PRELIMINARY EDITION

Burroughs

B 1700 SYSTEMS

REFERENCE MANUAL

COPYRIGHT © 1972

Burroughs Corporation

Detroit, Michigan 48232

\$4.00

This $\underline{\text{reprint}}$ also includes the information released under the following:

PCN 1057155-001 (9/8/72)

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Table of Contents

Section	Title		Page
	INTRODUCTION ' . '	′.•′·•	×iii
1	SYSTEM DESCRIPTION '.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	′	.1-1.
	General (. / · / · . / · . / · . / · . / · . / · . / · . / · . / · . / · . / · . / · . / · . / · . / · . / · .	/ ·•/ ·• · ·	.1-1.
	System Configuration '.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	/ · • / · • · ·	1-1
	Interpretary / / / / / / / / / / / / / / / / / / /	/ • / · • · ·	1-/
	Interpreters and the control of the	, • , • .	1 0
	Defined Field Concent' '.'.'.'.'.'.'.	/[•/[•]	1 - 0
	Delayed Binding / 1/1/1/1/1/1/1/1/1/1/	/.*/.* .	.1-9
	Delayed Binding' '.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'	/.•/.• .	.1-9
	Processing Unit' \'\.'\.\'\.\'\.\'\.\'\.\'\.\\\\\\\\\\	/. /	1-9
	I/O Interface / ././././././././././././././././././	/ . /	1-9
	Port Interchange ' .'.'.'.'.'.'.'.'.'.'.'.'.'.	/./	1-9
	Memory Control Unit' .'''''''''''	′. <u>.</u> ′	1-9
	Control Memory / ./ ./ ./ ./ ./ ./////	′.•′.•	.1-10
	Central System Cabinet.	/·•//·• ·	.1-10
	Console Control Panel	<u></u>	1-10
	Console Display Lights /	, •, •	.1-10
	Register Display Indicators' .' .' .' . Power Indicator' .' .' .' .' .' .' .' .' .' .' .' .' .'	, •, · · ·	.1-10
	m.,,,, m.,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	, ,	
	RUN Indicator'	/. <u>*</u> /. <u>*</u> .	.1-10
	BOT indicator' .'''''''''''	/. /	.1-10
	PARITY Indicator ' .''''''''	/ . 	1-10
	Console Controls '.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	′ • .' • • •	1-12
	CLEAR Switch / ./ / / / / / /	'••'·•	.1-12
	HALT Switch' .'''''''''''''	′ • .′ • • •	.1-12
	LOAD Switch	′·•′·• ·	.1-12
	Data Entry Switches' .''''''''	′·•′·• ·	.1-12
	INT (Interrupt) Switch / ./ ./ ./ ./ ./	′·•′ ·• ·	1-12
	START Switch / ./ . / . / . / . / . / . / . / . / .	<u>`</u> .•', ·• ·	.1-12
	INC-A: Switch ' a' a	, •/, • ·	.1-12
	READ/WRITE Switch	• •	.1-12
	Register Column and Register Row Select Switches' .''''''''''''	, ,	1 12
	Control MODE Switch' .''''''	,	.1-13
	CASSETTE REWIND' .'''''''''	, • , · • .	.1-13
	CASSETTE ON/OFF' .''''''''''	/. * /. * .	.1-14
	Console Printer/Keyboard / ././././//	/ . * / . *	.1-14
	Main Memory .''''''''''''	/. . /	.1-14
	Addressing ' .''''''''''''	/ . .	.1-14
	Parity Checking' '.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.		
	Memory Data Transfer ' .'.'.'.'.'.'.'.'.'.'.'.'.		
•	Read Operation / ./ . / . / . / . / . / . / . / . / .		

Section Title		Page
1 (cont)	Field Length '.''''''''''''.	.1 - 15
2	B 1710 PROCESSORS' .'''''''''''''	.2 - 1 .2 - 1 .2 - 1
	Processing Unit' .'''''''''''''	2-1
	Micro-Instruction Address Register (MAR)'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	2-2
	Top of the Address Stack (TAS) Register'.'. General Purpose Registers (L, T, X, Y)'.'. L Register '.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	.2-22-32-3.
	L Register /	.2-32-4.
	FB Portion of the F Register ' .' .' .'	.2 -4 .2 -4
	Base and Limit Registers (BR, LR)' .' .' .' .'	2-6
	CA and CB Registers' .''''''''''''	2-6 2-6 2-6
	CD Register' .''''''''''''.	.2 -7 .2 -7 .
	Register / '''''''''''''''''''''''''''''''''''	.2-7
	Tape Cassette Input (U) Register ' .' .' .' DATA Register' .' .' .' .' .' .' .' .' .' .' .' .' .'	2-8
	Command (CMND) Register' .'''''''''''''	.2-8
	(24-Bit Function Box)' '.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'	.2 - 9.
	AND/OR/EXCLUSIVE-OR (XANY, XEOY, XORY) Result Registers '.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	.2-10
	Result Registers ' .' . ' . ' . ' . ' . ' . ' . ' . ' .	2-11

Section	Title	Page
2 (cont)	Registers' .'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	.2-11 .2-12 .2-12
3	General'	3-1-1-2-3-3 3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-
	(24-Bit Function Box)' .' .' .' .' .' .' .' .' .' .' .' .'	.3-12

Section	Title	Page
3 (cont)	SUM Result Register' .''''''''''''	.3 - 13 .3 - 14
	Result Registers / ./ .////////	.3-14
	Complement X/Complement Y (CMP-X, -CMPY)	
	Result Register' .' .' .' .' .' .' .' .' .' .' .' .' .'	.3-15
	Register / ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ .	.3 - 15
	Binary Conditions (BICN) Register' .' .' .' .	.3 - 15
	X/Y Conditions (XYCN) Register '.'.'.'.	.3-16
	X/Y States (XYST) Register '.'.'.'.'.'.'.'''''''.	.3 - 16
	Section (Four-Bit Function Box)	.3-17
	Interrupt Conditions (INCN) Register / ./ ./ ./	
4	MICRO-INSTRUCTIONS / .///////// .	4-1
4	General' .'''''''''''''	4 - 1
	Micro-Instruction Description a'	4-1
	Register Move (1NNN) / ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ ./	.4 -2
	Scratchpad Move (2NNN) / ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ .	.4 - 3 · .4 - 4 ·
	Four-Bit Manipulate (3NNN) / ./ ./ ./ ./ ./ ./ ./ ./ ./	4-4
	Bit Test/Branch on True (5NNN) ' .' .' .' .' .' .' .'	.4 -6
	Skip When (6NNN) ' . ' . ' . ' . ' . ' . ' . ' . ' . '	4 -7
	Read/Write Memory (7NNN)	4-9
	Move 8-Bit Literal (8NNN)/ **********************************	.4 - 1 0
	Shift or Rotate T Register Left (ANNN) / ////	4-12
	Extract From T Register (BNNN) / ./ ./ ./ ./ ./ ./ .	4-12
	Branch ' .' . ' . ' . ' . ' . ' . ' . ' . ' .	.4-13
	Branch Forward (CNNN)	4-13
	Four-Bit Manipulate (3NNN) ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '	.4 - 13
		. 7
	Call Forward (ENNN)	.4 - 1 4 .4 - 1 4 A
	Call Reverse (FNNN)	.4-14
	Clear Register (03NN)' .'''''''''''	
	Shift/Rotate X or Y (04NN) '.'.'.'.'.'.'.'.'.	4-16
	Shift/Rotate X and Y (05NN)' .'''''''''	4-17
	Count FA/FL (06NN) ' •' •' • ' • ' • ' • ' • ' • ' • ' •	.4 - 18
	Exchange Doublepad Word (07NN) (4-19
	Scratchpad Relate FA (08NN)/ .//////	
	Monitor (09NN) / .////////////	4-20 4-20
	Bias (003N)' .'''''''''''''	
	Store F in Doublepad Word (004N) '.'.'.'.'.''	
	load F from Doublepad Word (005N)' a' a' a' a' a'	4-22
	Set CYF (006N) / ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ .	4-23
	Set CYF (006N) '.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	4 = 24 L = 2L
	TOTAL LEVEL DESCRIPTION OF A CONTRACT OF A C	· · · · · · · · · · · · · · · · · · ·

Section	Title			
4 (cont)	Normalize X (0003) '.'''''''''''''	.4-25 .4-25		
5	B 1726 I/O SUBSYSTEM /	.5-1. .5-1. .5-1. .5-1. .5-2.		
6	I/O DESCRIPTORS' .' .' .' .' .' .' .' .' .' .' .' .' .'	6-1 6-1 6-1 6-1 6-1 6-1 6-2 6-2 6-2 6-3 6-3 6-4		
7	INPUT/OUTPUT CONTROLLERS /	.7 = 2 .7 = 2 .7 = 2 .7 = 3		

Section	Title	Page
7 (cont)	Buffered Line Printer Controller ' .' .' .' .' .' .' .'	· .7 = 7.
, , , , , , , , , , , , , , , , , , , ,	Line Printer Write Descriptor' .'.'.'.'.'.	7-7
	Buffered Line Printer Controller '.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	.7 -8
	Line Printer Test Descriptor '.'.'.'.'.	.7 -8
	Line Printer Result Descriptor (./ ./ ./ ./ ./ .	· .7 - 8.
	Console Printer (SPO) Controller '.''''	7-9
	console Printer Read Descriptor	· ./=10
	Console Printer Write Descriptor '.'.'.'.	7-10
	Console Printer Test Descriptor'	./-11
	Console Printer Result Descriptor	
	Reader Sorter Controller / ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ .	7-11
	Read Reader Sorter Descriptor	
	Batch Count Descriptor	
	Pocket Light Descriptor' .'.'.'.'.'.'.'.'.'.	.7-17
	Reader Sorter Test Descriptor	7-17
	Reader Sorter Result Descriptor' .' .' .' .' .	7-17
	Disk Pack Controller (B 1726 Only) ' .' .' .' .' .'	.7-18
	Disk Pack Write Descriptor (1/2/2/2/2/2/2/2	· 7-18
	Disk Pack Read Descriptor' .'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	7-20
	Disk Pack Initialize Descriptor	7-20
	Disk Pack Verify Descriptor' .'''''''''''''	./-21
	Disk Pack Test Descriptor ////////	• -/-22
	Disk Pack Result Descriptor' '.'.'.'.'.'.	·
	Disk Cartridge Controller / / / / / / / / / / / / / / / / / /	· 7=24
	Disk Cartridge Read Descriptor / 1/2/2/2/2/	7-25
	Disk Cartridge Read Descriptor '.'.'.'.'. Disk Cartridge Write Descriptor'.'.'.'.	7-25
	Disk Cartridge Test Descriptr' .'.'.'.'.	.7 - 26
	Disk Cartridge Result Descriptor ' .'.'.'.	7-27
	Magnetic Tape Controllers' .' .' .' .' .' .' .' .' .' .' .' .' .'	.7-28
	7-Track Magnetic Tape Controller /	.7-28
	9-Track Magnetic Tape Controller / ./ ./ ./	
	Magnetic Tape Read Descriptor	· .7 - 28
	Magnetic Tape Write Descriptor	
	Magnetic Tape Erase Descriptor / ./ ./ ./ ./ .	
	Magnetic Tape Rewind Descriptor' .''''	
	Magnetic Tape Space Descriptor ' .' .' .' .' .' Magnetic Tape Test Descriptor' .' .' .' .' .'	
	Magnetic Tape Lock Descriptor' .'''''	
	Magnetic Tape Result Descriptor' .'''.	
	Data Communications Controllers'	
	Single-Line Controller	
	Line Adapters for Single-Line Controls / ./	
	Synchronous Line Adapter / ./ ./ ./ ./ ./ ./ ./ ./ ./ ./	
	Asynchronous Line Adapter	
	Direct Asynchronous Line Adapter ' .''	
	Adapter Characteristics / ./ . / . / . / . / . / . / . / . / .	

Section	Title	Page
7 (cdnt)	Data Communications Read Descriptor'.'.'. Data Communications Write Descriptor '.'. Data Communications Break Descriptor '.'. Data Communications Test Descriptor'.'.'. Data Communications Result Descriptor'.'.	.7 - 38 .7 - 38 .7 - 39
APPENDIX A	REGISTER SUMMARY / .'''''''''''''.	.A - 1
APPENDIX B	HEXADECIMAL-DECIMAL CONVERSION TABLE	.B - 1 ·
APPENDIX C	SYSTEM COMPARISON .' .' .'''''''' .	.C - 1 :
APPENDIX D	CONVERSIONS' .'''''''''	.D – 1
APPENDIX E	MICRO-INSTRUCTIONS / .'''''''''''''	E-1
APPENDIX F	I/O RESULT DESCRIPTORS ' .''''''''' .	.F - 1 ·
	List of Illustrations	
Figure	Title	Page
1-1 1-2 1-3 1-4 1-5 2-1 2-2 2-3 2-4 2-5 2-6 2-7 2-8	B 1712/B 1714 Systems Configuration'	.1-2 .1-8 .1-8 .1-11 .2-2 .2-3 .2-4 .2-5 .2-5 .2-6

List of Illustrations (cont)

Figure	Title	Page
2-11 2-11 2-12 3-4 3-5 3-7 3-11 3-15 3-16 4-12 3-16 4-12 4-14 4-14 4-14 4-14 4-14 4-14 4-14	BICN Register' XYCN Register' XYST Register' Micro-Instruction Fetch and Execution' A Register / / / / / / / / / / / / / / / / / / /	2-12 -12 -13 -14 -15 -16 -17 -17 -17 -17 -17 -17 -17 -17
4-15 4-16 4-17 4-18 4-19 4-20 4-21 4-22 4-23 4-24 4-25 4-26 4-27 4-28 4-29	Swap Memory (02NN) '.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	

List of Illustrations (cont)

Figure	Title	Page
.4-30	Normalize X (0003) '.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	.4 - 25
.4 = 3.1	No Operation (0000)/ ///////////////////////////////	4-26
5-1	I/O Subsystem / -/ -/ -/ -/ -/ -/ -/ -/ -/ -/ -/ -/ -	5 - 1
6-1	I/O Descriptor Format' (-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	.6 - 1
6-2	Sample Chain of I/O Descriptors' / · / · / · / · / · / · / · / · / · /	6-2
.6 - 3	Typical Magnetic Tape Chain' . 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	.6 = 5
.6 - 4 ·	Stop I/O Descriptor	6 -5
.7 - 1		./ = .
.7 - 2·	Card Read Test Descriptor' .'''''''''''''	.7 -2.
7-3	Card Punch I/O Descriptor' .''''''''''''	.7 - 3
.7 -4.	Card Punch Test Descriptor / ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ .	.7 -4.
.7 - 5	Paper Tape Read, Space, and Backspace I/O	
	Descriptor ' . ' ·	
.7 - 6	Paper Tape Rewind I/O Descriptor	.7 - 5·
.7 - 7	Paper Tape Test Descriptor	.7 - 6
.7 - 8	Line Printer Write Descriptor	
.7 - 9	Line Printer Skip Descriptor / ./ ./ ./ ./ ./ ./ ./ ./ ./ ./ ./	
.7 - 10	Line Printer Test Descriptor '	.7 - 8:
.7 - 1.1	Console Printer Read Descriptor'	
.7 - 12	Console Printer Write Descriptor / ./ ./ ./ ./ ./ ./	
.7 - 1 3	Console Printer Test Descriptor'	.7 - 1.1
.7 - 14	Reader Sorter I/O Descriptor	.7 - 15
.7 - 15	PS (Pocket Select) Portion of Reader Sorter I/O	
.7 - 16	Descriptor '.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	.7 - 15
.7 - 16	Read Reader Sorter Descriptor	.7-16
.7 - 18	Pocket Light Descriptor' (-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/	.7 - 16
.7 - 19	Reader Sorter Test Descriptor (1/1/1/1/1/1/1/1/1/1/1/1/1/1/1/1/1/1/1/	.7 - 17 .7 - 17
7-20	Nisk Pack Write Descriptor / /././././././././.	.7 - 17
7-21	Disk Pack Read Descriptor / / / / / / / / / / / / / / / / / / /	7-19
7-22	Disk Pack Initialize Descriptor' (-'-'-'-'-'-'-'-'-'-'-	.7 - 20
7-23	Disk Pack Verify Descriptor' / / / / / / / / / / / / / / / / / / /	.7 - 2.1
.7 - 24	Disk Pack Relocate Descriptor' (-/-/-/-/-/-/-/-	.7 - 2 2
7 - 25	Disk Pack Test Descriptor' And Andrew Control of the Control of th	.7 - 23
.7 - 26	Disk Cartridge Controller' .'''''''''''''	.7-24
.7 - 27	Disk Cartridge Read Descriptor / . / . / . / . / . / . / . / . / . /	.7 - 25
.7 - 28	Disk Cartridge Write Descriptor' .'.'.'.'.'.'.'.'.'.'.	.7 - 26
.7 - 2.9	Disk Cartridge Test Descriptor / ./././././././././	7-27
.7 - 30	Magnetic Tape Read Descriptor' '.'.'.'.'.'.'.'.'.'.	7-28
.7 - 3·1	Magnetic Tape Write/Erase Descriptor' .'.'.'.'.	.7 - 2.9
.7 - 3:2	Magnetic Tape Rewind Descriptor' .'''''''''''''	7-29
7 - 33	Magnetic Tape Space Descriptor ' .' .' .' .' . ' . ' . ' . ' . ' . '	.7 - 3.0
7-34	Magnetic Tape Test Descriptor' .' .' .' .' .' .' .' .' .' .' .' .' .'	.7-30
· 7 – 35	Magnetic Tape Lock Descriptor' .' .' .' .' .' .' .' .' .' .' .' .' .'	7-31
.7 - 36	Data Communications Read Descriptor' .''''	.7-38
7-37	Data Communications Write Descriptor (/ / / / / / / /)	.7-38
. 7 - 3 8	Data Communications Break Descriptor ' .'.'.'	7-39

List of Illustrations (cont)

Figure	Title	Page
7 - 39	Data Communications Test Descriptor' .' .' .' .' .' .' .' .' .''	.7 - 39
	List of Tables	
Table	Title	Page
.1-1 .1-2 .1-3 2-1 .3-1 .4-1 .7-1 .7-2 .7-3	System Comparison'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.'.	.1-2 .1-4 .1-13 .2-13 .3-19 .4-2 .7-12 .7-13 .7-37

INTRODUCTION

The B 1700 series of computers are small-to-medium-scale systems which utilize the latest data processing concepts, such as defined fields delayed binding and interpretive structure. The technique of micro-programing is used to accomplish most data manipulation. This concept brings to Burroughs Corporation users a new level of computer responsiveness for business and scientific problems, as well as in data communications. This responsiveness is available to even the smallest organization that has a requirement for electronic data processing. Speed and flexibility are the key words which describe this system. To support this, Burroughs Corporation offers:

- a. Bit addressability. Main memory is addressable to the individual bit. It has no preferred word or byte boundaries which are visible to the rest of the system.
- b. Memory modularity is provided, from a minimum of 16K bytes to over 40K bytes of main memory on the B 1712, 64K bytes on the B 1714, and 98K bytes on the B 1726.
- c. Higher-level programing languages available are: COBOL, FORTRAN, BASIC, RPG, and a Master Control Program (MCP).
- d. Flexibility of the system lies in its ability to be microprogramed. Micro-instructions operate on a set of hardware registers available to the micro-programmer. Sets of microinstructions called interpreters are supplied by Burroughs Corporation.
- e. On the B 1726 System two types of memory are available; main memory and control memory. Control memory is up to four times faster than main memory.

SECTION 1

SYSTEM DESCRIPTION

GENERAL.

This section contains an overall description of the B 1700 series of data processing systems without explaining the specifics of the hardware. The systems are designed for modularity utilizing monolithic circuitry. The basic B 1710 System consists of a processing unit, main memory, peripheral I/O controllers, console control panel, power supplies, and cabinets. The basic B 1720 System consists of a processing unit, port interchange, main memory, control memory, peripheral I/O controllers, maintenance console, power supplies, and cabinets.

SYSTEM CONFIGURATION.

The equipment is modular and can utilize any peripheral units within the scope of the I/O controllers described in this manual. System configurations and comparisons are given in figures 1-1 and 1-2, and in tables 1-1 and 1-2.

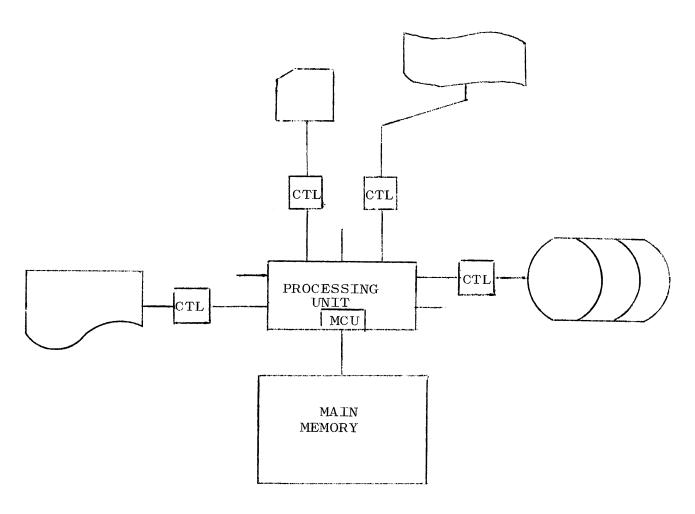


Figure 1-1. B 1712/B 1714 Systems Configuration

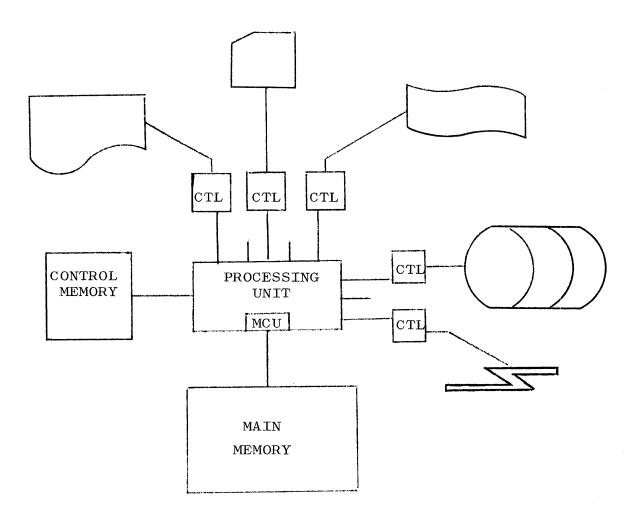


Figure 1-2. B 1726 System Configuration

Table 1-1
System Comparison

Characteristic	в 1712/в 1714	в 1726
Processors Clock rate Main memory Minimum size Maximum size Control Memory Size Port interchange Maximum number	1 2 MHz/4 MHz Yes 16,384 bytes 40,960/65,536 bytes No Not applicable No	1 6 MHz Yes 24,576 bytes 98,304 bytes Yes 4096 bytes Yes
of ports	Not applicable	2

Table .1-1 (cont)

System Comparison

Characteristic	B 1712/B 1714	B 1726
I/O controllers	Yes	Yes
Minimum number	1	1
Maximum number	8	8

Table .1-2.

System Configuration Chart

	B 1726	××××××××××××××××××××××××××××××××××××××
	В 1714	××× ××× ×× ×××××××××××××××××××××××××××
	B 1712	× × × × ××
3 0 0	Remarks	13 pocket 4/8/12/16 pocket 8 pocket 12 pocket 12 pocket 12 procket 132 prt. pos. 132 prt. pos. 132 prt. pos. 132 prt. pos. 132 prt. pos. 132 prt. pos. 132 prt. pos.
אשרכווו הסווו ושמו מרוסוו כוומן כ	Capabilities	80 col. 800 CPM 80 col. 1400 CPM 80 col. 300 CPM 96 col. 300 CPM 1000 DPM 600 DPM 600 DPM 600 DPM 600 DPM 700 LPM 700 LPM 700 LPM 700 LPM 700 LPM 700 LPM 750 LPM 750 LPM 750 LPM 750 LPM 750 LPM 750 LPM
	Description	Single Line Control Std. Asynch. Card Reader Card Reader Card Reader Card Reader Card Reader Reader Sorter Reader Sorter Reader Sorter Reader Sorter Reader Sorter Line Printer
	Unit Number	1351 9111 9112 9115 9116 9131-1 9135-2 9135-2 9136-6 9240-1 9240-1 9245-19 9245-19

Table .1-2 (cont)

System Configuration Chart

	Description	Capabilities	Remarks	B 1712	B 1714	B 1726
931.9-2.	Card Reader	•		×	×	×
93.19-4.	Punch High Speed	300/60 CPM			>	>
	Reader Punch	500/120 CPM			<	<
	Console Printer	10 CPS		×	×	×
9371-7	H-P-T DF Memory Bank	20ms avg access time; 231KB avg trsf rate.	7MB storage cap.	i		×
9371-14	H-P-T DF Memory Bank	40ms avg access time; 238KB avg trsf rate.	14MB storage cap.			×
9374-10	Add-on increment (for 9371-14)	40ms avg access time; 238KB avg trsf rate.	14MB storage cap.			×
9374-14	Add-on increment (for 9371-7)	20ms avg access time; 231KB avg trsf rate.	7MB storage cap.			×
9381-12	Magnetic Tape Unit	9 ch. 800 BPI	18KB - 2st.		×	×
9381-13	Magnetic Tape Unit	9 ch. 800 BPI NRZ	18KB - 3 st.		×	×
9381-14	Magnetic Tape	9 ch. 800 BPI	18KB - 4 s't.		×	×
9381-22	Magnetic Tape Unit	9 ch. 800 BPI NRZ	36KB - 2 st. Cluster		×	×
9381-23	Magnetic Tape	9 ch. 800 BPI	36KB - 3 st.		×	×
9381-24	Magnetic Tape Unit	9 ch. 800 BPI	36KB - 4 st.		×	×
93 90 - 3	Magnetic Tape	.7-channel tape	18/50 KC			×

Table .1-2 (cont)

System Configuration Chart

	Capabilities	Remarks	B 1712	B 1714	B 1726
1					
Reader Punch/	96 co'l.		×	×	×
	300/60 CFM 96 col.	6 stackers	×	×	×
	300/300/60 crm 2.3 megabyte	Single	×	×	×
	4.6 megabyte	Dual	×	×	×
	4.6 megabyte	Single	×	×	×
	9.2 megabyte	Dual	×	×	× :
	10KB Drive	2000	×	×	×
	95.5 megabyte	Dual			×

SYSTEM CONCEPT.

This series of computers can be described as systems that execute their programs under control of "micro" instruction's. It is impractical for the programmers to write micro-programs; therefore, various higher level languages have been developed for customer use on the system. It is not the intent of this section to describe these higher-level languages but to describe the relationship that exists between the micro-program and the higher-level language's.

S (secondary) language instructions are intermediate instructions which are equivalent to the machine language of a conventional system. For each S-instruction there exists a string of micro-instructions which interpretively execute the functions specified by that S-instruction.

It must be remembered that the S-language instruction does not directly cause the hardware to perform a function.

Because S-instructions are software defined by the sequence of micro-instructions, they may be as complex as the language designer requires. In most cases, S-instructions contain data addresses or base relative addresses, length of data fields, units of data, and an operation to be performed upon the data.

