
**CONTROL DATA®
1700 SYSTEM
MAINTENANCE MONITOR**

Volume 1 of 3

REFERENCE MANUAL

REVISION RECORD

REVISION	DESCRIPTION
01 (05-13-66)	Original Printing, preliminary edition.
02 (08-08-66)	Publications Change Order 14307. Reprint with revision which obsoletes all previous editions. Tests were updated and the following new tests were added: 1711/1712 Teletype, 1729 Card Reader, 1731 Magnetic Tape, 1706/1716 Buffered Data Channel and Coupling, Random Protect, 1700 SMM Edit Routine, and Enter Program.
A (05-01-67)	Manual released. Publication Change Order 16368. The following new tests are added: 0B (1718 Satellite Coupler Test), 0C (1742 Line Printer Test), and 3D (Enter Program). Other tests were extensively revised and updated. This edition obsoletes all previous editions.
B (09-14-67)	Publication Change Order 17146. To revise existing tests and add new tests. Introduction: page 5 revised. Description: pages 7, 12, 15, 18, 25, 26, 27, 30 and 35 revised. Pages 30-a and 30-b added. Tests: pages 90-1, 90-2, 100-7, 100-8, 100-10, 101-2, 101-7, 202-1, 202-7, 205-2, 206-6 thru 206-10, 207-3, 208-2 and 208-6 revised. Page 100-8a added. Tests sections: 102, 201, 203, 212, 213, and 214 added. Sections 102 Rev A, 201 Rev A and 203 Rev A removed.
C (02-28-68)	Publications Change Order 18929. To add 1728 Card Reader/Punch test, No. D.
D (06-11-68)	Publications Change Order 19818, to make miscellaneous publication corrections. Pages 37, 100-2, 100-18, 101-9, 102-7, 200-10, 201-6, 202-9, 203-7, 204-1, 204-12, 205-14, 206-9, 206-10, 207-4, 208-21, 210-4, 210-6, 211-13, and 215-23 revised. Pages 207-5 and 212-24 added.
E (01-06-69)	Manual Revised, Engineering Change Order 21307, publications change only. Information included through Edition 2.1. Pages 35, 90-1, 90-2, 90-6, 101-10, and 208-1 thru 208-21 revised; pages 30-c through 30-f, 51 through 60, 103, 216, 217, 218, 219, 220, 221, 222 and red tab dividers added. Manual divided into two volumes.
F (12-15-69)	Manual revised, Engineering Change Order 21883. This manual is complete through Edition 2.1.
G (02-15-70)	Manual revised. New tests are added and editorial corrections made. This manual is complete through Edition 2.2.
H (12-15-70)	Manuals revised. This publication is complete through Ed. 2.3. All previous editions are obsolete.
J (02-05-73)	Manuals revised. New tests are added and minor corrections are made. This publication is complete through Ed. 3.0.
K (09-20-73)	Manuals revised. Tests are added, deleted, and corrected.
L (02-01-74)	Manuals revised. Tests are added, deleted, and corrected. This publication is complete through Edition 3.1.
M (12-10-74)	Manuals revised. Tests are added and corrected. This publication is complete through Edition 3.1-1.
Publication No. 60182000	

REVISION LETTERS I, O, Q AND X ARE NOT USED

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4201 North Lexington Avenue
St. Paul, Minnesota 55112

© 1966, 1967, 1968, 1969, 1973, 1974, 1975, 1977, 1978
by Control Data Corporation
Printed in the United States of America

or use Comment Sheet in the back of this manual.

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
Front Cover	-	703-3	H	706-4	J	709-6	P	754-21	J
Title Page	-	703-4	H	706-5	J	709-7	P	754-22	J
ii	N	703-5	H	706-6	P	709-8	P	755-1	J
ii-a/ii-b	R	703-6	H	706-7	P	709-9	P	755-2	M
iii	R	703-7	H	706-8	P	709-10	P	755-3	L
iv	R	703-8	H	706-9	L	709-11	P	755-4	M
v/vi	P	703-9	H	706-10	P	709-12	P	755-5	J
vii	P	703-10	H	706-11	J	709-13	P	755-6	L
viii	R	703-11	H	706-12	J	753-1	J	755-7	J
ix	R	704-1	M	706-13	J	753-2	J	755-8	J
x	R	704-2	J	706-14	J	753-3	J	755-9	J
xi/xii	R	704-3	J	706-15	J	753-4	J	755-10	J
700-1	R	704-4	M	706-16	J	753-5	J	755-11	J
700-2	P	704-5	J	706-17	J	753-6	J	755-12	J
700-3	H	704-6	J	706-18	M	753-7	J	755-13	M
700-4	P	704-7	J	706-19	M	753-8	J	755-14	J
700-5	P	704-8	J	706-20	J	753-9	J	755-15	J
700-6	P	704-9	J	706-21	J	753-10	J	755-16	J
701-1	J	704-10	J	706-22	J	753-11	J	755-17	J
701-2	H	704-11	J	706-23	J	753-12	J	755-18	J
701-3	H	704-12	J	706-24	J	753-13	J	755-19	J
701-4	H	704-13	J	706-25	J	753-14	J	755-20	J
701-5	H	704-14	J	706-26	J	753-15	J	755-21	J
701-6	H	704-15	J	706-27	M	753-16	J	755-22	J
701-7	H	704-16	J	706-28	M	753-17	J	755-23	J
701-8	H	704-17	J	706-29	J	753-18	J	755-24	J
701-9	H	704-18	J	706-30	J	753-19	J	755-25	J
701-10	H	704-19	J	706-31	J	753-20	J	755-26	J
701-11	H	704-20	J	706-32	J	753-21	J	755-27	J
701-12	H	704-21	J	706-33	J	753-22	J	755-28	J
701-13	H	704-22	J	706-34	J	753-23	J	755-29	J
701-15	H	704-23	J	706-35	J	753-24	J	755-30	J
701-16	H	704-24	J	706-36	J	753-25	J	755-31	L
701-17	H	704-25	J	706-37	J	753-26	J	756-1	L
701-18	H	704-26	J	707-1	N	753-27	J	756-2	L
701-19	H	704-27	J	707-2	N	753-28	L	756-3	L
701-20	H	704-28	J	707-3	N	753-29	L	756-4	L
701-21	H	704-29	J	707-4	N	753-30	L	756-5	L
701-22	H	704-30	J	707-5	N	753-31	L	756-6	L
702-1	J	704-31	J	707-6	N	754-1	J	756-7	L
702-2	H	704-32	J	707-7	N	754-2	J	756-8	L
702-3	H	704-33	J	707-8	N	754-3	J	756-9	L
702-4	J	704-34	J	708-1	N	754-4	J	756-10	L
702-5	J	704-35	J	708-2	N	754-5	J	850-1	J
702-6	H	704-36	J	708-3	N	754-6	J	850-2	H
702-7	H	704-37	J	708-4	N	754-7	J	850-3	J
702-8	H	704-38	J	708-5	N	754-8	J	850-4	H
702-9	H	704-39	J	708-6	N	754-9	J	850-5	H
702-10	H	704-40	J	708-7	N	754-10	J	850-6	H
702-11	H	704-41	J	708-8	N	754-11	J	850-7	H
702-12	H	704-42	M	708-9	N	754-12	J	850-8	H
702-13	H	704-43	J	708-10	N	754-13	J	850-9	H
702-14	J	704-44	J	708-11	N	754-14	J	850-10	H
702-15	H	704-45	J	708-12	N	754-15	J	850-11	H
702-16	H	704-46	J	709-1	P	754-16	J	850-12	H
702-17	H	704-47	J	709-2	P	754-17	J	850-13	H
702-18	H	706-1	L	709-3	P	754-18	J	850-14	H
702-19	H	706-2	L	709-4	P	754-19	J	851-1	N
703-1	J	706-3	M	709-5	P	754-20	J	851-2	N
703-2	H								

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
851-3	N						
851-4	N						
851-5	N						
852-1	P						
852-2	P						
852-3	P						
852-4	P						
852-5	P						
852-6	P						
852-7	P						
852-8	P						
852-9	P						
852-10	P						
Comment Sheet	R						
Back Cover	-						

PREFACE

This manual is intended to serve as a reference aid for field and checkout personnel involved in the running of the CONTROL DATA® 1700 System Maintenance Monitor (SMM17).

This manual contains a detailed description of the operation and use of the monitor, instructions for the operator, restrictions, and necessary parameters. Detailed test descriptions are also included.

If information is required concerning the SMM17 QSE library, refer to SMM17 QSE Reference Manual, publication no. 60454710.

CONTENTS

VOLUME 1

SYSTEM FLOW CHART	xi
SMM17 DESCRIPTION	
I. SMM17 DEFINITION	1
A. General	1
B. Minimum System Configurations	1
C. Standard System Configurations	4
II. DESCRIPTION	5
A. System Loading and Initialization	5
B. System/Operator Communications	12
C. Monitor Error Codes	20
D. System Subroutines Available to Library Test	20
SUPPLEMENT A. Hand-Entered Bootstraps	A-1
SUPPLEMENT B. Quick Look Error Stops and Intermediate Loader Error Stops	B-1
SUPPLEMENT C. Monitor Error Codes	C-1
SUPPLEMENT D. SMM17 Programming Specifications	D-1
SUPPLEMENT E. Monitor Based Subroutines Programming Specification	E-1
SUPPLEMENT F. SMM17 Library Format	F-1
OPERATING PROCEDURES	
I. LOAD THE BOOTSTRAP	50-1
A. Executing Quick Look Command Tests	50-1
II. SETTING THE SYSTEM PARAMETERS	50-2
A. First Stop	50-2

B. Second Stop	50-3
C. Third Stop	50-5
D. Fourth Stop	50-5
E. Fifth Stop	50-6
III. SYSTEM INITIAL MESSAGE	50-6
A. Build Test List	50-7
IV. TEST EXECUTION	50-9
A. Controlling Test Execution	50-9

ROUTINES	MNEMONIC	NUMBER	SUPPORT CLASS	PAGE
SERVICE ROUTINES				
Printer/Teletype Dump	DMP	3B	II	100-1
Tape to Print Routine	LST	3C	II	101-1
3000 Channel Simulator Assembler	SAS	3D	II	102-1
SMM Edit Routine	EDT	3E	II	103-1
3000 Channel Simulator Program Update	UD3	2D	II	104-1
1700 Source/6000 TVC Update	UD1	57	II	105-1
1700 Source/6000 BUCAL Update	UD2	58	II	106-1
1700 MSMS Installation Program	INS	FD	II	107-1
1700 Disk Pack Call-Up Program	DPC	FC	II	108-1
Diagnostic Replace Program	REP	FB	II	109-1
Punch-A-Deck Program	PAD	69	II	110-1
Card Punch Verify Routine	CPV	F0	II	111-1
INTERNAL TESTS				
1700 Command Test	COM	01	II	200-1
1700 Memory Test	MEM	14	II	201-1
1700 Protect Test	RPT	09	II	202-1
1700 Quick Look Memory and Protect Test	QL2	FE	II	203-1
1774 System Controller Command Test	CAR	1B	II	204-1
1700 Memory Test	MY2	02	II	205-1
1700 Memory Test	MY1	12	II	206-1
1700 Command Test	CMD	1C	II	207-1
1784 MOS Memory Test	MOS	1E	II	208-1

ROUTINES	MNEMONIC	NUMBER	SUPPORT CLASS	PAGE
EXTERNAL TESTS				
Paper Tape Equipment				
1723 Paper Tape Punch Test	PTP	03	II	300-1
1721 Paper Tape Reader Test	PTR	04	II	301-1
1720-1 Paper Tape Punch Test	PT1	92	II	302-1
1720-1 Paper Tape Reader Test	PT2	93	II	303-1
Printing Equipment				
1711/12/13 Teletypewriter Test	TTY	05	II	350-1
1740/501, 1742 Line Printer Test	LP1	0C	II	351-1
FF524-A/1742-120/512 Printer Test	LP5	23	II	352-1
Card Equipment				
1728/430 Card Reader Punch Test	CRP	0D	II	401-1
1729-2 Card Reader Test	CR3	13	II	402-1
1726/405 Card Reader Test	CR2	17	II	403-1
1725-1/FH302 Card Punch Test	CPC	88	II	404-1
Magnetic Tape Equipment				
1731/601, 602, 612 Magnetic Tape Test	MT1	07	II	450-1
1731/601, 602, 612 Magnetic Tape Test	MT2	0E	II	451-1
1732/608, Magnetic Tape Test	MT3	15	II	452-1
1731/601, 602, 612-1732/608, 609 Special Magnetic Tape Test	MTS	1F	II	453-1
1731, 1732/60X Magnetic Tape Test	MTX	4A	II	454-1
1732-2/615-73/93, or 1732-3/616-72/93/95 Magnetic Tape Subsystem Test	PET	4B	II	455-1
VOLUME 2				
Data Channel				
1706 Data Channel Test	BD1	0A	II	500-1
1706/16 Data Channel Test	BD2	0F	II	501-1
Rotating Mass Storage				
1738/853/854 Disk Drive Test	DP1	08	II	550-1
1739 Cartridge Disk Drive Controller	CDD	78	II	551-1
BG504A/H Drum Controller Diagnostic	DRM	80	II	552-1
1738 Disk Quick Look Test	DP5	84	II	553-1
1733-1/1738/853, 854 and QSE 4730	DP3	27	II	554-1
1733-2 Multiple Cartridge Disk Driver Controller	MDC	7A	II	555-1

ROUTINES	MNEMONIC	NUMBER	SUPPORT CLASS	PAGE
Displays				
1745/6-1, 210 Display Test	DDC	40	II	600-1
1745/6-2, 211 Display Test	DDT	1D	II	601-1
1700/8000 Data Transfer Buffer Display	DTB	10	II	602-1
1744/274 Digigraphics Display Test	DIG	4F	II	603-1
1744/274 Digigraphics Display System	DG4	6F	II	604-1
CYBERDATA Key Entry Station Test	KEY	60	II	605-1
General Purpose Graphics Terminal (GPGT)	N/A	N/A	N/A	610-1
GPGT Troubleshooting Program	GT0	70	II	611-1
GPGT Command Test	GT1	71	II	612-1
GPGT Display Quality Test	GT2	72	II	613-1
GPGT Light Pen and Keyboard Test	GT3	73	II	614-1
GPGT Communications Test	GT4	74	II	615-1
GPGT Communications Test (12 Bit Interface)	GT5	75	II	616-1
GPGT Specification Verification Test	GT6	76	II	617-1

Optical/MICR Readers

1735/915 Optical Character Reader	OCR	35	II	650-1
1700/FF104/955 System Test	RX1	30	II	656-1
955-959/1700-1774-1784 Module Test	LDR/RX3	32/33	II	657-1
SC17/1700/FR101 MEM/COM/IFP Test	BC2	56	II	658-1
1700/FR101/955 Transport Test	RX4	34	II	660-1
SC/1700/FR101/FR113 Interface Test	BC3	59	III	661-1
929/1784 Diagnostic	LR1	31	II	662-1
234-14 Reader/Sorter	RST	61	II	663-1

