer data into Core Memory beyond address 63,999.

(6) Reg Add — The Reg Add (register address) indicator lights when the ZY character of an instruction is given an unassigned address.

13.3.2.6 INSTRUCTION ADDRESS INDICATORS. The Instruction Address indicators display the content of the Instruction Address register. Columns of indicators represent characters with the least significant bit at the bottom and the least significant character to the right. The ON or OFF state of an indicator is determined by the Core Memory address from which the next instruction will be retrieved.

13.3.2.7 INSTRUCTION REGISTER INDICATORS. The Instruction register indicators display the content of the Central Processor Instruction register. An indicator light will be ON or OFF to denote the state of the register element it is representing. The Central Processor mode of operation at any given time determines the state of the register elements.

a. Holders:

(1) C Holder — The C Holder indicators display the content of the Command (C Holder) register. An indicator will be ON or OFF, as determined by the command being executed.

(2) Z Holder — The Z Holder indicators display the content of the Device and Field Select (Z Holder) register. An indicator will be ON or OFF, as determined by the Z field of the instruction being executed.

(3) Y Holder — The Y Holder indicators display the content of the Device and Field Select (Y Holder) register. An indicator will be ON or OFF, as determined by the Y field of the instruction being executed.

(4) X Holder — The X Holder indicators display the content of the Index Modification (X Holder) register. An indicator will be ON or OFF, as determined by the X field of the instruction being executed.

b. Operand Address — The Operand Address indicators display the content of the Operand Address register. An indicator will be ON or OFF, as determined by the operand address of the instruction being executed.

(1) MASM — The MASM (multi-address stacking mode) indicator lights are ON when an instruction is being executed in the MASM mode.

(2) Indirect Add — The Indirect Add (indirect address) indicator light is ON to specify that the operand address is to be obtained from the memory address in the operand address portion of the instruction.

13. 3. 2. 8 INDEX REGISTER INDICATORS. The Index Register indicators display the content of the Index Modify (B) register. Columns of indicators represent characters with the least significant bit at the bottom and the least significant character to the right. An indicator will be ON or OFF as determined by the content of the last Index register used.

13. 3. 2. 9 A REGISTER INDICATORS. The A Register indicators display the content of the A register. Columns of indicators represent characters with the least significant bit at the bottom and the least significant character to the right. An indicator will be ON or OFF, as determined by the history of the register and the instruction being executed. The bits forming the exponent and signs of a floating-point number are repeated to the left of the main display.

13. 3. 2. 10 R REGISTER INDICATORS. The R Register indicators display the content of the R register. Columns of indicators represent characters with the least significant bit at the bottom and the least significant character to the right. An indicator will be ON or OFF, as determined by the history of the register and the instruction being executed. The bits forming the exponent and signs of a floating-point number are repeated to the left of the main display.

### 13. 3. 3 RIGHT SECTION CONTROLS AND INDICATORS.

The Right Section of the Central Processor Control Panel, figure 13-4, is primarily a maintenance panel that contains the following controls and indicators.

13. 3. 3. 1 PHASE INDICATORS. The Phase indicators (29) denote the operational steps through which the Central Processor passes during the execution of an instruction. Only one indicator light is ON at any given time.