S-instructions may completely specify a compiler-level language. Burroughs Corporation has defined the S-instructions and has written the necessary interpreters and compiler programs for several compiler languages: COBOL, FORTRAN, RPG (Report Program Generator), and BASIC. These compiler programs generate the necessary S-language instructions to perform the various operations specified by the higher level language.

INTERPRETER'S.

In addition to the S-language program (compiled user program), another program which is referred to as an interpreter is utilized. An interpreter has been developed for each of the high-level S-languages: COBOL, FORTRAN, RPG, and BASIC. It is the function of the interpreter to fetch the S-language instruction(s) from main memory, and interpret or execute them. The S-instructions are decoded and a series of micro-instructions are executed to cause the hardware to perform the function specified by the S-language instruction. On completion of the execution of a series of micro-instructions (representing an S-instruction), the interpreter fetches the next S-language instruction and the operation continues in this manner. The series of micro-instructions to perform each S-language instruction may be stored in control memory, main memory, or be called from disk.

The S-language programs and required interpreters are located and loaded from disk under control of the Master Control Program (MCP). On completion of the load, the MCP then passes control to the S-language program. The MCP is written in either SDL or micro-code. The SDL version (MCP II) requires the SDL interpreter or a portion of it to always be present in the system. For COBOL programs a COBOL interpreter is called and is in memory with the SDL interpreter, when required. The memory resident portion of MCP I is micro-coded; there-

fore, only the interpreter required for the user program need be present.

INTERRUPT SYSTEM CONCEPT.

The interrupt system is described as a "soft" interrupt system. That is, any interrupt that occurs does not cause any specific hardware function and is only recognized by the software. On recognition of the interrupt by the software, there are no specific rules for handling the interrupt and it is left solely to the discretion of the programmer. The handling of these interrupts will be discussed in the MCP Reference Manual (not yet available).

DEFINED FIELD CONCEPT.

An integral part of the micro-programming technique is the defined field concept (bit addressability). This assumes that storage is described in units of bit's. It is a generalized scheme for handling field lengths ranging from one bit to a large number of bits through the use of a bit address and bit length, regardless of how fields may cross hardware "byte" boundarie's. Special hardware has been implemented to achieve bit addressability and variable length data fields within main memor'y. The hardware contains logic to adjust the address and length values in the addressing register's. Micro-instructions in turn are capable of setting the processor control circuits to operate on different units of data and iterating the micro-instruction sequence until the field in main memory is exhauste'd. Data units may vary from one bit to 15 bit's. Field length may vary from one to 65,535 bit's.

Example: Assume a Write command to main memory from bit address 2014 (hexadecimal) with a field length of 12 bit's. the data to be written is three decimal (4-bit) digits with a value of 345. A Write forward results in the "3" beginning at bit address 2014 (see figure 1-3').

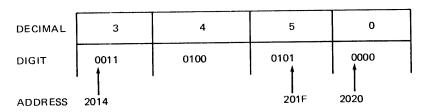


Figure 1-3. Write Forward

However, a Write reverse results in a "3" beginning at address 2008 (see figure 1-4%).

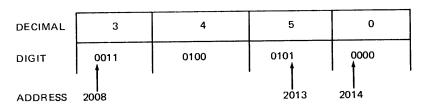


Figure 1-4. Write Reverse

Also note that a Write forward from bit address 2014 is equivalent to a Write reverse from bit address 2020.

DELAYED BINDING.

The noncommitment of resources until the actual need for these resources arises may be referred to as delayed binding. The proper utilization of resources enables the user to get the maximum system throughput per dollar investment.

Main memory is completely managed by the B 1700 Master Control Program. User code that is called into memory by the MCP may or may not be in one contiguous area. This allows for utilization of even small areas of memory.

Data space is within the area specified by the base and limit registers and is divided into two portions: overlayable and nonoverlayable. Program code is not in the area specified by the base and limit registers.

Compiler-generated code is completely independent of absolute memory locations and consequently is not restricted to particular areas of memory. Re-entrant code may exist; that is, one COBOL Compiler may be resident in memory for two or three compiles that are being accomplished, with each compile having a unique data area assigned to it.

Virtually infinite memory, proper control of input/output resources, bit addressability, and a flexible system architecture are other features which provide for proper utilization of resources on the B 1700 family of computers.

FUNCTIONS OF MAJOR UNITS.

PROCESSING UNIT.

The processing unit provides the control and combinatorial portion of the system along with various addressing and data manipulation registers.

I/O INTERFACE.

The I/O interface is connected to the processor and handles all communications between it and the I/O controllers on the B 1700 Systems. The I/O interface is capable of handling up to eight I/O controllers.

PORT INTERCHANGE.

The port interchange, if applicable, is the access path to main memory.

MEMORY CONTROL UNIT.

Main memory is connected to the memory control unit, which handles all memory access requests. The memory control unit resolves the addressing conflict between the bit-oriented data accesses and the byte-oriented parity checking. On the B 1710 Systems, the memory control unit is connected directly to the processor; on the B 1726, the memory control unit is connected to the port interchange which is connected to the processing unit.

CONTROL MEMORY.

Control memory, if present, is used to store the sequences of micro-instructions which perform macro-operations (COBOL and FORTRAN). Control memory is four times faster than main memory and is available in two sizes: 1024 or 2048 16-bit words. If sufficient control memory is not available, portions of main memory are utilized for storage of micro-instructions.

I/O CONTROLLER(S).

Each I/O controller controls the operation of the peripheral(s) connected to it. Each controller is a special-purpose device to interface with one type of peripheral. In most cases an I/O controller interfaces with a single device; however, in some cases a single I/O controller interfaces with two or more devices on a timed-shared basis (magnetic tape unit, disk files).

CENTRAL SYSTEM CABINET.

Connectable cabinets are used to contain all central system modules: memory, power supplies, and circulation fans.

CONSOLE CONTROL PANEL.

The console control panel is an integral part of the processor, consisting of a display panel plus various register- and data-manipulation switches. It is located on the front of the cabinet containing the processor. (See figure 1-5.)

CONSOLE DISPLAY LIGHTS.

REGISTER DISPLAY INDICATORS. These consist of a row of 24 lights which monitor the status of the 24-bit main bus of the processor. In the HALT state the register (or function) designated by the register select switches is displayed.

POWER INDICATOR. This switch/indicator is used to turn power on or off and to light the appropriate ON/OFF indicator.

RUN INDICATOR. This light indicates the processor is in the execute or fetch state, as opposed to the HALT or Read/Write main memory state.

STATE INDICATOR. This light, which is software-controlled, indicates that either control (MCP) or normal state operations are presently being executed. It is lit when the system is in control state.

BOT INDICATOR. This light indicates that the tape cassette is at the Beginning $\underline{0}$ f \underline{T} ape position. The light is under hardware control.

PARITY INDICATOR. This light indicates that an irrecoverable parity error has occurred while reading a tape cassette. The light is under hardware control.

CONSOLE CONTROLS.

CLEAR SWITCH. This is a pushbutton switch that clears the registers in the processor. On the B 1712 and B 1714, the switch clears the MAR(A), M, C, and U registers. On the B 1726, the switch clears all interrupts in the port interchange, memory control unit, and I/O controllers.

HALT SWITCH. This is a pushbutton switch which causes the processor to halt execution at the end of the present micro-instruction. The next micro-instruction to be executed is fetched and stored in the micro-instruction register (M register). Which register the register select switch is pointing to is displayed on the 24 register display lights.

LOAD SWITCH. This is a pushbutton switch that causes the data displayed on the register display lights to be entered in the register pointed to by the register select switches.

DATA ENTRY SWITCHES. These consist of a row of 24 toggle switches which are operative only when the processor is in the HALT state. The desired bit configuration may be entered into the register selected by the register select switches by using the appropriate toggle switches. The data entered is displayed on the register display lights. (Refer to READ/WRITE switch.)

INT (INTERRUPT) SWITCH. This is a toggle switch that causes a console interrupt condition to be set in the processor control register (C register). This interrupt is testable by the MCP, and the MCP decides what the appropriate system reaction is to be.

START SWITCH. This is a pushbutton switch that initializes various functions within the system. It is used in conjunction with the console MODE switch, the CLEAR pushbutton, and the READ/WRITE pushbutton.

INC-A SWITCH. This is a pushbutton switch which increments the control memory address register (A register) by one micro-instruction word address when displaying or writing control memory from the console. The register select switch must be set to MSM, and the console MODE switch must be set to TAPE.

READ/WRITE SWITCH. This is a pushbutton switch that enables manual writing or reading of data into memory. The processor must be in the halt state and the desired data address must be loaded into the A register. To read data the register select switches must be set up for (pointing to READ) a read operation. Depressing of the READ/WRITE switch will cause the data at the address in the A register to be displayed in the register display lights. For a write operation the desired destination address must be loaded in the A register. The Register Column and Register Row Select Switches must point to the WRIT position and the desired data to be written must be entered into the system via the 24 Data Entry Switches. Depression of the Read/Write Switch completes the Write operation

REGISTER COLUMN AND REGISTER ROW SELECT SWITCHES. The register COLUMN SELECT switch is a 4-position rotary switch which selects one of four columns of registers (or functions). (Refer to table 1-3.)

Table 1-3
Register Select Switch
Column-Row Matrix

Row	Column O	Column 1	Column 2	Column 3
0 1 2 3 4 5 6 7 8 9 A B C D E F	TA TB TC TD TE TF CA CB LA LB LC LD LC CD	FU FT FLC FLD FLE FLF BICN FLCN TOPM(*1) Reserved Reserved Reserved XYCN XYST INCN(*1) CPU	X Y T L A (MAR) M BR LR FA FB FL TAS CP MSM(*1) Console READ Console WRIT	SUM CMPX CMPY XANY XEOY MSKX MSKY XORY DIFF MAXS MAXM U MBR(*1) DATA CMND NULL

The register ROW SELECT switch is an 18-position rotary switch which selects one of 16 rows of registers (or functions). Two rows are repeated for convenience; they are rows 4 and 13 (D) in table 1-3. These two repeated rows are the most utilized.

The various registers are addressed within the micro-instructions or from the front console by a series of co-ordinates. The first co-ordinate is to select one of 16 rows of registers and the second co-ordinate then selects one of four columns. The particular register (or function) selected by the intersection of the column and row is displayed when the processor is in the HALT state.

CONTROL MODE SWITCH. The control MODE switch is a rotary switch with the following positions: STEP, RUN, and TAPE.

^{*1} Not available on B 1710 Systems.

When the switch is in the STEP position, pressing the START switch causes the micro-instructions in the M register to be executed and the next micro-instruction to be fetched.

When in the RUN position, pressing the START switch causes the continuous fetch and execution of micro-instructions from the address specified in the micro-instruction address register.

When in the TAPE position, pressing the START switch causes the tape cassette reader to start reading the mounted TAPE cassette. The processor then enters a fetch-execution phase wherein micro-instructions are accepted and executed directly from the tape cassette. The tape cassette may contain load routines, confidence checks, or maintenance test routines (MTR). Initial micro-instructions on the cassette tape can cause subsequent micro-instructions to be loaded into memory from any $\rm I/O$ device.

CASSETTE REWIND. This is a pushbutton switch which causes the tape cassette to rewind to the BOT (beginning of tape) mark.

CASSETTE ON/OFF. This is a toggle switch which controls power to the tape cassette unit.

CONSOLE PRINTER/KEYBOARD.

To operate with the Burroughs Master Control Program, a console printer/keyboard is required for communications between the Master Control Program and the system operator.

MAIN MEMORY.

Main memory is addressable to the individual bit and has variable operation lengths from 0 to 24 bits in parallel. The length, direction, and address of operation is under micro-program control, either up or down within the address continuum. Main memory has no preferred word or byte boundaries which are visible to the rest of the system.

Main memory is a logical part of the processor since most of its control logic is part of that unit. The processor is the only unit to which main memory interfaces. The actual storage medium is, however, physically separate from the processor and is modular in design to allow for expandable capacity.

ADDRESSING.

Main memory is addressed by a 24-bit absolute address, a 1-bit field direction indicator, and a 5-bit field length value. The address is modified by the field direction bit. The field length value specifies the number of data bits in the direction indicated to be accessed, and varies from zero (no operation) to 24. For field length values greater than 24 bits, address values may be incremented and micro-instruction sequences are re-executed until the entire field has been accessed.

PARITY CHECKING.

Reliability of data within main memory is maintained with a hardware parity checking technique. Main memory is physically byte oriented

with a parity bit for every eight data bits. The main memory control logic resolves addressing, parity checking, data rotation, and compaction as described in the following paragraphs.

The high-order 21 bits of an address point to a byte within main memory where the operation is to begin. This "key byte" and three more bytes above or below, depending on the field direction bit, are read into the read memory information register (R-MIR) in the memory control unit. Parity is always checked on all 36 bits within this register.

Next, the low-order three bits of the address along with the field length value and the field direction bit (both contained in the micro-instruction) determine what is passed to or from main memory.

A parity error on any operation sets a bit in the processor C register (CD bit 3 set to 1), and appropriate action is taken by the MCP.

MEMORY DATA TRANSFER

Data is transferred, right-adjusted, 24 bits (three bytes) is parallel to and from main memory. Any operation calling for less than 24 bits will have leading (left) zeros supplied by the memory control unit. The field length controls the main memory operation and leading zeros are ignored or generated as needed within the memory control unit.

READ OPERATION.

A 24-bit address, a 1-bit field direction sign, and a 5-bit field length value are received with the Read request by the memory control unit from the processor. The Read is non-destructive of the data in memory and parity is checked during the operation. The data is then stripped of parity, right-adjusted, and masked with leading zeros as it is moved into the processor.

FIELD LENGTH.

The 5-bit field length value indicates the length of the data, right-adjusted, to be read from or written into memory. The normal field length value range is from zero to 24 bits.

The value of zero leaves the data in main memory unchanged. A value of 25 or 26 implies a field length of 24 and forces correct parity or incorrect parity on data written during the Write or Swap operation. Field lengths greater than 24 are reserved.

Information formerly on this page was deleted by PCN 1057155-001.

SECTION 2

B 1710 PROCESSORS

GENERAL.

This section discusses the B 1710 processing unit(s), which consist of the control portion and the combinatorial portion of the system along with the various addressing and data manipulation registers.

PROCESSING UNIT

The processor is the general-purpose data manipulator of the system. It performs the basic micro-functions necessary for a variety of commercial, scientific, and data communications applications. These micro-functions are performed by a set of elementary operators called micro-instructions. Micro-instructions operate on strings of bits within registers that were formerly the responsibility of the hardware logic.

By stringing a series of micro-instructions together, a macro-operation comparable to that defined by an assembler or compiler-level instruction may be implemented. This micro-instruction string is an interpreter for a macro "S-instruction" operation. Micro-instruction strings are executed in a "read-only" mode and are stored in main memory.

PROCESSOR REGISTERS.

MICRO-INSTRUCTION (M) REGISTER.

The M register (micro-instruction register) is a 16-bit register used to hold and decode the current micro-instruction which has been obtained from main memory or the TAPE cassette. The M register is addressable as a 16-bit source or destination register. The right-most 16-bits, if more than 16 are transferred, are loaded into the M register. Data is always transferred right-adjusted and zero-filled to the left. Each micro-instruction to be loaded from memory is ORed into the M register. Thus, moves to the M register allow modification of the next micro-instruction without altering the original micro-instruction in main memory. The state of this register is decoded by the processor to enable various control levels to perform the operation called for by the current micro-instructions.

MICRO-INSTRUCTION ADDRESS REGISTER (MAR).

The MAR (micro-instruction address register) is a 19-bit register capable of addressing micro-instructions located in main memory. It addresses 16-bit micro-operators in main memory which are assumed to be located at bit addresses exactly divisible by 16. When used in this manner the low-order four bits of MAR are ignored. Thus, only the left-most 15 bits of MAR are actually utilized in addressing.

The MAR is capable of having binary increments from 0 through 4095 added to or subtracted from it, with a high-speed carry adder to facilitate micro-program branching. The MAR is automatically incremented by one word (16 bits) as each micro-instruction is fetched. Memory wraparound can occur and is permitted.

The MAR is addressable as a source or destination register. When addressed as a source, the low-order four bits are set to zero. When used as a destination, the low-order four bits of the source are stored in the low-order four bits of the MAR. They are, however, not significant. Direct addressing of 40,960 bytes on the B 1712 and 65,536 bytes on the B 1714 is possible through the MAR.

The MAR also serves to address main memory for the Read/Write memory micro-instruction. The address of the next micro-instruction is temporarily stored in a scratchpad holding register. The main memory address to be accessed is obtained from the FA register.

ADDRESS (A) STACK.

The A stack is a 16-element deep, 24-bit wide memory which operates as a push down stack, i.e., a last-in-first-out (LIFO) structure. Address wraparound occurs when more than 16 entries to the stack are made. Data is not destroyed on a removal from the stack. The micro-instructions Call and Move to TAS (top of A stack) result in a "put into" (push) stack request, whereas Exit and Move from TAS result in a "take out of" (pop) stack request.

Using this stack, the micro-routines operate in a normal call-return mode. This allows for highly shared micro-routines. The A stack is not intended to be used exclusively as an address stack; it has been made 24 bits wide to allow for operand storage. (See figure 2-1.)

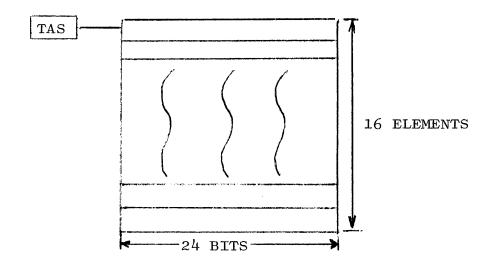


Figure 2-1. A Stack

TOP OF THE ADDRESS STACK (TAS) REGISTER. The TAS register (top of the A stack) is the 24-bit register which is currently on the top of the A stack. Moves to or from TAS result in data in the A stack being automatically moved to or from the A stack on a last-in-first-out basis.

GENERAL PURPOSE REGISTERS (L, -T, -X, -Y).

L REGISTER. The L register (figure 2-2) is a 24-bit general-purpose register. It is addressable as a source or as a destination register, either 24 bits in parallel or in 4-bit groups. Since the L register is addressable in 4-bit groups, its contents is available for analysis and alteration via the 4-bit function box. Manipulate, Skip, and Bit-Test-Branch micro-instructions may operate on the data in the L register. The L register is one of four registers (X, Y, L, T) capable of Read/Write operations with main memory.

	L/	4	L	В	L	.C		ĻD		L	.E	LF	
	3	0	3	0	3	0	3	-	0	3	0	3	0
١	23 🗖												- ol

Figure 2-2. L Register

T REGISTER. The T register (figure 2-3) is a 24-bit general-purpose register. It is addressable as a source or as a destination register, either 24 bits in parallel or in 4-bit groups. Since the T register is addressable in 4-bit groups, its contents is available for analysis and alteration via the 4-bit function box. Manipulate, Skip, and Bit-Test-Branch instructions can operate on the data in the T register, being one of the four registers (X, Y, T, L) capable of Read/Write operations with main memory. The T register is also capable of Shift/Rotate and Extract operations.

T	4	т	В	-	тс	т	D	Т	E	Т	F
3	0	3	0	3	0	3	0	3	0	3	0
23			-								- 0

Figure 2-3. T Register

X AND Y REGISTER'S. The X register and the Y register are 24-bit general purpose register's. They are used primarily to hold and act as sources for two of the three operands of the combinatorial section (the 24-bit function box) of the processor. The other operand is the CYF register (carry flip-flop'). The X and Y registers are both addressable as source and destination register's.

Both registers, along with the L register and the T register, are capable of Read/Write operations with main memory. Both registers are capable of the Shift/Rotate operation. The X register is affected by the Normalize operation.

FIELD (F) DEFINITION REGISTER.

The F register (field definition) specifies the address and lengths of data fields in main memory (see figure 2-4). It is a 48-bit register which is functionally divided into two portions: a 24-bit FA (field address) register and a 24-bit FB register which is divided into a 4-bit FU (field unit) register, a 4-bit FT (field type) register, and a 16-bit FL (field length) register.

FA	FU	FT	FLC	FLD	FLE	FLF
			3 0	3 0	3 0	3 0
	3 0	3 0	15	FL	-,,-	 0
23 ← FA ← 0	23 🕶		— FВ			 0

Figure .2-4. F Register

FA PORTION OF THE F REGISTER. The FA register holds the absolute main memory address of a data field and has the capability of addressing 524,488 bits (64KB) without regard to physical boundaries. FA may be addressed as either a source or destination register.

FB PORTION OF THE F REGISTER. FB may be considered as one 24-bit register or as various subregister's. The FU register holds descriptive information regarding the units of data which make up the addressed field in main memory (binary or 4-bit groups'). The FT register holds additional descriptive information. The FU and FT values do not affect the main memory operation. The FU and FT values are usually supplied by S-instruction's. The FU may also be used with the BIAS micro-instruction to set the CPU and CPL register's. The FL register holds the total length in binary of the field being operated on in main memory, and is capable of describing fields up to 65,536 bits in length.

FB may be addressed as either a source or destination register or addressed in part as FU (4 bits), FT (4 bits), and FL (16 bits). FL may also be addressed in 4-bit subfields (FLC, FLD, FLE, or FLF).

SCRATCHPAD.

A scratchpad of 16 words, each 48 bits wide, is provided to hold field descriptors of operands (see figure 2-5%). Scratchpad may be addressed as 16 48-bit words (SOO-S15) or 32 24-bit words (SOOA-S15A, SOOB-S15B%). Any word may be a source or a destination and may be used to hold pointers, values, or any other information.

The second half of the first 48-bit word (SOOB), along with FU and FL, is always entered as input to the 4-bit function box and used in certain decision-making logic. The field length value in the FL register is continually being compared against a corresponding portion (SFL) of the SOOB word to determine the relationship of their contents: high, low, equal, or zero. This relationship may then be used in the Bit-Test-Branch micro-instruction (refer to FLCN register). The FU and FL portions of the F register and the like portions (SFU and SFL) of SOOB

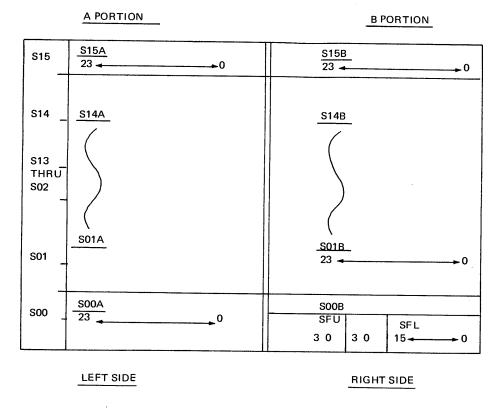


Figure .2-5. Schatchpad

may also be used in the combinatorial section to set the value in the CP portion of the C register.

FIELD LENGTH CONDITION (FLCN) REGISTER. The FLCN register (figure 2-6) is a 4-bit register which contains the result of a comparison between the FL portion of the F register and the corresponding portion of the first scratchpad word (right-most 16 bits of SOOB) which is called SF1. It has the following interpretation:

FL≠0 - FL not equal to 0
FL<SFL - FL less than SFL
FL>SFL - FL greater than SFL
FL=SFL - FL equal to SFL

FL=SFL	FL≫FL	FL≪SFL	FL≠0
3	2	1	0

Figure .2-6. FLCN Register

All 16 bits of FL are compared against the 16 bits of SFL. A one bit in a position of the FLCN register indicates that the respective condition is true.

BASE AND LIMIT REGISTERS (BR, -LR') . -

The base register (BR) and limit register (LR) may be used as source or destination registers. They are used for memory protection and for base relative addressing. Memory protection is provided for in the MCP by checking the main memory address in all memory Write operations with the BR and LR allowing the operation to take place only within these limits.

CONTROL (C) REGISTER . .

The C register (control) is a collection of independent registers which are utilized by the interrupt system of the processor and the combinatorial section (see figure 2-7). The C register is 24 bits wide and is divided into three 8-bit functional part's.

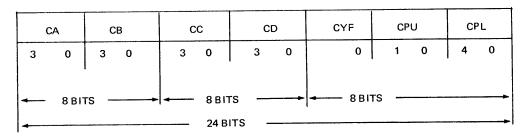


Figure 2-7. C Register

CA AND CB REGISTER'S. This portion of the C register is addressable in 4-bit units as CA and CB. These portions are available as general-purpose source and destination register's.

CC AND CD REGISTER'S. This portion of the C register is addressable in two 4-bit units defined as CC and CD. CC and CD are used for storage of the processor states and for processor interrupts as shown in the following paragraph's.

CC Register.

The bit meanings for the CC register are as follows:

- a. Bit 0 console interrupt.
- b. Bit 1 I/O service request.
- c. Bit 2 timer interrupt.
- d. Bit 3 console state.

The console interrupt bit is set from the console INT (interrupt) switch and allows the software to come to an orderly stop. The I/O service request interrupt bit is set by an I/o controller when it issues a service request to the processor. The micro-program must determine what caused the interrupt and transfers or receives data from the device. The timer interrupt bit is set by the hardware timer every 100 millisecond's. Once the timer interrupt bit is set, it stays set until the MCP resets it.

CD Register.

The bit meanings for the CD register are as follows:

a. · Bit 0 - reserved.

b. Bit 1 - reserved.

c. Bit 2 - reserved.

d. Bit 3 - memory parity error interrupt.

Any parity error detected during a main memory operation sets the memory parity error interrupt bit (CD bit 3').

All interrupts in the processor are soft and no reaction occurs as a result of an interrupt bit being set until the micro-program tests for such setting's.

CYF, CPU, AND CPL REGISTER'S. The least-significant bits of the C register are composed of the arithmetic unit carry flip-flop (CYF), the 2-bit arithmetic unit type (CPU), and the 5-bit combinatorial data length control (CPL). Moves from CP to a 24-bit register automatically inserts 16 leading zero's. Moves to CP from any register use only the least-significant eight bits of the field being move'd.

The Carry micro-instruction is supplied for manipulating CYF, thereby remembering the status of a carry or borrow beyond the length of an operand. For SUM and DIFF results, CYF is always an input to the combinatorial section along with the X and Y register's. CPU controls the arithmetic unit type of the combinatorial section and is addressable only as a destination register. The possible settings and their meanings are listed below:

<u>CPU</u>	<u>Data Type</u>
00	Binary
01	4-bit binary
10	Undef ined
11	Undef ined

CPL controls the length of the operands being entered as input to the combinatorial section and should be an integral multiple of the data type specified by CPU for valid arithmetic result's. The maximum allowable value of CPL is 24, which is an integral multiple of both data type's. Larger values are reserve'd.

MAXIMUM SIZE OF MAIN MEMORY (MAXS) REGISTER. The MAXS register is a 24-bit pseudo register which is set by a field engineer to indicate the maximum size of the installed main memory. The MAXS register is addressable as a source register only.

MAXIMUM SIZE OF CONTROL MEMORY (MAXM) REGISTER.

The MAXM register is a 24-bit pseudo register which is set by a field engineer to indicate the maximum size of the installed control (micro-instruction) memory. For this processor MAXM always contains a value of zero. The MAXM register is addressable as a source register only.

TAPE CASSETTE INPUT (U) REGISTER.

The U register is a 1.6-bit register used primarily to accumulate input (usually micro-instructions) from the console TAPE cassette. The U register is addressable only as a 1.6-bit source.