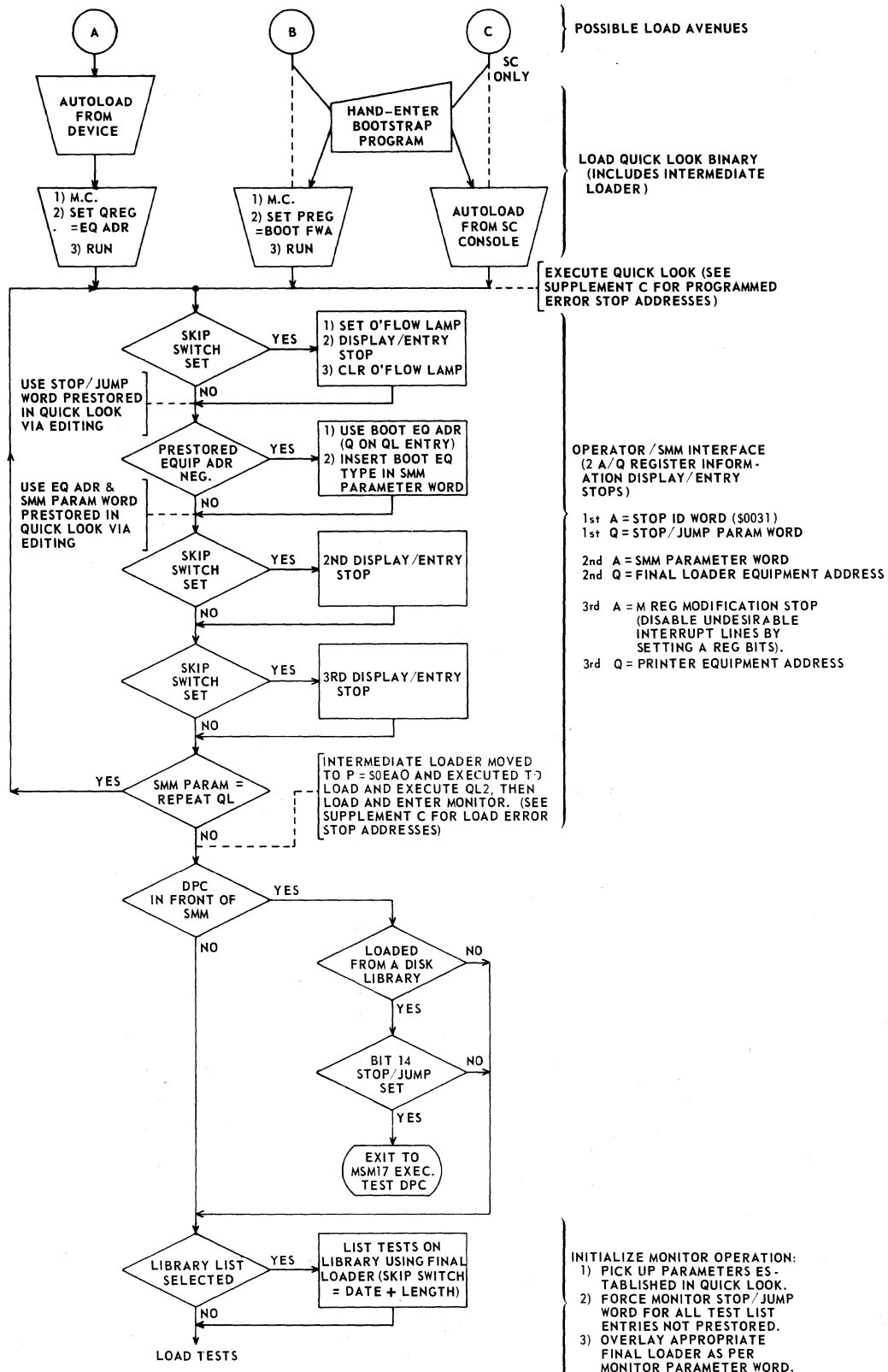
VOLUME 3

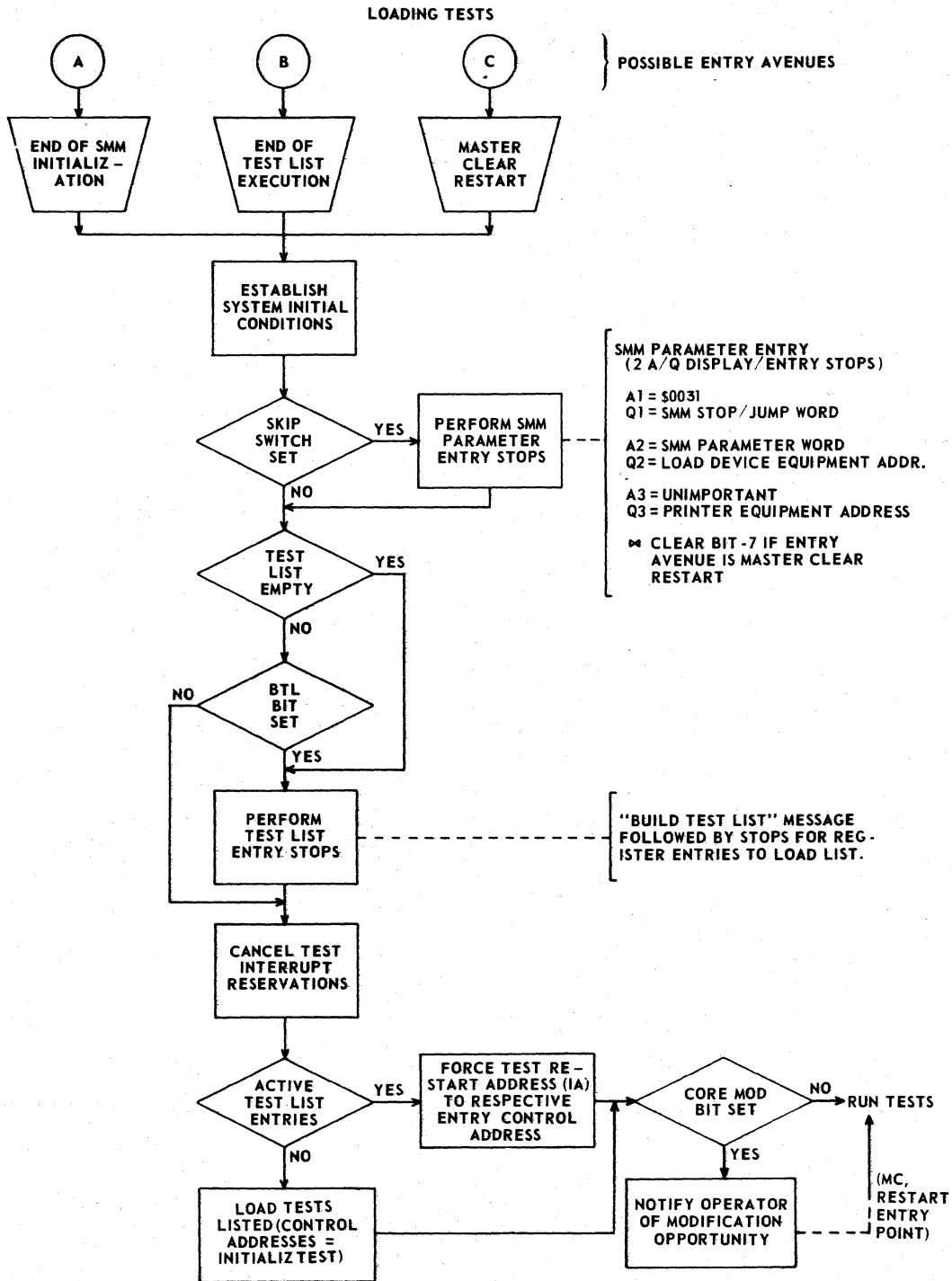
Communication Equipment

1718 Satellite Coupler Test	SC1	0B	II	700-1
1747/6000 Data Set Controller Test	DSC	11	II	701-1
1747 Data Set Controller Test	DS1	20	II	702-1
1749 Communications Terminal Test	CTC	43	II	703-1
1748-2 Multiplexer Controller CSPL Communications Adapter	MCC	48	II	704-1
DJ814A A/Q Communications Multiplexer (NUMOD)	AQM	36	II	706-1
1743-2 Asynchronous Communications Controller Test	ACC	86	II	707-1

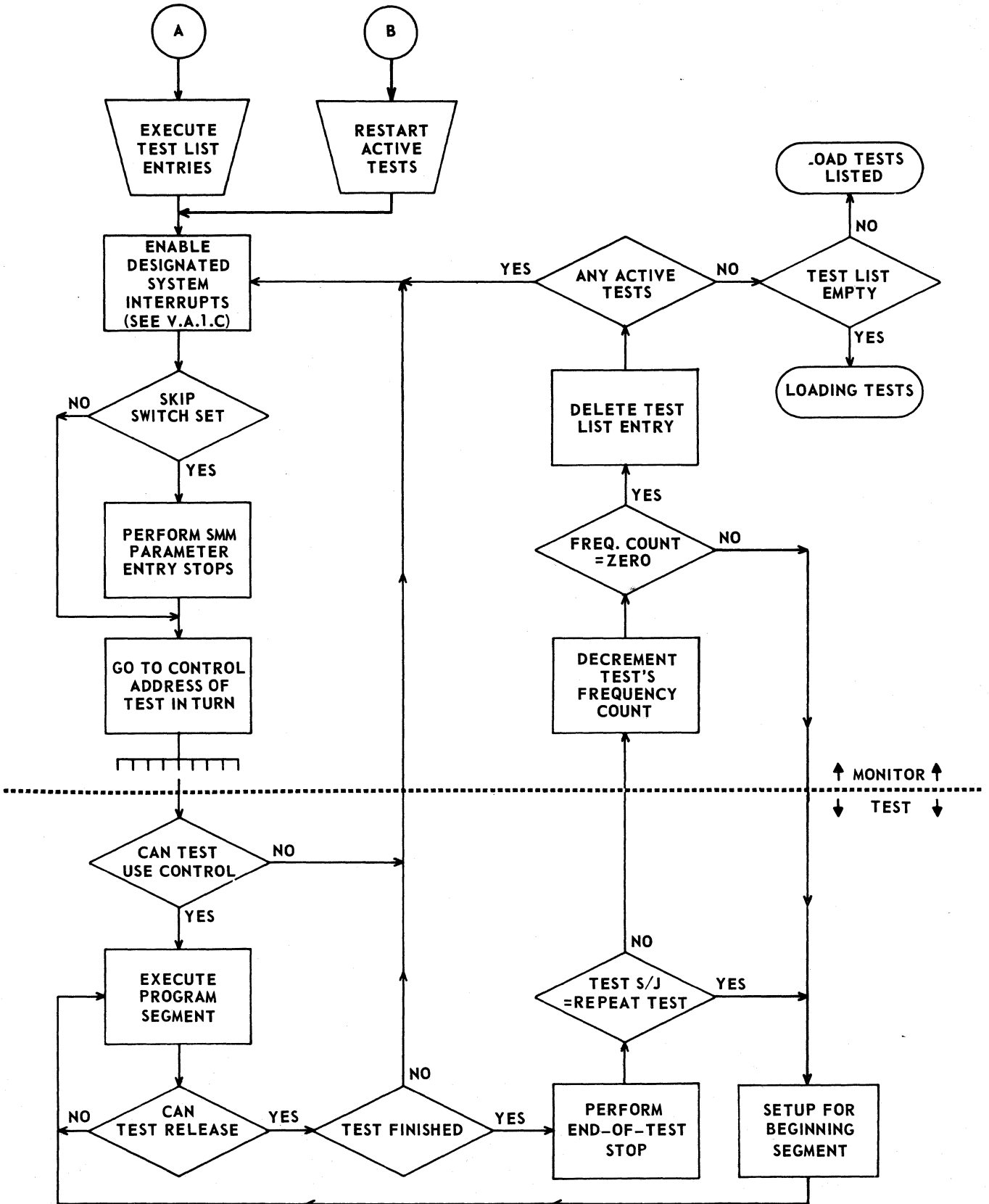
ROUTINES	MNEMONIC	NUMBER	SUPPORT CLASS	PAGE
1743-1 Synchronous Communications Controller Test	SCC	87	II	708-1
FJ505 Bisync Controller Test	BSC	89	II	709-1
Analog/Digital				
Event Counter Subsystem	CTR	81	II	753-1
Digital Input/Output Subsystem	DIO	83	II	754-1
IOM Mother Unit Diagnostic	IOM	90	II	755-1
1500 Series Remote Peripheral Controller Diagnostic	HOR	4C	II	756-1
Miscellaneous				
10126 Clock Test	CLK	42	II	850-1
10336-1 Real-Time Clock	RTC	91	II	851-1
1700 Hardware Floating Point Unit Diagnostic	HFP	8A	II	852-1

LOADING AND INITIALIZING SMM17





RUNNING TESTS





SMM17 DESCRIPTION

I. SMM DEFINITION

A. GENERAL

The 1700 System Maintenance Monitor (SMM17) is a set of programs designed to be run as an aid to maintain and check out the CONTROL DATA® 1700 series computer system. This set of programs is composed of a monitor and a collection of tests or diagnostic programs. The monitor is an executive routine which controls the running of the tests. Each test checks or diagnoses a particular peripheral equipment or some part of the main (1704/1714/1774/1784) computer.

B. MINIMUM SYSTEM CONFIGURATIONS

The SMM17 diagnostic system can be executed at varying levels of complexity, depending upon the peripheral complement of the computer system and the amount of core storage available to the diagnostic system. Thus, there are a number of minimum configuration requirements, depending upon the desired task at hand.

1. 4K Memory Mainframe, One Standard Input Device

A minimum system of this size allows the loading of one diagnostic at a time for execution. Only a portion of the diagnostic library can be loaded and executed in 4K of core. These tests are:

Command Test	CMD	1C
Paper Tape Punch Test	PTP	03
Paper Tape Reader Test	PTR	04
Random Protect Test	RPT	09
Memory Test	MEM	14
System Controller Test	CAR	1B
Magnetic Tape Test	MTS	1F
Dump-Printer/Teletype	DMP	3B
Tape to Print Utility	LST	3C

1700 Source/6000 TVC Update	UD1	57
1700 Source/6000 BUCAL Update	UD2	58
GPGT Troubleshooting Program	GT0	70
Digital Input/Output Subsystem	DIO	83

2. 8K Memory Mainframe, Standard Input and Output Device(s)

- a. Allows the loading of two selected library tests for concurrent execution. The size of these two tests must not exceed available core.
- b. Library work can be executed with the Edit program.
- c. Allows for the loading and execution of the preceding tests, plus the following tests.

1700 Memory Test (Loads in 4K and runs in 8K.)	MY2	02
1711/12/13 Teletypewriter Test	TTY	05
1731/601, 602, 612 Magnetic Tape Test	MT1	07
1738/853/854 Disk Drive Test	DP1	08
1706 Data Channel Test	BD1	0A
1718 Satellite Coupler Test	SC1	0B
1740/501, 1742 Line Printer Test	LP1	0C
1728/430 Card Reader Punch Test	CRP	0D
1731/601, 602, 612 Magnetic Tape Test	MT2	0E
1706/16 Data Channel Test	BD2	0F
1700/8000 Data Transfer Buffer Display	DTB	10
1747/6000 Data Set Controller Test	DSC	11
1700 Memory Test (Loads in 4K and runs in 8K.)	MY1	12
1729-2 Card Reader Test	CR3	13
1732/608/609-1732-2/658, 659 Magnetic Tape Test	MT3	15
1726/405 Card Reader Test	CR2	17
1700 Command Test	COM	01
1745/6-2, 31F Display Test	DDT	1D
1747 Data Set Controller Test	DS1	20
FF524-A/1742-120/512 Printer Test	LP5	23
1733-1/1738/853, 854 and QSE 4730	DP3	27
3000 Channel Simulator Program Update	UD3	2D
1700/FF104/955 System Test	RX1	30
1700/955 Module Test	RX3	33
1700/FR101/955 Transport Test	RX4	34

1735/915 Optical Character Reader	OCR	35
DJ814A A/Q Communications Multiplexer (NUM0D)	AQM	36
3000 Channel Simulator Assembler	SAS	3D
SMM Edit Routine	EDT	3E
1745/6-1, 210 Display Test	DDC	40
10126 Clock Test	CLK	42
1748-2 Multiplexer Controller CSPL Communications Adapter	MCC	48
1500 Series Remote Peripheral Controller Diagnostic	HOR	4C
SC/1700/FR101/FR113 Interface Test	BC3	59
CYBERDATA Key Entry Station Test	KEY	60
GPGT Light Pen and Keyboard Test	GT3	73
GPGT Communications Test	GT4	74
GPGT Communications Test (12-Bit Interface)	GT5	75
GPGT Specification Verification Test	GT6	76
Event Counter Subsystem	CTR	81
1738 Disk Quick Look Test	DP5	84

3. 12K Memory Main Frame, Standard Input and Output Device(s).
 - a. Allows all the capabilities of the 8K system configuration.
 - b. Allows the loading of all but a select few of the library tests for execution.
 - c. Allows for multiple loading and executing of most library tests.
 - d. The following tests require 12K of core memory to load.

1749 Communications Terminal Test	CTC	43
1739 Cartridge Disk Drive Controller	CDD	78
1733-2 Multiple Cartridge Disk Driver Controller	MDC	7A

4. 16K Memory Mainframe

Only the BG504A/H Drum Controller Diagnostic Test (DRM - 80 and also, DIG-4F, GT1-70, GT2-71) currently requires a minimum of 16K for execution.

5. 20K Memory Mainframe, Standard Input and Output Device(s).
 - a. Allows the capabilities of any of the smaller systems.
 - b. Allows the loading and executing of the following tests.

1744/274 Digigraphics Display Test	DIG	4F
1744/274 Digigraphics Display System	DG4	6F
GPGT Command Test	GT1	71
GPGT Display Quality Test	GT2	72

C. STANDARD SYSTEM CONFIGURATIONS

1. Mainframes

- a. 1704, 1714, 1774, 1784-1, 2
- b. Core sizes from 4K to 65K

2. Standard Input Devices

- a. Card Equipment
1726-405, 1728-430, 1729-2, 1729-3
- b. Magnetic Tape Equipment
1731, 1732-608/609, 1732-2-615-73, 93, 8000, 1732-3, 616, 617
- c. Rotating Mass Storage
1738/853-4, FA706/853-4, 1733-1, 1739/1733-2
- d. Paper Tape Equipment
1712, 1713, 1721, 1777

3. Standard Output Devices

- a. Card Equipment
1725-1
1728-430
- b. Magnetic Tape Equipment
1731, 1732-608/609, 1732-2-615-73, 93, 8000, 1732-3, 616-72, 92, 95
- c. Rotating Mass Storage
1738/853-4, FA706/853-4, 1733-1, 1739/1733-2
- d. Paper Tape Equipment
1720-1, 1723, 1777

II. DESCRIPTION

A. SYSTEM LOADING AND INITIALIZATION

1. General Techniques

Getting the monitor and tests read in requires three load operations. First, a Quick Look Command Test is loaded by an autoload or hand-entered Bootstrap. After executing two Quick Look Command Tests, an Intermediate Loader within Quick Look loads the monitor. The Final Loader will then load the tests. All references to these loaders will be made by the underlined terms above.

Following Quick Look execution, there are programmed stops for register display/entry of equipment definitions and operations information to be used by the monitor; Quick Look will terminate with the A register = \$0031 (monitor ID word), and the Q register = \$020D (system Stop/Jump parameter). Any other information at a stop is an error. Refer to Supplement B, C, or D for the P register address meaning.

The operator has control to force two or three programmed stops, depending upon the requirements of the system. If the SMM17 library resides on disk, the operator has the option of passing control to a mass storage maintenance system package after loading the SMM17 monitor. This loading sequence will be discussed in the Disk Pack Call-Up Program (DPC).

2. Parameter Settings

a. Stop/Jump Parameter Word

The Stop/Jump parameter is the first parameter word shown to the operator in the Q register during the first Monitor stop after Quick Look execution. This word is the software equivalent of hardware STOP and SKIP switches. During SMM17 initialization, each possible test list entry is assigned the Stop/Jump parameter designated during Quick Look execution. This parameter can then be changed to fit a specific testing condition.

The Stop/Jump parameter is always displayed in the first stop, Q register of any information stop series. Notation to identify this display is Q1. All displayed information, including the Stop/Jump parameter, will reference the active test or monitor initiating the information series. This active test or monitor is identified in the A register

of the first stop (A1 display) as the ID word.