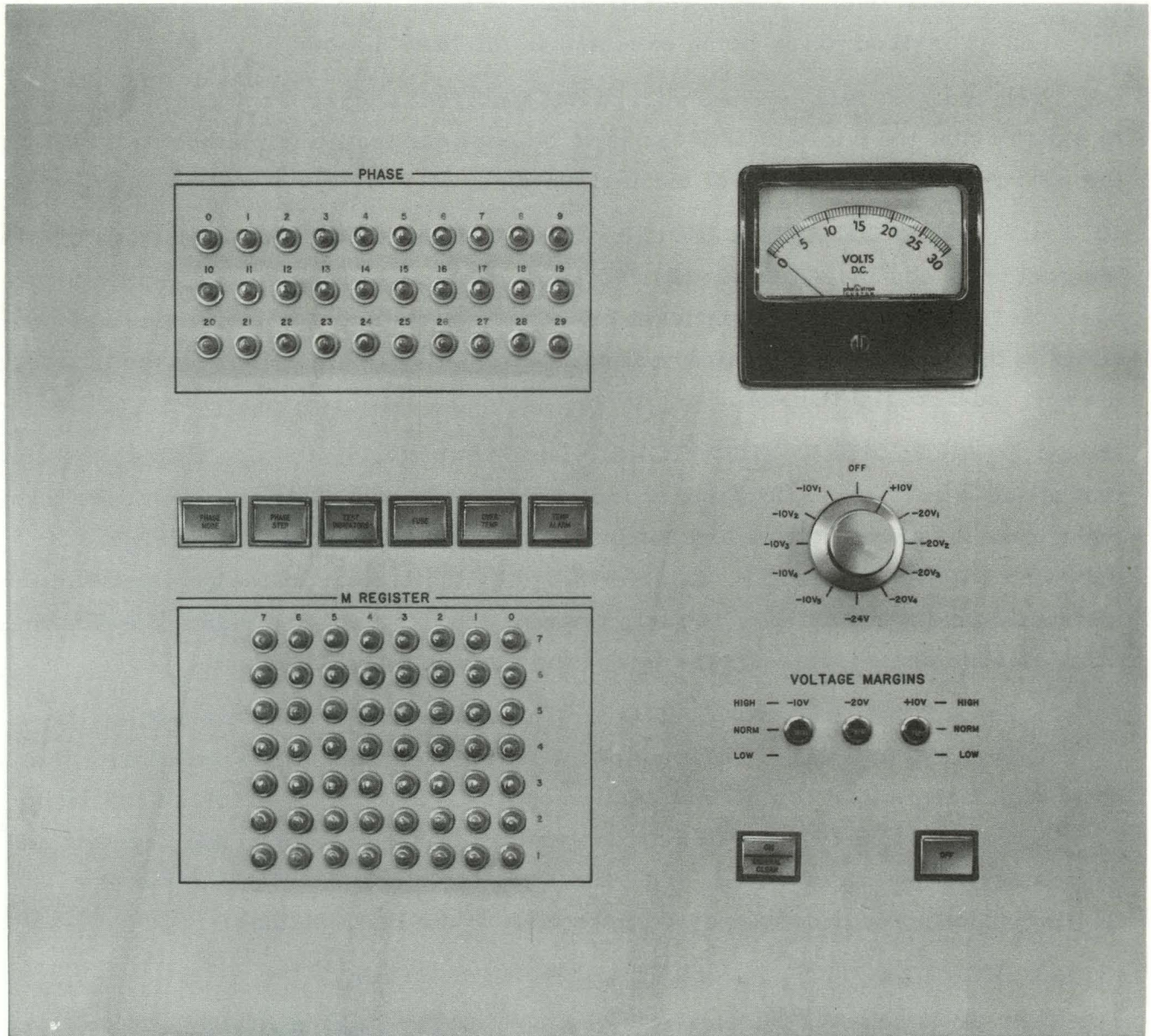


Figure 13-4. Central Processor Control Panel, Right Section

13.3.3.2 PHASE MODE BUTTON. This locking indicator switch causes the Central Processor to operate in Phase mode. In Phase mode, execution of each instruction is suspended between each phase.

13.3.3.3 PHASE STEP BUTTON. This momentary make indicator switch causes the Central Processor to advance from one phase to the next when in Phase mode.

13.3.3.4 TEST INDICATORS BUTTON. This momentary make indicator switch applies voltage to all indicators on the Central Processor Control Panel. This switch tests for burned-out lamps.

13.3.3.5 FUSE INDICATOR. The Fuse indicator light is ON when a fuse is blown in the Central Processor.

13.3.3.6 OVER TEMP INDICATOR. The Over Temp (over temperature) indicator light is ON when the air temperature in the cabinet exceeds the preset safe operating limit.

13.3.3.7 TEMP ALARM INDICATOR. The Temp Alarm (temperature alarm) indicator light is ON when the air temperature in the cabinet approaches the safe operating limit.

13.3.3.8 M REGISTER INDICATORS. The M Register indicators display the content of the Central Processor M register. Columns of indicators represent characters with the least significant bit at the bottom and the least significant character to the right. The ON or OFF state of an indicator is determined by the state of bits in the last Core Memory location which was operated upon.

13.3.3.9 POWER SUPPLY VOLTMETER. The Power Supply Voltmeter measures the output voltage of the power supply selected by the Power Supply Selector switch.

13.3.3.10 POWER SUPPLY SELECTOR SWITCH. The Power Supply Selector Switch is a sixteen position rotary switch. This switch connects the outputs of the Central Processor power supplies to the voltmeter.

13.3.3.11 VOLTAGE MARGINS SWITCHES. Three three-position switches used to monitor the High Margin, Low Margin, and Normal voltages of the power supplies.

13.3.3.12 ON/GENERAL CLEAR BUTTON. The On/General Clear button applies DC power to the electronic, logical, and power supply circuits of the Central Processor. This button also applies a DC voltage to return certain logical circuits to a

Section XIII  
Paragraph 13.3.3.13

predetermined state (General Clear). During operation of the Central Processor, the General Clear button can be used to return logic circuits to their initial state. The indicator light is ON when the button is depressed.

13.3.3.13 OFF BUTTON. The Off button removes DC operating voltage from the Central Processor electronic and logic circuits. When depressed, the indicator light is ON.

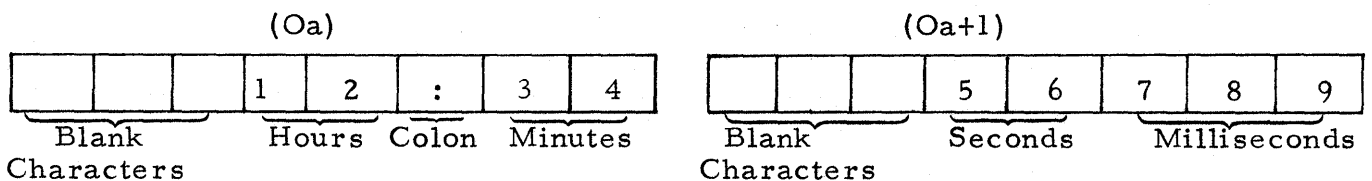
## SECTION XIV

### REAL-TIME CLOCK AND INTERVAL TIMER

The Real-Time Clock and the Interval Timer provide the Central Processor with an accurate time reference resolved to milliseconds. Either may be set or read at any time under program control.

#### 14.1 REAL-TIME CLOCK.

The Real-Time Clock may be loaded by a Load instruction and its content stored in memory by a Save instruction. For either the Load or Save instructions, reference is made to two successive memory locations starting at the effective operand address. The format is as follows:



The content of the blank character positions and the colon position do not affect the Real-Time Clock on a Load instruction, but are filled as shown by a Save instruction.

When the Real-Time Clock counts around to zero time, a Real-Time Clock Interrupt request is generated. Should the Central Processor access the Real-Time Clock as it is counting up (once per millisecond), a delay in instruction execution of up to 12 microseconds will occur. The Real-Time Clock content (exclusive of milliseconds) is displayed on the Central Processor Control Panel, figure 13-2.

#### 14.2 INTERVAL TIMER.

The Interval Timer has only a units-hours digit (an extra blank in the first word) and counts down instead of up, but is otherwise identical to the Real-Time Clock.

When the Interval Timer counts down to zero, an Interval Timer Interrupt request is generated.





## SECTION XV

## DISPLAY CONSOLES

The Data Processing System communicates on a real-time basis with a number of Display Consoles. These consoles are of three types designated the AN/FYA-2, AN/FYA-3 and AN/FYA-4, respectively. The three types differ principally in the aggregation of major functional components provided in each.