Only the micro-instruction Register Move may access the U register. In TAPE mode the contents of this register can be moved by a micro-program on the tape to the M register for execution.

DATA REGISTER.

The DATA register is a 24-bit pseudo register which can act as a source or destination. It is used to transfer data between the I/O devices and the processor. When used as a source it accepts 24 bits of data from the I/O controller. When used as a destination register, data is placed into the DATA register and is sent to the I/O controller.

COMMAND (CMND) REGISTER.

The CMND register (command) is a 24-bit pseudo register which can act as a destination only and is used to transfer I/O commands to the I/O device's.

NULL REGISTER.

The NULL register is a 24-bit pseudo register which contains zero's. Moves from NULL may be used for clearing various registers to zero's.

THE ARITHMETIC AND COMBINATORIAL SECTION (24-BIT FUNCTION BOX). The 24-bit arithmetic and combinatorial section (figure 2-8) is composed of a 24-bit arithmetic unit and a 24-bit combinatorial unit. It has as data inputs the contents of the X and Y registers and the carry indicator (CYF'). It also uses the CPU and CPL portions of the C register.

All results from the combinatorial section are generated immediately and are continuously available to the micro-programmer. A move to one of the input registers or an alteration of a value in the CYF, CPU, or CPL registers immediately generates a new result. The results are actessed by moving the contents of a result register to a destination register or by testing one of the 4-bit condition registers.

The results are most of the commonly used functions between two operand's. These include the AND, OR, EXCLUSIVE-OR, sum, carry out, difference, and borrow functions, and the set of equal to, greater than, and less than relational's. The results of the unary operations of complementation and masking are also available.

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

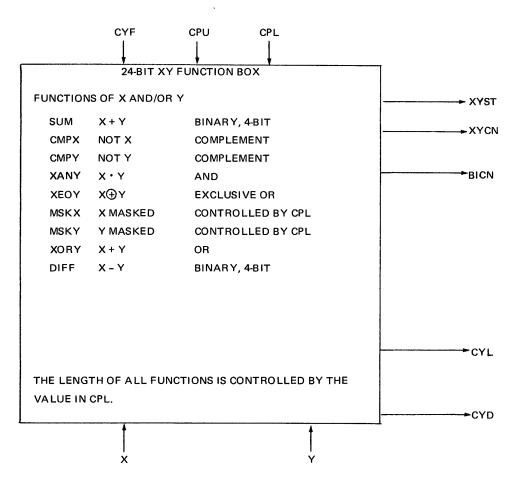


Figure 2-78. Arithmetic and Combinatorial Section (24-Bit Function Box)

<u>CPU</u>	<u>Unit Type</u>	Possible CPY Values	<u>Data Type</u>
00 01 10 11	1-bit operands 4-bit operands Undefined Undefined	1 to 24 4, -8, -1-2, -16, -2-0, -or 24	Binary .4-bit binary

For valid arithmetic operations, the operand length $\underline{\text{must}}$ be an exact multiple of the length of the unit specified by CPU.

Each of the following register contents are generated immediately and are available to the micro-programmer upon request.

SUM RESULT REGISTER. The sum of the X, Y, and CYF registers (X, Y, and CYF are inputs to the 24-b it function box) is produced and placed in the 24-b it pseudo sum

register. Zeros in the more-significant bit positions (left-most) of the 24-bit result are produced when the length given by CPL is less than 24. CYF should be set to zero at the start of a sum generation.

The added logic performs differently for the four possible values in the CPU register as follows:

- a. IF CPU = 00, the binary sum is produced.
- b. If CPU = 01, the decimal sum is produced by considering the input to be comprised of up to six 4-bit units. The 4-bit units are added decimally and a carry is propagated from one 4-bit unit to the next more significant 4-bit unit whenever the binary sum of two 4-bit units exceeds 9.
- 'c. If CPU = 10 or 11, the sum is not define'd.

DIFFERENCE (DIFF) RESULT REGISTER. The difference of the X, Y, and CYF registers (X, Y, and CYF are inputs to the 24-bit function box) is produced in the 24-bit pseudo DIFF register. Zeros in the more significant bit positions (left-most) of the 24-bit result are produced when the length determined by CPL is less than 24. Difference results are generated by adding the contents of X to the one's complement of both Y and CYF. The complement values are hardware generated and do not alter the values of Y or CYF.

The absolute value of (Y + CYF) may be greater than the value in X and produces a negative result. Such negative results are in complement form and are indicated by CYD (carry difference) = 1. If CYD = 0, the difference result value is a positive number. CYD is generated from all 24 bits of X and Y. It is not controlled by the value of CPL. CYD may be used to alter the value of CYF. Intermediate underflow may thus be remembered by CYF through iterations of a field.

The difference logic produces results under control of the CPU register as follows:

- 'a. If CPU = 00, the binary difference is produce'd. Negative results are expressed in 2's complement form when CYD = 1.
- b. If CPU = 01, the decimal difference is produced by considering the input to be comprised of up to six 4-bit units. The 4-bit units are subtracted and any borrow is propagated from one 4-bit unit to the next as in decimal subtraction. Negative results are expressed in 10's complement form when CYD = 1.
- c. If CPU = 10 or 11, the difference is not defined.

AND/OR/EXCLUSIVE-OR (XANY, XEOY, XORY) RESULT REGISTER'S. The results of the appropriate logical function, AND/OR/EXCLUSIVE-OR, of the X and Y registers (X and Y are inputs to the 24-bit function box) are produced and placed in the appropriate 24-bit pseudo register (XANY, XEOY, XORY, respectively). Zeros in the more significant bit positions are produced when the length determined by CPL is less than

24.

COMPLEMENT X/COMPLEMENT Y (CMPX, CMPY) RESULT REGISTER'S. The one's complement of the appropriate register X or Y (X or Y and CPL are inputs to the 24-bit function box) is produced and placed in the appropriate 24-bit pseudo register, CMPX or CMPY, respectively. Zeros in the more significant bit positions of the 24-bit result are produced when the length given by CPL is less than 24.

MASKED X/MASKED Y (MSKX, MSKY) RESULT REGISTER'S. The mask of the contents of the appropriate register X or Y (X or Y and CPL are inputs to the 24-bit function box) is produced and placed in the appropriate 24-bit pseudo register, MSKX or MSKY, respectively (see figure 2-9). The value of CPL determines the number of bits placed in MSKX or MSKY. Zeros in the more significant bit positions of the 24-bit result are produced when the length given by CPL is less than 24.

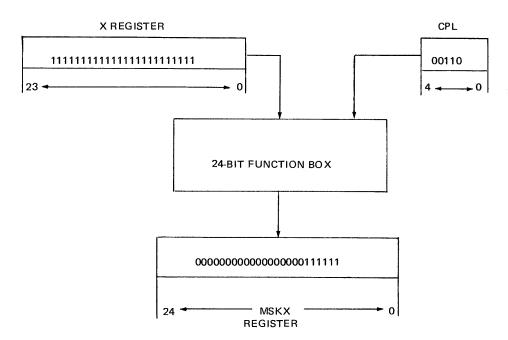


Figure 2-9. · MSKX Result Register

BINARY CONDITIONS (BICN) REGISTER.
The BICN register (figure 2-10) is a 4-bit pseudo register which may act as a source only and indicates the following binary conditions: CYL (carry out level), CYD (borrow out level), CYF (carry flag), and LSUY (least significant unit of Y).

LSUY	CYF	CYD	CYL
3	2	1	0

Figure 2-10. BICN Register

The LSUY is true if the least significant unit of the Y register is equal to 1 and CPU equals 00, or the least significant unit of the Y register is equal to 1001 (nine) and CPU is equal to 01. If CPU equals 10 or 1.1, LSUY is not defined.

The carry out and borrow out levels are a function of the inputs (X, Y, and CYF) and CPL. CYF may be manipulated by a special micro-instruction or as data in the high-order position of the CP register.

X/Y CONDITION (XYCN) REGISTER. The XYCN register (figure 2-11) is a 4-bit oseudo register which may act as a source only and contains the following relational conditions: X > Y (X is greater than Y), X < Y (X is less than Y), X = Y (X equals Y), and MSBX (the most significant bit of X).

MSBX	X = Y	x <y< th=""><th>x>Y</th></y<>	x>Y
3	2	1	0

Figure 2-11. XYCN Register

The relational conditions are based on the 24-bit binary value of both X and Υ . The MSBX is true if the most-significant bit of the X register, as designated by CPL, is a Υ 1.

X/Y STATE (XYST) REGISTER. The XYST register (figure 2-12) is a 4-bit pseudo register which may act as source only and contains the following relational conditions: $X \neq 0$ (X is not equal to zero), $Y \neq 0$ (Y is not equal to zero), Interrupt (INT is true if any of the following conditions contained by CC and CD are true: timer interrupt, I/O service request interrupt, console interrupt, and memory parity error interrupt), and LSUX (the least significant unit of X).

1	LSUX	INT	Y≠0	x≠0
	3	2	1	0

Figure 2-12. XYST Register

LSUX is true when the least significant unit of the X register is a 1 and CPU equals 00, or when the least significant unit of the X register equals 1001 (nine) and CPU equals 01. If CPU equals 10 or 11, LSUX is not defined.

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

FOUR-BIT ARITHMETIC AND COMBINATORIAL SECTION (FOUR-BIT FUNCTION BOX).
The 4-bit arithmetic and combinatorial section of the processor is used to generate most of the normally used functions between two 4-bit oper-

ands and can accept as input the contents of any one of the following 4-bit register's.

TA	T B	TC	TD ·	TE	TF
LA	LB	LC	LD	LE	LF
FU	F٢	FLC	FLD	FLE	FLF
CA	CB	CC	CD		
BICN	XYCN	XYST	FLCN		

A second input is obtained from the Four-Bit Manipulate micro-instruction itself. It has as possible results the commonly used functions between two operands. these include: SET, AND, OR, EXCLUSIVE-OR, Binary Modulo 16 Sum, and Binary Modulo 16 Difference functions. The results may only be directed to the same register which acted as the source. The sum and difference result can be tested for overflow and underflow, respectively.

The BICN, XYCN, XYST, and FLCN registers are available as source registers only.

The 4-bit combinatorial section also provides for the selective testing of any of the bits within the 4-bit registers and relative branching based on the results of the test (Bit-Test-Branch, micro-instructions). The Skip When micro-instruction also tests any combination of bits, up to four, and branches on the result.

Listed in table 2-1 is a summary of the various 4-bit conditions which are available to the micro-programmer. (Refer to table 1-3 for the registers and functions in the column-row matrix.)

Table 2-1
4-Bit Conditions

Register	Bit 3	Bit 2	Bit 1	Bit O
BICN XYCN XYST FLCN CC	LSUY MSBX LSUX FL = SFL Console state lamp	CYF X = Y INT FL > SFL Timer interrupt	CYD X < Y Y ≠ 0 FL < SFL I/O service interrupt request	CYL X > Y X ≠ 0 FL ≠ 0 Console interrupt
CD	Memory parity interrupt	Reserved	Reserved	Reserved

SECTION 3

B 1726 PROCESSOR

GENERAL.

This section discusses the B 1726 processing unit which consists of the combinatorial portion of the system along with the various addressing and data manipulation registers.

PROCESSING UNIT.

The processor is the general-purpose data manipulator of the system. It performs the basic micro-functions necessary for a variety of commercial, scientific, and data communications applications. These micro-functions are performed by a set of elementary operators called micro-instructions. Micro-instructions operate on strings of bits within registers that were formerly the responsibility of the hardware logic.

By stringing a series of micro-instructions together, a macro-operation comparable to that defined by an assembler or compiler-level instruction may be implemented. This micro-instruction string is an interpreter for a macro "S-instruction" operation (refer to S-Instructions). Micro-instruction strings are executed in a "read-only" mode and are usually stored in the high-speed control memory of the system. They may, however, be executed directly from the main memory of the system. Control memory may also be overlaid with micro-instructions from main memory.

CONTROL MEMORY.

This Read/Write memory resides within the processing unit and is used only to hold micro-instructions. If the set of micro-instructions in use exceeds available control memory, the excess micro-instructions are stored in, and can be executed from, main memory. The processor properly addresses the micro-instructions in either location (refer to Micro-Instruction Addressing). Micro-instructions which are being executed from main memory, however, result in a decrease in system throughput. This decrease occurs because control memory is four times faster than main memory. Micro-instruction sequences that are executed many times are located in control memory. Micro-instructions may be overlaid from main memory into control memory. The larger the amount of control memory the greater the system throughput.

Control memory size is 1024 or 2048 word's. Each word is 16 bits wide to accommodate one micro-instruction. The micro-programmer should locate the sequences of code according to their expected frequency of usage. The greater the frequency of usage, the lower their address in control memory. This is done to give the highest execution speed possible for the system, regardless of the installed size of control memory. Micro-instructions which normally would have been stored in the missing portion of control memory are located in available main memory and addressed directly. Control memory operations are overlapped with main memory operations except for the Read, Write, Dispatch, and Overlay micro-instruction's. Control memory operates in a "read only" mode, with the two exceptions of the Overlay micro-instruction and the console control panel in the halt/display mode.

MICRO-INSTRUCTION ADDRESSING.

To facilitate the fetch and execution of micro-instructions which are located in control memory and/or main memory, three hardware registers and certain decision-making logic have been implemented. The three registers are the A register (14-bit address), the TOPM register (top of control memory), and the MBR register (micro-instruction base). The addressing logic operates in the following manner.

As each micro-instruction is fetched, the A register is automatically incremented by 1. The micro-instruction just fetched is executed. Before fetching the next micro-instruction, the value in the A register is compared to (TOPM x 512'). The TOPM register normally contains a value equal to the number of bytes (MOD 1000) of control memory present on the system, so that multiplying TOPM by 512 yields the number of the first micro-instruction outside control memory.

If the value in the A register is less than (TOPM x 512), the next micro-instruction is fetched from control memory at the address given in the A register. The process then returns to the initial micro-instruction fetched.

If the value in the A register is equal to or greater than (TOPM x 512), the next micro-instruction is fetched from main memor/y. The bit address of that instruction is obtained from the following formula:

 $(A \times 16) + MBR$

The MBR register contains the base address above which micro-instructions in main memory are stored. The process then returns to the micro-instruction fetched. If the address generated by the above formula is equal to or greater than the value in the MAXS register, a memory-read-address-out-of-bounds interrupt occurs when the fetch is attempted.

Figure 3-1 illustrates the above sequence of operation's.

For example, suppose there are 4K bytes (2048 words) of control memory and 64K bytes (524,288 bits) of main memory installed on the system. In this case, the TOPM register should be set to 4 by the MCP to indicate the size of control memory. If the MBR register is set by the MCP to bit address 426,984, there is sufficient space for 4096 micro-instructions to be stored in main memory. Micro-instructions are fetched from control memory until the A register is equal to 2048 (TOPM x 512), indicating that the size of control memory has been exceeded. At this time, the processor begins to fetch micro-instructions from main-memory, starting at bit address 459,752 ((A x 16) +MBR). This process continues until the A register is reset by a micro-instruction or until the end of main memory is reached (((A x 16) + MBR) > MAXS). In the first case, a branch is executed to a different area of control or main memory; in the second case, a memory-read-address-out-of-bounds interrupt occur's.

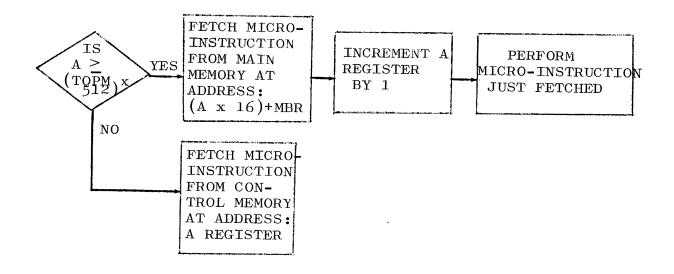


Figure 3-1. Micro-Instruction Fetch and Execution

Note the apparent discrepancy between the value in the MBR register and the address where the first micro-instruction is fetched from main memory. The area between the two addresses is exactly the size of the installed control memory (as indicated by (TOPM x 512)), and is available for data or code storage by S-instruction programs.

PROCESSOR REGISTERS.

MICRO-INSTRUCTION (M) REGISTER.

The M register (micro-instruction) is a 16-bit register used to hold and decode the current micro-instruction which has been obtained from control memory or main memory.

The M register is addressable as a 16-bit source or destination register. The right-most 16 bits, if more than 16 are transferred, are loaded. Each micro-instruction to be loaded from control memory is ORed with whatever may be in the M register. Thus, moves to the M register allow modification of the next micro-instruction without altering the original micro-instruction. The state of this register is decoded by the processor to enable various control levels to perform the operation called for by the micro-instructions.

CONTROL/MICRO-STRING MEMORY (MSM) REGISTER.

The MSM register is a 16-bit pseudo register which is addressable as a source when the console control panel is in a halt/display mode or as a destination register when the console control panel is in the tape mode. As a source register MSM contains the micro-instruction which is pointed to by the A register. It is addressable as a destination register by the Move 24-Bit Literal micro-instruction and by the Register Move micro-instruction in the tape mode.

ADDRESS (A) REGISTER.

The A register (address) is a 14-bit register capable of addressing up to 16,384 micro-instructions (each 16 bits in length) located in main memory and/or control memory (see figure 3-2). The A register is capable of having binary increments from 0 to 4095 added to or subtracted from it, with a high-speed carry adder to facilitate micro-program branching. The A register is automatically incremented by one as each micro-instruction is fetched.

BINARY ADDRESS			0	0	0	0
13	O 4		3			0
17	4	-)			Ü

Figure 3-2. A Register

The A register is addressable as a source or a destination. When addressed as a source, its contents are automatically multiplied by 16. When used as a destination, the right-most four bits of the source are lost. These functions are accomplished by the presence of four non-functioning bits appended to the low-order end of the register which are always zero. When used as a source, the A register is actually transferred as an 18-bit field, thus automatically accomplishing the multiplication by 16. As a destination, the source is moved right-justified into the 18-bit destination field, with the four low-order bits masked to zeros. These extra four bits cannot be addressed and are completely transparent to the micro-programmer.

TOP OF CONTROL MEMORY (TOPM) REGISTER.

The TOPM register (top of control memory) is a 4-bit register which may act as a source or a destination (see figure 3-3). This register is automatically multiplied by 512 and compared with the address in the A register to determine whether the next micro-instruction should be fetched from main memory or control memory. This automatic multiplication is accomplished by the presence of nine nonfunctioning bits which are always zero, appended to the low-order end. These extra nine bits cannot be addressed and are completely transparent to the micro-programmer. For each 1K bytes of control memory the TOPM register is incremented by 1.

0 1 0	0	0	0	0	0	0	0	0	0	0
3	О									
ТОРМ										

Figure 3-3. TOPM Register

MICRO-INSTRUCTION BASE REGISTER (MBR').

The MBR (micro-instruction base register) is a 24-bit register which may act as a source or a destination. It is used to contain the base address (modulo 16) above which all micro-instructions stored in main memory are located. The micro-instructions do not begin exactly at the address in the MBR register.

The address in main memory of the next micro-instruction to be fetched is equal to the contents of the A register times 16 added to the contents of MBR, or $(A \times 16) + MBR$.

ADDRESS (A) STACK.

The A Stack (figure 3-4) is a 32-element deep, 24-bit wide memory which operates as a push-down stack, i.e., a last in, first out (LIFO) structure. Address wraparound occurs when more than 32 entries to the stack are made. Data is not destroyed on a removal from the stack. The micro-instructions Call or Move to TAS (top of A stack) result in a "put into" (push) stack request, whereas Exit and Move from TAS result in a "take out of" (pop) stack request.

Using this stack, the micro-routines operate in a call-return mode which allows for highly shared micro-routines and reduces the control memory requirements. The A stack is not intended to be used extensively as an operand stack; it has been made 24 bits wide to allow for operand storage.

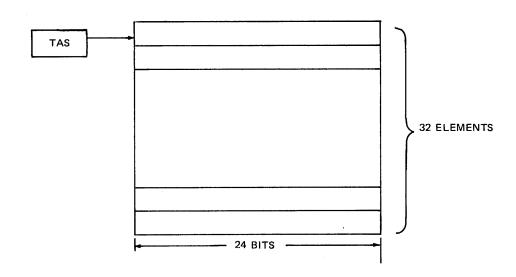


Figure 3-4. A Stack

TOP OF ADDRESS STACK (TAS) REGISTER.

The TAS register (top of the A stack) is the 24-bit register which is currently the top of the A stack. Moves to or from the TAS register result in data in the A stack being automatically moved to or from the A stack on a last in, first out basi's.

GENERAL PURPOSE REGISTERS (L, T, X, Y) ...

L REGISTER. The L register (figure 3-5) is a 24-bit, general-purpose register. It is addressable as a source or as a destination register, either 24 bits at a time or in 4-bit group's. Since the L register is addressable in 4-bit groups, its contents are available for analysis and alteration via the 4-bit function box. Manipulate, Skip, and Bit-Test-Branch micro-instructions may operate on the data in the L register. The L register is one of four registers (X, Y, L, T) capable of Read/Write/Swap operations with main memory.

			В	L	.c	L	D	LE		L	F
3	0	3	0	3	0	3	0	3	0	3	0
23 -											o

Figure 3-5. L Register

The Dispatch micro-instruction uses the contents of the L register as the 24-bit address of the first I/O description in main memory.

T REGISTER. The T register (figure 3-6) is a 24-bit general-purpose register. It is addressable as a source or a destination register, either 24 bits at a time or in 4-bit group's. Since the T register is addressable in 4-bit groups, its contents are available for analysis and alteration via the 4-bit function box. Manipulate, Skip, and Bit-Test-Branch instructions can operate on the data in the T register. It is one of the four registers (X, Y, T, L) capable of Read/Write/Swap operations with main memory. The T register is capable of Shift/Rotate and Extract operation's.

Т	та тв		TB TC TD			D	TE			TF	
3	0	3	0	3	0	3	0	3	0	3	0
23-				4 100	F.J.,						o

Figure 3-6. T Register

X AND Y REGISTER'S. The X register and the Y register are 24-bit, general-purpose register's. They are used primarily to hold and act as sources for two of the operands of the combinatorial section (input to the 24-bit function box) of the processor. The other operand is the CYF register (carry flip-flop'). The X and Y registers are addressable as source and destination register's.

Both registers, along with the L register and the T register, are capable of Read/Write Swap operations with main memory. Both registers are capable of the Shift/Rotate operation.

FIELD (F) DEFINITION REGISTER.

The F register (field definition) specifies the address and lengths of data fields in main memory (see figure 3-7). It is a 48-bit register which is functionally divided into two portions: a 24-bit FA (field address) register and a 24-bit FB register which is divided into a 4-bit FU (field unit) register, a 4-bit FT (field type) register, and a 16-bit FL (field length) register. It may be loaded, stored, or swapped 48 bits in parallel with scratchpad memory.

	FA	F	U	F	T	Fl	_C	FI	_D	FL	E	FL	.F
						3	0	3	0	3	o	3	0
		3	0	3	0	15-			F	L			→ 0
23-	— FA —→ 0	23 🕶		•		•		·					→ 0

Figure 3-7. F Register

FA PORTION OF THE F REGISTER. The FA register holds the absolute binary main memory address of a data field and has the capability of directly addressing 16,777,216 bits without regard to physical boundarie's. FA may be addressed as either a 24-bit source or destination register.

FB PORTION OF THE F REGISTER. FB may be considered as one 24-bit register or various subregisters: FU, FT, and FL. The FU register holds descriptive information regarding the units of data which make up the addressed field in main memory (binary, 4-bit, or 8-bit groups). The FT register holds additional descriptive information. The FU and FT values are usually supplied by S-instructions and the FU may also be used with the Bias micro-instruction to set the CPU and CPL register's. The FL register holds the total length in binary of the field being operated on in main memory, and is capable of describing fields up to 65,536 in length.

FB may be addressed as a source or destination or in part as FU (4 bits), FT (4 bits), and FL (16 bits). FL may also be addressed as 4-bit subfields (FLC, FLD, FLE, or FLF).

SCRATCHPAD.

A scratchpad of 16 words, each 48 bits wide, is provided to hold field descriptors of operands (see figure 3-8). Scratchpad may be addressed as 16 48-bit words (S00-S15) or 32 24-bit words (S00A-S15A, S00B-S15B). Any word may be a source or a destination and may be used to hold pointers, values, or any other information.

		A PORTIC	<u>N</u>		B POR	TION
S15	S15A 23		0	S15B 23		0
S14	S14A	•		S14B	•	
S13 THRU S02		· ·			•	
S01	S01A			S01B 23	one de la companya d	0
S00	S00A 23		0	S00B	ı— — —	.
				SFU 3 0	3 (SFL 0 15 0
		LEFT			RIGH	Ι

Figure 3-8. - Schatchpad

The second half of the first 48-bit word (SOOB), along with FU and FL, is always entered as input to the 4-bit function box and is used in certain decision-making logic. The field length value in the FL register is continually being compared against a corresponding portion (SFL) of the SOOB word to determine the relationship of their contents: high, low, equal, or zero. This relationship may then be used in the Bit-Test-Branch micro-instruction (refer to FLCN register). The FU and FL portions of the F register and the like portions (SFU and SFL) of SOOB may also be used in the combinatorial section to set the value in the CP portion of the C register.

FIELD LENGTH CONDITIONS (FLCN) REGISTER.
The FLCN register (figure 3-9) is a 4-bit register which contains the result of a comparison between the FL portion of the F register and the corresponding portion of the first schatchpad word (right-most 16 bits of SOOB) which is called SFL. It has the following interpretation:

 $FL \neq 0$ - FL not equal to 0

FL<SFL - FL less than SFL FL>SFL - FL greater than SFL FL=SFL - FL equal to SFL

FL=SFL	FL>SFL	FL <sfl< th=""><th>FL≠0</th></sfl<>	FL≠0
3	2	1	0

Figure 3-9. FLCN Register

All 16 bits of FL are compared against the 16 bits of SFL. A one bit in a position of the FLCN register indicates that the respective condition is true.

BASE AND LIMIT REGISTERS (BR, LR). The base register (BR) and limit register (LR) may be used as source or destination register's. They are used for main memory protection and for base relative addressing.

Memory protection is provided for in the MCP by checking the main memory address (in FA) on all Read/Write/Swap operations with the BR and LR allowing the operation to take place only within those limit's. All out-of-bound requests, whether allowed or not, are flagged in the CD portion of the C register (CD bits 0 and 1). Any Write operation or the write portion of a Swap operation outside the values in the base and limit registers is inhibited unless the out-of-bounds override bit is on in the C register (CD bit 2 = 1). Read out-of-bound operations are executed.

Memory protection is only provided on the main memory address in the FA register and not on any of the 23 adjacent bits when the value in the field length (FL) register is greater than one.

CONTROL (C) REGISTER.
The C register (control) is a collection of independent registers which are utilized by the interrupt system of the processor and the combinatorial section. The C register is 24 bits wide and is divided into three 8-bit functional parts (see figure 3-10).

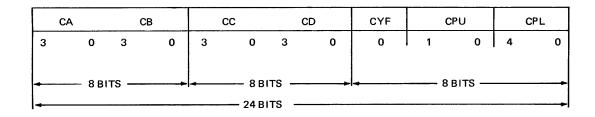


Figure 3-10. C Register

CA AND CB REGISTER'S. This portion of the C register is addressable in 4-bit units as CA and CB. These portions are available as general-purpose storage register's.