If the ID word indicates the active test is the monitor (that is, an ID of \$00XX), the Stop/Jump parameter of any test may be altered. This can be accomplished by changing the ID word to identify the desired test, and setting the Stop/Jump parameter in Q2 to that value desired for this new test - see operating procedures for a complete description.

Stop/Jump Parameter Bit Assignments

Bit 0 = Stop to enter test parameters	(Stop Type 1)
Bit 1 = Stop at end of test section	(Stop Type 2)
Bit 2 = Stop at end of test	(Stop Type 4)
Bit 3 = Stop on error	(Stop Type 8)
Bit 4 = Repeat conditions	(Jump Type 1)
Bit 5 = Repeat section	(Jump Type 2)
Bit 6 = Repeat test	(Jump Type 4)
Bit 7 = Repeat quick look	
Bit 8 = Omit typeouts (May be used in conjunction with one of the jumps for scoping purposes.)	
Bit 9 = Bias displayed return address of stop initiator	
1 = Display the actual core address	
0 = Display the listing address (relative to the test FWA)	
Bit 10 = Reenter test parameters	
Bit 11 = Preexecution correction stop	
Bits 12 to 15 = Not used	

The following paragraphs give a more detailed explanation of the Stop/Jump parameter bits.

- 1) Bit 0 = Stop to Enter Test Parameters = Stop 1
Setting bit 0 will cause a stop to occur during each test's initialization and when a test is restarted from its initial address. The stop allows the test's prestored parameters to be changed.
- 2) Bit 1 = Stop at End of Section - Stop 2
Setting bit 1 will cause a stop to occur after each test section has been completed.
- 3) Bit 2 = Stop at End of Test - Stop 4
Setting bit 2 will cause a stop to occur after each complete pass of the test.

- 4) Bit 3 = Stop on Error - Stop 8
Setting bit 3 will cause a stop to occur when a test or the monitor encounters an error. If no teletype is attached to the system and the line printer driver is not selected, this bit will be set by the monitor during initialization.
- 5) Bit 4 = Repeat Condition
Normally, bit 4 will be set only after an error stop. The failing loop will then be repeated with the same conditions until the bit is cleared.
- 6) Bit 5 = Repeat Section
Setting bit 5 will cause the current section to be repeated until the bit is cleared.
- 7) Bit 6 = Repeat Test
Setting bit 6 will cause the test to be repeated until the bit is cleared. The selected number of passes at load time (frequency count) will have no effect.
- 8) Bit 7 = Repeat quick look
This bit must be set during quick look's parameter entry stop to be effective.
- 9) Bit 8 = Omit Typeouts
Setting bit 8 will eliminate all typeouts. Not only will the end of test and error typeouts be eliminated, but the test's messages to the operator will also be omitted. If no teletype is attached to the system and the line printer driver is not selected, this bit will be set by the monitor during initialization, and information messages will be via the A and Q registers.
- 10) Bit 9 = Display Actual Memory Return Address
The test address to which control will return after the stop series is reported to the operator. This return address is displayed in the Q register of stop 2. If bit 9 is set, the address furnished will be the actual memory address. If cleared, the address furnished will be the address in the test's listing (that is, the program address relative to its load address).

11) Bit 10 = Reenter Test Parameters

If after the test has begun execution and a parameter change is desired, setting bit 10 will force a parameter stop after an end of section, end of test, or error stop. At this time, the operator may make whatever parameter changes are desired.

12) Bit 11 = Test Preexecution Correction Stop

Setting this bit will allow the operator an opportunity to change absolute core locations after all the tests identified in the test list have been loaded, but prior to execution of any test. A message is presented to the operator detailing how this is to be done. The A register entry is the core location to be changed, and the Q register entry is to be the new contents of the core location. The A register will be incremented by the monitor between each entry. This sequence is terminated by the operator by setting the A register entry to zero.

13) Bits 12 to 15 = Not Used

Each individual test may define these bits as required for special features or testing options. These bits being set will affect only the test identified by the A1 display, the stop ID word.

At any ID stop, the Stop/Jump parameter may be altered for that active test only. In the event of a monitor ID stop, ID = 00, the system's Stop/Jump may be changed or an individual test's Stop/Jump parameter may be changed. The Stop/Jump parameter entered in Q1 will replace the parameter of the monitor or test as identified in the ID word, the A1 display. (See operating procedures for a complete description.)

b. SMM Parameter Word

This parameter describes the operator desired system functions and operating modes to the SMM17 monitor.

1) SMM17 Parameter Bit Assignments

a) Bit 0 = List Library Contents

Setting this bit causes the monitor to list on the output device the tests contained within the library. If the SKIP switch is set, the format for this library list is:

BIN	QL		
00	SMM	03204	0D89
01	COM	03204	0787
.	.	.	.
.	.	.	.
. etc.	.	.	.

The left-most column is the test number, the next column is the three-letter test mnemonic, and the next column is the date of the last update to the test. The form of this date is MMDDY; where MM is the month, DD is the day, and Y is the last digit of the year. The right-most column is the length of the test. If the SKIP switch is not set, only the two left-most columns are reported.

This bit must be set during Quick Look's SMM parameter entry stop to be effective.

b) Bit 1 = Do not load hex corrections generated by Edit.

c) Bits 2,3 = Mainframe Memory Speed Selection

<u>Bits</u>	<u>3</u>	<u>2</u>	
	0	0	1704/14 - 1.1 microsecond
	0	1	1774 - 1.5 microsecond
	1	0	1784-1 - 900 nanoseconds
	1	1	1784-2 - 600 nanoseconds

d) Bit 4 = Select 713-120 Nonimpact Printer Line Feed/Carriage Return Delay. Setting this bit when generating hard copy on a 713-120 will allow ample time for a complete carriage return and align the left margin of messages.

e) Bit 5 = Type Messages in Noninterrupt Mode

Setting this bit forces continuous typeout of all messages, with testing inhibited until the message is complete.

Clearing this bit allows the timeouts to be driven by data interrupts, and testing is resumed between characters.

f) Bit 6 = Select Teletype Input Mode

Setting this bit selects the teletype input package so that all operator parameter inputs are accomplished through the TTY, and not through the registers. This bit must be set during Quick Look's SMM parameter entry stop. Otherwise, this software package will be overwritten. After this point, the TTY input capability may be selected and deselected at any monitor parameter stop, but it must be selected at load time to have this capability.

g) Bit 7 = Stop to Build a Test List

Setting this bit allows the operator to define which tests are to be loaded for execution. Refer to operating procedures for a detailed description of how to build the list.

Clearing this bit will enable the loading and execution of the prestored test list, which can be set up by the Edit program, or allow execution of previously loaded tests.

h) Bit 8 = Select the Monitor Based Subroutine (MBS) Software Package

Setting this bit forces the loading of a large selection of software packages which are used by selected tests. These subroutines provide efficient packages to do the following.

Copy and check equipment status.

Output functions to a device.

Monitor the equipment status and exit the routine only on a specific status condition or a timeout.

Transfer data to or from a device.

Select and deselect interrupt lines.

Process interrupts.

Report errors.

- i) Bits 9, 10, and 11 = Unit Number of the Tape or Disk Drive Containing the SMM17, V4.0 Library.

When a unit number other than the one selected in the Bootstrap must be used for loading tests, it will be modified as follows:

1. Do not change the unit number (bits 9, 10, and 11) until after the Build Test List Stop has occurred.
2. At the Build Test List Stop, set the SELECTIVE SKIP switch, clear A register, and set the RUN/STEP switch to RUN before entering tests to be loaded. This will force another series of monitor stops.
3. At the next A2 stop, make the desired unit number change, clear the SELECTIVE SKIP switch, run, and continue through the Build Test List sequence.

- j) Bits 12 through 15 = Final Loader Equipment Type, from which the Library Tests will be Loaded

Bits	15	14	13	12	
	0	0	0	1	Paper Tape
	0	0	1	0	Cards
	0	0	1	1	Mag Tape
	0	1	0	0	Disk Pack
	0	1	0	1	Cartridge Disk Pack
	0	1	1	0	1739 Cartridge Disk Pack
	0	1	1	1	SMD Disk Subsystem
	1	0	0	0	8000 Mag Tape

The loaders are preselected to match the bootstrap input device type, but may be changed to any other type only at the end of Quick Look execution. The contents of the Q register must also be changed to the correct equipment address at this time.

NOTE

Use caution when changing loaders. The MBS overlay packages, if required, will be loaded from the new loader type, and thus must be available.

B. SYSTEM/OPERATOR COMMUNICATIONS

1. Programmed Information Stops

Four types of stops for information display or entry during system execution may occur: parameter entry, end of test section, end of test, and error stops. These stops are handled by a monitor subroutine. Special test programming requirements may result in other peculiar stop possibilities. Such stops will be detailed in the deviating test's description. The A register contents on the first stop of a series of any type identifies the test number, series stop count, and the type of stop.

a. First Stop of a Series

Notation for the register contents in this stop is A1 for the A register, and Q1 for the Q register. For any first stop, the overflow lamp is always lit.

A register = XXYZ - Stop ID word

where XX = Stop initiator's test number

Y = Number of stops in the series (pairs of A and Q register displays, including this stop)

Z = Type of stop:

1 = Parameter stop

2 = End of section stop

4 = End of test stop

8 = Error stop

Q register = Initiator's Stop/Jump parameter

b. Successive Stop Information

1) Parameter Stop Series

A2/Q2, A3/Q3, ..., AX/QX conveys the test parameters required to correctly execute the test.

2) End of Section Stop

A2 = SS00 (SS is the section number)

Q2 = Return address

- 3) End of Test Stop
 - A2 = NNNN (NNNN is the test execution pass count)
 - Q2 = Return address
- 4) Error Stop
 - A2 = SSEE (SS is the test section number, and EE is the error code)
 - Q2 = Return address
 - A3/Q3 = Support information
through
AN/QN

2. Teletype Input Package

Bit 6 in the SMM parameter word will select the TTY input package. This must be selected when loading the monitor. Once selected, the package can be selected and deselected by setting and clearing bit 6 of the SMM parameter word.

The teletype input driver emulates the operator panel operations in data entry, register selection, setting the SKIP switch, doing a master clear, setting P register run, and inspecting and modifying memory. If using the 1711-1 Teletype, the BREAK light will set between inputs and must be cleared to allow the next input.

a. Data Entry Formats

The following three modes of data entry are chosen for each register entry by the operator.

- Hexadecimal entry. The data is not prefixed and any number of legal hexadecimal characters may be entered. The last four characters will be recognized as the data entry. Deviation from this will result in an error, and restart of the entry.
- Decimal entry. This data is prefixed by the letter N, and any number up to five legal decimal (0 through 9) numbers, right-justified and zero-filled may be entered.
- Bit position selects. This entry is prefixed by S, followed by legal decimal numbers corresponding to the respective bit positions separated by commas and in any order.

Example: S15,0,4,9,2,11 (CR) results in 8A15₁₆.

b. Displayed Data

The data being offered for change is typed out as follows:

XXXX / YYYY? BELL

where XXXX = Hex contents of the A register

YYYY = Hex contents of the Q register

c. Data Entry

The contents of a register is not altered until some data is typed on the TTY. A change is accomplished by selecting one of the three data entry modes and entering the desired data. After the registers have been typed out, as in step b, the first register input will be to the A register. If the contents of the register is not to be changed, go to the register select or entry termination.

d. Register Selects

- 1) Typing a / (slash) will store the contents of the A register, updated or not, and move input control to the Q register. If no input has been made before the slash, the A register will remain unchanged.
- 2) Typing a RUBOUT will void this entry and reset the input control to the beginning of the entry, the A register.
- 3) Data entry mode must be reselected for each register.

e. Entry Termination (Run)

Type a $\textcircled{\text{CR}}$ to emulate a run from the panel. The contents of the A and Q registers is now stored, and the next stop will occur.

f. Sweep and Modify CPU Memory

The operator may look and load memory locations by control from the TTY or CRT keyboard. While at any SMM stop, type LXXXX $\textcircled{\text{CR}}$, where XXXX is any legal hexadecimal memory address. SMM17 will respond with:

XXXX/NNNN?

XXXX is the specified address and NNNN is the contents of that address.

Typing a carriage return will result in SMM17 responding with:

XXXY/NNNN?

XXXY is the next location (XXXX + 1), and NNNN is the contents of this location.

To change the contents of a location, do the following:

LXXXX	(CR)	Operator's request to inspect address XXXX
XXXX / NNNN ?		CPU's presentation of the contents of XXXX
	/ MMMM	(CR) Operator's response to change location XXXX
XXXXY / NNNN ?		CR's presentation of location XXXX + 1
	(CR)	Operator's response, no change
XXXXZ / NNNN ?		CPU's presentation of location XXXX+ 2
	P1000	(CR) Restart execution at \$1000

g. Special Functions

- 1) Manual interrupt emulates the setting of the SELECTIVE SKIP switch and will result in a monitor stop the next time control is returned to the monitor. The Manual Interrupt button is not be used to emulate setting of the SELECTIVE SKIP switch while the TTY test resides in memory.
- 2) Master clear, set P, and go. After any typeout when pointing at the A register, the user may type PNNNN (NNNN is a number in any of the three data modes that will result in an address desired to jump to).

Example: P0 CR = Jump to zero and go
P900 CR = Jump to 900₁₆ and go
PN2304 CR = Jump to 900₁₆ and go.

- 3) A rubout will void the whole entry, including the P or N prefix.

h. Build Test List

<u>TTY Package Output</u>	<u>Operator Response</u>
0000 / 0000?	301/C1
0000 / 0000?	501/? 501/S0,4,9? 501/S07,0,4
0000 / 0000?	N257/0
0000 / 0000?	H
ERROR	
?0	
0341 / 020D?	P0
0301 / 00C1?	

3. Line Printer Package

Setting the line printer equipment code in monitor stop Q3 will select the system line printer for all message outputs. This capability reduces the time required to display operator information messages, error messages, end of section, end of test messages, and the library list output.

Setting bit 15 of the equipment address selects the 1742-30 or 1742-120 logic of this option. If bit 15 is not set, the 1742/501 logic is selected.

If the teletype input package is selected, in addition to the line printer driver package, outputs from this TTY input package displaying information to the operator for possible notification will go to the teletype. This prevents the situation of the operator first inspecting the line printer, and then responding with control statements on the teletype. In the event of a line printer failure, the messages will revert back to TTY or CRT.

The line printer package should not be selected when the line printer is under test.

4. Preexecution Correction Stop

Setting bit 11 of the SMM Stop/Jump word will allow the operator an opportunity to change core locations after all the tests selected during the Build Test operation have been loaded, but no test execution has yet taken place.

This program change format is similar to the program change format in the Edit package. After a descriptive message to the operator, the mainframe will halt with A and Q registers cleared.

The operator is to enter the address to be changed in the A register, the new contents of this address in the Q register, and run the computer. The specified address will be updated, the address will be incremented by one for possible sequential core location changes, and stop for an additional core change. The cycle will continue until the A register is cleared to zero as an address to be changed.

5. Load and Execution Automation

Tailoring a particular system's SMM library according to equipments available can result in a relatively automatic system load and execution (except for the bootstrap). Peripheral equipment addresses as well as test parameter words are easily installed in the library programs via test 3E (system editing program) or test FB (diagnostic replace program). (Monitor's parameter word and system Stop/Jump word are contained in Quick Look test FF.)

Operator understanding of SKIP switch and MANUAL INTERRUPT button use during system load and initialization (refer to first flowchart, this section), familiarity with the SMM/operator interface mechanisms (display/entry stop scheme, Stop/Jump and SMM parameter bit significance) will control efficient system usage.

CAUTION

When prestoring system parameters in Quick Look 1 (test FF), the line printer address must be preset to \$8000 and not \$0000 to disable the printer option. A zero value in Quick Look is illegal, and \$8000 will be masked and processed as a zero.

6. Worst Case Setups

Certain combinations of tests being run together are more effective at exercising the equipment than a test being run alone. Running several tests together may reveal equipment conflicts and critical timing considerations. This procedure may be helpful in simulating another system (that is, MSOS) which detects a failure, but is not adapted for error isolation. The following are examples.

- a. Running one or more copies of COM (Command Test) with a peripheral test (nonrun-alone) is an excellent method of introducing a random start-stop motion. Each additional copy of COM increases the delay between motion operations. The method has been especially effective with PTR Paper Tape Reader Test.
- b. Disk Pack Test DP1 may be loaded at various addresses (see IV.3) for checking direct storage access (DSA) addressing in various areas of core. Only the surface test section should be run during this check. Component isolation may require hand entry of a small, more specific, addressing exercise routine, once the failing area has been located.
- c. When the operating system (MSOS) fails without a clear indication of what happened, running several SMM tests together can, in some cases, simulate the failing conditions of the operating system. To determine what tests to run, observe which I/O operations were in progress when the operating system failed. Then, run the tests on those equipments that were active in the operating system failure. Also, the sections to be run should be selected to simulate the operating system failure. For instance, an assembly from a source tape fails, the tests to run would be a Magnetic Tape Test and the Disk Pack Test (DP1 with Sections 3 and 7 selected).

7. Test Restart

A particular sequence of system events causing an error condition may be recreated by a Master Clear Restart of the system. (Only those tests active in core at time of MC Restart will restart. Those having run to completion must be reloaded.) The following process is used.

a. Multiple Tests Loaded

- 1) Master clear.
- 2) If bit 7 of the SMM parameter word Build Test List is clear, go to 5.
- 3) Set SKIP switch to flag Monitor Parameter stop.
- 4) Clear bit 7 and SKIP switch.
- 5) Run.

b. Single Test Loaded

Any particular test may be restarted (after its initialization) during its program control cycle, if it is not executing the monitor's message subroutine, as follows:

- 1) Master clear.
- 2) Set P register = test IA (load address).
- 3) Run.