The AN/FYA-2 Console, Type A, contains all major components, these being:

- a. Electronic Typewriter (ET)
- b. Logic Keyboard Assembly (LKB)
- c. Multi-Color Display (MC)
- d. Console Printer (CP)
- e. Hard Copy Assembly (HC)

The AN/FYA-3 Console, Type B, contains items a, b, and c of the above. The AN/FYA-4 Console, Type C, contains items a and d.

The Data Processing System provides separate and distinct Interfaces for communication with each of the three major functional areas in the Display Consoles. These are:

- a. ET and Control Interface (including LKB)
- b. MC Display Interface (including Large Panel Displays)
- c. Console Printer Interface

For a technical description of the Interfaces and message formats, refer to Technical Memorandum TM-3511, 473L DPSS/ICSS Interface Description, Mitre Corp., 4 December 1962. The following paragraphs provide a brief description of the major functional areas of the Display Consoles. The Display Consoles are not part of the L-3055 System but are presented here to provide a better understanding of the L-3055 interface operation and capability.

## 15.1 ELECTRONIC TYPEWRITER (ET).

The ET display is a dynamic CRT display of alphanumeric data presented as 37 lines of 64 characters each in a 12-inch square area. The character set and coding is that for the L-3055 System. The CRT image, or page, is regenerated 60 times a second from internal Core Memory. There is additional storage capacity for two more pages, and any one of the three pages may be selected for viewing by the operator.

Operator controls include an alphanumeric keyboard with which to compose messages on the ET, and several control buttons and indicators, that interact with the L-3055 System through the I/O Interface.

The alphanumeric keyboard provides for composing and editing display messages by the console operator, off-line from the Central Processor. Messages may be any length up to a full page and are terminated by an EOM character. Normal typewriter functions are provided, such as space, backspace, and carriage return, as well as the additional capabilities of slue right, slue left, blank (erase), and erasure by overtyping. A visible cursor indicates the keyboard action position.

### 15.1.1 INPUT MESSAGES.

Some of the Display Console keys are used to transmit control messages, while others are used to transmit ET messages from the Display Console to the Central Processor. The key depression causes a Message Available signal to appear at the I/O Interface for selection by the Scanner. Details of the I/O Interface operation are given in Section V.

Control messages transmitted from the Display Console consist of four characters. The first character is a header character, designating the message as a control message and designating which of the ET display pages is being viewed. The second and third characters contain designation of the Display Console key and overlay designation, if the key is in the Logic Keyboard Assembly. The fourth character is an  $\neg$  (EOM) character. Provision is made to prevent any of the first three characters from being EOM. The EOM character is recognized in the I/O Interface as the termination of transmission.

ET messages transmitted from the Display Console consist of a variable number of characters from the display page being viewed. The first character of the message is a header character, designating the message as an ET message and designating

which of the display pages is being transmitted. The message terminates with the first EOM character, either from the display page, or automatically appended after the full page. (A full page is 2368 characters or 296 Central Processor words.)

Parity is checked both in the Display Console and in the Central Processor. Detection of bad parity terminates transmission, and sends an interrupt request to the Central Processor. The Display Console operator also is notified, and he attempts retransmission manually.

Message transmission is bit serial at a rate of 720,000  $\pm 15\%$  bits per second, as determined by the Display Console clock.

#### 15.1.2 OUTPUT MESSAGES.

The Central Processor program can initiate transmission of messages to the Display Console. Messages are either Control messages or ET messages, as distinguished by the first or header character.

Output Control messages consist of 16 characters including the header and last character, which is always an EOM character. The intermediate characters control various functions of the Display Console with specific bits. The format avoids the formation of an EOM character in the intermediate characters.

Output ET messages consist of a variable number of characters up to a full ET page, plus the header character. The message terminates on the first EOM character, as recognized by the ET. The header character is not part of the page that is displayed. If a full page is transmitted, the transmission terminates on the next character, whether or not it is an EOM.

The Display Console is always ready to receive an output message, unless it is not available; i. e., turned off, waiting with a message available or malfunctioning. In the case of an ET message, if the page being viewed is not cleared (all blank characters), a clear cycle, requiring 17 milliseconds, delays message transmission. Message transmission is bit serial at 720,000  $\pm 15\%$  bits per second, as determined by the Display Console clock.

Parity is checked in both the Display Console and in the Central Processor. Detection of bad parity, unless overridden, terminates transmission and sends an Interrupt request to the Central Processor. Retransmission of the message is a program function. When transmission of a message that contains a bad parity is desired, an

override is possible in both the I/O Interface and the Display Console. In the I/O Interface, the Ignore I/O Interface toggle, when ON, prevents bad parity from terminating message transfer. For the Display Console, special Control message and ET message header characters cause override of the parity check. The Display Console operator is notified when such a message is sent.

### 15.2 LOGIC KEYBOARD ASSEMBLY (LKB).

The LKB consists of an array of 34 control buttons on the Display Console with provision for an overlay carrying function labels. Each time a key is depressed, an input control message consisting of four characters is sent to the Central Processor. In the message, both the specific key and the specific overlay are identified.

The Central Processor can send 16 character output control messages that light up function labels for specific keys.

### 15.3 MULTI-COLOR DISPLAY (MC).

The MC presents images in color built up from alphanumeric characters and lines superimposed on a background slide. Display images are stored on photographic film and projected on a 12-inch square screen, which can display 37 lines of 64 characters each. Color is produced by using 3 color separation images, although only white and 3 other colors are intended to be used in the final displayed image.

Each character and the two ends of each line must have an XY coordinate position specified. The coordinates run from 0 to 1023 and are given as 10 bits in binary. With control, coordinates, and color specification, each displayed character and line-end requires 32 bits, which are carried in one Central Processor word. A special word serves to call up a background slide. An end of message (EOM) control code indicates transmission complete and initiates processing.