CC AND CD REGISTER'S. This portion of the C register is addressable in two 4-bit units defined as CC and CD. CC and CD are used for storage of the processor states and for processor interrupts as described in the following paragraph's.

CC Register.

The bit meanings for the CC register are as follows:

a. Bit 0 - console interrupt.

b. Bit 1 - I/O service request interrupt.

c. Bit 2 - timer interrupt.

d. Bit 3 - console state.

The console interrupt bit is set from the console INT (interrupt) switch and allows the software to come to an orderly stop. The I/O service request interrupt bit is set by an I/O controller when it issues a service request to the processor. The timer interrupt bit is set by the hardware timer every 100 millisecond's. Once the timer interrupt is set, it stays set until the MCP resets it. The console STATE light is also under control of the micro-programme'r.

CD Register.

The bit meanings for the CD register are as follows:

- a. Bit 0 memory Write/Swap address out of bounds interrupt.
- b. Bit 1 memory Read address out of bounds interrupt.
- c. Bit 2 memory Write/Swap address out of bounds override.
- d. Bit 3 memory parity error interrupt.

The out-of-bound signals used to set CD bits 0 and 1 are derived from logic which compares the contents of the FA register with the contents of the base (BR) and limit (LR) register's. The out-of-bounds override bit (CD bit 2 set to 1) allows a Write operation or the write portion of a Swap operation to be executed but does not inhibit the setting of the Write/Swap address out-of-bounds bit (CD bit 0). The Read cycle is never inhibited but sets the Read Address out-of-bounds bit (CD bit 1) if the address is not between the values in the base and limit register's. Any parity error detected during a main memory operation sets the memory parity error interrupt (CD bit 3'). All interrupts in the processor are soft and no reaction occurs as a result of an interrupt bit being set until the micro-program tests for such setting's.

CYF, -CPU, -AND CPL REGISTER'S. The least significant eight bits of the C register (CP) are composed of the arithmetic unit carry flip-flop (CYF), the 2-bit arithmetic unit type (CPU), and the 5-bit combinatorial data length control (CPL'). Moves from CP to a 24-bit register automatically insert 16 leading zeros, and moves to CP from any register use only the least significant eight bits of the field being moved.

The Carry micro-instruction is supplied for manipulating CYF and thereby remembering the status of a carry or borrow beyond the length of an operand. For SUM and DIFF results, CYF is always an input to the combinatorial section along with the X and Y register's. CPU is addressable only as a destination. CPU controls the arithmetic unit type of the combinatorial section. The possible settings and their meanings are listed below.

CPU	<u>Data Tyr</u>	<u>oe</u>
00	Binary	
01	4-bit binary	
10	Undef ined	
11	.8-bit numeric	(EBCDIC)

CPL controls the length of the operands being entered as input to the combinatorial section and should be an integral multiple of the data type specified by CPU for valid arithmetic result's. The maximum allowable value of CPL is 24, which is an integral multiple of all allowable data type's. Larger values are reserve'd.

MAXIMUM SIZE OF MAIN MEMORY (MAXS) REGISTER. The MAXS register is a 24-bit pseudo register which is set by a field engineer to indicate the maximum size of the installed main memory. The MAXS register is addressable as a source register only.

MAXIMUM SIZE OF CONTROL MEMORY (MAXM) REGISTER. The MAXM register is a 24-bit pseudo register which is set by a field engineer to indicate the maximum size of the installed control (micro-instruction) memory. The MAXM register is addressable as a source register only.

TAPE CASSETTE INPUT (U) REGISTER.

The U register is a 16-bit register used to accumulate input (micro-instructions) from the console tape cassette. The U register is addressable only as a 16-bit source register.

Only the micro-instruction Register Move may access the U register. If data is not yet available in the register, the micro-operator is delayed. During TAPE mode the contents of this register is moved by a micro-program to the M register for execution if the previous micro-instruction did not move the contents out.

DATA REGISTER.

The DATA register is a 24-bit pseudo register which can act as a source or destination. It is used to transfer data to and from the respective I/O controllers. When used as a source, the DATA register accepts 24 bits of data from the respective I/O controller. When used as a destination, the DATA register transfers up to 24 bits of data from a source in memory to the I/O controllers.

COMMAND (CMND) REGISTER. -

The CMND register (command) is a 24-bit pseudo register which can act as a destination only. It is used to transfer I/O commands to I/O devices on the respective I/O controller.

NULL REGISTER.

The NULL register is a 24-bit pseudo register which contains zero's. It may be addressed only as a source register. Moves from NULL may be used for setting various registers to zero's.

ARITHMETIC AND COMBINATORIAL SECTION (24-BIT FUNCTION BOX).
The 24-bit arithmetic and combinatorial section (figure 3-11) is composed of a 24-bit arithmetic unit and a 24-bit combinatorial unit. It has as data inputs the contents of the X and Y registers and the carry flip-flop (CYF'). It also uses CPU (control for the arithmetic unit) and CPL (the 5-bit variable operand length) from the CP portion of the C register.

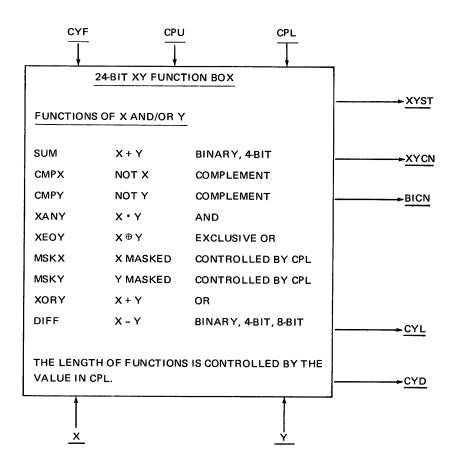


Figure 3-14. Arithmetic and Combinatorial Section

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

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

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

<u>CPU</u>	Unit Type	Possible CPL Values	Data Type
00 01 10	1-bit operands 4-bit operands Undefined	1 to 24 4, 8, 12, 16, 20, or 24	Binary 4 - bit binary
11	.8-b:it operands	8, 12, or 24	EBCD IC

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

Each of the following register contents is generated immediately and is available to the micro-programme'r.

SUM RESULT REGISTER.

The sum of the X, Y, and CYF registers (X, Y), and CYF are entered as inputs to the 24-bit function box) is produced and placed in the 24-bit pseudo SUM register. Zeros in the most-significant bit positions (left-most) of the 24-bit result are produced when the length given by CPL is less than 24. CYF should be set to zero at the start of a sum generation.

The adder logic performs differently for the four possible values in the CPU register as follows:

- a. If CPU = 00, the binary sum is produced.
- 6. If CPU = 01, the decimal sum is produced by considering the input to be comprised of up to six 4-bit units. The 4-bit units are added decimally and a carry is propagated from one 4-bit unit to the next most-significant 4-bit unit whenever the sum of two 4-bit units exceeds 9.

- 'c. If CPU = 10. the sum is undefined.
- d. If CPU = 11, the decimal sum is produced by considering the input to be comprised of up to three 8-bit unit's. The least-significant 4 bits of each unit are added in a manner similar to the 4-bit unit, including carrie's. The most-significant bits of each unit contain the inclusive-OR of the corresponding bits of the input's.

DIFFERENCE (DIFF) RESULT REGISTER.

The difference of the X, Y, and CYF registers (X, Y, and CYF are entered as inputs to the 24-bit function box) is produced and placed in the 24-bit pseudo DIFF register. Zeros in the most-significant bit positions (lef.t-most) of the 24-bit result are produced when the length given by CPL is less then 24. Difference results are generated by adding the contents of X to the one's complement of both Y and CYF. The complement values are hardware generated and do not alter the values of Y or CYF.

The absolute value of (Y + CYF) may be greater than the value in X and produces a negative result. Such negative results are in complement form and are indicated by CYD (carry difference) = 1. If CYD = 0, the difference result value is a positive number. CYD is generated from all 24 bits of X and Y and is not controlled by the value in CPL. CYD may be used to alter the value of CYF. Intermediate underflow may thus be remembered by CYF through iterations of a field.

The difference logic produces results under the control of the CPU register as follows:

- 'a. If CPU = 00, the binary difference is produce'd. Negative results are expressed in 2's complement form when CYD = 1.
- b. If CPU = 01, the decimal difference is produced by considering the input to be comprised of up to six 4-bit units. The 4-bit units are subtracted and any borrow is propagated from one 4-bit unit to the next as in decimal subtraction. Negative results are expressed in 10's complement form when CYD = 1.
- 'c. If CPU = 10, the difference is undefined.
- d. If CPU = 11, the decimal difference is produced by considering the input to be comprised of up to three .8-bit unit's. The least-significant 4 bits of each unit are subtracted in a manner similar to the 4-bit units including borrow's. The most-significant bits of each unit contain the inclusive-OR of the corresponding bits of the input's. Negative results are expressed in 10's complement form when CYD = 1.

AND/OR/EXCLUSIVE-OR (XANY, -XEOY, -XORY) RESULT REGISTER'S. The result of the appropriate logical function AND/OR/EXCLUSIVE-OR of the X and Y registers (X and Y are inputs to the 24-bit function box) is produced and placed in the appropriate 24-bit pseudo register (XANY, XEOY, -XORY). Zeros in the most-significant bit positions in the 24-bit

result are produced when the length given by CPL is less than 24.

COMPLEMENT X/COMPLEMENT Y (CMPX, CMPY) RESULT REGISTER. The one's complement of the appropriate register X or Y (X or Y and CPL are inputs to the 24-bit function box) is produced and placed in the appropriate 24-bit pseudo register, CMPX or CMPY, respectively. Zeros in the most-significant bit positions of the 24-bit result are produced when the length given by CPL is less than 24.

MASKED X/MASKED Y (MSKX, MSKY) RESULT REGISTER. The mask of the contents of the appropriate register X or Y (X or Y and CPL are inputs to the 24-bit function box) is produced and placed in the appropriate 24-bit pseudo register, MSKX or MSKY, respectively (see figure 3-12). The value of CPL determines the number of bits in MSKX or MSKY. Zeros in the most-significant bit positions of the 24-bit result are produced when the length given by CPL is less than 24.

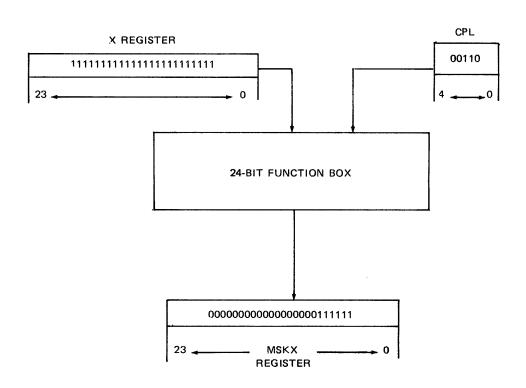


Figure 3-12. MSKX Register

BINARY CONDITIONS (BICN) REGISTER.
The BICN register (figure 3-13) is a 4-bit pseudo register which may act as a source only and indicates the following binary conditions: CYL (carry out level), CYD (borrow out level), CYF (carry flag), and LSUY (least significant unit of Y).

LSUY	CYF	CYD	CYL
3	2	1	0

Figure 3-13. BICN Register

The LSUY is true if: the least significant unit of the Y register is equal to 1 and CPU equals 00, or the right-most 4 bits of the least significant unit of the Y register is equal to 1001 (nine) and CPU is equal to 01 or 1/1. If CPU is equal to 10, LSUY is undefined.

The carry out and borrow out levels are a function of the inputs (X, Y, and CYF) and CPL. CYF may be manipulated by a special micro-instruction or as data in the high-order position of the CP register.

X/Y CONDITIONS (XYCN) REGISTER. The XYCN register (figure 3-14) is a 4-bit pseudo register which may act as a source only and contains the following relational conditions: X>Y (X is greater than Y, X<Y (X is less than Y), X=Y (X equals Y), and MSBX (most significant bit of X).

MSBX	X=Y	X <y< th=""><th>x>Y</th></y<>	x>Y
3	2	1	0

Figure 3-14. XYCN Register

The relational conditions are based on all the binary values of 24 bits of both X and Y. MSBX is true if the bit pointed to by CPL is a 1.

X/Y STATES (XYST) REGISTER.

The XYST register (figure 3-15) is a 4-bit pseudo register which may act as a source only and contains the following relational conditions: $X \neq 0$ (X is not equal to zero); $Y \neq 0$ (Y is not equal to zero); INT (interrupt) is true if any of the following conditions reflected by INCN, CC, and CD are true: missing port device, port interrupt, timer interrupt, I/O service request interrupt, console interrupt, memory parity error interrupt, and memory Write/Swap address out-of-bounds interrupt; LSUX (the least significant unit of X).

LSUX is true when the least significant unit of the X register is a 1 and CPU equals 00 or when the low-order 4 bits of the least significant unit of the X register equals 1001 (nine) and CPU is equal to 01 or 11. If CPU equals 10, LSUX is undefined.

LSUX	INT	Y#0	X ≠ 0
3	2	1	0

Figure 3-15. XYST Register

FOUR-BIT ARITHMETIC AND COMBINATORIAL SECTION (FOUR-BIT FUNCTION BOX). The 4-bit arithmetic and combinatorial section of the processor is used to generate most of the normally used functions between two 4-bit operands and can accept as an input the contents of any one of the following 4-bit registers. A second input is obtained from the Four-Bit Manipulate micro-instruction itself.

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

It has as possible results most of the functions between two operands. These include: SE-T, AND, OR, EXCLUSIVE-DR, binary modulo 16 sum, and binary modulo 16 difference functions. The results may only be directed to the same register which acted as the source. The sum and difference results can be tested for overflow and underflow, respectively.

The BICN, XYCN, XYST, FLCN, and INCN registers are available as source registers only.

The 4-bit combinatorial section also provides for the selective testing of any of the bits within the 4-bit registers and relative branching based on the results of the test (Bit-Test-Branch micro-instruction). The Skip When micro-instruction also tests any combination of bits, up to four, and branches on the result.

INTERRUPT CONDITIONS (INCN) REGISTER.

The INCN register (figure 3-16) is a 4-bit register which contains the interrupt conditions of particular interface lines between the processor and port interchange.

PORT MISSING DEVICE	PORT HIGH PRIORITY INTERRUPT	PORT INTERRUPT (NORMAL)	PORT LOCKOUT
3	2	1	0

Figure .3-16. INCN Register

Listed in table 3-1 is a summary of the various 4-bit conditions which are available to the micro-programme'r. (Refer to table 1-3 for the registers and functions in the column-row matrix.)

Table 3-1.
4-Bit Conditions

Register	Bit 3	Bit 2	Bit 1	Bit 0
BICN	LSUY		CYU	
XXCN	MSBX		>- ×	
XYST	LSUX		0 £ k	
E CN	FL=SFL		FL SFL	FL≠0
INCN	Port	Port	Port interrupt	
	device missing			
ည	Unassigned	Timer interrupt	I/O interrupt	Console interrupt
5	Memory parity	Memory	Memory	Memory
	error interrupt	Write/Swap	Read address	Write/Swap
		memory address	out-of-bounds	address
		out-of-bounds	interrupt	out-of-bounds
		override		interrupt

SECTION 4

MICRO-INSTRUCTIONS

GENERAL.

Micro-instruction routines can be related to read-only memories of past systems that contain a hard set of micro-programs which control most of the action taken by the processor. Because read-only memories are expensive and difficult to change once a system has been designed for a certain environment, it is, in most cases, destined to stay suited for that environment.

With the advent of Read/Write control memories and micro-programming techniques, Burroughs Corporation now provides the ability to change the system architecture to whatever is desired. By changing the microprogram (interpreter), a system efficient in COBOL can be changed to a system efficient in FORTRAN.

MICRO-INSTRUCTION DESCRIPTION.
All micro-instructions are 16 bits in length and are placed in the M register (micro-instruction) for execution. The M register (figure 4-1) is 16 bits long and is divided into four 4-bit sections which are not addressable as individual register's.

4 BITS			4 BITS			4 BITS			4 BITS						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							_16 B	ITS							

M REGISTER BIT NUMBER

Figure 4-1. M Register

The micro-instructions are referred to by name and hexadecimal value within the specific portion of the M register. For example, the Register Move micro-instruction has a hexadecimal value of 1 in the first 4-bit portion of the M register; the micro in machine code for a Register Move is 1. The Halt micro-instruction has a hexadecimal value of 1 in the last 4-bit portion of the M register; the micro in machine code for a Halt command is 0001. The other portions of the M register (if not part of machine code) are used for the selection of source and destination register and also for any variants associated with the respective micro-command's.

Source and destination register selection and other variables are determined by the hexadecimal value placed in the remaining 4-bit section's. For the Register Move command, these hexadecimal values point to the desired register as shown in table 4-1. If the hexadecimal value in the second 4-bit portion is equal to a 1100 (row C) and the most-significant two bits of the third 4-bit portion are equal to 01 (column 1), XYCN is selected as the source register for the Register Move micro-instruction.

Table 4-1

Register Select Switch
Column-Row Matrix

Row	Column 0	Column 1	Column 2	Column 3
0 1 2 3 4 5 6 7 8 9 A B C D E F	TA TB TC TD TE TF CA CB LA LB LC LD LE CC CD	FU FT FLC FLD FLE FLF BICN FLCN TOPM(*2) Reserved Reserved Reserved XYCN XYST INCN(*2) CPU	X Y T L A(MAR) M BR LR FA FB FL TAS CP MSM(*2) Console READ Console WRIT	SUM CMPX CMPY XANY XEOY MSKX MSKY XORY DIFF MAXS MAXM U MBR(*2) DATA CMND NULL

REGISTER MOVE (1NNN).

The function of this micro-instruction (figure 4-2) is to move the contents of the source register to the destination register. If the move is between registers of unequal lengths, the data is right justified with left zero bits supplied or with data truncated from the left, whichever is appropriate. The source register is unaffected unless it is also the destination.

MS and CPU are excluded as source register's.

When M is used as a destination register, the operation is changed to a Bit-OR which modifies the next micro-instruction to be execute'd. It does not modify the micro-instruction as stored in memory.

BICN, FLCN, XYCN, XYST, and INCN are excluded as destination register's.

MSM is not available as a destination register in STEP and RUN modé. It is permitted as a destination register in TAPE mode only.

² Not available on B 1710 Systems

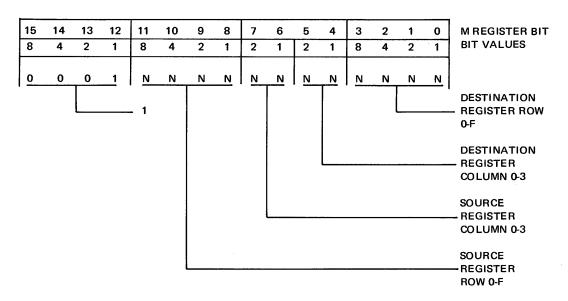


Figure 4-2. Register Move (1NNN)

The U register is excluded as a source register in STEP and TAPE mode but is permitted as a source register in the RUN mode.

All registers and pseudo registers in column select three are excluded as destination registers except MBR, -CMND, and DATA.

When CMND or DATA is designated as a source register, CMND and DATA are prohibited from being destination register's.

SCRATCHPAD MOVE (2NNY).

The function of this micro-instruction (figure 4-3) is to move the contents of either scratchpad memory to the selected register or the contents of the selected register to scratchpad memory as designated by the micro-instruction. The contents of the source register/scratchpad word is unchanged by the move.

MSM, U, and CPU are excluded as source register's.

When M is used as a destination register, the operation is changed to a Bit-OR which modifies the next micro-instruction to be executed but does not modify the micro-instruction as stored in memory.

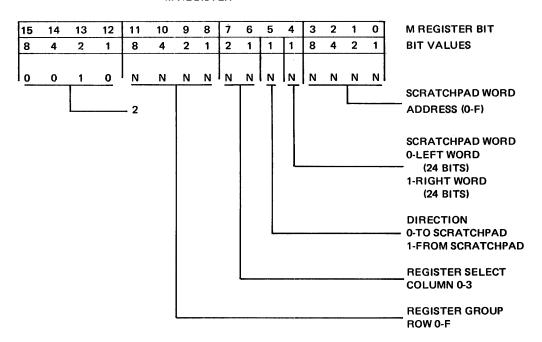


Figure 4-3. Schatchpad Move (2NNN)

BICN, FLCN, XYCN, XYST, INCN, and MSM are excluded as destination register's.

All registers and pseudo registers in column select three are excluded as destination registers except MBR, CMND, and DATA.

FOUR-BIT MANIPULATE (3NNN).

The function of this micro-instruction (figure 4-4) is to perform the Manipulate operation as specified by the micro-instruction variants on the addressed 4-bit register utilizing the literal specified in MF.

Only two columns of registers are available to the Four-Bit Manipulate micro-instruction (columns 0 and 1). Bit seven contains either a 0 or 1 to enable selection of the registers in columns 0 or 1 as expressed in table 4-1.

Bits four through six designate the specific operation to be performed on the addressed register and its relationship with the 4-bit literal expressed in bits zero through three. The possible operations permitted by respective variant values are given below.

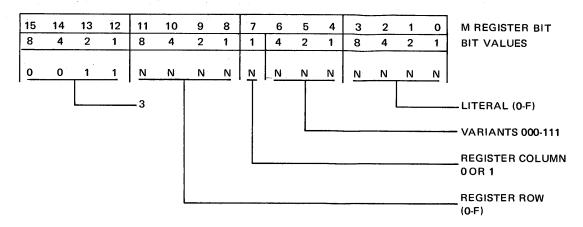


Figure 4-4. Four-Bit Manipulate (3NNN)

<u>Variant</u>	Operation
000	Set the addressed register equal to the value of the literal.
001	Set the addressed register to the logical AND of the addressed register and the literal.
010	Set the addressed register to the logical OR of the addressed register and the literal.
011	Set the addressed register to the logical EXCLUSIVE OR of the addressed register and the literal.
100	Set the addressed register to the binary sum of the addressed register and the literal.
101	Set the addressed register to the binary difference of the addressed register and the literal.
110	Set the addressed register to the binary sum of the addressed register and the literal, and skip the next micro-instruction if a carry is produced.
111	Set the addressed register to the binary difference of the addressed register and the literal, and skip the next micro-instruction if a borrow is produced.

BIT TEST BRANCH ON FALSE (4NNN).

The function of this micro-instruction (figure 4-5) is to test the designated bit (0-3) within the specified register and branch relative to the next instruction by the displacement magnitude value if the displacement sign bit is zero. If the displacement sign bit is one, a displacement value of zero is assumed and the next micro-instruction in line is executed. A displacement value indicates the number of 16-bit words from the next in-line micro-instruction. This value may not exceed 15 (hexadecimal F).

M REGISTER 7 0 M REGISTER BIT 15 12 11 10 9 R 4 3 14 13 5 2 1 BIT VALUES 4 2 8 4 2 8 4 2 1 8 1 1 2 1 1 1 Ν 0 0 Ν Ν Ν Ν Ν Ν Ν Ν LITERAL, RELATIVE DISPLACEMENT MAGNITUDE (0-F) DISPLACEMENT SIGN 0 - POSITIVE 1 - NEGATIVE REGISTER BIT TO BE TESTED 0-3 IN THE ADDRESSED REGISTER REGISTER COLUMN 0 OR 1 REGISTER ROW 0-F

Figure 4-5. Bit Test/Branch on False (4NNN)

Only the 4-bit registers shown in columns 0 and 1 in table 4-1 may be used as the addressed register on which this micro-instruction may act.

BIT TEST/BRANCH ON TRUE (5NNN).

The function of this micro-instruction (figure 4-6) is to test the designated bit (0-3) within the specified register and branch relative to the next instruction by the signed displacement value if the displacement sign bit is one. If the displacement sign bit is zero, a displacement value of zero is assumed and control passes to the next in-

line micro-instruction. a displacement value indicates the number of 16-bit words (micro-instructions) from the next in-line micro-instruction. This value may not exceed 15 (hexadecimal F).

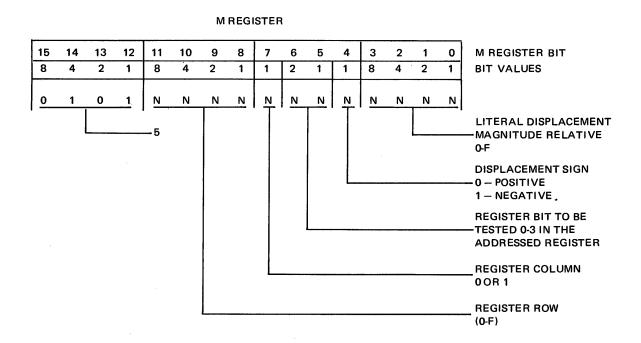


Figure 4-6. Bit Test/Branch on True (5NNN)

Only the 4-bit registers shown in columns 0 and 1 in table 4-1 may be used as the addressed register on which this micro-instruction may act.

SKIP WHEN (6NNN).

The function of this micro-instruction (figure 4-7) is to test only the bits in the designated register that are referenced by the 1 bits in the mask contained in the micro-instruction, to ignore all others, and to perform the action as specified by the variants.

The possible actions which may be specified by the 3-bit variant field are as follow's.

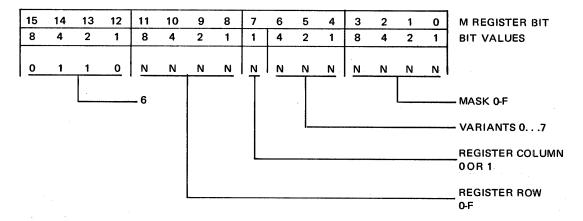


Figure 4-7. Skip When (6NNN)

<u>Variant</u>	<u>Operation</u>
000	If any of the referenced bits are 1, skip the next micro-instruction.
001	If all the referenced bits are 1, skip the next micro-instruction.
010	If the register is EQUAL to the mask, skip the next micro-instruction.
011	If all the referenced bits are 1, skip the next micro-instruction and CLEAR the referenced bits to zero.
100	If any of the referenced bits are 1, do not skip the next micro-instruction.
101	If all the referenced bits are 1, do not skip the next micro-instruction.
110	If the register is EQUAL to the mask, do not skip the next micro-instruction.
111	If any of the referenced bits are 1, do not skip the next micro-instruction.

NOTE

If the mask equals 0000, the (any) result is false. The skip is made if the variant is a 001 or 011 but is not made for 101 and 111.

BICN, FLCN, XYCN, XYST, and INCN are excluded as operand registers when the variant is equal to 011 or 111.

Only the 4-bit registers shown in columns 0 and 1 in table 4-1 may be used as the addressed register on which this micro-instruction is to act.

READ/WRITE MEMORY (7NNN).

The function of this micro-instruction (figure 4-8) is to move the contents of the addressed register to the main memory location that is specified in FA (field address register) or move the contents of the main memory location specified by FA to the addressed register, depending on the direction bit in the instruction.