8. Quick Look 1 and 2 Command Tests

Quick Look 1 and 2 are abbreviated command tests which check out the instruction set of the 17X4 before the diagnostic monitor is loaded. Quick Look 1 and 2 are automatically loaded and executed each time the monitor is loaded. Quick Look 2 cannot be called as an individual test.

a. Quick Look 1 Command Test

After loading, Quick Look 1 executes in the following way.

- 1) Control is given to Quick Look 1 at either location \$0 or \$7, depending upon autoloading the diagnostic system (\$0) or using bootstrap loader (\$7).

- 2) A checksum is performed as part of the initialization. Whenever the checksum is not equal to zero, an error stop will occur with the P register equal to \$001E.
- 3) Quick Look 1 can be executed even if a checksum error exists. Set the RUN/STEP switch to RUN in order to ignore the error. False error stops may occur when executing Quick Look with these conditions. A checksum error denotes either that Quick Look is not residing in core correctly or that the instructions listing the checksum have failed.
- 4) Whenever an error stop occurs in Quick Look 1, the address displayed in P register indicates the error type. (Refer to Quick Look Errors, Supplement B.)
- 5) Running after an error stop has occurred will cause the failing portion of Quick Look 1 to be repeated.
- 6) Quick Look 1 is programmed to allow restart by a Master Clear and Run. All instructions are executed and checked in Quick Look, except ECA and DCA.
- 7) After execution, Quick Look 1 will stop for parameter inputs if the SKIP switch is set. After this processing, Quick Look 2 will be loaded and executed.

b. Quick Look 2 Command Test

This test checks the CPU's ability to discretely address each memory location, to use various modes of memory addressing, to detect and process system interrupts, to utilize system program protect, and to hold certain data patterns in memory.

Verification of the results of these operations is done by comparison to anticipated results. The following are in the same order as they appear in the program:

- S register test
- Addressing modes test
- Protect test
- Interrupt forcing test
- Memory pattern test
- Bootstrap and intermediate loader restore routine

Refer to QL2 for complete description and operating procedures of Quick Look 2.

C. MONITOR ERROR CODES

During execution of any of the library diagnostics, the monitor is continuously checking system conditions. An example might be memory parity errors, protect violations, unidentified interrupts, or a large group of system failures not directly related to any one diagnostic. This group of errors is always identified by an ID word (identification word in the A1 display location), consisting of an error stop of 00X8 (x defines the number of pairs of register displays in the error message). These monitor error messages can occur any time after Quick Look Command Test has completed and the diagnostic monitor has been loaded.

A specific list and definition of these monitor error codes is listed in Supplement C.

D. SYSTEM SUBROUTINES AVAILABLE TO LIBRARY TESTS

The diagnostic monitor contains a large number of subroutines available to a diagnostic in order to minimize redundant code throughout the library tests.

1. Conversion of hexadecimal numbers to ASCII codes.
2. Generating random numbers.
3. Fixed time delay.
4. Random time delay.
5. Set program protect bits.
6. Clear program protect bit.

Items 3 through 6 are contained within the monitor's MBS overlay. If a test chooses to use these subroutines, the MBS overlay must always be loaded concurrently with the test.

SUPPLEMENT A HAND ENTERED BOOTSTRAPS

- A. Bootstraps are included for the following input devices:
1. 1712/13 Teletype Paper Tape Readers - 1704, 1714
 2. 1712/13 Teletype Paper Tape Readers - 1774
 3. 1721/1777 Paper Tape Readers
 4. 1731/32 - 601/8 Magnetic Tape (7-Track)
 5. 8000 Prepared Mag Tapes on 1732-608/1731 - 601
 6. 1732 - 609 Magnetic Tapes (9-Track)
 7. 1726/405, 1728, 480, 1729 Card Readers
- B. The equipment address in each bootstrap is indicated by WESD. The operator should enter the equipment address of the applicable loading device.
1. Hand-Entered Bootstrap Procedure
 - a. Master clear.
 - b. If on an SC-1700 System Controller, press AUTOLOAD PROTECT button.
 - c. Press P register select button.
 - d. Set P register to \$XFEO (X = number of highest bank).
 - e. Set ENTER/SWEEP to ENTER.
 - f. Press X register select button.
 - g. Clear X register with clear button.
 - h. Enter word of bootstrap in X (see specific bootstrap program).
 - i. Push RUN/STEP switch to STEP.
 - j. Repeat steps g, h, and i until bootstrap has been entered.
 - k. Return ENTER/SWEEP switch to center position.
 - l. Master clear.
 - m. Set SELECTIVE STOP and SKIP switches.
 - n. Make input device READY.
 - o. Press P register select button.
 - p. Set P register to \$XFEO (first word address of bootstrap).

- q. Place RUN/STEP switch in RUN to execute the bootstrap, loading and entering Quick Look.

2. Autoload Procedure

a. Device Autoload

- 1) Master clear.
- 2) Make input device ready.
- 3) Press device AUTOLOAD button.
- 4) Set SELECTIVE STOP and SKIP switches.
- 5) If input device equipment address for direct input (Director Status 1) is \$0181 or edit prestored value, go to step 6. Otherwise, enter director status value in Q register.
- 6) Place RUN/STEP switch in RUN to execute Quick Look.

b. Console Autoload (SC-1700 System Controller only)

- 1) If bootstrap program is not in core, do hand entry (steps 1a through 1j).
- 2) Make input device ready.
- 3) Set SELECTIVE STOP and SKIP switches.
- 4) Press AUTOLOAD button to execute the bootstrap, loading and entering Quick Look.

c. 8000 Magnetic Tape Autoload Procedure

- 1) Master clear the system.
- 2) Place the library tape on an 8000 unit 0 at load point, and make the unit ready.
- 3) Press the 8000 AUTOLOAD button.
- 4) If the equipment address for the 8000 tape is prestored in Quick Look, go to 6.
- 5) Set Q = equipment address (director function zero) for the 8000 tape controller.
- 6) Run.

1712/13 TELETYPE PAPER TAPE - 1704, 1714

XFE0	68FE	TTBOOT	STA*	*-1
XFE1	5803	LDR	RTJ*	FRM
XFE2	011C		SAN	DATA+1
XFE3	1802		JMP*	FRM+1
XFE4	1802	FRM	NUM	\$1802
XFE5	6CF9		STA*	(TTBOOT-1)
XFE6	E000		LDQ	=N\$WESD WESD=EQUIP ADDR
XFE7	WESD		<input type="checkbox"/>	i. e. 0091
XFE8	0A20		ENA	\$20
XFE9	03FE		OUT	-1
XFEA	0DFE		INQ	-1
XFEB	CCF3		LDA*	(TTBOOT-1)
XFEC	02FE		INP	-1
XFED	1CF6		JMP	(FRM)
XFEE	58F5	DATA	RTJ*	FRM
XFEF	0FC8		ALS	8
XFF0	58F3		RTJ*	FRM
XFF1	6CED		STA*	(TTBOOT-1)
XFF2	0102		SAZ	ENDBT-*-1
XFF3	D8EB		RAO*	TTBOOT-1
XFF4	18F9		JMP*	DATA
XFF5	1007	ENDBT	JMP-	QL ENTRY

The highest core bank for autoloader area is indicated by X.

NOTE

Operator instructions on next page.

- * 1. Select TTS mode on teletype and load paper tape into teletype paper tape reader.
- * 2. Refer to Loading SMM with Hand-Entered Bootstrap and follow that procedure.
- * 3. Set switch on teletype paper tape reader to RUN position. Clear BREAK light on teletype if light is illuminated.
4. Quick Look will be loaded and executed (see first flow chart). SMM17 will be loaded.
 - * At that time manually stop the reader, select K mode on teletype.
- * 5. SMM17 ED. X.X will be typed out. Load teletype paper tape reader with tests to be loaded.
 - * Select TTS mode on teletype. Set switch on the teletype paper tape reader to RUN position.
6. PROGRAM PROTECT light will flash while test is being loaded.
 - * When loading has completed, select K mode.
7. Test heading will be typed out. Take reader out of RUN position.
8. Test(s) will be executed.

NOTE

Steps under 4, 5, 6, and 7 refer to the 1712 Teletype and are to be ignored if teletype is a 1713.

1712/13 TELETYPE PAPER TAPE - 1774

XFE0	0000	FRM	NUM	0	
XFE1	0000		NUM	0	
XFE2	0000		NUM	0	
XFE3	C000	TTBOOT	LDA	=N\$1809	
XFE4	1809				
XFE5	68FB		STA*	TTBOOT-2	SET UP JUMP INSTRUCTION
XFE6	0844		CLR	A	
XFE7	68FA		STA*	TTBOOT-1	START COUNTER
XFE8	58F7	LDR	RTJ*	FRM	
XFE9	011A		SAN	DATA+ 1	SKIP AFTER LEADER IS FINISHED
XFEA	6CF7		STA*	(TTBOOT-1)	
XFEB	E000		LDQ	=N\$91	
XFEC	0091				
XFED	0A20		ENA	\$20	
XFEE	03FE		OUT	- 1	
XFEF	0DFE		INQ	- 1	
XFF0	CCF1		LDA*	(TTBOOT-1)	GET FIRST HALF OF WORD
XFF1	02FE		INP	-1	INPUT ONE FRAME OF DATA
XFF2	1CED		JMP*	(FRM)	
XFF3	58EC	DATA	RTJ*	FRM	
XFF4	0FC8		ALS	8	MOVE FIRST FRAME TO UPPER 8 BITS
XFF5	58EA		RTJ*	FRM	
XFF6	6CEB		STA*	(TTBOOT-1)	STORE FULL WORD IN MEMORY
XFF7	0102		SAZ	ENDBT	EXIT ON FIRST ALL ZERO WORD
XFF8	D8E9		RAO*	TTBOOT-1	INCREMENT ADDRESS POINTER
XFF9	18F9		JMP*	DATA	CONTINUE READING WORDS
XFFA	1007	ENDBT	JMP-	7	

The highest memory bank for autoload area is indicated by X.

NOTE

Operator instructions on previous page.

1731/32 - 601/8 MAGNETIC TAPE (7-TRACK)

XFE0	68FE	MTBOOT	STA*	*-1
XFE1	E000		LDQ	=N\$WESD WESD=EQUIP ADDR
XFE2	WESD	EQUIP	<input type="checkbox"/>	i. e. 1382
XFE3	C000		LDA	=N\$405
XFE4	0405		<input type="checkbox"/>	
XFE5	03FE		OUT	-1 Select Unit 0 and Binary
XFE6	09FB		INA	-4
XFE7	0DFE		INQ	-1
XFE8	03FE		OUT	-1 Rewind
XFE9	0F42		ARS	2
XFEA	03FE		OUT	-1 Motion
XFEB	0DFE		INQ	-1
XFEC	0A00		ENA	0
XFED	1807		JMP*	MT2
XFEE	02FE	MT1	INP	-1
XFEF	0FC6		ALS	6
XFF0	BCEE		EOR*	(MTBOOT-1)
XFF1	0109		SAZ	ENDBT-*-1
XFF2	7CEC		SPA*	(MTBOOT-1)
XFF3	D8EB		RAO*	MTBOOT-1
XFF4	02FE	MT2	INP	-1
XFF5	7CE9		SPA*	(MTBOOT-1)
XFF6	02FE		INP	-1
XFF7	0FCA		ALS	10
XFF8	BCE6		EOR*	(MTBOOT-1)
XFF9	7CE5		SPA*	(MTBOOT-1)
XFFA	18F3		JMP*	MT1
XFFB	1007	ENDBT	JMP-	QL ENTRY

The highest core bank for autoloader area is indicated by X.

NOTE

Load the SMM17 Magnetic Tape on unit 0 and ready the unit.

3.0

1732 - 609 MAGNETIC TAPE (9-TRACK)

XFE0	68FE	MTBOOT	STA*	*-1
XFE1	E000		LDQ	=N\$WESD WESD=EQUIP ADDR
XFE2	WESD	EQUIP	<input type="checkbox"/>	i. e. 1382
XFE3	C000		LDA	=N\$444
XFE4	0444		<input type="checkbox"/>	
XFE5	03FE		OUT	-1 SELECT UNIT 0
XFE6	09BB		INA	-\$44
XFE7	0DFE		INQ	-1
XFE8	03FE		OUT	-1 REWIND
XFE9	0F42		ARS	2
XFEA	03FE		OUT	-1 MOTION
XFEB	0DFE		INQ	-1
XFEC	02FE	MT1	INP	-1
XFED	6CF1		STA*	(MTBOOT-1)
XFEE	0102		SAZ	ENDBT-*-1
XFEF	D8EF		RAO*	MTBOOT-1
XFF0	18FB		JMP*	MT1
XFF1	1007	ENDBT	JMP-	QL ENTRY

The highest core bank for autoload area is indicated by X.

1721/1777 PAPER TAPE READER

XFE0	68FE	PTBOOT	STA*	*-1
XFE1	E000		LDQ	=N\$WESD WESD=EQUIP ADDR
XFE2	WESD		<input type="checkbox"/>	i. e. 00A1
XFE3	0A20		ENA	\$20 START MOTION
XFE4	03FE		OUT	-1
XFE5	0DFE		INQ	-1
XFE6	02FE	LDR	INP	-1
XFE7	0112		SAN	DATA+1
XFE8	18FD		JMP*	LDR
XFE9	02FE	DATA	INP	-1
XFEA	0FC8		ALS	8
XFEB	02FE		INP	-1
XFEC	6CF2		STA*	(PTBOOT-1)
XFED	0102		SAZ	ENDBT-*-1
XFEE	D8F0		RAO*	PTBOOT-1
XFEF	18F9		JMP*	DATA
XFF0	1007	ENDBT	JMP-	QL ENTRY

The highest core bank for autoload area is indicated by X.

1731/32-601/8, 1732-2 MAGNETIC TAPE (7 TRACK)

XFE0	68FE	MTBOOT	STA*	*-1
XFE1	E000		LDQ	=N\$WESD WESD=EQUIP ADDR
XFE2	WESD	EQUIP		i. c. 1382
XFE3	C000		LDA	=N\$414
XFE4	0414		Select unit,	556 BPI, & Binary
XFE5	03FE		OUT	-1
XFE6	C000		LDA	=N \$401
XFE7	0401		Rewind,	Clear Controller
XFE8	ODFE		INQ	-1
XFE9	03FE		OUT	-1
XFEA	0F42		ARS	2
XFEB	03FE		OUT	-1 Read motion
XFEC	ODFE		INQ	-1
XFED	0A00		ENA	0
XFEE	1807		JMP*	MT2
XFEF	02FE	MT1	INP	-1
XFF0	0FC6		ALS	6
XFF1	BCED		EOR*	(MTBOOT-1)
XFF2	0109		SAZ	ENDBT-*-1
XFF3	7CEB		SPA	(MTBOOT-1)
XFF4	D8EA		RAO*	MTBOOT-1
XFF5	02FE	MT2	INP	-1
XFF6	7CE8		SPA*	(MTBOOT-1)
XFF7	02FE		INP	-1
XFF8	0FCA		ALS	10
XFF9	BCE5		EOR*	(MTBOOT-1)
XFFA	7CE4		SPA*	(MTBOOT-1)
XFFB	18F3		JMP*	MT1
XFFC	1007	ENDBT	JMP-	QL ENTRY

X designates the highest core bank for autoloading area.

NOTE

For units other than 0, address XFE4 bits 7, 8, and 9 and SMM parameter stop A register bits 9, 10, and 11 must be set to the unit number with the SMM library.

1731/32 - 601/8 MAGNETIC TAPE (7-TRACK) *USING EDIT SPECIAL FORMAT F)

XFE0	68FE	MTBOOT	STA	-1	
XFE1	E000		LDQ	= N\$WESD	
XFE2	WESD				
XFE3	C000		LDA	= N\$ 405	
XFE4	0405				
XFE5	03FE		OUT	-1	
XFE6	09FB		INA	-4	
XFE7	ODFE		INQ	-1	
XFE8	03FE		OUT	-1	Rewind
XFE9	0F42		ARS	2	
XFEA	03FE		OUT	-1	Motion
XFEB	ODFE		INQ	-1	
XFEC	0A00		ENA	0	
XFED	1807		JMP	MT2	
XFEE	02FE	MT1	INP	-1	
XFEF	0F42		ARS	2	
XFF0	BCEE		EOR	(MTBOOT-1)	
XFF1	010A		SAZ	ENDBT-*-1	
XFF2	7CEC		SPA	(MTBOOT-1)	
XFF3	D8EB		RAO	MTBOOT-1	
XFF4	02FE	MT2	INP	-1	
XFF5	0FCA		ALS	10	
XFF6	7CE8		SPA	(MTBOOT-1)	
XFF7	02FE		INP	-1	
XFF8	0FC4		ALS	4	
XFF9	BCE5		EOR	(MTBOOT-1)	
XFFA	7CE4		SPA	(MTBOOT-1)	
XFFB	18F2		JMP	MT1	
XFFC	1007	ENDBT	JMP -	QL ENTRY	

*For tapes prepared on 8000 Mag Tapes.

1732/1732-2 - 609 MAGNETIC TAPE (9-TRACK)

XFE0	68FE	MTBOOT	STA*	*-1
XFE1	E000		LDQ	=N\$WESD WESD=EQUIP ADDR
XFE2	WESD	EQUIP	<input type="checkbox"/>	i. e. 1382
XFE3	C000		LDA	=N\$44C
XFE4	044C		<input type="checkbox"/>	
XFE5	03FE		OUT	-1 SELECT UNIT 0
XFE6	09B3		INA	-\$400-\$44C
XFE7	0DFE		INQ	-1
XFE8	03FE		OUT	-1 REWIND
XFE9	0F42		ARS	2
XFEA	03FE		OUT	-1 MOTION
XFEB	0DFE		INQ	-1
XFEC	02FE	MT1	INP	-1
XFED	6CF1		STA*	(MTBOOT-1)
XFEE	0102		SAZ	ENDBT--1
XFEF	D8EF		RAO*	MTBOOT-1
XFF0	18FB		JMP*	MT1
XFF1	1007	ENDBT	JMP-	QL ENTRY

The highest core bank for autoloader area is indicated by X.