Display images are stored in the MC Buffer Store by the Central Processor for transmission, word by word, to the MC display, paragraph 9.3.4. As words are transmitted, the image is built up, character by character, on the film.

After exposure, the film is processed and moved to the projection station. While a new image is transmitted and exposed, the old image continues to be projected.

15.4 CONSOLE PRINTER (CP).

The Console Printer is an electric typewriter with a format similar to that of the Electronic Typewriter (ET); i. e., a 64 character line and message termination by an EOM character. The CP is used to make hard copy of ET display images. Messages are stored in the CP Buffer Store by the Central Processor for transmission, character by character, to the CP. Parity is checked by the CP, but no notification is given to the Central Processor of detected bad parity.

15.5 HARD COPY ASSEMBLY (HC).

The Hard Copy Assembly is a device that photographs an MC display image for a permanent record. The HC functions off-line in the Display Console.



## APPENDIX A

## AN/FYQ-11 CHARACTER SET

CENTRAL PROCESSOR CODE				LINE PRINTER & TYPEWRITER CHARACTERS	(1) FIELDATA CHARACTERS	(3) CARD READER/PUNCH ROW (S)
BITS	7	65	4321			
	1	11	0000	(Blank)	(Space)	(Blank)
	0	11	0001	A	A	12-1
	0	11	0010	B	B	12-2
	1	11	0011	C	C	12-3
	0	11	0100	D	D	12-4
	1	11	0101	E	E	12-5
	1	11	0110	F	F	12-6
	0	11	0111	G	G	12-7
	0	11	1000	H	H	12-8
	1	11	1001	I	I	12-9
	1	11	1010	)	)	12-8-4
	0	11	1011	. (Period)	.	12-8-3
	1	11	1100	(2) □	□	11-8-2
	0	11	1101	(2) (Underline)	(Underline)	12-8-6
	0	11	1110	(2) ≠	!	11-8-5
	1	11	1111	(2) ?	?	12-8-7
	0	10	0000	:(Colon)	:	12-8-5
	1	10	0001	J	J	11-1
	1	10	0010	K	K	11-2
	0	10	0011	L	L	11-3
	1	10	0100	M	M	11-4
	0	10	0101	N	N	11-5
	0	10	0110	O	O	11-6
	1	10	0111	P	P	11-7
	1	10	1000	Q	Q	11-8
	0	10	1001	R	R	11-9
	0	10	1010	(Minus)	(Minus)	11
	1	10	1011	*	*	11-8-4

Appendix A

CENTRAL PROCESSOR				LINE PRINTER	(1) FIELDATA	(3) CARD
CODE				& TYPEWRITER	CHARACTERS	READER/PUNCH
BITS	7	65	4321	CHARACTERS		ROW (S)
	0	10	1100	(2) ^ (Up)	(Upper case)	11-8-7
	1	10	1101	(2) \$	\$	11-8-3
	1	10	1110	(2) v (Down)	(Lower case)	12-8-2
	0	10	1111	(2) ⊕	⊕	0-8-5
	0	01	0000	+	+	12
	1	01	0001	/	/	0-1
	1	01	0010	S	S	0-2
	0	01	0011	T	T	0-3
	1	01	0100	U	U	0-4
	0	01	0101	V	V	0-5
	0	01	0110	W	W	0-6
	1	01	0111	X	X	0-7
	1	01	1000	Y	Y	0-8
	0	01	1001	Z	Z	0-9
	0	01	1010	(	(	0-8-4
	1	01	1011	(Comma)	(Comma)	0-8-3
	0	01	1100	(2) #	" (Quote)	0-8-2
	1	01	1101	(2) ° (Degree)	(Ignore)	6-8
	1	01	1110	(2) ;	;	11-8-6
	0	01	1111	(2) [	(Line Feed)	0-8-7
	1	00	0000	∅	∅	0
	0	00	0001	1	1	1
	0	00	0010	2	2	2
	1	00	0011	3	3	3
	0	00	0100	4	4	4
	1	00	0101	5	5	5
	1	00	0110	6	6	6
	0	00	0111	7	7	7
	0	00	1000	8	8	8
	1	00	1001	9	9	9
	1	00	1010	=	=	8-3
	0	00	1011	(Apostrophe)	(Apostrophe)	8-4
	1	00	1100	(2) >	>	8-2



CENTRAL PROCESSOR			LINE PRINTER	(1) FIELDATA	(3) CARD	
CODE			& TYPEWRITER	CHARACTERS	READER/PUNCH	
BITS	7	65	4321	CHARACTERS	ROW (S)	
	0	00	1101	(2) <	<	8-5
	0	00	1110	(2) ]	(Carriage Return)	0-8-6
	1	00	1111	(2) 7 (EOM)	(Master Space)	8-7

- NOTES: (1) As translated by the L-119 Module.  
(2) Low speed print characters (Line Printer).  
(3) This code is compatible with IBM 026 Type-H print set.



## APPENDIX B

## ZY ADDRESSABLE REGISTERS

(SAVE (\$), LOAD (0) and COPY (=) instructions)

NAME	MEMORY ADDRESS (if APPLICABLE)	ZY	APPLICABLE COMMANDS	REMARKS
Index Register #1	LSH of Address 0001	01	⊕, ⊕, =	
Index Register #2	LSH of Address 0002	02	S, ⊕, =	
Index Register #3	LSH of Address 0003	03	S, ⊕, =	
Index Register #4	LSH of Address 0004	04	S, ⊕, =	
Index Register #5	LSH of Address 0005	05	S, ⊕, =	
Index Register #6	LSH of Address 0006	06	S, ⊕, =	
Index Register #7	LSH of Address 0007	07	S, ⊕, =	
Index Register #8	LSH of Address 0008	08	S, ⊕, =	
Index Register #9	LSH of Address 0009	09	S, ⊕, =	
Index Register #10	LSH of Address 0010	0=	S, ⊕, =	
Index Register #11	LSH of Address 0011	0'	S, ⊕, =	