M REGISTER 15 13 14 12 10 9 8 7 M REGISTER BIT 8 4 2 1 1 4 2 1 2 1 1 16 8 4 2 **BIT VALUES** 1 Ν Ν N Ν Ν N Ν N N N MAIN MEMORY FIELD LENGTH (0-23) FIELD DIRECTION 0 - POSITIVE 1 - NEGATIVE ADDRESSED GENERAL PURPOSE REGISTER 00-X REGISTER 01-Y REGISTER 10-T REGISTER 11-L REGISTER VARIANTS MOVE DIRECTION BIT 0-TO REGISTER 1-TO MEMORY

Figure 4-8. Read/Write Memory (7NNN)

If the value specified in the main memory field length is less than 24, the data from memory is right justified into the addressed register with left-most zero bits supplied while data from the register is truncated from the left when placed into memory. Values above 24 in the field length are reserved. The contents of the source is unchanged.

If the value of the main memory field length as given in the microinstruction is zero, the value in the CPL portion of the C (control) register is utilized instead.

The possible variant conditions are as follows: (FA = field address register and FL = field length register.)

<u>Variant</u>	Condition
000	No incrementation
001	Binarily increment FA (FA†)
010	Binarily increment FL (FL+)
011	Binarily increment FA and decrement FL (FA↑-FL↓)
100	Binarily decrement FA and increment FL (FA+-FL+)
101	Binarily decrement FA (FA+)
110	Binarily decrement FL (FL+)
111	Binarily decrement FA and FL (FA+-FL+)

MOVE 8-BIT LITERAL (8NNN).

The function of this micro-instruction (figure 4-9) is to move the 8-bit literal contained in the micro-instruction to the destination register. If the move is between registers of unequal lengths, the data is right justified with left zero bits supplied. Only registers X, Y, T, L, A, BR, LR, FA, FB, FL, TAS, and CP can be specified. The register row number is assumed to be two. MSM can be specified as a destination in TAPE mode only.

MOVE 24-BIT LITERAL (9NNN).

The function of this micro-instruction (figure 4-10) is to move the 24-bit literal given in the micro-instruction to the destination register. If the move is between registers of unequal lengths, the literal is truncated from the left. The least-significant 16 bits of the literal to be moved must be contained in the next word of memory following the Move 24-Bit Literal micro-instruction.

M REGISTER

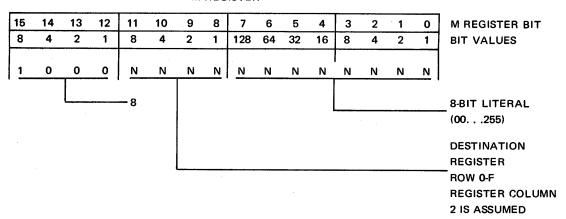


Figure 4-9. Move 8-Bit Literal (8NNN)

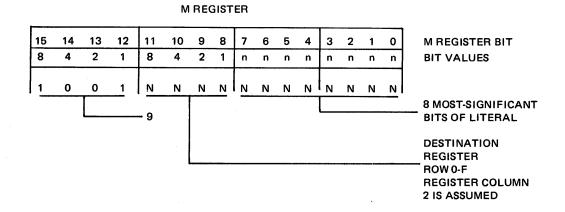


Figure 4-10. Move 24-Bit Literal (9NNN)

Only the registers X, Y, T, L, A, BR, LR, FA, FB, FL, TAS, and CP can be specified. The register row number is assumed to be two. MSM (only permitted on B 1726) can be specified as a destination register in the TAPE mode only.

NOTE
The least-significant 16 bits of the literal are found automatically by binarily incrementing the address (A) register by 16 (10000).

SHIFT OR ROTATE T REGISTER LEFT (ANNN).

The function of this micro-instruction (figure 4-11) is to shift or rotate the T register left by the number of bits specified and then move the 24-bit result to the destination register. If the move is between registers of unequal lengths, the data is right justified with data truncated from the left. For the shift operation, zero fill on the right and truncation on the left occur. The contents of the source register is unchanged unless it is also the designated destination.

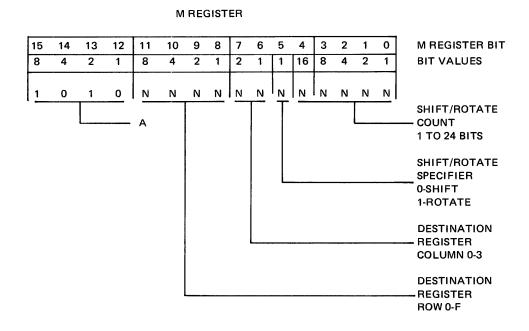


Figure 4-11. Shift or Rotate T Register Left (ANNN)

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

When M is used as a destination register, the operation is changed to a Bit-OR which modifies the next micro-instruction to be executed but does not modify the micro-instruction as stored in control memory.

BICN, FLCN, XYCN, XYST, INCN, and MSM are excluded as destination register's.

EXTRACT FROM T REGISTER (BNNN).

The function of this micro-instruction (figure 4-12) is to rotate the T-register left by the number of bits specified and then extract the

number of bits specified and move the results to the destination register. If the extract bit count is less then 24, the data is right justified with left zero bits supplied. The contents of the T register is unchanged unless it is also the designated destination. A rotate value of 24 is equivalent to zero.

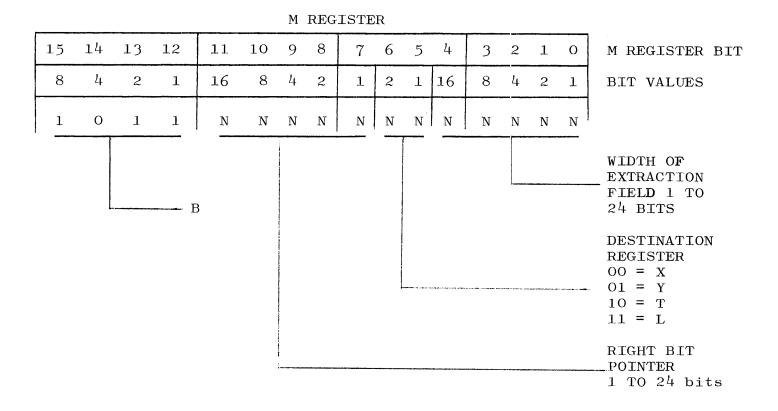


Figure 4-12. Extract from T register (BNNN)

If the right bit pointer value is equivalent to zero, the value expressed in CPL is used instead.

BRANCH.

BRANCH FORWARD (CNNN).

The function of this micro-instruction (figure 4-13) is to fetch the next micro-instruction from the location obtained by binarily adding the displacement value given in the micro-instruction to the address of the next in-line micro-instruction. The displacement value indicates the number of 16-bit words.

BRANCH BACKWARD (DNNN).

The function of this micro-instruction (figure 4-14) is to fetch the next micro-instruction from the location obtained by binarily subtracting the displacement value given in the micro-instruction from the address of the next in-line micro-instruction. The displacement value indicates the number of 16-bit words.

M REGISTER

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	M REGISTER BIT
I	8	4	2	1	(2	048)										(0)	BIT VALUES
	1	1	0	0	N	N	N	N	N	N	N	N	N	N	N	N	
				C													BINARY RELATIVE DISPLACEMENT VALUE 04095

Figure 4-13. Branch Forward (CNNN)

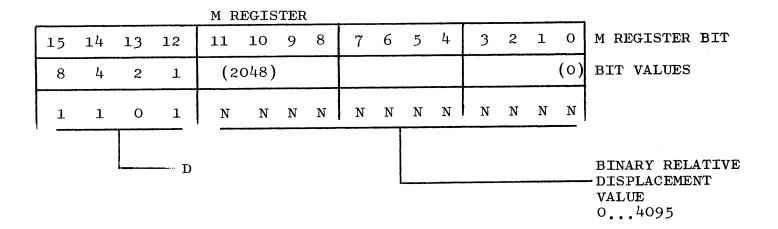


Figure 4-14. Branch Backward (DNNN)

NOTE

For both of the above micro-instructions the address of next in-line micro-instructions is determined by hardware by binarily incrementing the address (A) register by 16 (10000).

CALL.

CALL FORWARD (ENNN).

The function of this micro-instruction is to push the address of the next in-line instruction into the ASTACK and then Fetch the next micro-instruction from the location obtained by binarily adding the displacement value given in the micro-instruction to the address of the next in-line micro-instruction. The displacement value indicates the number of 16-bit words.

NOTE

When the A address is stored in the A stack it is multiplied by 16 and stored as a bit address.

M REGISTER 15 14 13 12 11 10 5 M REGISTER BIT 3 2 1 8 4 2 1 (2048)(0)BIT VALUES 1 0 1 N N N N N N N N BINARY RELATIVE ADDRESS MAGNITUDE - E DISPLACEMENT VALUE 0...4095

Figure 4-14A. Call Forward (ENNN)

CALL REVERSE (FNNN).

The function of this micro-instruction is to push the address of the next in-line instruction into the ASTACK and then fetch the next micro-instruction from the location obtained by binarily subtracting the displacement value given in the micro-instruction from the address of the next in-line micro-instruction. The displacement value indicates the number of 16-bit words.

NOTE

When the A address is stored in the A stack, it is multiplied by 16 and stored as a bit address.

M REGISTER

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	M REGISTER BIT
8	4	2	1	(2	048)			·		í	- :				(0)	BIT VALUES
1	1	1	1	N	N	N	N	N	N	N	N	N	N	N	N	
	F													- AD DI	DRE SPL	Y RELATIVE SS MAGNITUDE ACEMENT O4095

Figure 4-14B. Call Reverse (FNNN)

SWAP MEMORY (02NN).

The function of this micro-instruction (figure 4-15) is to swap up to 24 bits of data from main memory with the data in the specified register. If the value of the main memory field specified is less than 24 bits, the data from memory is right justified into the specified register with left zero bits supplied while the data from the register is truncated from the left as it is placed into memory.

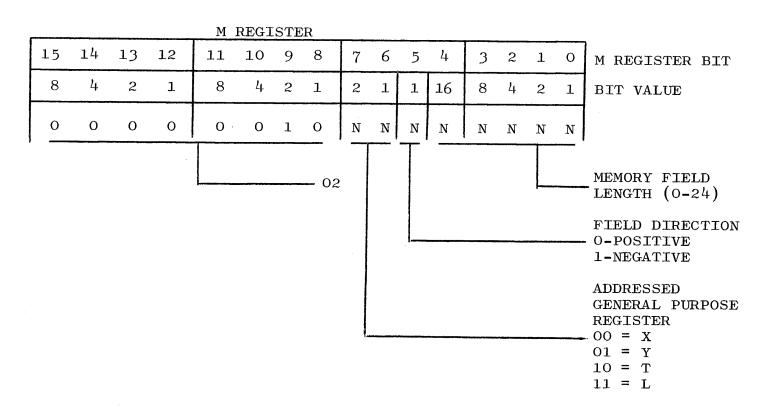


Figure 4-15. Swap Memory (02NN)

The field address register (FA) contains the address of the main memory location affected while the field direction sign and field length are given in the instruction. If the value of the main memory field length given in the instruction is zero, the value contained in CPL is used.

NOTE

This micro-instruction is available only on the B 1726 System.

CLEAR REGISTER (03NN).

The function of this micro-instruction (figure 4-16) is to clear the designated register to zero if its respective flag bit is set. Registers not designated are unaffected.

NOTE

This micro-instruction is available only on the B 1726 System.

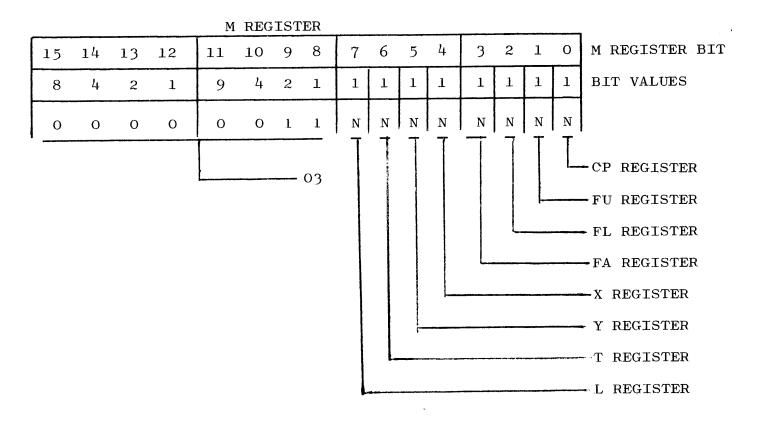


Figure 4-16. Clear Register (03NN)

SHIFT/ROTATE X OR Y (04NN). The function of this micro-instruction (figure 4-17) is to shift or rotate the X or Y register either left or right by the number of bits specified by the shift/rotate count portion of the micro-instruction. If the value of the shift/rotate count given in the micro-instruction

is zero, the shift or rotate is determined by CPU as follows.

$\underline{\text{CPU}}$	Shift/Rotate Count
00	1 bit
01	4 bits
10	Undefined
11	8 bits (not available on B 1710 Systems)

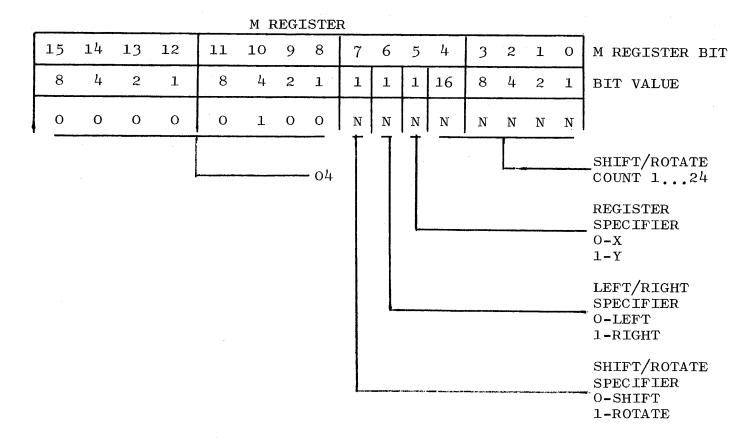


Figure 4-17. Shift/Rotate X or Y (04NN)

On the shift left the most-significant bits are truncated and zero bits occur on the right. On the shift right the least-significant bits are truncated and zero bits occur on the left.

SHIFT/ROTATE X AND Y (05NN).

The function of this micro-instruction (figure 4-18) is to concatenate the X and Y registers to become one 48-bit register with the X register being the left-most or most-significant. The combined registers are either shifted or rotated left or right by the number of bits specified. If the shift/rotate count is zero, the amount to be shifted or rotated is determined by the contents of CPU as follows:

CPU	Shift/Rotate
00	1 bit
01	4 bits
10	Undefined
11	8 bits (not available on B 1710 Systems)

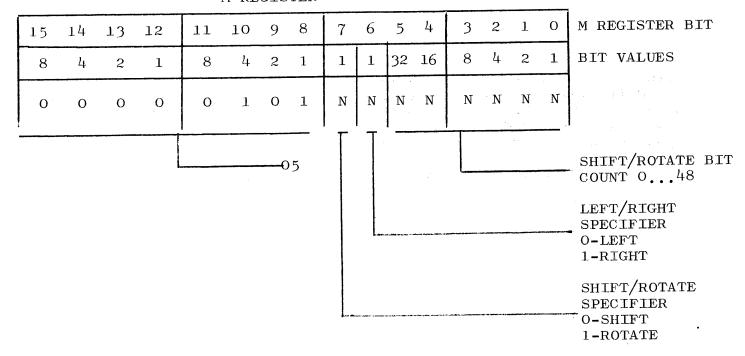


Figure 4-18. Shift/Rotate X and Y (05NN)

If the micro-instruction indicates a shift left operation, zero fills on the right and truncation on the left occur. For the shift right operation, zero fills on the left and truncation on the right occur.

COUNT FA/FL (06NN).
The function of this micro-instruction (figure 4-19, depending on the variant field, is to binarily increment or decrement the designated register(s) by the value of the literal contained in the micro-instruction or by the value of CPL if the literal is zero. The field address (FA) register and the field length (FL) register are the only two registers affected by this micro-instruction. Results of values of greater than 2Y for FL are undefined.

						M R	EGI	STEF							ı		
ſ	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	M REGISTER BIT
	8	4	2	1	8	4	2	1	4	2	1	16	8	4	2	1	BIT VALUES
	0	0	0	0	0	1	1	О	N	N	N	N	N	N	N	N	
1	06																-LITERAL (0-31)
										L							COUNT VARIANTS

Figure 4-19. Count FA/FL (06NN)

Neither overflow nor underflow of FA is detected. The value of FA may exceed its maximum value and wraparound. Overflow of FL is also not detected. The value of FL may exceed its maximum value and wraparound. Underflow of FL is detected and does not wraparound, in which case a binary value of zero is left in FL.

The possible variant conditions are as follows.

<u>Variant</u>	Condition
000	No operation
001	Binarily increment FA (FA+)
010	Binarily increment FL (FL+)
011	Binarily increment FA and decrement FL (FA+-FL+)
100	Binarily decrement FA and increment FL (FA+-FL+)
101	Binarily decrement FA (FA+)
110	Binarily decrement FL (FL+)
111	Binarily decrement FA and FL (FA+-FL+)

EXCHANGE DOUBLEPAD WORD (07NN).

The function of this micro-instruction (figure 4-20) is to move the 48 bits of the field (F) definition register to a holding register (refer to note) and then move 48 bits from a source register in scratchpad to the F register. After this is accomplished the 48 bits are taken from the holding register and placed into the specified destination register in scratchpad. A doublepad word is equal to 48 bits of scratchpad memory.

M REGISTER

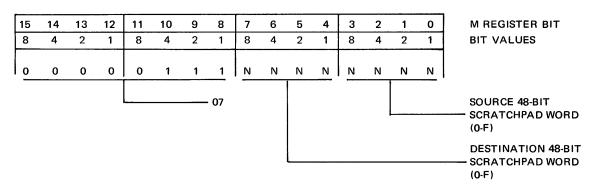


Figure 4-20. Exchange Doublepad Word (07NN)

NOTE

The holding register is imbedded in the hardwaré. It is not specified within the micro-instruction and is not accessible to the micro-programmer.

SCRATCHPAD RELATE FA (08NN).

This micro-instruction (figure 4-21), depending on the sign of operation, binarily adds or subtracts the contents of the left half of the addressed portion of scratchpad memory (scratchpad locations S01A

through S15A) to/from the contents of the 24-bit field address (FA) register, and places the results in FA.

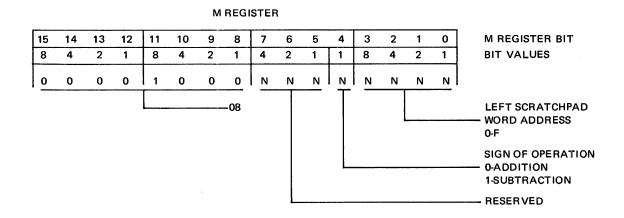


Figure 4-21. Scratchpad Relate FA (08NN)

MONITOR (09NN).

This micro-instruction (figure 4-22) skips to the next sequential instruction. During the time this skipping is being executed the two least-significant bits of the micro-instruction are decoded and certain levels which are testable by a field engineer are set.

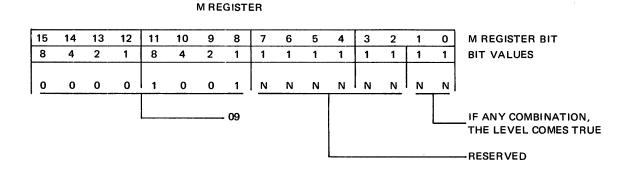


Figure 4-22. Monitor (09NN)

CASSETTE CONTROL (002N).

The function of this micro-instruction (figure 4-23) is to control the cassette tape movement.

M REGISTER

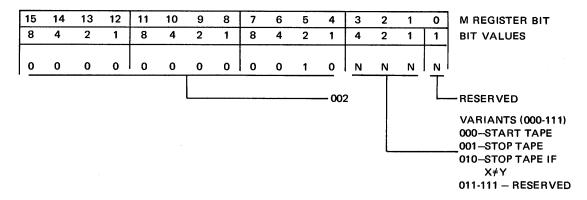


Figure 4-23. Cassette Control (002N)

All tape stop variants cause the cassette tape to halt in the next available tape gap, not immediately. If the processor is in the TAPE mode and a stop tape cassette condition occurs, the processor also halt's.

BIAS (003N).

The function of this micro-instruction (figure 4-24) is to set the CPU register to the value of 1, 3, or 0 depending on whether the value of the field unit (FU) portion of the F register is equal to 4, 8, or any other value 0-F, respectively.

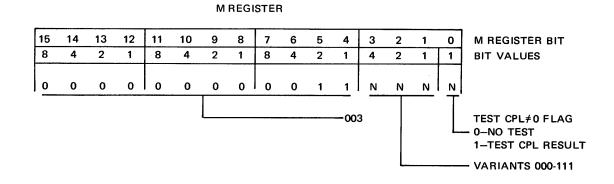


Figure 4-24. Bias (003N)

Depending on the variants the value of CPL is set to the value denoted or to the smallest of the values listed below.

<u>Variant</u>	<u>Values</u>							
000	FU							
001	24 or FL							
010	24 or SFL							
011	24 or FL or SFL							
100	NOP							
101	24 or CPL or FL							
110	NOP							
111	Reserved							

When the variant field is equal to 010, the value of CPU is determined by SFU instead of FU. SFU is contained in the first four bits of scratchpad word SOOB. If the test flag is equal to 1 and CPL is not equal to zero, the next micro-instruction is skipped.

STORE F IN DOUBLEPAD WORD (004N').

The function of this micro-instruction (figure 4-25) is to store the contents of the F register (FA portion = 24 bits and FB portion = 24 bits) in the designated scratchpad word. FA goes to the left half (A) of the scratchpad word and FB to the right half (B) of the scratchpad word.

M REGISTER 13 11 M REGISTER 9 6 5 3 2 1 0 8 4 2 1 8 4 1 2 8 4 2 1 8 4 2 1 BIT VALUES lο 0 0 0 0 0 0 0 n 1 O 0 | N Ν Ν -004 SCRATCHPAD DOUBLE WORD ADDRESS

Figure 4-25. Store F into Doublepad Word (004N)

NOTE

This micro-instruction is available only on the B 1726 System.

LOAD F FROM DOUBLEPAD WORD (005 N).

The function of this micro-instruction (figure 4-26) is to move all 48 bits of the designated scratchpad word to the F register. The left half (A) of the doublepad word is placed into the FA portion of the F register, and the right half (B) of the doublepad word is placed in the

FB portion of the F register.

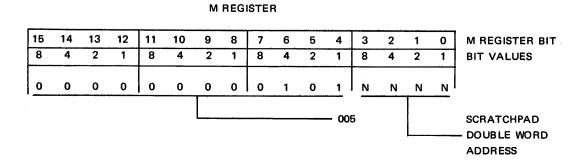


Figure 4-26. Load F from Doublepad Word (005N)

NOTE
This micro-instruction is available only on the B 1726 System.

SET CYF (006N). The function of this micro-instruction (figure 4-27) is to set CYF (carry flip-flop) as specified by the variant.

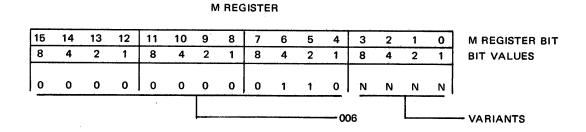


Figure 4-27. Set CYF (006N)

The available variants are represented below; all others are reserved.

<u>Variant</u>	Action									
0000	Set CYF to 0.									
0001	Set CYF to 1.									
0100	Set CYF to CYL (carry total for sums).									
1000	Set CYF to CYD (carry borrow from difference).	,								

CYL is generated under control of the length in CPL, and CYF is an input to the arithmetic logic along with the X and Y register's.

HALT (0001).
The function of this micro-instruction (figure 4-28) is to halt the sequence of executable micro-instruction's.

M REGISTER

M REGISTER BIT BIT VALUES O n n O 0 | O

Figure 4-28. Halt (0001)

The system halts with the next in-line micro-instruction in the M register and the address of the second in-line micro-instruction in the address (A) registers. Whatever register is pointed to by the console controls is displayed on the register display lights.

OVERLAY CONTROL MEMORY (0002). The function of this micro-instruction (figure 4-29) is to overlay control memory from main memory. This is desired because of faster micro-instruction execution time in control memory.

M REGISTER

M REGISTER BIT 8. BIT VALUES Я 0 | n 0 | ń

Figure 4-29. Overlay Control Memory (0002)

The address for data from main memory is obtained from the field address (FA) register, and the address where data is to be placed into control memory is contained in the L register. The length of the data overlay bits is taken from the FL register, and overlaying occurs until FL=0 or the maximum amount of control memory is exceeded.

The execution of the micro-instructions proceeds as follows:

- a. A is moved to TAS.
- b. L is moved to A.
- c. The first 16 bits are read from main memory and stored in control memory. Register FL is decremented by 16 bits, FA incremented by 16 bits, and A incremented by one word.
- d. Step c is repeated until one of the two above mentioned ending conditions are met.

NOTE
This instruction is available only on the B 1726 System.

NORMALIZE X (0003).

The function of this micro-instruction (figure 4-30) is to shift the X register left while counting FL down until FL is equal to zero or the bit in X which is referenced by CPL is equal to one. Zero bits are shifted into the least-significant end of X during the shift operation.

	2						₹ ,	STEF	REG	М						
										- :	2.5					
M REGISTER BI	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
BIT VALUES	1	2	4	8	1	2	4	8	1	2	4	8	1	2	4	8
	1	1	0	0.	0	0	0	0	0	0	0	0	.0	0	0,	0

Figure 4-30. Normalize X (0003)

If CPL is equal to 1, the least-significant bits of X are referenced; if CPL is equal to 24, the most-significant bits of X are referenced.

NO OPERATION (0000)

The function of this micro-instruction (figure 4-31) is to reserve 16 bits of memory (the length of one micro-instruction) and to utilize one machine cycle.

When this micro-instruction is encountered, a fall-through to the next sequential micro-instruction occur's.

M REGISTER

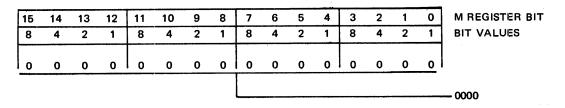


Figure 4-31. No Operation (0000)

SECTION 5

B 1726 I/O SUBSYSTEM

GENERAL.

The I/O subsystem on the B 1726 System is designed to enhance simultaneity of I/O operation. The MCP creates the various I/O descriptors and sends an interrupt message to the desired I/O controller. At the completion of the I/O event, either normal or abnormal, the I/O controller returns the result descriptor which contains descriptive information concerning the I/O operation just completed. Immediately following the result descriptor there may exist the address (link) of another I/O descriptor which is initiated by the MCP. This provides for chaining of I/O's.

I/O COMPONENTS.

The I/O subsystem has an ordered relationship among its components. From peripheral device to main memory the order is as follows. A peripheral device (disk cartridge, printer, etc.) interfaces and is controlled by an I/O controller which interfaces to the processor. Up to eight I/O controllers may interface with the processor which is connected to main memory. (See figure 5-1.)