1726/405, 1728/430, 1729 CARD READER

XFE0	0AFE	CRDBOOT	ENA	-1
XFE1	68FD		STA*	*-2
XFE2	E000		LDQ	=N\$WESD WESD=EQUIP ADDR
XFE3	WESD	EQP1	<input type="checkbox"/>	i. e. 0621
XFE4	C812	J430	LDA*	FCN
XFE5	03FE		OUT	-1
XFE6	0DFE	J405	INQ	-1
XFE7	0206		INP	EXT
XFE8	0FC8		ALS	8
XFE9	D8F5		RAO*	CRDBOOT-1
XFEA	6CF4		STA*	(CRDBOOT-1)
XFEB	02FE		INP	-1
XFEC	BCF2		EOR*	(CRDBOOT-1)
XFED	6CF1	INT	STA*	(CRDBOOT-1)
XFEE	E8F4	EXT	LDQ*	EQP1
XFEF	02FE		INP	-1
XFF0	0FCB		ALS	11
XFF1	0131		SAM	EOP-*-1
XFF2	18F3		JMP*	J405
XFF3	CCEB	EOP	LDA*	(CRDBOOT-1)
XFF4	0102		SAZ	ENDBT-*-1
XFF5	18EE		JMP*	J430
XFF6	0081	FCN	<input type="checkbox"/>	
XFF7	1007	ENDBT	JMP-	QL ENTRY

NOTE

Tests selected to be run under SMM17 should be placed in the input tray of the card reader in the same order as selected. If not in the same order, the loader will skip cards until it finds the test selected or detects a Hopper Empty condition.

The highest core bank for autoloader area is indicated by X.

When using 405 card reader locations XFF5 should be 18F0, and location XFF6 should be 0401.

LOADER ON FIRST TWO CARDS FOR 1726/405

0F00	02FE	NEW	INP	-1	Column 1 (7-9)
0F01	02FE		INP	-1	Column 2
0F02	02FE		INP	-1	Column 3
0F03	02FE		INP	-1	Column 4
0F04	02FE	LOOP	INP	-1	Column 5 (Data) 9, 13, ---
0F05	0FC4		ALS	4	
0F06	6828		STA*	TEM405	
0F07	02FE		INP	-1	Column 6 (Data) 10, 14 ---
0F08	6827		STA*	TEM405+1	
0F09	0F48		ARS	8	
0F0A	B824		EOR*	TEM405	
0F0B	6C20		STA*	(BUF)	
0F0C	D81F		RAO*	BUF	
0F0D	C822		LDA*	TEM405+1	
0F0E	A000		AND	=N\$00FF	
0F0F	00FF				
0F10	0FC8		ALS	8	
0F11	681D		STA*	TEM405	
0F12	02FE		INP	-1	Column 7 (Data) 11, 15 ---
0F13	681C		STA*	TEM405+1	
0F14	0F44		ARS	4	
0F15	B819		EOR*	TEM405	
0F16	6C15		STA*	(BUF)	
0F17	D814		RAO*	BUF	
0F18	C817		LDA*	TEM405+1	
0F19	A000		AND	=N\$F	
0F1A	000F				
0F1B	0FCC		ALS	12	
0F1C	6812		STA*	TEM405	
0F1D	02FE		INP	-1	Column 8 (Data) 12, 16, ---
0F1E	B810		EOR*	TEM405	
0F1F	6C0C		STA*	(BUF)	
0F20	D80B		RAO*	BUF	
0F21	0D01		INQ	1	D=1
0F22	02FE		INP	-1	Status
0F23	0FCB		ALS	11	Check EOP
0F24	0132		SAM	EOP- *-1	Yes EOP
0F25	0DFE		INQ	-1	D=0

LOADER ON FIRST TWO CARDS FOR 1726/405 (Cont'd)

0F26	18DD		JMP*	LOOP	
0F27	0FCA	EOP	ALS	10	
0F28	0133		SAM	EXT405-**-1	
0F29	0DFE		INQ	-1	
0F2A	18D5		JMP*	NEW	
0F2B	0000	BUF	NUM	\$0	
0F2C	1001	EXT405	JMP-	1	Start Quick Look
0F2D	0000	TEM405	NUM	\$0	
0F2E	0000		NUM	0	

This loader is on the first two cards of the SMM17 card deck. If these two cards are not available, SMM17 can be loaded by entering this loader starting at location 0F00. After loader has been entered, set P = 0F00 and enter in Q the 1726 equipment address for data transfer (D=0) and run.

SUPPLEMENT B

QUICK LOOK ERROR STOPS AND INTERMEDIATE LOADER ERROR STOPS

All error stops in Quick Look are unconditional stops.

The following is a list of addresses (ADDR) that appear in P register whenever instructions fail in Quick Look. The instruction associated with each address is listed next to it. The instruction that failed with the address may not have caused the error, but the instruments that simulated the instruction could have created the error. A listing should be consulted to further determine the cause of the error.

<u>Error Stop ADDR</u>	<u>Restart ADDR</u>	<u>Instruction</u>
1E		CHSM*
26	23	18XX, + Δ NOT EXECUTED
29	23	JUMP EXECUTED INCORRECTLY
2C	29	1CXX, +Δ NOT EXECUTED
3D	23	JUMP EXECUTED INCORRECTLY
34	30	1400, + IMMEDIATE OPERAND NOT EXECUTED
38	23	JUMP EXECUTED INCORRECTLY
3B	38	18XX, -Δ NOT EXECUTED
3E	23	JUMP EXECUTED INCORRECTLY
42	3F	1CXX, -Δ NOT EXECUTED
45	23	JUMP EXECUTED INCORRECTLY
49	45	1400, - IMMEDIATE OPERAND NOT EXECUTED
4C	49	A REGISTER BIT 15 NOT SET OR SAM FAILED ON A MINUS REGISTER
4F	49	A REGISTER BIT 15 NOT SET OR SAP FAILED ON A MINUS REGISTER
54	50	SKIP A ZERO FAILED ON A SPECIFIC BIT-FAILING BIT IN A REGISTER
56	50	SKIP A NONZERO FAILED ON A SPECIFIC BIT-FAILING BIT IN A REGISTER
5C	59	SKIP A ZERO FAILED ON A ZERO REGISTER
5F	59	SKIP A MINUS FAILED ON A POSITIVE REGISTER
62	59	SKIP A NONZERO SKIPPED ON A ZERO REGISTER
64	59	SKIP A POSITIVE FAILED TO SKIP ON A POSITIVE REGISTER
67	64	Q REGISTER BIT 15 NOT SET OR SQM FAILED ON A MINUS REGISTER

<u>Error Stop ADDR</u>	<u>Restart ADDR</u>	<u>Instruction</u>
6A	64	Q REGISTER BIT 15 NOT SET OR SQP FAILED ON A MINUS REGISTER
6F	6B	SKIP Q ZERO FAILED ON A SPECIFIC BIT-FAILING BIT IN THE Q REGISTER
71	6B	SKIP Q NONZERO FAILED ON A SPECIFIC BIT-FAILING BIT IN THE Q REGISTER
77	74	SKIP Q ZERO FAILED ON A ZERO REGISTER
7A	74	SKIP Q MINUS FAILED ON A POSITIVE REGISTER
7D	74	SKIP Q NONZERO FAILED ON A ZERO REGISTER
7F	74	SKIP Q POSITIVE FAILED ON A POSITIVE REGISTER
85	7F	BOTH SWS AND SWN SKIPPED
87	7F	BOTH SWS AND SWN FAILED TO SKIP
8C	88	SET A OR ADD BIT 0 FAILED
8F	88	A SINGLE BIT ADD FAILED
94	88	ADD ZERO VALUE FAILED
98	88	SUBTRACT -0 FAILED
A1	9D	INDIRECT ADDRESSING FAILED, - Δ
A5	9D	INDIRECT ADDRESSING FAILED, + Δ
AD	A8	RTJ FAILED TO STORE AN ADDRESS
B0	A8	RTJ STORED AN INCORRECT VALUE
BA		EOR
C2		EOR
C9		LDA
D1		AND
D9		AND
E3		STA
ED		STA
F4		SPA - ODD PARITY
FA		SPA
109		SPA - EVEN PARITY
110		SPA
117		TRQ A
11F		LDQ
12A		STQ
134		RAQ
13F		ADQ
14D		SET M or TRM A

<u>Error Stop ADDR</u>	<u>Restart ADDR</u>	<u>Instruction</u>
154		TRQ A
15C		TRQ M
163		TRM Q
169		TRA Q
171		TRA M
176		SOV
179		SNO
17F		SOV
186		SNO
18B		SPE or SNP
18F		SPE or SNP
195		INP or INT REJ
19C		OUT or INT REJ
19F		EXT or INT REJ
1AB		TCA Q
1B2		TCA M
1BC		TCM A
1C3		TCM Q
1CB		TCQ A
1D2		TCQ M
1DF		TCB A
1E7		TCB Q
1EE		TCB M
1FA		EAM A
201		EAM Q
208		EAM M
212		EAQ A
217		EAQ Q
21E		EAQ M
22D		EAB A
235		EAB Q
23D		EAB M
246		LAM A
24C		LAM Q
252		LAM M
25F		LAM A
266		LAM Q
26E		LAM M

Q = EQADR

<u>Error Stop ADDR</u>	<u>Restart ADDR</u>	<u>Instruction</u>
274		LAQ A
278		LAQ Q
27E		LAQ M
288		LAQ A
28D		LAQ Q
295		LAQ M
2A3		LAB A
2AB		LAB Q
2B3		LAB M
2C0		CAM A
2C7		CAM Q
2CF		CAM M
2D9		CAQ A
2DE		CAQ Q
2E6		CAQ M
2F4		CAB A
2FC		CAB Q
304		CAB M
311		TRB A
319		TRB Q
321		TRB M
326		SNF
32A		SPF
332		ARS
33C		LRS
343		QRS
34E		LLS
35A		MUI
366		MUI
371		DVI
376		INA
37B		INQ
386		AAM A
38E		AAM Q
396		AAM M
39E		AAQ A

<u>Error Stop ADDR</u>	<u>Restart ADDR</u>	<u>Instruction</u>
3A3		AAQ Q
3AB		AAQ M
3BA		AAB A
3C2		AAB Q
3CA		AAB M
3D2		SQSQ*
3D8		SOV
3E1		ADDER
3EA		RTJ
3F1		RTJ
3F9		TRA A
3FF		TRQ Q
408		TRM M
40E		TCA A
414		TCQ Q
41D		TCM M or TRM A

*CHSM = Checksum Error. Ignore if on master clear restart

SQSQ - Square Square Failed

ADDER = Adder Failed

The following is a list of the address in the P register when an Address mode fails in Quick Look, the hexadecimal value for the Address mode, whether the delta portion of the instruction is zero or non-zero and the Address modes.

<u>ADR</u>	<u>HEXVAL</u>	<u>DELTA</u>	<u>ADDRESS MODE</u>
46F	0	≠0	No Address mode
475	0	=0	No Address mode
47A	4	≠0	Indirect
480	4	=0	Indirect
486	2	≠0	Q index
48D	2	=0	Q index
494	1	≠0	I index
49C	1	=0	I index
4A2	C	=0	Relative, Indirect
4A8	6	≠0	Indirect, Q index
4AF	6	=0	Indirect, Q index
4B7	3	≠0	Q Index, I index
4C0	3	=0	Q Index, I index
4C7	9	≠0	Relative, I index
4CF	9	=0	Relative, I index
4DE	5	≠0	Indirect, I index
4E6	5	=0	Indirect, I index
4EC	A	≠0	Relative, Q index
4F3	A	=0	Relative, Q index
4F9	E	≠0	Relative, Indirect, Q index
500	E	=0	Relative, Indirect, Q index
508	7	≠0	Indirect, Q and I index
511	7	=0	Indirect, Q and I index
518	D	≠0	Relative, Indirect, I index
520	D	=0	Relative, Indirect, I index
528	B	≠0	Relative, Q and I index
531	B	=0	Relative, Q and I index
539	F	≠0	Relative, Indirect, Q and I index
542	F	=0	Relative, Indirect, Q and I index
549	A	=0	Relative, Q index

(Continued on next page)

Monitor stops:

- STOP 1 P = \$042E, A = SMM ID, Q = Monitor STJP
- STOP 2 P = \$043B, A - SMM Parameter, Q = Final Loader Equip Addr
- STOP 3 P = \$0445, A = Complement of Mask Reg. Q = Equip Addr Printer
 Option

Intermediate loader problems are flagged as follows:

Paper Tape:

1. P=\$0EE1 = checksum error. Run to ignore and set P=\$0ECE to retry data block involved (tape must be moved to beginning of RBD block, see Supplement F).
2. Hang in loop P=\$0EE4 to P=\$0EEE on alarm status previous to frame read.

Card:

1. P=\$0F0F, Q=zero = checksum error, status 1 in A register
2. P=\$0EDA, Q=Equip Addr = load error, status 1 in A register
Reload last card and run.

Magnetic Tape:

- P=\$0EFE = Alarm, status A register. Run to reread RBD block.
Set P=\$0F01 and run to ignore.
- P=\$0F2F = Internal or external rejects; run to retry.

Disk Pack: (85X)

- P=\$0EF6 = Alarm, status in A register; run to ignore.

Cartridge Disk:

- P=0EF9 = Alarm, status in A register; run to ignore.

Storage Module Drive:

- P=\$0F1B = Error stop; run to retry.

8000 Mag. Tape:

- P=\$0EF6 = Error stop; run to retry.
Set P=\$0EFA to ignore, error bits in A register

NOTE

Listings for the intermediate loaders can be found within the Quick Look 1 listings.



SUPPLEMENT C

MONITOR ERROR CODES

The stop identification word (A1) = \$00X8 for all monitor errors (X = 2 or 3). One of the following codes will appear in the A register on the second stop of the series (A2) and have the described significance (a value of 0 for A3 will be displayed as \$8000 due to program flag restrictions):

<u>Code</u>	<u>Description</u>
01	Memory parity error. A3 = interrupted program address.
02	Protect fault. A3 = interrupted program address.
03	Monitor message. Clear computer PROTECT switch(es).
04	Unrequested interrupt. A3 = decimal line number.
05	Interrupt line request conflict. A3 = requestor's IA Q3 = assigned interrupt processor's address. (Master clear from error stop, and restart at test IA for recovery.)
06	Teletype rejected data after data interrupt.
07	MBS could not return to test after interrupt within the time specified by the test.
08	MBS test selected, but MBS routine was not loaded.
09	The selected initial address for a test is invalid.
10	Load area insufficient for test. Retried after tests in core ran; aborted if none (A3 = test/PREG, Q3 = assigned IA).
11	Loader encountered illegal RBD clock ID, A3 = ID word and Q3 = code of load error, if any. Load aborted.
12	No RBD transfer block found. Load aborted.
13	Checksum error (card or paper tape). Attempt reload.
14	Paper tape reader alarm, A3 = status. Attempt reload.
15	Card reader alarm, A3 = status. Attempt reload.
16	Improper card reader EOP, A3 = column count. Try reload.
17	Load tape alarm error, A3 = status and Q3 = times error recurred. (Load retry is aborted on 8th, and successful if less.)
18	Disk pack alarm, A3 = status and Q3 = times error recurred. (Load retry is aborted on 50th, successful if less.)
19	Cartridge disk alarm, A3 = status and Q3 = times error recurred. (Load retry is aborted on 50th, successful if less.)
1A	Overlay loading error. If MBS overlay, reload system.

1B Mass storage subsystem alarm on seek
1C SMD disk subsystem timeout on busy
1D SMD DA alarm
1E SMD not ready or on cylinder
1F SMD drive fault is up
20 Reject from disk adapter
26 No tape file mark
27 Library device reject A3 = A register
Q3 = Q register
28 Library device not ready
Nonstd Code Power fail interrupt. System halts with the A=0018, Q=007F
register contents. Restore power, if required. Master clear
and run.

<u>ADDR</u>	<u>INST</u>
0188	TCA Q
018F	TCA M
0199	TCM A
01A0	TCM Q
01A8	TCQ A
01AF	TCQ M
01BC	TCB A
01C4	TCB Q
01CB	TCB M
01D7	EAM A
01DE	EAM Q
01E5	EAM M
01EF	EAQ A
01F4	EAQ Q
01FB	EAQ M
020A	EAB A
0212	EAB Q
021A	EAB M
0223	LAM A
0229	LAM Q
022F	LAM M
023C	LAM A
0243	LAM Q
024B	LAM M
0251	LAQ A
0255	LAQ Q
025B	LAQ M
0265	LAQ A
026A	LAQ Q
0272	LAQ M
0280	LAB A
0288	LAB Q
0290	LAB M
029D	CAM A
02A4	CAM Q

<u>ADDR</u>	<u>INST</u>
02AC	CAM M
02B7	CAQ A
02BC	CAQ Q
02C4	CAQ M
02D2	CAB A
02DA	CAB Q
02E2	CAB M
02E7	CLR A
02EA	CLR Q
02EE	CLR M
02FB	TRB A
0303	TRB Q
030B	TRB M
0310	SNF
0314	SPF
031C	ARS
0326	LRS
032D	QRS
0335	ALS
033D	QLS
0348	LLS
0354	MUI
0360	MUI
036B	DVI
0370	INA
0375	INQ
0380	AAM A
0388	AAM Q
0390	AAM M
0398	AAQ A
039D	AAQ Q
03A5	AAQ M
03B4	AAB A
03BC	AAB Q
03C4	AAB M

<u>ADDR</u>	<u>INST</u>
3CD	SQSQ*
3D3	SOV
3DC	ADDER
3E5	RTJ
3EC	RTJ
41A	EIAT*
440	TRA A
446	TRQ Q
44F	TRM M
455	TCA A
45B	TCQ Q
464	TCM M or TRM A

*CHSM = Checksum Error. Ignore if on master clear restart

SQSQ = Square Square Failed

ADDER = Adder Failed

EIAT = Extended Indirect Addressing Test Failed

The following is a list of the address in the P register when an Address mode fails in Quick Look, the hexadecimal value for the Address mode, whether the delta portion of the instruction is zero or non-zero and the Address modes.