## Appendix B

NAME	MEMORY ADDRESS (if APPLICABLE)	ZY	APPLICABLE COMMANDS	REMARKS
Index Register #12	LSH of Address 0012	0>	S, ⊕, =	
Index Register #13	LSH of Address 0013	0<	S, ⊕, =	
Index Register #14	LSH of Address 0014	0]	S, ⊕, =	
Index Register #15	LSH of Address 0015	0⌒	S, ⊕, =	
Index Register #16		0+	S, ⊕, =	Numeric bits of 4 characters and bits 35 and 36.
Index Register #17		0/	S, ⊕, =	Numeric bits of 4 characters and bits 35 and 36.
Real-Time Clock		0S	S, ⊕, =	Numeric bits of 9 characters stored in 2 con- secutive add- resses as ---12:34 ---56789
Interval Timer		0T	S, ⊕	Numeric bits of 8 characters stored in 2 con- secutive add- resses as ---1:23 ---45678
A Register		0U	S, ⊕, =	
Instruction Address Register		0V	S, ⊕, =	
Operand Address Holder		0W	=	Operand Add- ress stored after modifi- cation.

NAME	MEMORY ADDRESS (if APPLICABLE)	ZY	APPLICABLE COMMANDS	REMARKS
Uni-Record Interrupt Entry Register	MSH of Address 0008	0H	\$, ⊕, =	
Disc Memory Inter- rupt Entry Register	MSH of Address 0009	0°	\$, ⊕, =	
Error Interrupt Entry Register	MSH of Address 0010	0X	\$, ⊕, =	
Input Interrupt Entry Register	MSH of Address 0011	0Y	\$, ⊕, =	
Real Time Interrupt Entry Register	MSH of Address 0012	0Z	\$, ⊕, =	
Other Processor Interrupt Entry Register	MSH of Address 0013	0(	\$, ⊕, =	
Interrupt Return Address Register	MSH of Address 0014	0;	\$, ⊕, =	
R Register		0,	=	
Uni-Record Interface Word Counter		0#	\$, ⊕	
Central Processor Return Address Register	MSH of Address 0001	11	\$, ⊕	
Disc Memory Inter- face Return Address Register	MSH of Address 0002	12	\$, ⊕	
Uni-Record Inter- face Return Address Register	MSH of Address 0003	13	\$, ⊕	
I/O Interface Return Address Register	MSH of Address 0004	14	\$, ⊕	
X Counter		15	\$	X Counter stored before modification

## Appendix B

NAME	MEMORY ADDRESS (if APPLICABLE)	ZY	APPLICABLE COMMANDS	REMARKS
Manual Switch Status		17	\$	
Disc Memory Inter- face Instruction Register		1+	\$	X and Z char- acters set to Ø
Uni-Record Inter- face Instruction Holder		1/	\$	X character Ø
I/O Interface Instruction Holder		IS	\$	Instruction re- constructed before storing.
Disc Memory Inter- face Instruction Location Register		IT	\$	Numeric Bits and bits 35 and 36.
Uni-Record Inter- face Instruction Location Register		IU	\$	Numeric Bits and bits 35 and 36.
I/O Interface Instruction Address Register		IV	\$	Numeric Bits and bits 35 and 36.
Input Scanner		IW	\$	Z and Y char- acters only.

## APPENDIX C

TOGGLES AND LINES AFFECTED BY  
TEST (T) AND SET (V) INSTRUCTIONS

	<u>BASIC ZY ADDRESS</u>	<u>NOTE*</u>	<u>PARAGRAPH</u>
MASTER INTERRUPT	40	1	11
MASTER INTERRUPT IGNORE	20		
I/O Interface Interrupt	42	2	11.7
I/O Interface Interrupt Ignore	22		
I/O Interface Not Busy	39	D	11.7.1
L-119 Message Available	2,	D	11.7.2
Display Console Message Available	2#	D	11.7.3
Display Console Message Available Ignore	70		
Output L-119 Available	2(	D	11.7.4
Real-Time Interrupt	43	2	11.8
Real-Time Interrupt Ignore	23		
Real-Time Clock	3(	D	11.8.1
Interval Timer	3,	D	11.8.2
Other Processor Interrupt	44	D	11.9
Other Processor Interrupt Ignore	24		
Disc Memory Interface Interrupt	3[	2	11.10
Disc Memory Interface Interrupt Ignore	25		
Disc Memory Interface Not Busy	2Y	D	11.10.1
Disc Memory Unit Not Busy	2Z	D	11.10.2
Display Depleted	35	D	11.10.3
Uni-Record Interface	3W	2	11.11
Uni-Record Interface Ignore	3X		
Uni-Record Interface Not Busy	3Y	D	11.11.1
Operator	3Z	D	11.11.2

\* Refer to Notes on page C-6

Appendix C

	<u>BASIC ZY ADDRESS</u>	<u>NOTE</u>	<u>PARAGRAPH</u>
Error Interrupt	41	1	11
Error Interrupt Ignore	21		
Computer Error Interrupt	U0	1	11.12
Computer Error Interrupt Ignore	2U		
Computer Data Transfer Error Interrupt	V0	2	11.12
Computer Data Transfer Error Interrupt Ignore	2V		
M Register	V1	D	11.12.1
R Register	V2	D	11.12.2
A Register	V3	D	11.12.3
Q Register	V4	D	11.12.4
Arithmetic	V5	D	11.12.5
R Repeater	V6	D	11.12.6
Computer Arithmetic Error Interrupt	W0	2	11.13
Computer Arithmetic Error Interrupt Ignore	2W		
Overflow	W1	D	3.1.4, 11.13.1
Floating-Point Overflow	W2	D	3.1.4, 11.13.2, 4.6
Non-Numeric	W3	D	3.1.4, 11.13.3
Convert Overflow	W4	D	3.1.4, 11.13.4
Computer Program Error Interrupt	X0	2	11.14
Computer Program Error Interrupt Ignore	2X		
Instruction	X1	D	3.2.4, 11.14.1
Illegal Address	X2	D	3.2.1.2, 11.14.2
End of Memory	X3	D	3.2.1, 11.14.3
Register Address	X4	D	11.14.4
Interface Error Interrupt	+0	1	
Interface Error Interrupt Ignore	2+		
Disc Memory Interface Error Interrupt	/0	2	11.5
Disc Memory Interface Error Interrupt Ignore	2/		