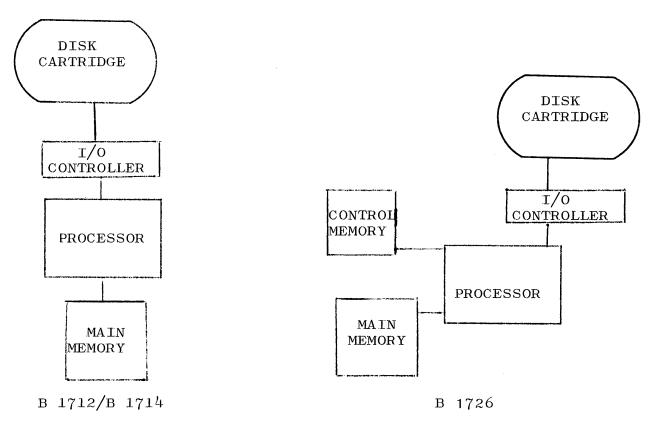


Figure 5-1. I/O Subsystem

I/O CONTROLLERS.

The peripheral I/O controllers are physically separate, logically inde-

pendent devices. The peripheral side of each I/0 controller is unique for each type of device.

Each I/O controller is a special-purpose device specifically designed to interface one type of peripheral to the system. In most cases an I/O controller interfaces to a single device; in a few cases a single I/O controller interfaces with two or more identical devices on a time-shared basis (i.e., serial disk drives or serial tape drives). I/O controllers have individual hardware buffers to aid in simultaneity of I/O operations.

I/O CONTROLLER FUNCTIONS.

The functions performed by an I/O controller are to accept an I/O descriptor from the processor and to determine if it is a valid I/O descriptor. In the case of an invalid I/O descriptor an appropriate result descriptor is returned. Otherwise, the controller signals the processor that it is now ready to receive data. Hardware translation between the peripheral data codes and the internal memory code is performed by the I/O controller (i.e., BCL card code to EBCDIC).

At the conclusion of an I/O operation the I/O controller returns a result descriptor to the processor.

SECTION 6

I/O DESCRIPTORS

GENERAL.

The purpose of input/output descriptors is to communicate control information to the various I/O controllers which perform the desired operation on the peripheral.

I/O DESCRIPTOR FORMAT.

Each I/O descriptor (figure 6-1) consists of a series of 24-bit field's. The number of these 24-bit fields varies with the peripheral to which the I/O descriptor is directed. These differences are explained in section 7.

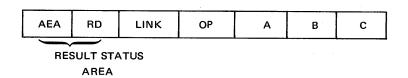


Figure 6-1. I/O Descriptor Format

The fields shown above are each 24 bit's. The maximum size of any I/O descriptor is 168 bits; the minimum size is 96 bit's. Fields A, B, and C may be omitted depending on the I/O descriptor initiated and the peripheral involved. The various fields are defined in the following paragraph's.

RESULT STATUS AREA.

The result status area (RSA) consists of two 24-bit fields: the actual ending address (AEA), which is the ending binary address + 1 of the last data bit affected by the operation; and a 24-bit result descriptor (RD), which indicates the status of the just terminated I/O operation. The RSA is returned by the I/O controller at the completion of each I/O operation.

LINK.

The LINK field contains an address which points to the beginning address of a result descriptor which is contained in the next desired I/O descriptor. Linking is further explained later in this section.

OP.

The operation (OP) field contains the desired I/O command and its variant's.

Δ.

The A field contains the beginning binary address of the input/output main memory area.

- B. The B field contains the ending binary address + 1 of the input/output main memory area.
- C.
 The C field is used in conjunction with disk I/O's only and contains the absolute binary disk addres's.

I/O DESCRIPTOR CHAINING.

The ability to link or chain together I/O descriptors provides for maximum utilization of the processor and peripheral devices. Shown in figure 6-2 is a sample chain of I/O descriptors which are created by the MCP because a user program has requested three output print buffer's.

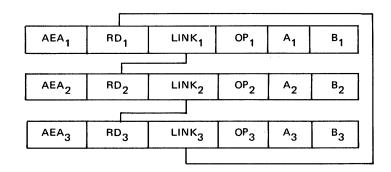


Figure 6-2. Sample Chain of I/O Descriptors

LINK-sub1 contains the address of RD-sub2, LINK-sub2 contains the address of RD-sub3, and LINK-sub3 contains the address of RD-sub1. This forms a continuous chain of I/O descriptors, one associated with each of the requested buffer's.

The address provided to the I/O controller by the processor contains the address of RD-sub1. This action locks in the particular I/O controller and device to the proper chain of I/O descriptor's. At the address provided, the I/O-complete bit of that result descriptor is examined; if it is a zero, the I/O descriptor is executed and a result descriptor is returned to location RD-subl. If no exception condition has occurred, the I/O descriptor pointed to by LINK-sub2 is executed, and so on, thus providing a continuous chain of I/O descriptor's. If the I/O-complete bit is set to one when examined, a pause of a specified number of milliseconds occurs and re-examination of the I/O complete-bit occur's. The I/O-complete continues to be examined until it becomes zero; in this manner execution of the I/O descriptors is under MCP control. The MCP can get ahead of the actual execution of the I/O descriptors; that is, the MCP may be releasing buffers faster than the I/O descriptors can be executed. The I/O controllers can never get ahead of the MCP because each I/O controller remains locked

to each I/O descriptor until the I/O is complete. A stop I/O descriptor is provided by which any I/O controller may be stopped from executing an I/O descriptor. If an error condition occurs on any given I/O operation, that particular chain is broken and the MCP is notified.

This process is completely under control of the MCP. The MCP examines the I/O-complete bits and performs the above operation.

DISK FILE AND DISK PACK CHAINING OPERATION.

Since disk I/O descriptors all contain an absolute disk address, there is no danger of an operation being performed out of logical sequence as long as the controller does not attempt to execute a descriptor that is already complete. Also, since disk addresses may be in the chain for many different programs and files, the controller must not wait for any given descriptor to be released. On discovering an I/O descriptor that is already complete, the control must proceed to the next I/O descriptor immediately.

As a consequence of this mode of operation, the controller could conceivably be continually searching the same chain of I/O descriptors for one that is executable and therefore could be using all available memory accesses. For this reason, the test/pause I/O descriptor has been implemented.

In summary, once a disk file is opened and its descriptors are linked into the subsystem chain, they may or may not be executable depending on the setting of the I/O-complete bit in the result descriptor, which in turn is dependent on the read, write, and seek instructions in the object program.

MAGNETIC TAPE OPERATION.

Because of the serial nature of magnetic tape operation, it is slightly more complex than that of disk. There is one chain of I/O descriptors for the entire tape subsystem, similar to the disk operation previously described. The descriptors must, however, be executed in logical sequence. To accomplish this, the lock descriptor has been implemented in the tape controls only.

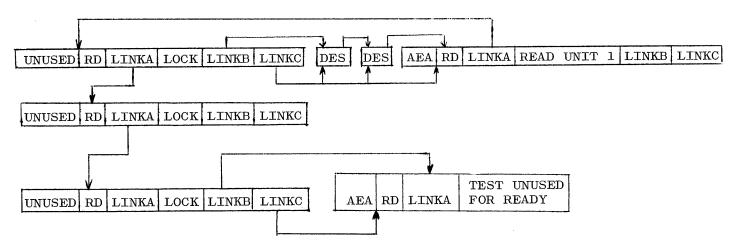
The lock descriptor contains three link fields, as shown in figure 6-3.

There is one lock descriptor in the subsystem chain for each tape unit in the system. All I/O descriptors for that unit are linked, directly or indirectly, from the lock descriptor for the unit. LINK-A of the lock descriptor always contains the memory address of the lock descriptor for the next unit in the subsystem or the address of a pause descriptor. There is one pause descriptor for the entire subsystem. LINK-B of the lock descriptor always contains the memory address of the first, physical I/O descriptor for the unit. This field is initialized by the MCP when the file is opened. LINK-C contains the memory address of the logically next I/O descriptor to be executed on the unit. LINK-C is initialized by the MCP but is maintained by the magnetic tape controller.

STOP I/O DESCRIPTOR.

 $\overline{All\ I/0\ controllers}$ are capable of receiving and recognizing a stop I/0 descriptor. When this descriptor is received, the I/0 controller stops operation and idles. The main use of this descriptor is to allow the MCP the ability to terminate a chaining operation.

The format for a stop I/O descriptor is shown in figure 6-4.



LOCK DESCRIPTOR

LINKA ALWAYS POINTS TO NEXT LOCK DESCRIPTOR LINKB ALWAYS POINTS TO FIRST NON-LOCK DESCRIPTOR. LINKC ALWAYS POINTS TO CURRENT NON-LOCK DESCRIPTOR.

Figure 6-3. Typical Magnetic Tape Chain

vvv	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
3 BITS	21 BITS

VVV = 111 - STOP I/O COMMAND

R = RESERVED

Figure 6-4. Stop I/O Descriptor

DISPATCH OF I/O DESCRIPTORS.

The Dispatch operation is used to send the address of an I/O descriptor to a device on an I/O port. Information is also received from the I/O port via this operation.

SECTION 7

INPUT/OUTPUT CONTROLLERS

GENERAL.

This section describes the I/O controllers which may be connected to the B 1700 series of computer's. All input/output operations are initiated but not executed by the processor. The execution of any specified I/O operation is accomplished by the respective I/O controller's. The I/O operation may be executed simultaneously with a processor operation or other previously initiated I/O operation's. The type of I/O operation is determined by an I/O descriptor that is transferred to the I/O controller. At the conclusion of an I/O operation, a result descriptor that specifies any exception conditions and other pertinent information is returned by the I/O controller. All I/O's are initiated by the Burroughs Master Control Program.

CARD READER CONTROLLER.

The card read I/O controller executes the card reader as initiated by the MCP. It reads data into ascending memory locations beginning with the location specified by the A addres's.

CARD READ DESCRIPTOR (EBCDIC).

This I/O descriptor controller (figure 7-1) is capable of translating the standard EBCDIC card code into the internal 8-bit EBCDIC code. When reading EBCDIC, 256 card-hole combinations are valid. All invalid combinations are translated to the ? character. The card is read into ascending memory locations beginning with the location specified by the A address and continuing to, but not including, the terminal location specified by the B addres's. The contents of each card column occupies one 8-bit character position in memory. Reading is also terminated after 80 columns of information have been transferred to main memory.

vvv	Т	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	Α	В	
3 BITS	1	20 BITS	24 BITS	24 BITS	

Figure 7-1. Card Read I/O Descriptor

Referring to figure 7-1:

- VVV = 000 card read operation codé.
- T = 0. card read EBCDIC. Ъ.
- T = 1 card read binary. Ć.
- R = reserved.
- é. A = beginning address of data.
- B = ending address of data + 1.

CARD READ DESCRIPTOR (BINARY). The card read controller is also capable of a binary read operation in which each hole in a card column is interpreted as a one bit and each non-hole is interpreted as a zero bit. The binary card image is stored in ascending memory locations beginning at the locations specified by the A address continuing until, but not into, the end location specified by the B address, or until 80 columns of information have been

read, whichever occurs firs't. Each column is stored as 12 bits of data in main memory. (See figure 7-1.)

CARD TEST.

This I/O descriptor (figure 7-2) returns a result descriptor indicating that the test operation has been completed and the card reader is ready (operation complete) or the reader is not ready (operation not complete or exception). The test result descriptor also contains the controller ID.

vvv	TT	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
3 BITS	2	19 BITS

Figure 7-2. Card Read Test Descriptor

Referring to figure 7-2:

VVV = 100 - test card reader and controller.

b. TT = 00 - result store immediately.

ć. TT = 10 - result is delayed until card reader is ready.

 \dot{d} . TT = 01 - result is stored when card reader is not ready.

R = reserved.

CARD READ RESULT DESCRIPTOR.

At the completion of a card read operation, the controller returns a result descriptor which is described as follows:

Bit 0 - I/O completé.

Bit 1 - exception condition (any bit on in bits 3-16).

Bit 2 - not read'y. ć.

ď. Bit 3. - validity errof.

Bit 4 - memory access error. é.

Bit 5 - reserved. f.

Bit 6. - read check. ģ.

h. Bits 7-16 - reserved.

Bits 17-23 - controller ID = 0101010 (test I/O only).

CARD PUNCH CONTROLLER.
The card punch I/O controller executes card punch I/O descriptors and punches data from ascending memory locations beginning at the location specified by the A address of the operand until 80 columns of information are punched. It allows EBCDIC card punch operation and a binary punch operation.

Information is transferred to the punch bit-serially for each punch position, with 12 separate 80-bit transfers being required to punch a card. Transfer of a given row to the punch consists of 40 memory accesses, obtaining punch information for two columns per access. The number of memory accesses required to punch a card is 480. A full card is always punched. The source program need not be concerned with the transfer of data to the punch.

The controller provides a punch echo check which checks if fewer than 80 data bits are received for each row or if fewer than 12 row cycles are counted. It also checks whether the number of punched holes agrees with the number of bits in the original data received by the I/O control unit.

CARD PUNCH DESCRIPTOR (BINARY).

A card is punched from ascending main memory locations beginning with the location specified by the A address (see figure 7-3). Punching is terminated after the punching of 80 columns, where each hole in A to be punched in a card column is represented by a one bit in memory.

vvv	т	şs	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	А	
3 BITS	1	2 BITS	18 BITS	24 BITS	

Figure 7-3. Card Punch I/O Descriptor

Referring to figure .7-3:

- a. VVV = 010 card punch operation command.
- 6. T = 0 card punch EBCDIC.
- c. T = 1 card punch binary.
- d. SS = 00 normal stacker.
- é. SS = 01 auxiliary stackér.
- f. SS = 10 error stacker.
- g. R = reserved.
- ĥ. A = beginning address of data.

CARD PUNCH DESCRIPTOR (EBCDIC).

A card is punched from ascending main memory locations beginning with the address specified in A (see figure 7-3). One column is punched for each EBCDIC memory character and 80 columns are always punched.

CARD PUNCH TEST DESCRIPTOR.

The card punch and card punch controller are tested, and a result descriptor containing the controller ID number is returned (see figure 7-4).

vvv	Т	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
3 BITS	1	20 BITS

Figure 7-4. Card Punch Test Descriptor

Referring to figure 7-4:

- a. VVV = 100 test the card punch and controller.
- b. T = 0 the result is stored immediately.
- \acute{c} . T = 1 -the result is delayed until the reader is ready.
- á. R = reserveá.

CARD PUNCH RESULT DESCRIPTOR.

At the completion of an I/O operation the I/O controller retains a result descriptor, for the just-completed I/O, in the second 24 bits of the result status area. The result descriptor information for the card punch is as follows:

- a. Bit 0 I/O completé.
- b. Bit 1 exception condition (any bit on in bits 3-16).
- ć. Bit 2 not ready.
- d. Bit 3 punch check error.
- 'e. Bit 4 memory access erro'r.
- f. Bit 5 memory parity error.
- g. Bits 6-16 reserved.
- h. Bits 17-23 controller ID = 0000100 (test I/O only).

PAPER TAPE READER CONTROLLER.

This controller stores one 8-bit character of data per memory cycle. If the number of characters specified is not read, spaced, or back-spaced because of termination by the reader, the short record bit (bit 9 of the result descriptor) is set. If the termination is caused by the end-of-tape or beginning-of-tape reflective strip, the appropriate EOT or BOT is set in the result descriptor.

PAPER TAPE READ DESCRIPTOR.

A record is read from the paper tape reader into ascending memory locations beginning at the location specified by the A address and continuing to, but not into, the address specified by the B address (see figure 7-5). If the T (type) variant of the read I/O contains a zero, eight bits received from the reader are transferred to memory via a BCL/EBCDIC translator. If the T variant contains a one, no translation occurs and eight bits received from the reader are transferred to memory.

vvv	т	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
3 BITS	1	20 BITS



Figure 7-5. Paper Tape Read, Space, and Backspace I/O Descriptor

Referring to figure .7-5:

- VVV = 000 paper tape read operation command.
- VVV = 101 paper tape spacé.
- VVV = 110 paper tape backspace.
 T = 0 translate BCL to EBCDIC and check for odd parity. ď.
- T = 1 no translation and check for even parity. é.
- R = reserved for unit addres's.
- A = beginning address of data.
- B = ending address of data + 1.

PAPER TAPE SPACE DESCRIPTOR.

The paper tape is spaced forward the number of characters specified unless stopped by an end-of-tape condition. The number of characters is specified by the difference between the A and B addresses of the I/O descriptor (figure 7-5).

PAPER TAPE BACKSPACE DESCRIPTOR.

The paper tape is spaced backward the number of characters specified unless stopped by a beginning-of-tape condition. The number of characters is specified by the difference between the A and B addresses of the I/O descriptor (figure 7-5).

PAPER TAPE REWIND I/O DESCRIPTOR.

The paper tape is rewound to the beginning-of-tape position (see figure 7-6').

vvv	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
3 BITS	21 BITS

Figure 7-6. Paper Tape Rewind I/O Descriptor

Referring to figure 7-6:

- 'a. VVV = 011 paper tape rewind operation comman'd.
- b. R = reserved for unit addres's.

PAPER TAPE TEST I/O DESCRIPTOR. This descriptor (figure 7-7) tests the designated unit and returns a result descriptor.

vvv	Т	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
3 BITS	1	20 BITS

Figure 7-7. Paper Tape Test Descriptor

Referring to figure .7-7:

- a. VVV = 100 tests the paper tape reader and controller.
- b. T = 0 the result is stored immediately on completion.
- ć. T = 1 the result is delayed until the reader is ready and it is not rewinding.

PAPER TAPE READ RESULT DESCRIPTOR.

At the completion of an I/O operation, the I/O controller returns a result descriptor, for the just-completed I/O, in the second 24 bits of the result status area. The result descriptor for the paper tape reader operation is:

- a. Bit 0 I/O completé.
- b. Bit 1. exception condition (any bit on in bits .3-16).
- c. Bit 2 not ready.
- d. Bit 3 tape parity error.
- e. Bit 4 memory access error.
- f. Bit 5 reserved.
- g. Bit 6 end of tape.
- h. Bit 7 beginning of tape.
- 'i. Bit & reserved.
- j. Bit 9 short record.
- k. Bit 10 rewinding.
- 1. Bits 1.1-16 reserved.
- \hat{m} . Bits 17-23 controller ID = 0001110 (test I/O only).

UNBUFFERED TRAIN PRINTER CONTROLLER.

This controller provides printer speeds of 400 and 750 LPM with 132 characters per line. It provides code conversion from internal EBCDIC to a binary code which is related to the graphic position for the standard 16, 48, 64, and 96 chain's. If memory access is missed, the control waits until the same graphic is presented to it a second time and then continues, thus providing automatic recovery.

This controller also includes a quick release feature which terminates the operation and enables the start of paper motion as soon as all non-blank characters have been printed. Printing speed with this feature is dependent on the number of blank characters; however, the effective rate is usually greater than the rated speed.

BUFFERED LINE PRINTER CONTROLLER.

This controller provides for EBCDIC to BCL translation on the data as it is transferred from memory to the printer. The I/O controller contains a one-character buffer which receives information from memory serially by character and transfers the information serially by character to the printer buffer. Transfer of paper motion control information takes place after the printer buffer is filled. The I/O controller is capable of loading the printer buffer, while paper motion is occurring, in response to a previous I/O descriptor. After the paper motion of the previous command is completed, format information is received for the information in the buffer, which can now be printed. The controller provides for printer speed options of 300, 400, 475, 700, and 750 LPM with 132 characters per line. A 1040 LPM buffered line printer controller is available on the B 1726.

LINE PRINTER WRITE DESCRIPTOR.

A line of data is printed from ascending memory locations beginning at the memory location specified by the A address (see figure 7-8). The length of the line is determined by the number of printer columns, 80, 120. 132, or until the location specified by the B address is reached, whichever occurs first. Spacing or skipping takes place after printing, and skipping takes precedence over spacing.

vvv	ı	тттт	RRRRRRRRRRRRRR	U	Α	В	
3 BITS	1	4 BITS	15 BITS	1	24 BITS	24 BITS	

Figure 7-8. Line Printer Write Descriptor

Referring to figure 7-8:

- a. VVV = 010 printer write operation command.
- b. I = 1. inhibit reporting of EOP and skip to channel one if this operation calls for single or double spacing.
- c. TTTT = 0000 no paper advance.
- d. TTTT = 1110 single space after printing.
- é. TTTT = 1111. double space after printing.
- f. TTTT = 0001. skip to channel 1 after printing.
- g. TTTT = 1100 skip to channel 12 after printing.
- h. TTTT = 1101 skip to next channel after printing.
- i. R = reserved.
- j. U = 0 printer unit 1.

- k. U = 1 printer unit 2.
- 1. A = beginning address of data.
- \hat{m} . B = ending address of data + 1.

LINE PRINTER SKIP DESCRIPTOR. Skipping is controlled by bits 4-8 as in the write descriptor. Skipping takes precedence over spacing. The skip I/O descriptor format is the same as for the printer write (see figure 7-9).

vvv	ı	тттт	RRRRRRRRRRRRRR	U	
3 BITS	1	4 BITS	15 BITS	1	

Figure 7-9. Line Printer Skip Descriptor

The printer space or skip operation command has VVV equal to 101. Variables are the same as for the write I/O descriptor.

LINE PRINTER TEST DESCRIPTOR.

The status of the designated unit is tested and a result descriptor is returned which contains the controller ID (see figure 7-10).

vvv	Т	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	U
3 BITS	1	19 BITS	1

Figure 7-10. Line Printer Test Descriptor

Referring to figure 7-10:

- a. VVV = 100 test the printer and the printer controller.
- b. T = 0 result is stored immediately.
- ć. T = 1 result is delayed until the printer is ready and paper motion is stopped.
- d. R, U same as for the write I/O descriptor.

LINE PRINTER RESULT DESCRIPTOR.

At the completion of the I/O operation the controller returns a result descriptor to the second 24 bits of the result status area. The result descriptor information is as follows:

- á. Bit 0 I/O completé.
- b. Bit 1. exception condition (any bit on in bits .3-16').
- c. Bit 2 not ready.

```
Bit 3. - code parity.
ď.
    Bit 4 - reserved.
′е.
    Bit 5. - memory parity error.
午。
    Bit 6 - end of page.
    Bits 7-8-9:
h.
        000 - 64 character set (test I/O only).
    1)
        001 - 48 character set.
    2)
        010 - 16 character set.
    3)
        011 - 96 character set.
    4)
        100 - 192 character set.
    5)
    Bits 1.0-11:
Ί.
        00. - 860 LPM or 400 LPM (test I/O only).
    1)
        01 - 300 LPM.
       10 - 600 LPM.
    3)
        11. - 1100 LPM.
    4)
    Bit 12 - paper in motion (test I/O only).
    Bit 13. - motor-on test (test I/O only).
γ.
    Bits 14-15:
1.
    1) 00. - 132 columns per print line (test I/O only).
       10 - 120 columns per print line.
    3) 11. - 80 columns per print liné.
    Bit 16 - reserved.
    Bits 17-23 - controller ID = 0010000 (test I/O only).
'n.
```

CONSOLE PRINTER (SPO) CONTROLLER. This I/O controller is used to control the console printer. The control provides a bidirectional translator which translates EBCDIC to USASCII (also referred to as ASCII and ANSCII) for the 128 USASCII code's. Translation for codes not specified is undefined.

Information between the console printer and the controller is transferred serially by bit, whereas it is transferred serially by character between the controller and main memory.

The controller can be "input request enabled," at which time the controller is sensitive only to the USASCII ENQ code from the keyboard. When the controller receives this code, an interrupt is set. The ENQ signal is remembered and is reported to main memory. Once reported, the interrupt remembering an ENQ signal is cleared.

Upon receipt of a read descriptor, the controller automatically generates a signal to light an indicator on the console printer and to await an input message from the systems operator. The controller is sensitive to the USASCII ETX (end of text) in an output message, which is translated to EBCDIC and transferred to main memory.

The controller is also sensitive to the USASCII NAK code which is also translated and stored in main memory. The NAK and ETX signals terminate a read operation.

The A and B addresses of the operand must define an integral number of 8-bit character's.

Data entered into the console printer is first sent to the I/O controller and then is retransmitted (echo back) to the teletypewriter for printing.

Console printer I/O descriptors are described in the following paragraph's.

CONSOLE PRINTER READ DESCRIPTOR.

The message being typed on the typewriter keyboard is read into ascending memory locations beginning with the location specified by the A address and continuing until an ETX code is detected, or up to the location specified by the B address (see figure 7-11). The ETX code is stored following the data stream only if the ETX signal terminates the operation.

vvv	Т	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR	Α	В	
3 BITS	1	20 BITS	24 BITS	24 BITS	

Figure 7-11. Console Printer Read Descriptor

Referring to figure 7-11:

á. VVV = 000 - console printer read operation command.

b. T = 0 - EBCDIC translation.

ć. T = 1 - no translation.

d. R = reserved.

é. A = beginning address of data.

f. B = ending address of data + 1.

CONSOLE PRINTER WRITE DESCRIPTOR.

A message is typed on the console printer for ascending memory locations beginning with the location specified by the A address and continuing until an ETX code is detected, or up to the location specified by the B address (see figure 7-12).

vvv	Т	RRRRRRRRRRRRRRRRRRRR	А	В	
3 BITS	1	20 BITS	24 BITS	24 BITS	

Figure 7-12. Console Printer Write Descriptor

The console printer write command has VVV equal to 010. Other var.i-ables are the same as for the read descriptor.

CONSOLE PRINTER TEST DESCRIPTOR.

The I/O controller and console printer are tested, and a result descriptor which contains the controller ID is returned (see figure 7-13).

VV	V	Т	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
3 BI	rs	1	20 BITS

Figure 7-13. Console Printer Test Descriptor

Referring to figure 7-13:

- a. VVV = 100 console printer test.
- b. T = 0 store result descriptor immediately.
- c. T = 1 store result descriptor when ENQ is sensed.
- á. R = reserveá.

CONSOLE PRINTER RESULT DESCRIPTOR.

The result descriptor for the console printer is as follows:

- a. Bit 0 I/O completé.
- b. Bit 1 exception condition (any bit on in bits 3-17).
- c. Bit 2 not ready.
- d. Bit 3 NAK received.
- é. Bit 4 reserved.
- f. Bit 5. memory parity error.
- 'g. Bit 6 attempt to exceed maximum addres's.
- g. Bit 6 attempt to ex h. Bit 7 ENQ received.
- 'i. Bits 8-16 reserve'd.
- (i. Bits 1.7-23 controller ID = 0010110 (test I/O only).

READER SORTER CONTROLLER.

The reader sorter controllers are used to execute I/O operations on reader sorters. One result descriptor is returned for each I/O operation initiated. The reader sorter control can store one character per memory access. The time between memory access is approximately 420 micro-seconds for an MICR read. The A and B addresses of the I/O descriptor must specify an integral number of 8-bit characters. Demand read operation is not permitted.

The controls provide an MICR-USASCII-8 translator which translates the first 16 USASCII codes (0000 0000 to 000 1111 inclusive) to EBCDIC as shown in table 7-1.