<u>ADR</u>	<u>HEXVAL</u>	<u>DELTA</u>	<u>ADDRESS MODE</u>
46F	0	≠0	No Address mode
475	0	=0	No Address mode
47A	4	≠0	Indirect
480	4	=0	Indirect
486	2	≠0	Q index
48D	2	=0	Q index
494	1	≠0	I index
49C	1	=0	I index
4A2	C	=0	Relative, Indirect
4A8	6	≠0	Indirect, Q index
4AF	6	=0	Indirect, Q index
4B7	3	≠0	Q Index, I index
4C0	3	=0	Q Index, I index
4C7	9	≠0	Relative, I index
4CF	9	=0	Relative, I index
4DE	5	≠0	Indirect, I index
4E6	5	=0	Indirect, I index
4EC	A	≠0	Relative, Q index
4F3	A	=0	Relative, Q index
4F9	E	≠0	Relative, Indirect, Q index
500	E	=0	Relative, Indirect, Q index
508	7	≠0	Indirect, Q and I index
511	7	=0	Indirect, Q and I index
518	D	≠0	Relative, Indirect, I index
520	D	=0	Relative, Indirect, I index
528	B	≠0	Relative, Q and I index
531	B	=0	Relative, Q and I index
539	F	≠0	Relative, Indirect, Q and I index
542	F	=0	Relative, Indirect, Q and I index
549	A	=0	Relative, Q index

(Continued on next page)

Stops at P=\$0559 and P=\$0566 are for SMM Stop/Jump word (first Q) and SMM parameters, Final loader Eq. Adr. (2nd A/2nd Q) respectively (see first flow chart in the beginning of the section).

Intermediate loader problems are flagged as follows:

Paper Tape:

1. P=\$0EB8 = checksum error. Run to ignore, set P=\$0EA3 to retry data block involved (tape must be moved to beginning of RBD block, see Supplement B)
2. Hang in loop P=\$0EBB to P=\$0EC5 on alarm status previous to frame read.

Card:

1. P=\$0EB1, Q=Zero = checksum error
2. P=\$0EB1, Q=Eq. Adr. = load error, status 1 in A-register
For either, re-load last card and Run.

Mag. Tape:

P=\$0EC9 = Alarm, status A register. Run to re-read RBD block.
Set P=\$0ECD and Run to ignore.

Disk Pack:

P=\$0EC7 = Alarm, status in A-register. Run to ignore.

Cartridge Disk:

P=\$0ECA = Alarm, status in A-register. Run to ignore.

SUPPLEMENT D

SMM1700 PROGRAMMING SPECIFICATIONS

I. INTERFACING WITH SMM

A. TEST STRUCTURE

1. The NAM Card Format:

- a. Columns 2-4 = NAM
- b. Columns 12-14 = (Test Mnemonic) i.e. COM
- c. Column 15 = A (for run alone), Zero (for multiplexed with other test), or P (if the test requires the MBS overlay).
- d. Columns 16-17 = (Test Number) i.e. 01
- e. Columns 25-26 = Month revised, i.e. 06 = June
- f. Columns 27-28 = Date revised, i.e. 15 = 15th
- g. Column 29 = Last digit of year revised, i.e. 2 = 1972
- h. Columns 31-62 = Copyright data as follows:
COPYRIGHT CONTROL DATA CORP 1970
- i. Column 64 = Blank or Zero (0); relocation is to be done by the monitor. (Bias value NOT calculated by the test).

Example:

NAM COM001 06152 COPYRIGHT CONTROL DATA CORP 1970

2. a. The test must not contain an ORG statement to an absolute address.
- b. An ORG * statement (not ORG*) immediately preceding the identification (ID) word of the test's parameter data block is required to facilitate prestored parameter modification by the editing routine. (I.A.4c)
- c. An ORG statement resulting in program address reversal is used at the beginning of an overlay routine and flags the editing routine to insert an overlay marker block in the edited library. The overlay marker block both terminates primary load and identifies the overlay on a subsequent overlay call. Overlay calls are of the form RTJ-

(OVLAY) with A REG = Overlay Number. The routine returns to the user at P + 1 for a successful load, or P + 2 on an unsuccessful load (terminal load error encountered). The LWA of the test must be set to a value sufficient to accommodate the longest overlay to prevent over-laying the next test.

The last program address of the final overlay must be equal to the LWA of the entire test. This is used by the assembler to determine program length for RBD NAMBLK information.

3. All references to the monitor must be made through the use of tags that are equated to absolute low core locations. This block of EQU cards must follow the NAM card. The equate deck structure is indicated in I.I (equates).
4. The first locations of the test must contain the following:
 - a. INITIAL ADDRESS (IA) = A one-word jump (JMP*) instruction to the Master Clear restart address. The tag for this jump instruction should be the same 6 characters used in columns 12-17 of the NAM card.
 - b. IA+1 through IA+3 = An ALF statement of the 6 characters contained in columns 12-17 of the NAM card, i. e., ALF 3, COM001.
 - c. IA+4 = An ADC statement of the Parameter Entry ID word, i. e., ADC IDWRD.
 - d. IA+5 = An ADC statement of the entry to the initialization routine. The monitor always transfers control to the test at IA+5, therefore the test must ensure that its next return address is stored at IA+5 prior to returning control to the monitor.
 - e. IA+6 = Equipment address for Director Status 1. This location may be prestored with the equipment address; if so, it may be changed or entered during loading of the test.

EXAMPLE

COM001	JMP*	PARAM	PARAMETER ENTRY
	ALF	3, COM001	
PARADR	ADC	TAG	TAG=PARAM ID WORD
RETURN	ADC	INITIAL	INITIALIZATION
EQUIP	NUM	0	

5. After IA+6 the test will have subroutines and sections. The following is a list of possible subroutines.
 - a. Repeat Condition
 - b. Repeat Section
 - c. Repeat Test
 - d. Section Selection
 - e. Parameter Routine
 - f. End of Test
 - g. Error Reporting
 - h. Return Control to Monitor
 - i. Status
 - j. Function Selection
 - k. Data Output
 - l. Data Input
 - m. Interrupt Processor
 - n. Counters
6. The last portion of the test will be initialization followed by the END card.

B. INITIALIZATION

1. This portion of the test can be used for a data storage area by the test.
2. This routine is entered via a transfer of control from the monitor to the address (ADC) at IA+5.
3. The following are functions this routine performs:
 - a. Convert IA and frequency count to ASCII, and store in title message.

EXAMPLE

```
LDA      =XCOM001
RTJ-     (HEXASC)
STQ      Location in title for upper 2 characters
STA      Location in title for lower 2 characters
LDQ-     TSACTV
```

LDA- TSFREQ-1,Q
RTJ- (HEXASC)
STA Location in title for frequency count

- b. Change location IA+5 to the address (parameter entry routine) where control is to be returned to the test by monitor. (Run-alone test does not return control to monitor until completion of execution.)
- c. Type out program name, initial address, and frequency count.
- d. Execute RTJ- (CONTROL) to return control to the monitor.
- e. After control is returned to the test the parameter entry routine is executed.
- f. Interrupt line is requested, i. e., RTJ- (REQINT)
- g. Reset return address (IA+5) to the address of test section control.
- h. Execute RTJ- (CONTROL) to exit initialization except for run-alone tests.

C. CONTROL

1. Loading

- a. After loading all tests called or a run-alone test is loaded, the tests are then sequentially given control at location IA+5. After initializing itself the test returns control via RTJ- (CONTROL). This process is continued until all tests are initialized.

2. Execution

- a. Control is passed to a test indirect through IA+5 during execution.
- b. After the test receives control from the monitor, the test must decide whether control can be accepted, that is, is the unit under test capable of continuing testing at this time. If control cannot be accepted by the test, control is returned to the monitor.
- c. The test returns control to the monitor via an RTJ- (CONTROL). Before passing control to the monitor, the test must store the address where the monitor should return control at test location IA+5.
- d. The test should return control to the monitor as often as possible, but not at times that could cause conflicts due to interaction of tests, creating false errors. A run-alone test does not return control to the monitor until it has completed execution. The test must not execute an RTJ- (CONTROL) while in its interrupt processor.

- e. When test has completed execution, the restarting address is to be stored in IA+5, then an RTJ- (EXIT) is executed. If the frequency number was greater than 1, control is returned at the return address through IA+5.
- f. Restarts may be made by Master Clear, set P equal to IA and RUN.

D. STOP AND JUMP

The test must use the conditional stop and jump routines provided by the monitor. Conditional stops and jumps are selected by the operator setting appropriate bits in the Stop/Jump (S/J) parameter word. Each test is provided with its own (SJ) parameter word. This word is displayed in Q whenever a conditional stop is encountered. The test CANNOT ENTER THE STOP AND JUMP ROUTINE WHILE IN INTERRUPT STATE.

1. Conditional Stops in Stop/Jump Parameter

- a. SMM17 provides four programmed stops specified by bits 0 through 4 of Stop/Jump parameter.
 - 1) Stop 1 (Bit 0) - Stop to enter parameter.
 - 2) Stop 2 (Bit 1) - Stop at End of Test section.
 - 3) Stop 4 (Bit 2) - Stop at End of Test.
 - 4) Stop 8 (Bit 3) - Stop on error.
- b. Test does a RTJ- (STOP) with the Identification (ID) word and display information in consecutive locations starting at return address plus one. If the Stop switch is set and the operator selected the stop by setting the appropriate bit in the Stop/Jump parameter word, a stop will occur in the monitor. The format in which data is displayed must conform to the specifications of the individual stop. The first stop will always display the Identification Word in A and the Stop/Jump parameter word in Q.

1) STOP 1 - Stop to enter parameters

Bit 0 of Stop/Jump parameter and ID words must be set for Stop 1 to be executed. Upon stopping, parameters are entered into A and Q in consecutive order until all parameters designated by the number of stops in the ID word have been entered. The display of data begins after the first stop. The data that is presently in the parameter area will be displayed. A set of commonly used parameters (normally for executing without operator intervention) must be prestored in the parameter area, or stored during editing.

Test Bias = The initial address of the test.

EXAMPLE

LDA BIAS The Initial Address
RTJ- (STOP) To stop routine
JMP* LABEL
ORG *
NUM \$XYZ ID word (SMM adds 10_{16})
 XX = test number
 Y = number of stops (excluding ID and S/J
 stop)
 Z = type of stop (Bit 0 set for Parameter stop)
PARAM NUM \$AAAA Test parameter area (maximum
 NUM \$BBBB number of parameters = 28_{10})
LABEL Continue
If ID = \$0C11
SMM displays: A = 0C21
 Q = Stop/Jump
 A = AAAA
 Q = BBBB

2) STOP 2 - Stop at End of Section

Bit 1 of Stop/Jump parameter and ID word must be set for Stop 2 to be executed.

EXAMPLE

LDA BIAS Load A with test bias to stop routine
RTJ- (STOP) ID word (SMM adds 20_{16})
JMP* LABEL
NUM \$XYZ XX = test number
 Y = number of stops (excluding ID and S/J
 stop)
 Z = type of stop (Bit 1 set for End of Section
 stop)
NUM \$SS00 Section number
LABEL Continue
If ID = \$1502 and section number = \$06 SMM displays:
 A = 1522
 Q = Stop/Jump
 A = 0600
 Q = Return address

3) STOP 4 - Stop at End of Test

Bit 2 of Stop/Jump parameter and ID words must be set for Stop 4 to be executed.

EXAMPLE

RAO*	PASS	Update pass count
LDA	BIAS	Load A with test bias to stop routine
RTJ-	(STOP)	
JMP*	LABEL	
NUM	\$XXYZ	ID word (SMM adds 20 ₁₆)

XX = test number
Y = number of stops (excluding ID and S/J stop)
Z = type of stop (Bit 2 set for End of Test stop)

PASS	NUM	\$0005	Pass count
LABEL	Continue		

If ID = \$0C04 SMM displays:

A = 0C24
Q = Stop/Jump
A = 0005
Q = Return address

4) STOP 8 - Stop on error

Bit 3 of Stop/Jump parameter and ID words must be set. The display of data begins at the third stop.

EXAMPLE #1

LDA	BIAS	Load A with test bias to stop routine
RTJ-	(STOP)	
JMP*	LABEL	
NUM	\$XXYZ	ID word (SMM adds 20 ₁₆)

XX = test number
Y = number of stops
Z = type of stop (Bit 3 set for error stop)

NUM	\$SSEE	Section/Error numbers
		SS = Section error occurred in
		EE = Error number
NUM	\$AAAA	Test Data 1
NUM	\$BBBB	Test Data 2
LABEL	Continue	

If ID = \$4F18, section number = 07, and error number = \$21 SMM displays:

A = 4F38
 Q = Stop/Jump
 A = 0721
 Q = Return Address
 A = AAAA
 Q = BBBB

- c. The monitor determines the return address. A register is loaded with the address bias value before entering the stop routine. The monitor determines whether the return address displayed is to be the listing address or memory address (bit 9 of S/J parameter word). The return address is the location within the test from which the test stop routine was called.
- d. When a common error reporting routine is used a location is provided in the stop routine for the return address. The return address is established by an RTJ entry to the common error stop routine which is determined and stored by the test, in the location provided. Before storing, the test must check bit 9 of the Stop/Jump parameter and determine if it is to be stored with or without bias value. The A register is cleared when the call to RTJ- (STOP) is made.

EXAMPLE 2

ERROUT	NUM	0	Common error stop routine entry
	LDA*	ERROUT	Get return address (biased)
	LDQ-	STJP	
	QLS	6	Check bit 9
	SQM	STRTA--1	Bit 9 set, store with bias
	SUB	BIAS	Bit 9 not, set subtract bias
STRTA	STA	RTNADR	Store return address
	CLR	A	Clear A
	RTJ-	(STOP)	
	JMP*	LABEL	
	NUM	\$XXYZ	ID word (SMM adds 20_{16})
			XX = test number
			Y = number of stops
			Z = type of stop (bit 3 set for error stop)

	NUM	\$SSEE	Section/Error numbers
			SS = section error occurred in
			EE = error number
RTNADR	NUM	\$RRRR	Return address
	NUM	\$AAAA	Test data 1
	NUM	\$BBBB	Test data 2
	LABEL	Continue	

2. Conditional Jumps in Stop/Jump Parameter

a. SMM provides 12 programmed jumps specified by bits 4 through 15 of Stop/Jump parameter.

- 1) Jump 0 (Bit 4) - Repeat conditions
 - 2) Jump 1 (Bit 5) - Repeat section
 - 3) Jump 2 (Bit 6) - Repeat test
 - 4) Jump 3 (Bit 7) - Not used
 - 5) Jump 4 (Bit 8) - Omit typeout
 - 6) Jump 5 (Bit 9) - Bias "Return Address " display
 - 7) Jump 6 (Bit 10) - Re-enter parameters
 - 8) Jump 7 (Bit 11)
 - 9) Jump 8 (Bit 12)
 - 10) Jump 9 (Bit 13)
 - 11) Jump 10 (Bit 14)
 - 12) Jump 11 (Bit 15)
- } Optional jumps determined by individual test

- b. Test does an RTJ- (JUMP) with the correct jump mask in A. If the appropriate bit in A register matches the Stop/Jump parameter, control is returned to P (P = return jump address in the test). If a comparison is not found, control is returned to P+1.

EXAMPLE

ENA	\$10	Repeat conditions bit set?
RTJ-	(JUMP)	
JMP*	LABEL	Yes - Set up and repeat condition just executed
Continue		No - Continue on to next condition

- c. Care must be taken to check conditions of Stop/Jump word in logical order, that is, Stop at End of Section (Bit 1) should be checked before Repeat Section (Bit 5) is checked.

3. Stop and Skip Switches

Selective stop and skip switches are used to bring about certain conditions in the operation of the monitor and tests. The tests cannot use these switches for any other purpose than defined in these specifications.

- a. Selective Stop Switch should always be set.
- b. Selective Skip Switch is set by the operator to display the monitor parameters: Stop/Jump Parameter, SMM Parameter and Equipment Address for the resident loader.

E. INTERRUPTS

Test must request permission to select an interrupt line from SMM17. This request cannot be done while in interrupt state. SMM17 gives the test permission to use a particular interrupt line after checking for a previous selection of the same line. SMM17 DOES NOT SELECT THE INTERRUPT.

- 1. Test requests permission to select an interrupt line by executing the following procedure:
 - a. Load appropriate interrupt line parameter into the A register, i. e., line 6 equals bit 6 set. This interrupt line parameter is from the parameter area.

- b. Load the equipment address for status in Q register for the unit to be tested. (Contents of test initial address +6).
- c. Test does a RTJ- (REQINT).
- d. The location following the return jump instruction must contain the address of the test's interrupt processor.