	<u>BASIC ZY ADDRESS</u>	<u>NOTE</u>	<u>PARAGRAPH</u>
Vertical Parity Internal	/1	D	11.15.1
Horizontal Check Internal	/2	D	11.15.2
Flag Parity	/3	D	11.15.3
Fixed Address	/4	D	11.15.4
M Register	/5	D	11.15.5
Write	/6	D	11.15.6
Instruction	/7	D	11.15.7
Hang Up	/8	D	11.15.8
Transmission Parity	/9	D	11.15.9
Out of Temperature	/=	D	11.15.10
Slow Down	/'	D	11.15.11
Operand Address	/>	D	11.15.12
Count	36	D	11.15.13
I/O Interface Error Interrupt	30	2	11.16
I/O Interface Error Interrupt Ignore	2S		
Data Parity	S1	D	11.16.1
M Register Parity	S2	D	11.16.2
Hang Up	S3	D	11.16.3
Operand Address	S4	D	11.16.4
Uni-Record Interface Error Interrupt	T0	2	11.17, 7.6
Uni-Record Interface Error Interrupt Ignore	2T		
On-Line Data	T1	D	11.17.1
On-Line Eavesdrop	T2	D	11.17.2
On-Line Mechanical	T3	D	11.17.3
M Register Parity	T4	D	11.17.4
Horizontal Check	T5	D	11.17.5
Instruction	T6	D	11.17.6
Overload	T7	D	11.17.7
Data Link Parity	T8	D	11.17.8
Data Link Mechanical	T9	D	11.17.9
Device Unavailable	T>	D	11.17.10

Appendix C

	<u>BASIC ZY ADDRESS</u>	<u>NOTE</u>	<u>PARAGRAPH</u>
Operand Address	T<	D	11, 17, 11
<b>DISC MEMORY STATUS</b>			
Disc Memory Interface Busy	3#	1	9.7
Disc Memory Overload	//		9.7
Disc Memory Twin Compare	/S		9.7
Disc Memory Operative	/U		9.7
Disc Memory Display Backup	34		9.7
Disc Memory Unit Busy Status	37	3	9.7
<b>I/O INTERFACE</b>			
I/O Interface Busy	3°	1	
Step Scanner	38	3	5.3
<b>UNI-RECORD INTERFACE</b>			
Uni-Record Interface Busy	3;	1	7.6
Device Operative	45	1	
Device Busy	46	1	
Device On-Line	47	1	
Start of Tape	48	1	
End of Tape	49	1	
Write Inhibit	4=	1	
Transfer Stop in Record	4'	1	
Encountered File Mark	4>	1	
Even Parity	3>	1	
<b>CONTROL TYPEWRITER</b>			
Tab Inhibit	3=	4	13.2.2.2
<b>SYSTEM STATUS</b>			
Control Status	59	5	2.2.3;12.4.1
Request Memory Switch	31		2.2.3
Memory Switch Status	32	1	2.2.3
Disc Memory Trunkline Coupling Mode	5=	5	12.4.2
Other Disc Memory Busy	5'	5	12.4.3
Other Disc Memory Lockout	5>	5	12.4.4

	<u>BASIC ZY ADDRESS</u>	<u>NOTE</u>	<u>PARAGRAPH</u>
General Purpose	5<	5	
Interrupt Other Processor	3'	3	11.9
<b>CENTRAL PROCESSOR MODES</b>			
Auto Error Mode	57		
MASM Mode	50		3.5
Count +16	5(		3.5
Count -16	5,		3.5
Count +17	5#		3.5
Count -17	5°		3.5
PDPM Mode	51		3.4
Search Mode	52		4.3
HI Comparison	55		4.3
LO Comparison	56		4.3
EQ Comparison	57		4.3
Equal to Complement	58		4.3
Core Memory Twin Mode	53		2.2.5
TAG SWITCH	5J	1	2.4.5.4;13.1
BREAKPOINT 1	5/	1	3.2.10;13.1
2	5S	1	3.2.10;13.1
3	5T	1	3.2.10;13.1
4	5U	1	3.2.10;13.1
5	5V	1	3.2.10;13.1
6	5W	1	3.2.10;13.1
7	5X	1	3.2.10;13.1
8	5Y	1	3.2.10;13.1
9	5Z	1	3.2.10;13.1
PROGRAM MEMO 0	4+		3.2.10
1	4/		3.2.10
2	4S		3.2.10
3	4T		3.2.10
4	4U		3.2.10
5	4V		3.2.10

	<u>BASIC ZY ADDRESS</u>	<u>NOTE</u>	<u>PARAGRAPH</u>
PROGRAM MEMO 6	4W		3.2.10
7	4X		3.2.10
8	4Y		3.2.10
9	4Z		3.2.10
Any	4(		3.2.10

Where no Note is given, a toggle is designated that can be set and reset by a V instruction and tested true or false by a T instruction.

NOTE 1: Designates a composite signal that can be tested true or false by a T instruction.

NOTE 2: Designates a composite signal that can be tested true or false by a T instruction. In addition, a V instruction will reset (but not set) all constituent Detail Error toggles.

NOTE 3: Designates a function that can be set true (but not reset) by a V instruction.

NOTE 4: Designates a function that can be set true or reset false by a V instruction.

NOTE 5: Designates a status function that can be tested true or false by a T instruction. The function can be set true or reset false by a V instruction only by the Central Processor that has Switch Control Status true (ZY = 59).