Table 7-1

Reader Sorter
MICR/USASCII-8 Translation

Sorter Symbol	Sorter Code	I/O Controller Internal Code	System Internal Graphic	Internal EBCDIC Code
0 1 2 3 4 5 6 7 8 9 Amount (S2) Transit (S2) On-Us (S3) Hyphen (S4) S5 Cannot read	000 0000 000 0001 000 0010 000 0011 000 0100 000 0110 000 0111 000 1000 000 1011 000 1010 000 1111 000 1100 000 1101	0011 0000 0011 0001 0011 0010 0011 0011 0011 0100 0011 0101 0011 0110 011 0111 0011 1000 0011 1001 0010 0100 0011 1100 0010 0111 0010 0111	0 (F0) 1 (F1) 2 (F2) 3 (F3) 4 (F4) 5 (F5) 6 (F6) 7 (F7) 8 (F8) 9 (F9) # (7B) 0 (7C) • (60) • (7D) ? (6F)	1111 0000 1111 0001 1111 0010 1111 0011 1111 0100 1111 0110 1111 0111 1111 1000 1111 1001 0111 1011 0111 1100 0111 1010 0111 1010 0111 1010 0111 1101 0111 1101

The MICR control provides a USASCII/EBCDIC translator which translates the center 64 USASCII codes (010 0000 to 101 1111 inclusive) to the corresponding EBCDIC code's. (Refer to table 7-2.) In addition, the USASCII code 001 1010 for SUB is translated to the USASCII-8 code 0001 1010 for SUB.

Validity checking for MICR reading is as follows:

- á. Amount field 1st and 12th character for amount symbols and intervening 10 characters for decimal digit's.
- b. Transit field 40th and 50th characters for transit symbols and intervening nine characters for decimal digits except for hyphen in position 45.

Table 7-2

Reader Sorter
USASCII/EBCDIC Translation

Sorter Symbol	Sorter Code	EBCDIC Code	Sorter Symbol	Sorter Code	EBCDIC Code
Blank Vertical bar # \$ % & ! (()) * + ! - / 0 1 2 3 4 5 6 7 8 9 : ; < = >	010 0000 010 0001 010 0010 010 0011 010 0100 010 0101 010 0110 010 1010 010 1001 010 1010 010 1011 010 1100 010 1111 010 1100 010 1111 011 0000 011 0011 011 0110 011 0110 011 0110 011 0110 011 1010 011 1011 011 1000 011 1011 011 1010 011 1011 011 1010	0100 0000 0100 1111 0111 1111 0111 1011 0101 1011 0100 1100 0101 0000 0111 1101 0100 1101 0100 1101 0100 1110 0100 1110 0110 0000 0100 1011 0110 0000 1111 0001 1111 0010 1111 0100 1111 0110 1111 0110 1111 0110 1111 1010 1111 1010 1111 1010 0101 1110 0101 1110	a A B C D E F G H I J K L M N O P Q R S T U V W X Y Z L \]	100 0000 100 0001 100 0010 100 0011 100 0100 100 0101 100 0110 100 1001 100 1001 100 1010 100 1101 100 1110 100 1111 101 0000 101 0011 101 0110 101 0110 101 0110 101 0110 101 0110 101 0110 101 0110 101 0110 101 0110 101 1010 101 1010 101 1010 101 1010 101 1010 101 1010 101 1010 101 1100 101 1100 101 1100 101 1100 101 1100 101 1100 101 1100 101 1100 101 1100 101 1110	0111 1100 1100 0001 1100 0010 1100 0010 1100 0100 1100 0101 1100 0110 1100 1001 1100 1001 1101 0001 1101 0101 1101 0101 1101 0101 1101 0101 1101 0101 1101 0101 1101 0101 1110 0101 1110 0101 1110 0100 1110 0101 1110 0101 1110 0101 1110 0100 1110 1000 1110 1000
? Cannot read (SUB)	011 1111 001 1010	0110 1111		101 1111	0110 1101

NOTE

Leading unreadable characters are not stored and are not reported as error's.

Formatting for MICR reading is accomplished by storing all data except for hyphens in the ON-US field into descending memory locations until the first transit symbol is detected and then storing blanks until the

40th character location is reached. The transit symbol code is stored in the 40th character position followed by the remaining data. In no case is data stored more often than every eight micro-seconds.

In flow mode, the appropriate operation complete (OC) bit is inspected just prior to the too-late-to-read time. If the OC bit is false, the read takes place and the link address is used to obtain a new set of addresses for the next read operation. If an OC bit is true, all subsequent read operations associated with that station are discontinued. However, the other read station, if active, continues to read and store data. The feeder is stopped by the presence of the OC bit for either read stations or by a too-late-to-pocket-select condition or by a halt variant contained in the read OP code.

Pocket select and interrupt information, if present, are obtained from memory via a swap operation just prior to the too-late-to-pocket-select time. The controller first swaps a lockout pattern of 24 zero bits and then determines whether valid information is present. The controller then stores the pocket select result information. If valid information is not present, the feeder is stopped; the current item and all subsequent items are rejected. The too-late bit is reported in the pocket select result descriptor area immediately. A new initiate can be accepted while the controller is rejecting previous items.

An interrupt can be requested at the completion of any read from the first station, the second station, or both. A pending interrupt does not prevent the storing of subsequent read data but does cause any subsequent request for an interrupt to be ignored.

If an unencoded item is encountered, it is reported as such in the read result descriptor. The read area in memory is unaffected.

If a batch ticket is encountered, it is reported as such in the read result descriptor for the read area in which the batch ticket data has been stored.

If underspaced, overlength, or double items are detected, the control does <u>not</u> read the fault item(s) but rejects them <u>without</u> any result indication. The feeder is not stopped and subsequent items are read normally. However, if the fault item is a batch ticket, it is reported in the read result descriptor as a rejected item. The read information is not valid.

If an empty hopper, full pocket, or sorter stop button is detected, all items in motion are processed. The not-ready bit is reported in the pocket select result descriptor area when the last item read is pocketed.

If a jam or missort is detected, the sorter stops the feeder and the control rejects all items in the feed line which have not been read. The jam-missort bit is reported in the pocket select result descriptor area when the last item read is pocketed. After a pocket light descriptor is sent to the sorter, the sorter is placed in a not-ready status and accepts only pocket light operations.

The I/O descriptor for a reader sorter operation is shown in figure 7-14. Each field is 24 bits wide.

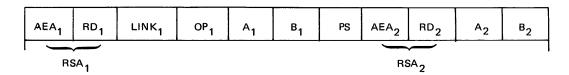


Figure 7-14. Reader Sorter I/O Descriptor

Referring to figure 7-14:

- a. Read station oné.
 - 1) AEA-sub1 actual ending addres's.
 - 2) RD-subl result descriptor.
 - 3) LINK-subl link addres's.
 - 4) OP-sub1 operation code and variant's.
 - 5) A-sub1 start addres's.
 - 6) B-sub1 ending address of data + 1.
 - 7) PS pocket select information and result.
- b. Read station two (present only if variants specify a read for both stations).
 - 1) AEA-sub2 actual ending addres's.
 - 2) RD-sub2 result descriptor.
 - 3) A-sub2 start address of data.
 - 4) B-sub2 ending address of data + 1.

POCKET SELECT INFORMATION. Pocket select information, if present, is obtained from the 24-bit PS portion of the I/O descriptor (see figure 7-15).

VV	PPPPP	RRRRRRRRRRRRRRRRR
2 BITS	5 BITS	17 BITS

Figure 7-15. PS (Pocket Select) Portion of Reader Sorter I/O Descriptor

Referring to figure 7-15:

- 'a. VV = 00 valid information is present and a normal pocket selection is performed.
- b. VV not equal to 00 valid pocket select information is not present and the control waits and again checks if valid information is present.

- PPPPP = 0000.0-11111. from four to 32 pockets are available for selection.
- d. R = reserved.

READ READER SORTER DESCRIPTOR.

Read the data from the appropriate read station(s) into descending memory locations starting at the location specified by the B address of the I/O descriptor minus one and continuing until, but not into, the location specified by the A address plus one or until all data from the item has been read, whichever occurs first. (See figure 7-16.)

vvv	R	С	D	Е	F	G	Н	1	J	R	н	RRRRRRRRRR
3 BITS	1	1	1	1	1	1	1	1	1	1	1	10 BITS

Figure 7-16. Read Reader Sorter Descriptor

Referring to figure 7-16:

- VVV = 000 read reader sorter operation command.
- R = reserved.
- C = 1 read first station.
- G = 1 read second station.
- é. CG = 11 - read both station's.
- D = 1. format and validity check first station. 午。
- H = 1 format and validity check second station.
- DH = 11 format and validity check both station's. 'n.
- E = 1 do not store delimiter, first station.
- I = 1 do not store delimiter, second station.
- k. F = 1 delimiter = blank, first station. 1. J = 1 delimiter = blank, second station.
- H = 1 halt the feeder.

BATCH COUNT DESCRIPTOR.

The batch counter in the sorter is advanced by on'e. Flow must be stopped and all items must be pocket-selected. (See figure 7-17.)

	vvv	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
24	BITS	21 BITS

Figure 7-17. Batch Count Descriptor

Referring to figure 7-17, VVV is equal to 101, and R indicates reserved.

POCKET LIGHT DESCRIPTOR.

The light in the pocket specified (0-30) in the I/O descriptor is illuminated. Flow must be stopped and all items must be pocket-selected. (See figure 7-1%.)

vvv	NNNNN	RRRRRRRRRRRRRRR
3 BITS	5 BITS	16 BITS

Figure 7-18. Pocket Light Descriptor

Referring to figure 7-18:

- a. VVV = 010 pocket light enable operation command.
- 5. NNNNN = 00000-11111 light specific pocket q through 32 (0-31).
- ć. R = reserved.

READER SORTER TEST DESCRIPTOR.

The reader sorter and controller are tested, a not-ready, jam, or missort condition is reported, if present, and an appropriate result descriptor is returned. (See figure 7-19.)

vvv	Т	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
3 BITS	1	20 BITS

Figure 7-19. Reader Sorter Test Descriptor

Referring to figure 7-19:

- 'a. VVV = 100 test reader sorter I/O controller and reader sorter.
- 6. T = 0 store result immediately.
- ć. T = 1 store result when reader sorter is ready.
- d. R = reserved.

READER SORTER RESULT DESCRIPTOR.

At the completion of the operation, the result status information is stored at the specified location. The result descriptor information is as follows:

- a. Bit O operation completé.
- b. Bit 1 exception condition (any bit on 3 through 16 set).
- c. Bit 2 not ready.

- d. Bit 3 unencoded document.
- é. Bit 4 memory access error.
- f. Bit 5 cannot read document.
- g. Bit 6 amount field error.
- h. Bit 7 on-us field error.
- i. Bit 8 transit field error.
- j. Bit 9 double document's.
- k. Bit 10 too late to read.
- 1. Bit 11 jam.
- m. Bit 12 missor't.
- n. Bit 13 batch ticket (no item left in path to be read).
- o. Bit 14 flow stopped (no items left in path to be read).
- p. Bit 15 empty hopper, full stacker, stop.
- q. Bit 16 reserved.
- r. Bits 17-23 control ID = 0010100 (test I/O only).

DISK PACK CONTROLLER (B 1726 ONLY).

This I/O controller provides for the attachment of disk packs to the system. The maximum capacity of each pack is 30 million byte's.

The controller can be used with two disk pack drives in a 1 \times 2 configuration.

The disk pack control can store or fetch 24 bits of data per memory access and is capable of receiving and recording information at a five megabit per second rate.

Correctness of information in main memory is verified by the port interchange. If a parity error is detected during the fetch of any portion of an I/O descriptor, a message is returned by the controller to the processor containing the address and 24 of the bits which have caused the memory parity error. Any error detected during data transfer causes the return of an appropriate result descriptor.

The disk pack controller assumes a linked list of I/O descriptor's. Once a control is initiated with an address which points to a result descriptor, the control reads the first two bits (00) of the result descriptor. The control fetches that I/O descriptor and executes it. If not, the control exits by fetching the link address to the next result descriptor.

Each disk pack consists of 10 disk platters; each has 20 data surfaces numbered 00-19. There are 33 available segments on each surface in a cylinder except the first, on which five alternate segments are reserved. These alternate segments are used for relocating segments which had been unrecordable in their original location. Continuous operation from segment to segment, track to track, and cylinder to cylinder is permitted, but continuous operation from disk pack unit to unit is not.

DISK PACK WRITE DESCRIPTOR.

Data is written to the disk pack starting at the given file address (C) from ascending memory locations beginning at the location specified by the A address continuing to but not from the end location specified by

the B address. Zero bits are written to complete the last segment. (See figure 7-20.)

vvv	RR	E	С	Т	NNN	RRRRRRRRR	บบบบ	'	A	В	, C
3 BITS	2	1	1	1	3 BITS	9 BITS	4 BITS		24 BITS	24 BITS	24 BITS

Figure 7-20. Disk Pack Write Descriptor

Referring to figure 7-20:

- a. VVV = 010 disk pack write operation command.
- b. R = Reserved
- c. E = 0 enable automatic restore after seek error.
- d E = 1 disable automatic restore after seek error.
- e. C = 0 do not parity-check segment written.
- f. C = 1 parity-check all segments written (to be specified if required).
- g. T = 0 segment mode.
- h. T = 1 track mode.
- i. NNN = 000 normal segment address.
- j. NNN = 001 spare segment #1 on selected track.
- k. NNN = 010 spare segment #2 on selected track.
- 1. NNN = 011 spare segment #3 on selected track.
- m. NNN = 100 spare segment #4 on selected track.
- n. NNN = 101 spare segment #5 on selected track.
- o. UUUU = 0-15 unit number.
- p. R = Reserved
- q. A = beginning address of data.
- r. B = ending address of data + 1.
- s. C = binary file address.

DISK PACK READ DESCRIPTOR.

Data is read from the disk pack starting at the given file address (C) into ascending memory locations beginning at the location specified by the A address and ending at but not in the end location specified by the B address. A complete segment need not be stored but is parity-checked. (See figure 7-21.)

			-								
vvv	M	R	E	С	${f T}$	NNN	RRRRRRRR	บบบบ	A	В	C
3 BITS	1	1	1	1	1	3 BITS	9 BITS	4 BITS	24 BITS	24 BITS	24 BITS

Figure 7-21. Disk Pack Read Descriptor

Referring to figure 7-21:

- a. VVV = 000 disk pack read operation.
- b. M = 1 normal read operation.
- c. M = 0 if reading a relocated segment, the address field returned contains all ones.
- d. C = 0 enable error correction.
- e. C = 1 disable error correction.
- f. Other variables are the same as for the disk pack write descriptor.

DISK PACK INITIALIZE DESCRIPTOR.

Segment addresses, data (all zeros), and gaps in all tracks are written starting after the index pulse on the track decoded from the given file address continuing through the entire track, cylinder, or pack. (See figure 7-22.) The data consist of a 16-bit pattern obtained from ascending memory locations at the specified A address. This 10-bit pattern is repeated 90 times throughout each data segment.

vvv	R	R	E	R	Т	PP	RRRRRRRRR	บบบบ	Ī	A	В	`	C
3 BITS	1	1	1	1	1	2	10 BITS	4 BITS		24 BITS	24 BITS		24 BITS

Figure 7-22. Disk Pack Initialize Descriptor

Referring to figure 7-22:

a. VVV = 011 - disk pack initialize operation.

- b. E = 0 enable automatic restore after seek error.
- c. E = 1 disable automatic restore after seek error.
- d. T = 0 segment mode.
- e. T = 1 track mode.
- f. PP = 00 entire pack.
- g. PP = 01 cylinder only.
- h. PP = 10 track only.
- i. PP = 11 undefined.
- j. UUUU = 0-15 unit number.
- k. R = reserved.

DISK PACK VERIFY DESCRIPTOR.

The disk pack is read and checked for address errors and information parity errors beginning with the first segment after the index pulse on the track decoded from the given file address, continuing through the entire track, cylinder, or pack. The positions of segments are verified by counting from the index on each track.

The binary file address of the detected error(s) is reported into ascending memory locations beginning at the location specified by the A address plus 16. The data is checked by comparing the 16-bit pattern at the location specified by the A address with each 16-bit group of data in the segment.

NOTE
Relocated segments are also checked.

vvv	R	R	E	R	Т	PP	RRRRRRRRR	טטטט	A	В	C
3 BITS	1	1	1	1	1	2	10 BITS	4 BITS	24 BITS	24 BITS	24 BITS

Figure 7-23. Disk Pack Verify Descriptor

Referring to figure 7-23:

- a. VVV = 001 verify disk pack operation.
- b. E = 0 enable automatic restore after seek error.
- c. E = 1 disable automatic restore after seek error.
- d. T = 0 segment mode.

- e. T = 1 track mode.
- f. PP = 00 entire pack.
- g. PP = 01 cylinder only.
- h. PP = 10 track only.
- i. PP = 11 undefined.
- j. UUUU = 0-15 unit number.
- k. R = reserved.

DISK PACK RELOCATE DESCRIPTOR.

The segment address designated by the given file address (C) is flagged with an error configuration, and is rewritten on track 00 in the space sector specified by N_{\bullet} . Both the original address field and the relocated address field on track 00 are located by counting from the index on each track.

NOTE

An error configuration is one byte of binary "ones" with clock pulses omitted, followed by one byte of binary "ones" (with clock pulses), followed by one byte of binary "ones" with clock pulses omitted. Also the Sync Code is changed to eight zeros.

The error configuration is written in the address field of the designated sector. The standard test data pattern is also written in the relocated segment data field. (See figure 7-24.)

VVV	R	R	E	R	Т	NNN	RRRRRRRR	טטטט	A	В	С
3 BITS	1	1	1	1	1	3 BITS	8 BITS	4 BITS	24 BITS	24 BITS	24 BITS

Figure 7-24. Disk Pack Relocate Descriptor

Referring to figure 7-24:

- a. VVV = 101 relocate segment operation.
- b. E = 0 enable automatic restore after seek error.
- c. E = 1 disable automatic restore after seek error.
- d. T = 0 segment mode.
- e. T = 1 track mode.

- f. NNN = 1-5 indicates the spare segment in 28 through 32 on track 00.
- g. R = reserved.

DISK PACK TEST DESCRIPTOR.

The disk pack drive is tested for the following conditions (see figure 7-25):

- a. Readiness of drive.
- b. Busyness of drive.
- c. Timeout of seek operation.
- d. Seek complete status.
- e. Seeking status.
- f. Control identification.
- g. Write lockout.
- h. Drive type.

-						
VVV	E	E	Р	$^{\mathrm{P}}_{1}$	RRRRRRRRRRRR	บบบบ
3 BITS	1	1	1		13 BITS	4 BITS

Figure 7-25. Disk Pack Test Descriptor

Referring to figure 7-25:

- a. VVV = 100 disk pack test operation.
- b. EEP = 000 return a result immediately. Do not pause.
- c. EEP = 100 return a result only if the drive is ready and in a seek complete status; otherwise, unlock and fetch the next descriptor.
- d. EEP = 010 returns a result only if the drive is present else unlock and fetch the next descriptor.
- e. EEP = 001 if no I/O descriptor has been executed since the last encounter with a test descriptor with EEP = 001, unlock and then pause one to two milliseconds before proceeding to fetch the result descriptor field of the next descriptor. If an I/O descriptor has been executed, do not pause; unlock and then proceed immediately to fetch the result descriptor field of the next descriptor. Do not store a result in either case.
- f. $P_1 = 1$ put the drive off time for pack removal.
- g. R = reserved.

DISK PACK RESULT DESCRIPTOR.

At the completion of a disk pack I/O operation, the following result descriptor information is returned:

- a. Bit 0 operation complete.
- b. Bit 1 exception condition.
- c. Bit 2 not ready.
- d. Bit 3 read data parity error; write data parity error.
- e. Bit 4 memory access error.
- f. Bit 5 memory parity.
- g. Bit 6 write lockout operation not performed.
- h. Bit 7 transmission parity error; read data error corrected.
- i. Bit 8 overrun.
- j. Bits 7-9:
 - 1) 000 not present.
 - 2) 001 20 surface 203 track.
 - 3) 010 20 surface 406 track.
- k. Bit 10 address parity error or verify error.
- 1. Bit 11 seek timeout.
- m. Bit 12 seek status.
- n. Bit 13 control no. 0...1.
- o. Bit 14 seeking; seek initiated flag; wrong cylinder.
- p. Bit 15 segment address error.
- q. Bit 16 reserved.
- r. Bits 17-23 controller ID = 0011110 (test I/O only).

DISK CARTRIDGE CONTROLLER.

This I/O controller provides for the attachment of disk cartridge drives to the system (see figure 7-26).

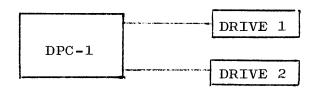


Figure 7-26. Disk Cartridge Controller

The controller can store and fetch 16 bits of data per memory access and is capable of handling a data transfer rate from and to a drive of 1.55 and 3.10 megabits per second.

Segment to segment, track to track, and cylinder to cylinder operation is permitted but continous operation from unit to unit is not permitted.

Bad tracks should not be addressed by the software in any routines other than initialization routines. Bad tracks are not detected by the control, and reading and writing on or over bad tracks produces an undefined result.

DISK CARTRIDGE READ DESCRIPTOR.

Data is read from the disk cartridge starting at the given file address (C) into ascending memory locations beginning at the location specified by the A address and ending at but not in the end location specified by the B address. A complete segment need not be stored but is parity-checked. (See figure 7-27.)

VVV	R	M	R	S	RRRRRRRRRRRR	UUU	A	В	C	,
3 BITS	1	1	1	1	12 BITS	3 BITS	24 BITS	24 BITS	24 BITS	

Figure 7-27. Disk Cartridge Read Descriptor

Referring to figure 7-27:

- a. VVV = 000 disk cartridge read.
- b. MS = 00 normal read.
- c. MS = 01 undefined.
- d. MS = 10 read immediately following the segment pulse.
- e. MS = 11 undefined.
- f. UUU = 000-111 disk cartridge unit number.
- g. R = reserved.
- h. A = beginning address of data.
- i. B = ending address of data + 1.
- j. C = binary absolute disk address.

DISK CARTRIDGE WRITE DESCRIPTOR.

Data is written to the disk cartridge starting at the given file address (C) from ascending memory locations beginning at the location specified by the A address continuing to but not from the end location specified by the B address. Zero bits are written to complete the last segment. (See figure 7-28.)

vvv	R	М	R	S	RRRRRRRRRRRRR	עטע	A	В	С	
3 BITS	1	1	1	1	14 BITS	3 BITS	24 BITS	24 BITS	24 BITS	

Figure 7-28. Disk Cartridge Write Descriptor

Referring to figure 7-28:

- a. VVV = 010 disk cartridge write.
- b. MS = 00 normal write.
- c. MS = 01 undefined.
- d. MS = 10 write immediately following the segment pulse.
- e. MS = 11 undefined.
- f. Other variables are the same as for the disk cartridge read descriptor.

DISK CARTRIDGE TEST DESCRIPTOR.

The designated controller is tested for the following conditions:

- a. Readiness of cartridge drive.
- b. Write lockout.
- c. Timeout of seek operation.
- d. Seek status.
- e. Seeking status.

f. Control identification.

The disk cartridge test descriptor is shown in figure $.7-2^{\prime}9$.

-	vvv	MS	RRRRRRRRRRRRRRR	บบบ
	3 BITS	2	16 BITS	з вітѕ

Figure 7-29. Disk Cartridge Test Descriptor

Referring to figure 7-29:

- 'a. vvv = 100 disk cartridge tes't.
- b. MS = 00 return a result immediately unconditionally.
- 'c. MS = 01. do not return any result unless ready.
- d. MS = 01. do not return any result but pause one to two milliseconds before fetching the next I/O descriptor.

DISK CARTRIDGE RESULT DESCRIPTOR.
The disk cartridge result descriptor is defined as follows:

- a. Bit 0 operation completed.
- b. Bit 1. exception condition.
- c. Bit 2 not ready.
- d. Bit 3 read parity error.
- 'e. Bit 4 memory access error.
- f. Bit 5 memory parity error.
- g. Bit 6 write lockout operation not performed (test).
- h. Bit 8 overrun.
- 7. Bits 7-8-9:
 - 1) 000 not present (test').
 - 2) 100 2200 BPI, 203T (test).
- j. Bit 10 reserved.
- k. Bit 11 seek timeout operation not performed.
- 1. Bit 12 seek status (test').
- m. Bit 13 reserved.

- 'n. Bit 14 - seeking (test).
- **′o**. Bit 15. - reserve'd.
- Bit 16 reserve'd.
- Bits 17-23 controller ID.

MAGNETIC TAPE CONTROLLER'S.

7-TRACK MAGNETIC TAPE CONTROLLER.

This controller is used with all 7-track free-standing transport's. It has the capability of operating at 200, 556, and 800 BPI densitie's. A maximum of six transports can be used on the controlle'r. The ability to perform non-stop forward operations is also included.

9-TRACK MAGNETIC TAPE CONTROLLER.

This controller is used with any 9-track free-standing transports and 9-track magnetic tape cluster's. It has the capability of operating at 1600 and 800 BPI density.

> NOTE 200 and 556 BPI are not provide'd.

A maximum of six transports can be used on the controller.

The magnetic tape I/O descriptors are described below.

MAGNETIC TAPE READ DESCRIPTOR.

This operation reads data from magnetic tape in either a forward or backward direction into ascending or descending memory locations as specified by the A address and continuing until the end location specified by the B address (see figure 7-30).

vvv	Т	С	FFF	PPPP	RRRRRRRR	υυυυ	Α	В	
3 BITS	1	1	3 BITS	4 BITS	8 BITS	4 BITS	24 BITS	24 BITS	ı

Figure 7-30. Magnetic Tape Read Descriptor

Referring to figure .7-30:

- a. VVV = 000 read operation command.
- T = 0 read forward.
- T = 1 read reverse. C = 1 current track. ď.
- FFF = 000 through 111 track number. é.
- PPPP = 0110 800 BPI even parity 7-track. f.
- PPPP = 0111 800 BPI odd parity 7-track. ģ.
- PPPP = 1011 800 BPI odd parity 9-track. **h**.
- PPPP = 0010 556 BPI even parity 7-track.

- j. PPPP = 0011. 556 BPI odd parity 7-track.
- k. PPPP = 0000 200 BPI even parity 7-track.
- 1. PPPP = 0001 200 BPI odd parity 7-track.
- m. PPPP = 1111. 1600 BPI odd parity 9-track (phase encoded).
- n. PPPP = 0100 (as selected by switch even parity).
- 6. PPPP = 0101. (as selected by switch odd parity).
- ρ. R = reserved.
- 'q. A = beginning address of dat'a.
- Υ . B = ending address of data + 1.
- 's. UUUU = 0000 through 1111. tape unit numbe/r.

MAGNETIC TAPE WRITE DESCRIPTOR.

This operation writes a data to tape moving in the forward direction from ascending memory locations beginning at the location specified by the A address and continuing until the location specified by the B address. (See figure 7-31.)

MAGNETIC TAPE ERASE DESCRIPTOR.

Magnetic tape is erased in the forward direction on the designated unit. The number of characters erased is the number of characters which would be written for the same A and B addresse's. No memory space is used, but memory cycle time is used. (See figure 7-31.)

1	vvv	E	Т	RRRR	FFFF	RRRRRRRR	บบบบ	Α	В	
	3 BITS	1	1	4 BITS	4 BITS	8 BITS	4 BITS	24 BITS	24 BITS	

Figure 7-31. Magnetic Tape
Write/Erase Descriptor

Referring to figure 7-31:

- a. VVV = 010 tape write operation command.
- b. E = 1 erase.
- c. T = 1 write tape mark.
- d. All other variables are the same as in tape read.

MAGNETIC TAPE REWIND DESCRIPTOR.