EXAMPLE

LDA	INTLIN	Interrupt line to be requested
LDQ	EQUIP	Equipment address
RTJ-	(REQINT)	To request interrupt
ADC	INTPRO	Address of interrupt processor

Continue

- e. Only one test can request a given interrupt line. The test must be able to handle the common interrupt line or multiple interrupt lines if the unit under test has this feature. The test interrupt processor must determine which interrupt occurred in the situation where more than one unique equipment interrupt is on the same line.
 - f. The functions described in a through c above must be performed when a Master Clear restart is executed.
2. Interrupt processing is accomplished in the following manner:
- a. SMM saves the contents of A, Q, I and M registers.
 - 1) Interrupt on lines two to fifteen - SMM stores the interrupt processor address for the line that interrupted in the I register.
 - 2) Interrupt on line 1 - SMM checks for an interrupt condition for each unit in the slow speed package. If a station is found that interrupted, SMM stores the interrupt processor address for that unit in the I register and passes interrupt control to it. (See step c.) If no slow-speed unit interrupted, the interrupt processor address for that line is stored in the I register and interrupt control is passed to it (See step c.)
 - 3) Interrupt is on line 0 - and there has been a request for that line, SMM passes interrupt control to it. (See step c.) If no request has been made, control is passed to the SMM17 parity and protect fault routine for error output.
 - b. SMM clears the interrupt mask bit corresponding to the interrupt, enables interrupts and does a return jump indirect to the contents of the I register with Q equal to the exit interrupt value for that line.

- c. The test processes the interrupt and passes interrupt control back to SMM via the return address with the exit interrupt value in Q.
- d. SMM restores A, Q, I and M registers and exits interrupt state.

F. MESSAGES

Test communications to the operator must use the monitor TYPEOUT routine for outputting messages. Typical messages would include program name and initial address and operator intervention messages such as Clear Protect Switch. This routine should not be used while in interrupt state.

Message TYPEOUT routine types out ASCII code packed two characters per word on teletype and is executed by using the following procedure:

1. Reset return address (IA+5) to loop address.
2. Monitor will set TTY BUSY switch (Bit 0) when the test enters the TYPEOUT routine.
3. Load A with the first word address of data to be typed out.
4. Load Q with the number of words to be typed out. Set Bit 15 if additional calls to TYPEOUT will be done to complete one line of print in the message. Bit 15 must be cleared for the final call to TYPEOUT to properly terminate the message.
5. Do an RTJ- (TYPEOUT). Monitor will clear BUSY switch Bit 0 to inform other tests when they may use TYPEOUT routine.

EXAMPLE

	LDA	=XLABEL0	
	SFA	RETURN	(IA+5)
LABEL0	ENA	1	
	AND-	INFORM	Check BUSY switch
	SAZ	LABEL1-* -1	subroutine to return
	RTJ-	(CONTROL)	control to monitor
LABEL1	LDA	=XDATA	Data address (biased)
	ENQ	DATA1 - DATA+1	Number of words
	RTJ	(TYPEOUT)	TYPEOUT routine
	JMP*	CONTINUE	
DATA	NUM	\$8D0A	CR, LF
	ALF	X, MESSAGE X	To be typed MESSAGE
DATA1	NUM	\$8D0A	CR, LF

G. AVAILABLE SUBROUTINES

The following are subroutines available to the test. These subroutines cannot be used while in interrupt state.

1. Convert hexadecimal numbers to ASCII codes.
 - a. Load A with numbers to be converted.
 - b. Do a RTJ- (HEXASC).
 - c. When control is returned to subprogram, Q equals the upper two characters and A equals the lower two characters.
2. Generate Random Numbers
 - a. Load Q with the number of words to be generated.
 - b. Load A with the first word address of area where numbers are to be stored.
 - c. Do a RTJ- (GENRAN).

H. INFORM WORD

The information word is a monitor low core location which contains the Busy switches, M register size and core size. The information word is divided as follows:

1. Bit 0 Busy switch for teletype
2. Bit 1 Busy switch for paper tape reader
3. Bit 2 Busy switch for paper tape punch
4. Bit 3 Busy switch for card reader
5. Bit 4 Busy switch for 1716, 06 No. 1
6. Bit 5 Busy switch for 1716, 06 No. 2
7. Bit 6 Busy switch for 1716, 06 No. 3
8. Bit 7
9. Bit 8
10. Bit 9
11. Bit 10
12. Bit 11 Mask Size
 - a. 0 = 4 Bit mask
 - b. 1 = 16 Bit mask

13. Bit 12 to 15 - Core size

- a. 0000 = 4K
- b. 0001 = 8K
- c. 0010 = 12K
- d. 0011 = 16K
- e. 0100 = 20K
- f. 0101 = 24K
- g. 0110 = 28K
- h. 0111 = 32K

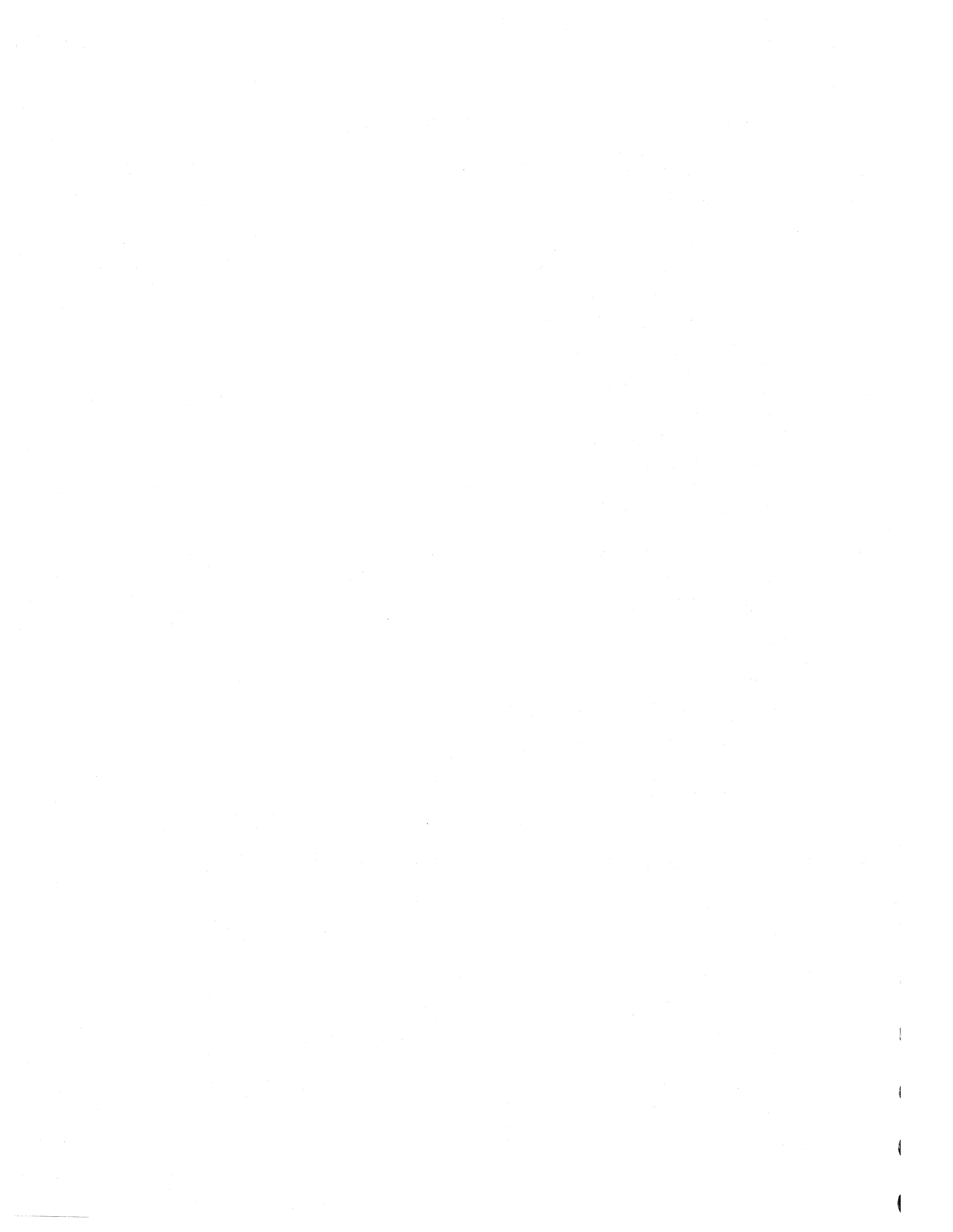
I. EQUATES

List of monitor location equates for test use are:

EQU	CONTROL (1)	RETURN SMM CONTROL ADDRESS
EQU	STOPX(CONTROL+1)	ALL STOPS AND ERROR TYPEOUTS
EQU	EXIT(STOPX+1)	END OF TEST-PASS EXIT
EQU	REQINT(EXIT+1)	REQUEST INTERRUPT ADDRESS
EQU	FCLRINT(REQINT+1)	FAKE CLEAR INTERRUPT ROUTINE
EQU	JUMP(REQINT+2)	CHECK SKIP SWITCH FOR PARAM ENTRY
EQU	GENRAN(JUMP+1)	RANDOM NUMBER GENERATOR
EQU	TYPEOUT(GENRAN+1)	MESSAGE TYPEOUT ROUTINE
EQU	TTYBZY(TYPEOUT+1)	ROUTINE TO WAIT TTY NOT BUSY
EQU	HEXASC(TYPEOUT+2)	HEX TO ASCII CONVERSION
EQU	OVRLAY(HEXASC+1)	OVERLAY CALL ROUTINE
EQU	RELPOS(OVRLAY+1)	A/Q RELATIVE POSITION TEST
EQU	MAINL(RELPOS+1)	MANUAL INTERRUPT ROUTINE
EQU	SETMASK(MAINL+53)	M REGISTER RESET VALUE
EQU	STJP(SETMASK+1)	STOP/JUMP PARAMETER WORD
EQU	LASTVALU(STJP+1)	MONITOR AREA LWA+1
EQU	LASTAD(STJP+2)	AVAILABLE LOAD CORE FWA
EQU	LDLCORE(LASTAD+2)	BANK0 LAST ADDRESS
EQU	LDL1COR(LDLCORE+1)	BANK1 LAST ADDR
EQU	INFORM(LDL1COR+1)	BK0 SIZE, MASK SIZE, BUSY SWITCH
EQU	SMMCNT(INFORM+13)	SMM PARAMETER WORD
EQU	BIT00(SMMCNT+21)	XXXXXXXXX CONSTANT TABLE XXXXXXXXX
EQU	BIT0(BIT00)	
EQU	BIT1(BIT0+1)	
EQU	BIT2(BIT1+1)	
EQU	BIT3(BIT2+1)	

EQU BIT4(BIT3+1)
 EQU BIT5(BIT4+1)
 EQU BIT6(BIT5+1)
 EQU BIT7(BIT6+1)
 EQU BIT8(BIT7+1)
 EQU BIT9(BIT8+1)
 EQU BIT10(BIT9+1)
 EQU BIT11(BIT10+1)
 EQU BIT12(BIT11+1)
 EQU BIT13(BIT12+1)
 EQU BIT14(BIT13+1)
 EQU BIT15(BIT14+1)
 EQU H0000(BIT15+1)
 EQU HFFFF(H0000+1)
 EQU H000F(HFFFF+1)
 EQU H00F0(H000F+1)
 EQU H0F00(H00F0+1)
 EQU HF000(H0F00+1)
 EQU H00FF(HF000+1)
 EQU HFF00(H00FF+1)
 EQU HFFF0(HFF00+1)
 EQU H0FFF(HFFF0+1)
 EQU HFF0F(H0FFF+1)
 EQU HF0FF(HFF0F+1)
 EQU H7FFF(HF0FF+1)
 EQU H7F00(H7FFF+1)
 EQU H0780(H7F00+1)
 EQU H007F(H0780+1)
 EQU H2020(H007F+1)
 EQU TACTV(H2020+6)
 EQU TSFREQ(TACTV+1)

TABLE INDEX TO TEST IN CONTROL
 TEST, FREQUENCY TABLE



EQU FN(\$10)
EQU MNTRST(\$11)
EQU CKST(\$12)
EQU RECKST(\$13)
EQU ERROR(\$14)
EQU RDMLY(\$16)
EQU FIXDLY(\$17)
EQU SETPP(\$18)
EQU CLRPP(\$19)
EQU RD(\$1A)
EQU WR(\$1B)
EQU HOG(\$1C)
EQU MSINIT(\$20)
EQU RINT(\$21)
EQU DSELIN(\$22)
EQU SELIN(\$23)

The test must not use any low-core locations for storage use. Low-core is restricted to monitor use. This does not include location \$00FF, i. e., I register.

II. TEST DESIGN SPECIFICATION

A. SECTION STRUCTURING

1. Sections should be structured so as to allow any section to be executed by itself or following any other section. Section must be repeatable and, therefore, conditions must be initialized before each pass. The routine for selecting

sections should allow for the addition of more sections in the future. Sections should call on common subroutines as often as possible. Control must be returned to the monitor between sections.

2. One section should be singled out as an interrupt test although other sections may also use interrupts.
3. At least one section should check buffered data transfers in case the equipment is on a 1706 or 1716.
4. Conditions must be repeatable whether an error occurred or not. Each condition should be a complete entity in itself. All initial conditions must be included in the loop. Channel and non-channel considerations must be taken into account. Control must be returned to the monitor between conditions.
5. The sections must be written such that a Master Clear on the computer can be done at any time. After the Master Clear and restarting the test at IA, the test must still execute correctly.

B. STATUS

1. A full status word compare must always be used. Status should be taken and two compares of the bits made, check for those bits that are expected to change and then a check for those bits that are not expected to change.
2. Status operation should be done in a subroutine rather than in line code.
3. Both the channel and equipment status should be checked when applicable. The channel status must be taken before the equipment status. (If an equipment status attempt is made when the channel is Busy, a reject will occur.)
4. When test is in Wait mode because of positioning, I/O, timing, etc., the program should be monitoring the full status word, not only the single condition bit. Abnormal and other erroneous status conditions could be missed if this check is omitted.

C. HANGING

1. The test must ensure that the I/O channel is protected from hanging up. Some form of time out along with status checking (full word) should be used.
2. To prevent hanging the system, the test must not hang on an INPUT or OUTPUT instruction i. e., INP -1. All rejects should use some form of time out or retry counts along with status checking.

D. RETRY CONSIDERATIONS

The test should have two selectable modes of operations where applicable.

1. Report all errors and allow normal repeat condition.
2. When detecting a data or parity error, repeat the Read or Write operation N times before reporting error. If retry corrected the error, update the number of retries required.

The test can repeat an erroring operation N times before reporting the malfunction. This is usually done only on reading and writing of data. The number of times the retry procedure was used must be updated and reported when the test is completed.

E. ERROR REPORTING

All available pertinent information should be reported in the error display. This information should be gathered at the closest possible time to the moment of the erroring condition. Examples of error information to report follow:

1. Channel conditions - All status
2. Equipment conditions - All status
3. Data written and data read
4. Timing
5. Retries
6. Address at which error was detected
7. Contents of pertinent registers
8. Interrupts selected and expected
9. Critical locations in test

F. TIMING CONSIDERATIONS

1. Running Time
 - a. Total time to execute the entire test with the prestored parameters should be available in the test writeup.
 - b. Total time to execute each section of the test should be available in the test writeup.
2. Timing of Operations
 - a. Timing must take into consideration the differences in configurations, that is, with a channel or without a channel.
 - b. For accurate timing in critical portions of the test, it may be necessary to run this portion by itself, that is, not with other tests.
 - c. Timing must take into consideration the differences between the 1700 processors (1704/1714, 1774, and 1784) regarding the differences in in-

struction execution time. A test must execute properly on either of the processors unless it is designated to be restricted to a particular processor. Bits 2 and 3 of the SMM parameter denote the various memory speeds of the processors as follows:

Conditions of		Processors	Memory Speed
Bits 3	2		
0	0	1704/1714	1.1 microseconds
0	1	1774	1.5 microseconds
1	0	1784	900 nanoseconds
1	1	1784	600 nanoseconds

G. TAG REFERENCES

Tags must always be used by the test in place of relative numeric values when referencing test and monitor locations. Involved are the SKIP, I/O, and all Memory Reference instructions. If tags are not used, adding and/or deleting op-codes from the source could necessitate updating the relative numeric values. Also, relative numeric values should not be used with tags.

Example:

Tag References		Relative Numeric (Do Not Use)	
SAZ	MEM001-*-1	SAZ	3
INP	PTR004-*	INP	3
JMP*	MTA430	JMP*	*+4
LDA	LPA500	LDA	TAG+6
RAO	(COM250)	RAO	(TAG+5)
ADD-	BIT0	ADD-	\$D4

H. MASTER CLEAR RESTART CONSIDERATIONS

1. Test will be designed to accomplish Master Clear Restart in the following sequence:
 - a. Master Clear, set P register to the test initial address, and Run. IA contains a Jump (JMP*) instruction to the Parameter Entry Routine. This is to allow the operator to change parameters on restarting the test regardless of the condition of bit 10 Stop/Jump word.
 - b. Subsequent to the execution of Parameter Entry Routine the test must request interrupt line from monitor, i. e. RTJ- (REQINT).
2. After completion of the above functions the test transfers control to the test section control routine and continues execution.

I. DESIGN CAUTION

Test will not be designed to have more than one level of indirect addressing due to 65K mode restrictions.