NOTE D: Designates a Detail Error toggle that can be set or reset by a V instruction and tested true or false by a T instruction.

## APPENDIX D

## LAP MNEMONICS

<u>MNEMONIC</u>	<u>C</u>	<u>Z</u>	<u>Y</u>	<u>PARAGRAPH</u>	<u>MNEMONIC</u>	<u>C</u>	<u>Z</u>	<u>Y</u>	<u>PARAGRAPH</u>
AND	^	∅	∅	4.4.4	FSB	S	∅(Blank)		4.6.2
ADD	A	∅	∅	4.1.1	FMLT	M	∅(Blank)		4.6.5
ADM	A	+	∅	4.1.1	FDIV	D	∅(Blank)		4.6.6
ADVF	)	∅	:	8.3.3	FRAD	+	∅(Blank)		4.6.3
ADVR	)	∅	∅	8.3.3	FMIN	-	∅(Blank)		4.6.4
BPHTR	Z	5	+	4.5.7	FLAG	<	1	∅	10.2.4
BRING	B	∅	∅	4.2.1	FNOP	[	1	∅	10.5.1
BKSF	(	∅	:	8.3.3	FADM	A	+(Blank)		4.6.1
BKSR	(	∅	∅	8.3.3	FSBM	S	+(Blank)		4.6.2
CXM	Y	∅	∅	4.3.4	FMLTM	M	+(Blank)		4.6.5
COMPA	E	∅	∅	4.3.1	FDIVM	D	+(Blank)		4.6.6
COMPN	Q	∅	∅	4.3.2	FRADM	+	+(Blank)		4.6.3
COMPNM	Q	+	∅	4.3.2	FMINM	-	+(Blank)		4.6.4
CBD	P	∅	∅	4.4.5	HTR	Z	5	[	4.5.7
CDB	P	∅	:	4.4.5	HOLD	H	∅	∅	4.2.3
COMPF	Q	∅	:	4.6.7	JUMP	U	∅	∅	4.5.2
COMPFM	Q	+	:	4.6.7	JERK	J	∅	∅	4.2.4
COMB	C	∅	∅	4.2.2	KEEP	K	∅	∅	4.2.5
CPYRG	=	∅	∅	4.2.8	LDXR	⊕	∅	∅	4.2.6
CPOUT	#	∅	∅	10.4.2	LMULT	M	∅	∅	4.15
CLRA	B	∅	∅	4.2.1	LMLTM	M	+	∅	4.15
DIV	/	∅	∅	4.1.8	LDIV	D	∅	∅	4.17
DIVM	/	+	∅	4.1.8	LDIVM	D	+	∅	4.17
DSKFIL	F	∅	∅	9.2.1	LEFT	L	∅	∅	4.2.9
DISOUT	°	∅	∅	10.4.1	LLFT	L	+	∅	4.2.9
FCOMP	:	∅	∅	4.3.3	LRGT	R	+	∅	4.2.10
FAD	A	∅(Blank)		4.6.1	LDRG	⊕	∅	∅	4.2.6

## Appendix D

<u>MNEMONIC</u>	<u>C</u>	<u>Z</u>	<u>Y</u>	<u>PARAGRAPH</u>	<u>MNEMONIC</u>	<u>C</u>	<u>Z</u>	<u>Y</u>	<u>PARAGRAPH</u>
LDKM	^	∅	∅	10.5.2	REW	[	∅	∅	8.3.3
MULT	G	∅	∅	4.1.6	RDT	I	∅	∅	8.2.1
MULTM	G	+	∅	4.1.6	RDTXR	I	∅	:	8.2.1
MINUS	-	∅	∅	4.1.4	RDCD	I	∅	∅	8.8.2
MINM	-	+	∅	4.1.4	RDCDXR	I	∅	:	8.8.2
NOP	N	∅	∅	4.5.1	RCLC	+	∅	∅	10.6.2
OR	∨	∅	∅	4.4.3	RCLI	>	∅	∅	10.6.1
OBS	1	1	∅	10.1.2	RECAL	;	∅	∅	10.6.3
OBBK	9	1	∅	10.2.3	SVXR	\$	∅	∅	4.2.7
OBRK	5	1	∅	10.3.2	SET	V	:	∅	4.4.1
PRTCTL	O	∅	:	8.10.1	SBA	#	+	∅	4.4.2
PRNT	O	∅	∅	8.10.1	SUB	S	∅	∅	4.1.2
PCHCTL	O	∅	:	8.9.2	SUBM	S	+	∅	4.1.2
PNCH	O	∅	∅	8.9.2	SSP	#	∅	6	4.4.2
PICK	2	1	∅	10.1.1	SSM	#	+	6	4.4.2
PKBK	=	1	∅	10.2.1	SVRG	\$	∅	∅	4.2.7
PKRK	6	1	∅	10.3.1	TXD	□	∅	:	4.5.6
PKBF	]	1	∅	10.2.5	TXI	□	∅	∅	4.5.6
PKRF	S	1	∅	10.3.4	TION	T	∅	:	4.5.3
PURGE	3	1	∅	10.1.3	TIOF	T	∅	∅	4.5.3
PGBK	'	1	∅	10.2.2	TBAT	*	∅	∅	4.5.4
PGRK	7	1	∅	10.3.3	TCHM	*	+	∅	4.5.4
PGBF	⌋	1	∅	10.2.6	UFAD	A	∅	:	4.6.1
PGRF	T	1	∅	10.3.5	UFSB	S	∅	:	4.6.2
RESET	V	∅	∅	4.4.1	UFMLT	M	∅	:	4.6.5
RBA	#	∅	∅	4.4.2	UFDIV	D	∅	:	4.6.6
RAD	+	∅	∅	4.1.3	UFRAD	+	∅	:	4.6.3
RADM	+	+	∅	4.1.3	UFMIN	-	∅	:	4.6.4
RIGHT	R	∅	∅	4.2.10	UFADM	A	+	:	4.6.1
RCYCL	R	1	∅	4.2.10	UFSBM	S	+	:	4.6.2
RIO	I	∅	∅	5.10.1	UFMLTM	M	+	:	4.6.5
RTYP	>	V	∅	13.2.2.1	UFDIVM	D	+	:	4.6.6
RTYPC	>	∅	∅	13.2.2.1	UFRADM	+	+	:	4.6.3