Magnetic tape is rewound on the designated unit to the beginning of tape (BOT). After the operation is initiated, a result descriptor is returned with the I/O complete bit set ON without waiting until the rewind is complete. (See figure 7-32.)

vvv	RRRRRRRRRRRRRRRR	ับบบบ
3 BITS	17 BITS	4 BITS

Figure 7-32. Magnetic Tape Rewind Descriptor

Referring to figure 7-32:

- a. VVV = 011 tape rewind operation command.
- b. R = reserved.
- ć. UUUU = 0000-1111. tape unit number.

MAGNETIC TAPE SPACE DESCRIPTOR.

Magnetic tape is spaced forward or backward by the number of records specified by bits 13-20 of the tape space I/O descriptor (001 through 255 physical records) unless tape is stopped by detection of an end of file (EOF), end of tape (EOT), or beginning of tape (BOT) condition. (See figure 7-33.)

vvv	т	RRRR	PPPP	RRRRRRR	S	UUUU
3 BITS	1	4 BITS	4 BITS	7 BITS	1	4 BITS

Figure 7-33. Magnetic Tape Space Descriptor

Referring to figure 7-33:

- a. VVV = 110 space operation command.
- b. S = 0 space to next end of file (EOF).
- c. S = 1 space one record.
- d. All other variables are the same as in the tape descriptor.

MAGNETIC TAPE TEST DESCRIPTOR. This descriptor tests the unit specified and returns a result descriptor (see figure 7-34).

vvv	тт	RRRRRRRRRRRRRR	υυυυ
3 BITS	2	15 BITS	4 BITS

Figure 7-34. Magnetic Tape Test Descriptor

Referring to figure .7-34:

- a. VVV = 100 test operation command.
- b. TT = 00 always return a result descriptor.
- ć. TT = 10 return a result descriptor only if the unit is ready.
- d. IT = 01 return a result descriptor only if the unit is not ready.
- 'e. All other variables are the same as for the tape read

descriptor.

MAGNETIC TAPE LOCK DESCRIPTOR. There is one lock descriptor in the subsystem chain for each magnetic tape unit (see figure 7-35).

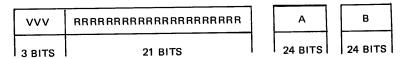


Figure 7-35. Magnetic Tape Lock Descriptor

Referring to figure 7-35:

- vvv = 101 magnetic tape lock operation.
- R = reserve'd. ъ.
- A = address of first non-lock descriptor. ć.
- B = address of current non-lock descriptor.

MAGNETIC TAPE RESULT DESCRIPTOR. At the completion of an I/O operation the following result descriptor information is returned:

- Bit 0 I/O complete. 'n.
- Bit 1. exception condition (any bit on in bits 3-16). **b**.
- Bit 2 not ready. C.
- Bit 3 tape parity error. ď.
- Bit 4 memory access error. e.
- Bit 5. memory parity error. Bit 6. end of tape (EOT). 午。
- Bit 7. beginning of tape (BOT'). h.
- Bit & write lockout. i.
- Bit 9 end of file (EOF').
- Bit 10 tape is rewinding. **%**.
- Bit 11 spacing of three feet of blank tape (timeout). 1.
- Bit 12 cyclic redundancy check. 'n.
- Bits 13-15 track in error (000-111). 'n.
- Bit 16 reserved. **6.**
- Bit 17. short record (incomplete record'). **p**.
- Bits 1.8-23 reserved. ∕a•

The magnetic tape test result descriptor is as follows:

- a. Bit 0 - I/O complete.
- Bit 1. exception condition (any bit on in bits 3-16). 6.
- Bit 2 not ready. ′c .
- ď. Bit 3. - bus'y.
- Bits 4-5 reserved. e.

- f. Bit 6 end of tape.
- g. Bit 7. beginning of tape.
- h. Bit 8 write lockou't.
- i. Bit 9 reserved.
- 'j. Bit 10 rewinding.
- k. Bit 11 reserved.
- 7. Bits 1.2-14:
 - 1) 000 7-track 200 BPI.
 - 2) 001 7-track 556 BPI.
 - 3) 011 7-track 800 BPI.
 - 4) 101 9-track 800 BPI.
 - 5) 111 9-track 1600 BPI.
- m. Bits 15-16 reserved.
- n. Bits 17-23 controller ID.
 - 1) ID #1 0110010.
 - 2) ID #2 0110000.
 - 3) ID #3 0110100.
 - 4) ID #4 0110110.
 - 5) ID #5 0111000.

NOTE

The cyclic redundancy check is a means of generating another check character during write operations and provides error correcting capabilities. During a read, the generated check character must agree with the one that has been written.

DATA COMMUNICATIONS CONTROLLER'S.

A data communications controller provides for certain basic functions described below with non-standard functions peculiar to particular line disciplines provided for by the specific line adapter. Basic functions of data communications controllers are:

- a. I/O initiate handling.
- b. I/O descriptor fetch for an adapter.
- ć. Provides for storage in an adapter for address of result status area, beginning of data address, and end of data address + 1.

- d. Initiating an adapter to the appropriate operating modé.
- é. Servicing an adapter request for character transfer's.
- 作. USASCII-7 to/from EBCDIC translation as selected by variant.
- g. Result descriptor and interrupt handling.

The data communications controller accepts all I/O initiate descriptors directed to it regardless of the state of an adapter. If an I/O descriptor is directed to a busy adapter, the control forces the adapter to the new operating state, destroying all traces of the previous operation including any pending interrupt's.

On the controller a parity error or a memory access error does not terminate a read or write operation. On a read, an access failure results in a loss of the character. On a write, an access failure results in the transmittal of a sync character on a synchronous line and a steady marking condition on an asynchronous line.

SINGLE-LINE CONTROLLER.

A single-line controller provides connection between a single I/O channel and a single communications line. A line adapter is required to equip the single-line controller for use with a particular type of remote I/O device. The adapter determines code sensitivity, transmission rate, and character length in bits.

LINE ADAPTERS FOR SINGLE-LINE CONTROL'S. The adapters are input/output devices that provide the terminal connections between a telephone or telegraph facility and an I/O channel via the single-line control's.

During input, all adapters receive data from the line serially by bit. When a bit has been stored, the adapter signals the control to accept the bit. During output, all adapters receive data serially by bit from the I/O control. As each bit is transmitted, the adapter requests the next bit from the control.

The adapters provide for both a receive and a transmit break. On receipt of a break I/O descriptor, the adapter terminates the operation in process, if any, and transmits the break signal. It does not return a result descriptor for an operation in process. The result descriptor for the break is returned after completion of the break operation.

A time-out feature is provided for the I/O controller/adapter hookup when using leased lines or direct line operation and provides a read time-out which detects the absence of a data character for a period of time greater than one second. The time-out starts immediately after the receipt of the I/O descriptor. The time-out can be inhibited programmatically. The time-out period can be changed by a field engineer to a 20-second period.

Time-out also occurs if the clear-to-send signal from the data set is not returned within 20 seconds after the request to send. This time-out cannot be inhibited.

Three types of standard line adapters are available.

- a. Standard synchronous line adapter.
- b. Standard asynchronous line adaptef.
- c. Standard direct asynchronous line adapter.

Standard line adapters provide for the attachment of the following terminals:

- a. Teletypes (models 33, 35, and 37).
- b. Burroughs Input and Display. (B 9353').
- c. CRT Input and Display (B 9352).
- d. Burroughs T.C-Serie's.
- e. Burroughs T.U-Serie's.
- f. Burroughs RT-Serie's.

SYNCHRONOUS LINE ADAPTER.

The standard synchronous line adapter(s) provide for a connection of the listed terminals via the following common carrier services and data sets. Data sets equivalent to those listed can be used but must first be approved and specified. The clock is supplied by the data set.

Speed (BPS)	<u>Service</u>	Data Set	Burroughs <u>Data Sets</u>
2000	Dial	WE 201A3	TA 713/TA 783
2400	Leased	WE 201B1	
4800	Leased	WE 203	TA 713/TA 783
9600	Leased	WE 203	

ASYNCHRONOUS LINE ADAPTER.

Standard asynchronous line adapter(s) provide for a connection of the listed terminals via the following common carrier services and data set's. Data sets equivalent to those listed can be used but must first be approved and specified. The adapters may be modified to accommodate any one of the listed transmission speed's.

Speed (BPS)	Service	<u>Data Set</u>	Burroughs <u>Data Sets</u>
150	Dial	WE 103A	TA 713/753
150	Leased	WE 103F	
300	Dial	WE 202C	TA 713/753
300	Leased	WE 202D	
600	Dial	WE 202C	
600	Leased	WE 202D	TA 713/753

Speed (BPS)	<u>Service</u>	<u>Data Set</u>	Burroughs <u>Data Sets</u>
1200 1200 1800	Dial Leased Leased	WE 202C WE 202D WE 202D	TA 713/753 TA 783

DIRECT ASYNCHRONOUS LINE ADAPTER.

The standard direct asynchronous line adapter provides for a connection of the listed terminals via a 2-wire direct connect interface. The adapter may be modified to accommodate any of the following listed transmission speeds (in BPS):

- 'n. 110.
- 150. ъ.
- ć. 300.
- 1200. ď.
- é. 1800.
- 午。 2400.
- ģ. 4800.
- 9600. h.

ADAPTER CHARACTERISTIC'S.

Transmission code is USASCII-7. The seven data bits are transmitted least significant bits first followed by one parity bit. For synchronous operation, character parity is odd. For asynchronous operation, character parity is even. The adapter performs the parity generation and checking. One start bit and one stop bit frame the seven bits plus parity that make up the character; therefore, for each character of data. -10 bits are sent down the line.

A block check code (BCC) is normally transmitted following all messages that contain a start code. The BCC is formed by taking the modulo two sum on each data bit following a start code, up to and including an end code, and is transmitted after the end code with appropriate character parity.

Both EBCDIC and USASCII-8 codes are capable of being received from memory by the line adapter's. EBCDIC code must be translated to USASCII-8 by the control. USASCII-8 is translated to USASCII-7 in the adapter by ignoring the high order bit. On input, USASCII-7 is translated to USASCII-8 in the adapter and then translated to EBCDIC by the control if requested by a variant in the I/O descriptor.

The adapter assembles one character of data before requesting service from the I/O controller.

The following code sensitivity is provided by the adapter's.

<u>Function</u>		<u>Code</u>
Start End Positive Negative Enquiry Sync		SOH, STX ETX, ETB ACK, ENQ, BEL, ETX, ETB NAK, EOT ENQ SYNC

Two SYNC codes must be received before a synchronous adapter is sensitive to any other code's. SYNC codes are not stored in main memor'y. After the receipt of the last leading SYNC code, an adapter stores in memory all subsequent codes, except the SYNC character, up to but not including a block check character. SYNC characters are not included in a BCC sum. At least four SYNC codes must be transmitted prior to data. Four SYNC codes are automatically generated by the I/O controller prior to transmittal of data.

A control code listed under the classifications of positive responses and negative responses is also defined to be an ending code if it is not preceded by a code classified as a start code.

The control characters ETX and ETB are not considered to be a positive response unless followed by a correct block check character (BCC).

The negative response code EOT is a response code on a read only and never on a write. To enable an on-hook condition at a remote site, the adapter must receive a message such as DLE, EOT, ENQ. To enable an on-hook condition at the central site, the adapter must receive an I/O descriptor with the disconnect variant.

The receipt of an ENQ code or a ringing condition is recognized by the adapter when the adapter is in a test and wait state. It is not remembered or recognized in any other state.

A variant in the I/O descriptor is provided to ignore EOT as an ending code for read. This variant allows the controller and the system to which it is attached to act as a terminal and receive a polling sequence from other system's. No hardware recognition of addresses is provided, however.

The SLC I/O controller and line adapters are designed to operate over the communications facilities defined in table 7-3 and table 7-4.

Table 7-3

Transmission Facility Classification

Designation	Bandwidth	Transmission Rate	Remarks
Narrowband	Variable - generally up to 300 Hz	15.0-300 bits/sec	Generally private line except for TELEX and TWX services
Voiceband	Nominal 4 kHz	600/1200/1800/2000/ 2400 bits/sec and higher with WE or COAM modems	Private and dial lines
Wideband	48 kHZ and up	40.8 K bits/sec and higher	Generally private line except DATA- PHONE 50 service from the Bell System

Table 7-4
Conditioning

Voiceband	Conditioning	Delay Distortion	Band
3002 3002	None C1	1750 us 1000 us 1750 us	800-2600 Hz 1000-2400 Hz 800-2600 Hz
3002	C2	500 us 1500 us	1000-2600 Hz 600-2600 Hz
3002	C4	3000 us 300 us 1500 us	500-2800 Hz 1000-2600 Hz 600-3000 Hz

DATA COMMUNICATIONS READ DESCRIPTOR.

Data is read from the remote device in ascending memory locations beginning with the location specified by the A address. Reading is continued until a control code denoting the end of text is detected. In no case is the data stored in the ending location specified by the B address. The complete message read must be terminated by a control code. (See figure 7-36.)

vvv	TTT	w	RR	D	СС	RRRRRRRR	υυυυ	А	В
3 BITS	3 BITS	1	2	1	2	8 BITS	4 BITS	24 BITS	24 BITS

Figure 7-36. Data Communications Read Descriptor

Referring to figure 7-36:

- VVV = 000 data communications read operation.
- TTT = 000 no translation. ъ.
- TTT = 010 USASCII-7 to EBCDIC translation. ć.
- W = 0 do not disable time-out timef. ď.
- W = 1. disable time-out time'r. ́е.
- D = 1. ignore EOT as a response cyclie. Ŧ.
- CC = 00 normal I/O chaining.
- CC = 01 I/O chaining if negative response is received. **h**.
- CC = 10 I/O chaining if positive response is received. Ί.
- R = reserve'd.
- UUUU = .0-15. adapter number. **k**.
- 1. A = beginning address of data.
- B = ending address of data + 1.

DATA COMMUNICATIONS WRITE DESCRIPTOR.

Data is written to the remote device from ascending memory locations beginning with the location specified by the A address and continuing until a control code which denotes end of text is detected or up to the location specified by the B addres's. (See figure 7-37.)

vvv	TTT	RRRRRRRRRRRRR	UUUU	А	В
3 BITS	3 BITS	14 BITS	4 BITS	24 BITS	24 BITS

Figure 7-37. Data Communications Write Descriptor

VVV is equal to 010 for the data communications write operation. variables are the same as for the read descriptor.

DATA COMMUNICATIONS BREAK DESCRIPTOR. Send a break signal to the remote terminal or go on hook as specified by the variants in the I/O descriptor's. (See figure 7-3%.)

VVV	RRR	Т	RRRRRRRRRRRR	υυυυ
3 BITS	3 BITS	1	13 BITS	4 BITS

Figure 7-38. Data Communications Break Descriptor

Referring to figure 7-38:

- 'a. VVV = 111 data communications break.
- b. T = 0 send break and do not disconnect.
- c. T = 1. disconnect and do not break.
- $6. \quad UUUU = .0-15. adapter number.$
- é. R = reserved.
- f. D = address of result status area.

DATA COMMUNICATIONS TEST DESCRIPTOR.

Test the adapter identification and for a received ENQ, and return a result descriptor which contains the adapter I/0. (See figure 7-39.)

vvv	Т	RRRRRRRRRRRRRR	υυυυ
3 BITS	1	16 BITS	4 BITS

Figure 7-39. Data Communications Test Descriptor

Referring to figure 7-39:

- 'a. VVV = 100 test the I/O controlle'r.
- 6. VVV = 111. test the I/O controller, return a result descriptor, and go idle.
- \dot{c} . T = 1 wait until ENQ is received before completing I/O.
- d. R = reserved.
- 'e. D = address of result status are'a.

DATA COMMUNICATIONS RESULT DESCRIPTOR.

The data communications result descriptor is as follows:

- a. Bit 0 I/O completé.
- b. Bit 1. exception conditions (any bit on in bits 3-16').

- ć. Bit 2 not ready.
- d. Bit 3 parity error character or BCC.
- é. Bit 4 memory access error.
- f. Bit 5 memory parity error.
- g. Bit 6 time-ou't.
- h. Bit 7. break.
- i. Bit & ending control code expected but not received.
- 'j. Bit 9 chaining terminated.
- k. Bit 10 reserve'd.
- 1. Bit 11 loss of clear to send.
- m. Bit 12 carrier los's.
- n. Bit 13 reserved.
- 6. Bit 14 off hook (dial%
- p. Bit 15 ringing or enquiry received.
- a. Bit 16 reserved.
- r. Bits 1.7-23 adapter ID (test I/O only).
 - 1) 1000000 = adapter not present.
 - 2) 10nnnnn = leased or direct connect.
 - 3) linnnnn = switched line.

NOTE

The variable nnnnn is defined as:

00010 - standard line adapter

01000 - Teletype adapter

00100 - ACU adapter

APPENDIX A

REGISTER SUMMARY

MICRO-INSTRUCTION CONTROL REGISTERS.

Mnemonic	Width	Function Punction	Usage	<u>Remarks</u>
А	19	Address reģ.	Source/dest.	2000 KB
MAR (1700)	14	Address re'g.	Source/dest.	64 KB
М	16	Micro-inst.	Source/dest.	
TAS	24	Top of stack	Source/dest.	Pseudo
ТОРМ	4	Top ct1. mem.	Source/dest.	Not av1. on B 1710 Series
MBR	24	Mem. base reg.	Source/dest.	Not av1. on B 1710 Series

MAIN MEMORY CONTROL REGISTER'S.

<u>Mnemonic</u>	<u>Width</u>	<u>Function</u>	<u>Usage</u>	<u>Remarks</u>
BR	24	Base reg.	Source	
LR	24	Limit reģ.	Source	
FA	24	Field Adr.	Source/dest.	B 1700 64 KB ma'x
FL	16	Field lgth.	Source/dest.	FLC/D/E/F
CP	8	Arith. ct1.	Source/dest.	CYF, CPU, CPL
CYF	1	Carry flag	Source/dest.	
СРИ	2	Arithunit	Dest.	00 = binary 01 = 4-bit 11 = 8-bit
CPL	5	Data length	Source/dest.	Ma⁄k. va⁄1. 24
MAXS	24	Main me⁄m. size	Source	
MXAM	24	Control me'm. size	Source	
FLCN	l ₄	Fld. lgth. cond.	Source/dest.	

INTERRUPT CONTROL REGISTER'S.

<u>Mnemonic</u>	<u>Width</u>	<u>Function</u>	Usage	Remarks
CC	4	In tr. cond.	Source/dest.	0 = console intr. 1 = I/O ser. rq. 2 = Clock intr. 3 = State flag
CD	4	In t'r. -con'd.	Source/dest.	0 = Wrt. out bd's. 1 = R'd. out bd's. 2 = Out bd's. ovride. 3 = R'd. par. er.
INCN	4	Intr. cond.	Source/dest.	0 = Disp. lkout. 1 = Disp. intr. 2 = Priority disp. 3 = Missing port

GENERAL PURPOSE REGISTER'S.

Mnemonic	<u>Width</u>	<u>Function</u>	<u>Usage</u>	<u>Remarks</u>
X Y T L CA CB	24 24 24 24 4 4		Source/dest. Source/dest.	Inp. to funct. box Inp. to funct. box Used by DISPATCH

INPUTS TO 24-BIT FUNCTION BOX.

Mnemonic	Width	Function	Usage	<u>Remarks</u>
X Y C YF CPU CPL	24 24 1 2 5	Input Input Carry flag Arith. unit Data length	Source/dest. Source/dest. Source/dest. Dest. Source/dest.	

OUTPUTS FROM 24-BIT FUNCTION BOX.

<u>Mnemonic</u>	<u>Width</u>	<u>Function</u>	<u>Usage</u>	<u>Remarks</u>
B I C N	<i>l</i> ∔	Binary cond.	Source	
X Y C N	<i>L</i> ‡	X and Y cond.	Source	

Mnemonic	Width	<u>Function</u>	<u>Usage</u>	<u>Remarks</u>
XYST CYL CYD	4 1 1	X and Y states Carry out level Borrow out level	Source Source Source	

AVAILABLE FROM 24-BIT FUNCTION BOX.

Mnemonic	<u>Width</u>	Function	Usage	Remarks
SUM D IFF XORY XE OY XANY	24 24 24 24 24	X + Y X - Y X OR Y X EXOR Y X AND Y	Source Source Source Source Source	AND CYF AND CYD
CMP X CMP Y MSK X MSK Y	24 24 24 24	Compl. X Compl. Y Mask X Mask Y	Source Source Source Source	AND CPL AND CPL AND CPL AND CPL

MISCELLANEOUS REGISTER'S.

Mnemonic	Width	Function	<u>Usage</u>	<u>Remarks</u>
U NULL DATA CMND	16 24 24 24	Cassette input I/O data I/O command	Source Source Source/dest. Dest.	Contains zeros

APPENDIX B

HE XADECIMAL - DECIMAL

CONVERSION TABLE

The table in this appendix provides for direct conversion of decimal and hexadecimal numbers in the ranges:

<u>He xadec i mal</u>	<u>Decimal</u>
000 to FFF	0 to 4095

For numbers outside the range of the table, add the following values to the table figure's.

<u>Hexadecimal</u>	<u>Decimal</u>
1000	4096
2000	8192
3000	12288
4000	16384
5000	20484
6000	24576
7000	28672
8000	32768
9000	36864
A000	40960
B 000	45056
C000	49152
D000	53248
E000	57344
F000	61440

3 2 2 7.9 9.5 351 لعة 4.5€ ပ 27 4.3 5.9 7.4 25 41 41 57 8 4 6 8 8 4 6 8 88 • 39 55 _ • 'n 36 37.1 ~ ~ 32 48 4 4 OVO 0 E O OBO V V 100

APPENDIX B (cont'd)

543 559 671 703 735 751 767 799 815 831 1007 1023 798 814 830 606 734 926 958 974 990 u 989 925 941 957 541 557 797 668 ပ 538 602 794 922 • 665 681 793 809 8.89 985 921 537 • 664 568 728 792 920 • 855 791 983 _ 822 534 550 566 918 982 548 662 • 981 725 741 757 789 805 821 • 564 596 788 804 820 • 787 803 595 851 883 979 ~ APPENDIX B (cont'd) 658 978 722 754 786 ~ 785 801 817 977 529 657 784 800 816 912 976 992 528 688 592 2E0 290 240 280 300 360 370 210 220 230 200 350 350 390 380 310 320

APPENDIX B (cont'd)

	٥		~		4	S	9	2	•	6	¥	60	U	o	ผ	ě.
004	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
410	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1021	1052	1053	1054	1055
420	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1046	1067	1068	1069	1070	1071
430	1072	1073	1074	1075	1076	1077	1078	1079	1080	1001	1082	1083	1084	1085	1086	1087
440	1088	1089	1090	1001	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
450	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
394	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
470	1136	1137	1136	1139	1140	1141	1142	1143	1104	1145	1146	1147	1148	1149	1150	1151
480	1152	1153	1154	1155	1156	1157	1158	1159	1166	1161	1162	1163	1164	1165	1166	1167
490	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1163
4 A O	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
9 RO	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
00	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1221	1228	1229	1230	1231
00.4	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E0	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
4 F O	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
200	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
510	1296	1597	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
520	1312	1313	1314	1315	1316	1317	1318	1319	1320	1351	1322	1323	1324	1325	1326	1327
530	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
540	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
550	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
260	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
570	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
580	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
290	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1414	1435	1436	1437	1438	1439
5 A O	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B0	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
200	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
500	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E0	1504	1505	1506	1507	1500	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
550	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535

۱. w 1564 9 2 9 1 U 5 2 9 1 1562 • 1561 Ò 2008 • ~ **2** 155**8** ø 1557 • m APPENDIX B (cont'd) ~ 1553 700 760 7 A O 7 B O 6 A O 6F0 7.10 00

222? w ပ • ~ • • ~ APPENDIX B (cont'd) 2258 2128 810 820 BAO OHO 0 E O 8F0 006 9E0

۱. w U 306€ 26A2 • • • ~ • • ~ APPENDIX B (cont'd) 2578 N 28.97 800 BEOBFO ACO AUO A E O AFO 84C B60 B70 890 BA0 A00 A10 A40 A 60 A 7 0 A80 A90 AAG A 80 A20 A30

ပ 34A2 33A6 • • • 3.349 ~ 35.39 APPENDIX B (cont'd) 33.45 C 2 0 ¢30 CAO CB0 CEO CFO DAO

لما

 w U ⋖ € -39.42 ø 36.69 \$923 m APPENDIX B (cont'd) ~ 3601 -F C 0 FLO F 40 F 50 F 60 5 F C EAO E E O F 80 F 90 FAO £20 E40 E50 E60 E80 E90 EC0 F00 F10 F20 F30 F70 F 80 £ 30 € 80 EF0

APPENDIX C

SYSTEM COMPARISON

	B 1712/B 1714	B 1726
MSM	No	Yes
ТОРМ	No	Yes
MBR	No	Yes
INCN	No	Yes
A stack	16 elements	32 elements
BR & LR	MCP checking only	CD reg. set out of bounds
CPU	00 = binary 01 = 4 bit	00 = binary 01 = 4 bit 11 = 8 bit
MA XM	0	0,4,8
DATA/CMND	Utilized	Utilized
XYST/conditions	Timer, I/O service request, console inter-rupt, memory parity bits	1700 plus, swap out of bounds, port interrupt, missing port
Swap memory micro	No	Yes
Clear register micro	No	Yes
Store F into doublepad word micro	Мо	Yes
Overlay M-string micro	Мо	Yes
Control memory	No	Yes
Port interchange	No	Yes
Bufferer I/O controllers	Yes	Ye s
Processor clock rate	2 MHz/4 MHz	6 MHz

B 1712/B	1714	B 1726

4 clocks Read cycle 4 clocks Write cycle 6 clocks 6 clocks 24 KB 16 KB Main memory minimum 96 KB 32 KB/56 KB Main memory maximum

NOTE

6 MHz = 167 nanoseconds (B 1726) 4 MHz = 250 nanoseconds (B 1714) 2 MHz = 500 nanoseconds (B 1712) 1 KB = 1000 bytes 1 byte = 8 bits

APPENDIX D

CONVERSIONS

POWERS OF TWO. BASE 10.

2	<u>**n</u>	n		2**	a		n
1 2 4 8 16	1248 16248 12564 12562 0248 0962 1384	n 0 1 2 3 4 5 6 7 8 9 10 11 12 13	1 2 4 8 17	1 2 4 8 16 33 67 134 268 536 073 147 294 589 179	0487 19487 19877 19877 1987 1987 1987 1987 1987	5762 1008	n 201 22 23 24 25 26 27 28 9 3 3 3 3 3 3 4
32	768	15	34	359	738	368	35
65 131 262 524	536 072 144 288	16 17 18 19	68 137 274 549	719 438 877 755	476 953 906 813	736 472 944 888	36 37 38 39
7 44	200	1 2	フサブ	100	017	000	ノノ

NOTE 2**n means 2 raised to the power of n.

BINARY/HEXADECIMAL/DECIMAL.

Binary	<u>Hexadecimal</u>	<u>Decimal</u>
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	$\iota_{\!\scriptscriptstyle{+}}$	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1,001	9	9
1010	А	10
1011	В	11
1100	С	12
1101	D	13

APPENDIX D (cont)

<u>Binary</u>	<u>He xadecimal</u>	<u>Decimal</u>
1110	E	14
1111	F	15