<u>MNEMONIC</u>	<u>C</u>	<u>Z</u>	<u>Y</u>	<u>PARAGRAPH</u>	<u>MNEMONIC</u>	<u>C</u>	<u>Z</u>	<u>Y</u>	<u>PARAGRAPH</u>
UFMINM	-	+	:	4.6.4	WRITE	∅	1	∅	10.1.4
WIO	∅	∅	∅	5.10.2	WROB	4	1	∅	10.3.6
WTYP	<	∅	∅	13.2.2.2	WROBS	8	1	∅	10.2.7
WEF	∟	∅	∅	8.3.3	XECR	X	∅	∅	4.5.5
WRT	∅	∅	∅	8.3.1	XECA	X	+	∅	4.5.5
WRTCTL	∅	∅	:	8.3.2					





## APPENDIX E

## CENTRAL PROCESSOR FUNCTIONAL BLOCK DIAGRAM

Figure E-1 is a functional block diagram of the Central Processor. Eight character and four character parallel data transfer is represented by the ribbon flow paths. Character serial and bit serial data transfer is represented by the solid lines. Control lines are dotted. No attempt has been made to represent logic toggles and gates.

Physically the Central Processor contains Interface Sections that can perform data transfers while the Central Processor program continues. The operation of the Interface Sections is described in several sections of this manual. The Arithmetic and Program Control Sections are involved, at least in initial access and recognition, of all system instructions.

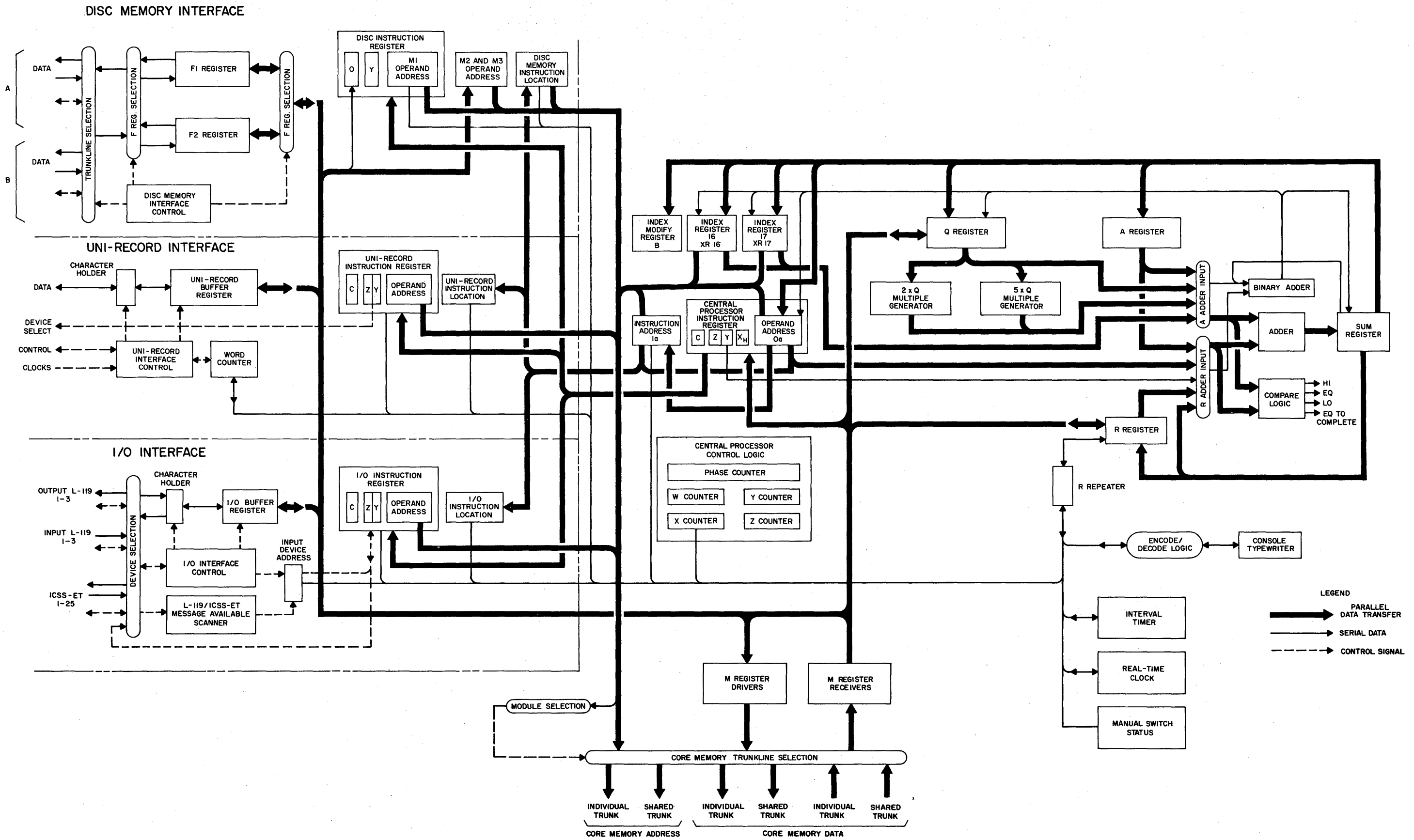


Figure E-1. Central Processor Functional Block Diagram



LIBRASCOPE DIVISION

 **GENERAL  
PRECISION**

INFORMATION SYSTEMS GROUP

808 WESTERN AVE • GLENDALE 1, CALIFORNIA