I. DESIGN CAUTION

If a BZS or BSS statement is used in the test, it must be followed by a DAT statement of equal value followed by a NOP to establish the proper length in the NAMBLK. This is used in the disk_pack loader to determine the number of data blocks to skip to find the next NAMBLK when searching through the library.

```
Example:  BZS      BLOCK (500)
          DAT      AIDDPLDR (DAEND-BLOCK+1)
          DAEND    NOP      0
```

J. Special Considerations for Tests Designed under SMM17, Version 2.3, requiring the test to calculate bias.

1. The first locations of the test are the same as the above except:
 - a. IA+5 = An ADC statement of the Return Jump instruction to the initialization routine.
 - b. IA+7 = The value for calculating the test bias. This is the address of this location, i.e., ADC*. The Bias Value is calculated during initialization and stored at IA+7. During initialization the Bias Value is added to certain instructions for relocation.
2. Initialization of the test is same as above except:
 - a. The initialization routine is entered via a Return Jump from the address specified by IA+5.
 - b. The following are functions this routine must perform:
 - 1) Bias must be computed for adding the relocation factors to certain instructions. Example of computing bias:

```
INITIAL  NUM  0      Return Jump
          LDA* INITIAL Address of this routine
          SUB  IA+7   Bias value
          STA  IA+7   This is the test bias value
```

- 2) Add relocation factor to the following instructions:
 - a) All - ADC's
 - b) All instructions in the form "=XLABEL".
 - c) Two word absolute

EXAMPLE

```
LDA  IA+7  Bias value
LDQ  TAG
AAQ  Q
STQ  TAG
```


SUPPLEMENT E MONITOR BASED SUBROUTINES
PROGRAMMING SPECIFICATION

I. INTRODUCTION

A. GENERAL

1. The collection of monitor based standardized subroutines, to be referred to as MBS, provides the basis for a pseudo language for diagnostics.
2. Using MBS results in the following:
 - a) Tests are shorter; redundant coding is nearly eliminated.
 - b) Tests are standardized; error reporting, except for data errors, is done through the monitor.
 - c) The writing of tests is simplified giving the writer time to design a more thorough diagnostic.
3. MBS provides an environment for multiplexing tests. The multiplex philosophy is opposite standard SMM17 techniques in that it will try to trade off control every other time an I/O subroutine is called unless an interrupt condition exists or an inhibit trade flag (HOG) is set. Multiplexing tests this way results in trading off control at a much higher rate thereby more closely simulating 1700 MSOS.

B. 1700 MSOS FOREGROUND/BACKGROUND SIMULATION

1. During SMM17 initialization the protect option message will be output to the operator if the MBS bit was set in the SMM control word (bit 08).
2. In this mode the monitor is protected (foreground) and the tests are not protected (background).
3. Each MBS call results in a legal protect fault similar to 1700 MSOS. Non MBS protect faults will produce an error. This mode of operation is transparent to the diagnostic and does not impose special restrictions to the programmer.

MBS tests cannot execute with standard SMM17 library tests in the Protect mode.

II. INTERFACING WITH MBS

A. TEST STRUCTURE

1. The ALF card at IA+1 must contain the letter P following the three letter test mnemonic. For example test 80 (BG504 drum test) would be:

ALF 3, DRMP80

2. IA+11 contains an ADC for the address of the error/operation file. This is a 93 word file that is located at the end of the test. It is constantly being updated by MBS with I/O data such as expected and actual status, type of operation being performed, and timing information. It also contains equipment codes and interrupt data. All MBS error information is pulled from the tests error file.
3. IA+12 describes the number of error files for the test. This number is typically zero for one file. If the test was talking to two devices, then IA+12 would be incremented by one prior to performing I/O on the second device. This process would continue for as many devices as required.
4. IA+13 contains the unit/station number when testing controllers that have sub-devices connected. This information will be output with the clear text message to be available on a future SMM17 release.
5. The error file has the following structure:

ERFILE	NUM	0	FWA of error file
	NUM	0	Status 1 address set by MBS
	NUM	\$00XX	Status 2 director bits 0 = none
	NUM	\$00YY	Status 3 director bits 0 = none
	NUM	\$00ZZ	Status 4 director bits 0 = none
	NUM	0	Channel Address status set by MBS
	NUM	0	Equipment address set by MBS
	BSS	(TSDATA)	Remaining file data

The programmer must supply director bits or station codes necessary to copy status 2, 3, and 4 if they exist. The MBS initialize routine overlays the status 1 location with the address supplied to SMM17 at test load time. The remaining status addresses will be made if XX, YY, and ZZ are non-zero. The address to read channel address status will be set by MBS if a 1706/1716 was specified. The equipment number for the device will be set by MBS. The error file should be located at the end of the test. The value for TSDATA will be found in the EQU deck. Consecutive files must be located back-to-back.

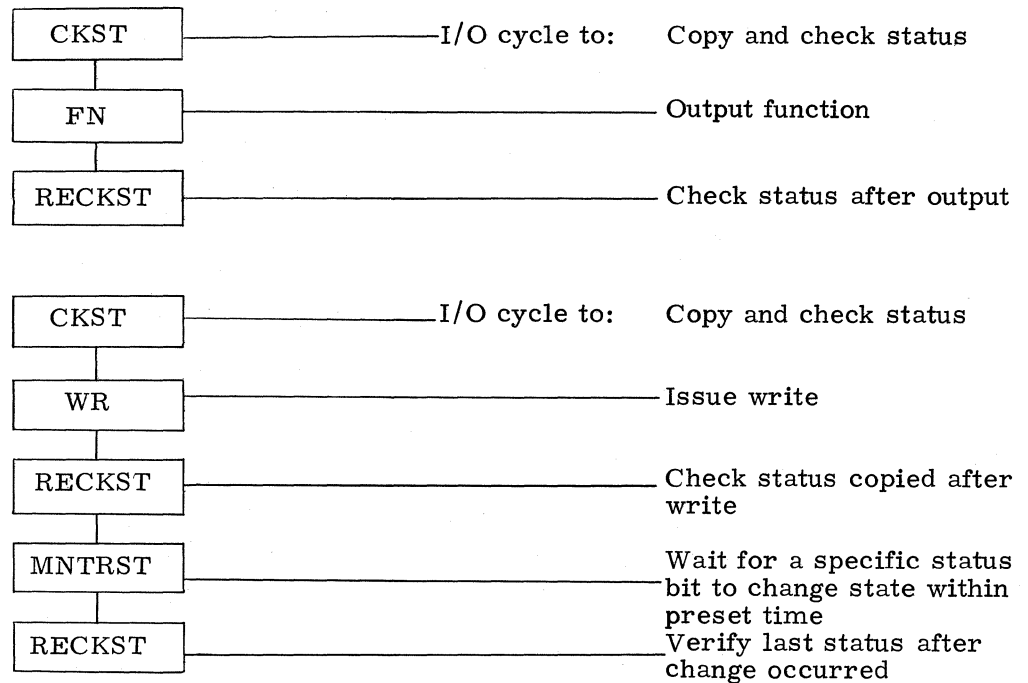
B. AVAILABLE SUBROUTINES

1. The following subroutines are available for I/O:
 - a. CKST - Check status
 - b. RECKST - Recheck status
 - c. MNTRST - Monitor status
 - d. FN - Function
 - e. RD - Read
 - f. WR - Write
 - g. RINT - Recognize interrupt
2. The following subroutines are available for utility:
 - a. FIXDLY - Millisecond delay
 - b. RDMLY - Random millisecond delay
3. The following subroutines are available for control:
 - a. MSINIT - MBS initialize
 - b. SETPP - Set protect bit
 - c. CLRPP - Clear protect bits
 - d. SELIN - Select interrupt
 - e. DSELIN - Deselect interrupts
 - f. HOG - Inhibit test multiplex
 - g. ERROR - Report error

C. SUBROUTINE STRUCTURE

1. The most efficient use of the I/O subroutines is to group them in an I/O cycle.
2. An I/O cycle is the group of calls in the sequence required to issue a function, read or write to a specific device. Typically there is one I/O cycle per test but there is no limit. The I/O cycle is generally located near the test front end or before this first section.
3. The function of each section will be to set up the I/O cycle for the type of I/O required (FN, RD, WR). There is no limit of passes through the I/O cycle per section.

4. I/O cycle call examples:



D. INTERRUPT PROCESSING

1. The condition of any given interrupt line can be in four modes:
 - a. Illegal - This line has not been selected by a test.
 - b. Expected - This line has been selected by a test, but the hardware interrupt has not occurred.
 - c. Active - This line has received a hardware interrupt, it has been pre-processed by the monitor but it has not been processed by the test selected it.
 - d. Process - This line has caused control to be given to the test that selected it and that test is now processing this interrupt.
2. A line is in the ILLEGAL mode if an interrupt occurs prior to selecting it or after the DSELIN call, deselect interrupt, has been issued.
3. MBS will handle a maximum of four lines per test. The input parameter for interrupt line assignment must have the following format:

B1	L1	B2	L2
B3	L3	B4	L4

Must be a two-word entry

B (X) = Bit position in director status word associated with assigned line.

L (X) = Assigned interrupt line.

EXAMPLE:

4500 Controller has one interrupt on line 5 which correlates to
0000 status bit (E. O. P.).
3F4F Status bits 3, 4 and 5 (data, E.O.P. and alarm) are on
5F00 line 15.

4. The call format for DSELIN is:

CALL

RTJ- (DSELIN)
+0 NUM \$XXXX X=bit number(s)

PARAMETERS

+0 Bit number(s)

The bit position of the expected interrupt in the equipment status word; must agree with interrupt input parameter(s).

RETURN

- a) This interrupt line is now illegal.
- b) The Interrupt Mark register for this bit is set.
- c) The timer is cleared.

EXAMPLE RTJ- (DSELIN)
 NUM \$0018 Clear interrupt lines for
 data and E. O. P.

5. To make an interrupt(s) legal, the select interrupt call is used.

CALL

	RTJ-	(SELIN)	
+0	NUM	\$XXXX	X=Bit number(s).
+1	NUM	\$tttt	t=System (not hardware) times.
+2	ADC		Processor address.
+3	ADC		Interrupt parameter FWA.

PARAMETERS

+0	Bit position in equipment status word (same as DSELIN).
+1	A maximum allowable time in milliseconds between Active and Process modes, max = 7FFF.
+2	Process routine address.
+3	First word address of two-word interrupt line assignment from parameters input area.

RETURN

Interrupts occurring will be in the Expected mode.

EXAMPLE

	RTJ-	(SELIN)	
	NUM	\$0010	Select line associated with
	NUM	\$0100	bit 04. Wait 256 milli-
	ADC	PROCES	seconds to process test.
	ADC	INTWRD	Process routine address
			first word of interrupt data.

6. The recognize interrupt routine is used by a diagnostic to process an expected interrupt.

CALL

	RTJ-	(RINT)		
	+0	NUM 0	Time Limit	
	+1	NUM 0	Director 1 Status Mask	
	+2	NUM 0	Director 1 Status Value	
	+3	NUM 0	Channel Status Mask	
	+4	NUM 0	Channel Status Value	
	+5	NUM 0	Channel Address Mask	
	+6	NUM 0	Channel Address Value	
Optional	{	+7	NUM 0	Director 2 Status Mask
		+8	NUM 0	Director 2 Status Value
	+9		Error Return	
	+10		Normal Return	

PARAMETERS

- +0 Time Limit - Defines the maximum time allotted for the monitor to receive the interrupt. This time limit is to be in 1-millisecond increments, between the limits of 0 and 32,767 (\$7FFF) milliseconds.
- +1 +3 Status Mask - Defines those bits which are to be included in each status check while waiting for the interrupt to occur. Status is not copied or checked after the interrupt occurs.
- +5 +7
- +2 +4 Status Value - Defines the expected status value of masked status being checked.
- +6 +8
- +9 The monitor will return control to the diagnostic at +9 if any of the masked status is not exactly as specified by the respective status values, or the time limit specified in +0 has been exceeded. This must be a one-word instruction.
- +10 When the selected interrupt occurs, the monitor returns control to the test's interrupt processor routine in the format of a return jump to the address specified in +2 of the select interrupt routine call. An indirect jump through this entry point to exit the processor routine will result in returning test execution to this point, the normal return at +10. For devices that do not have a second director status, error and normal return will be at +7 and +9.

NOTE

For equipments that will not be connected to a 1706/1716, the +3, +4, +5, and +6 parameters may be replaced with status 3 and 4 mask/value fields. Such devices connect directly on DSA.

EXAMPLE

	RTJ-	(RINT)	
+0	NUM	10000	Wait 1 Second
+1	NUM	\$FFFF	Status 1 Mask
+2	NUM	\$0019	Status 1 Value
+3	NUM	\$FDBF	Status 3 Mask
+4	NUM	\$0021	Status 3 Value
+5	NUM	\$FB7F	Status 4 Mask
+6	NUM	\$0063	Status 4 Value
+7	NUM	\$7FFE	Status 2 Mask
+8	NUM	\$139C	Status 2 Value
+9	JMP*	ER8000	Error Return
+10	JMP	TAG	Continue at Normal Return

E. ERROR DETECTION - USE OF STATUS CHECKING

1. Error detection will be done in the monitor. The most often used routines will be those that check status, monitor status, or recheck status. As a result a diagnostic test should make efficient use of the status checking routines. If the reader refers to the description of the three types of status routines available, the following should be clear.
2. Each time a diagnostic performs a single operational or control function on a device, it is strongly recommended that it make use of all three status routines. In order to make use of these it is advisable that the test writer be certain of the equations for each bit in all status words. This information will serve the user in determining masks for the status routines and more important, in determining expected results from certain operations. The equations must be accurate and are made from the prints of a device and its controller.

3. It may also be helpful to list all computer originating function codes and to correlate those functions to the list of status bit equations. This list would then contain a function code followed by a set of bits which would be present if that function code were issued to a device.
4. With these two lists the diagnostic programmer should have the information necessary to make a call to any of the three status routines. The diagnostic should never check status without using a monitor routine. Before performing an operation, the diagnostic should call the check status routine to make sure the equipment can accept the operation. Next it should call the re-check status routine to determine the status that was copied immediately following the last operation. Finally, the diagnostic should call the monitor status routine which will continually check for a certain condition to be met, e.g., EOP either until that condition is met or a specified amount of time has passed. (Note that this time is specified by the diagnostic and it is up to the programmer to determine this time.) After performing all these status checks, there remains one final status check. The diagnostic should perform a RECKST after an MNTST to assure that the final status copied (just after the MNTST found the bit it was waiting for) is the one expected.
5. This series of status checks will provide the diagnostic with one of the most thorough means available to verify any operational or control function of a device. If the programmer ignores these routines, he will not have a good test under SMM17.

F. STATUS ROUTINES

1. Check Status

This routine copies and checks all the status of the device under test. The copied status is masked and compared to the expected values as specified by the call. Director 1 status, buffer channel status, and current channel address parameters must be included in the call. Director 2 status may be included if available from the device under test.

CALL

	RTJ-	(CKST)			
	+0	NUM	0	Director 1 Status Mask	
	+1	NUM	0	Director 1 Status Value	
	+2	NUM	0	Channel Status Mask	
	+3	NUM	0	Channel Status Value	
	+4	NUM	0	Channel Address Mask	
	+5	NUM	0	Channel Address Value	
Optional	{	+6	NUM	0	Director 2 Status Mask
		+7	NUM	0	Director 2 Status Value
	+8			Error Return	
	+9			Normal Return	

PARAMETERS

- +0 +2 Status Mask - Defines those bits which are to be included in each status check.
- +4 +6
- +1 +3 Status Value - Defines the expected status value of each masked status being checked.
- +5 +7
- +8 Error Return - The monitor return control to the test at +8 if any of the masked status is not exactly as specified by the respective status values. The instruction located at CALLPT+8 must be a one-word instruction.
- +9 The monitor returns control to the test at +9 if all the masked status values are as specified.

NOTE

For equipments that will not be connected to a 1706/1716, the +2, +3, +4, +5 parameters may be replaced with status 3 and 4 mask/value fields. Such devices connect directly on DSA.

EXAMPLE

Device has four levels of status and connect directly on DSA.

	RTJ	(CKST)	
+0	NUM	\$DB7F	Status 1 Mask (ignore bits 7, 10, and 13)
+1	NUM	\$0019	Expected Value (EOP, data and ready)
+2	NUM	\$FFFF	Status 3 Mask
+3	NUM	\$1234	Expected Value
+4	NUM	\$FFFF	Status 4 Mask
+5	NUM	\$D636	Expected Value
+6	NUM	\$7FEO	Status 2 Mask (ignore bits 0-4, 15)
+7	NUM	\$0020	Expected Value
+8	JMP*	ERE000	Error Return
+9	JMP	CKE000	Normal Return

2. Monitor Status

This routine will copy and check all requested status while waiting for a specific status bit to change state within a specified time. The last status(s) copied which saw the bit change is not verified. To verify the last status copied the user must call the recheck status routine, RECKST.

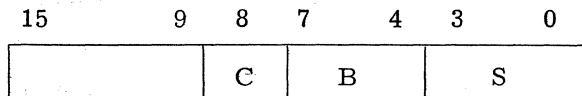
CALL

	RTJ-	(MNTRST)		
	+0	NUM	tttt Time Limit	
	+1	NUM	0CBS Status Control Word	
	+2	NUM	mmmm Director 1 Status Mask	
	+3	NUM	vvvv Director 1 Status Value	
	+4	NUM	mmmm Channel Status Mask	
	+5	NUM	vvvv Channel Status Value	
	+6	NUM	mmmm Channel Address Mask	
	+7	NUM	vvvv Channel Address Value	
Optional	{	+8	NUM	mmmm Director 2 Status Mask
		+9	NUM	vvvv Director 2 Status Value
	+10		Error Return	
	+11		Normal Return	

PARAMETERS

+0 Time in milliseconds to wait for condition to occur.
Maximum time = \$7FFF.

+1 Status Control word = 0CBS



0 = Not used

C = Condition

1 = Wait for status bit to go off

0 = Wait for status bit to go on

B = Bit position in status word

S = Status word

0 = Director 1 Status

2 = Director 2 Status

4 = Channel Status (Director 3 Status)

7 = Channel Address Status (Director 4 Status)

+2, +4 Status Mask - Defines those bits which are to be included
+6, +8 in each status mask.

+3, +5 Status Value - Defines the expected status of each masked
+7, +9 status being checked.

If a 1706/1716 will not be used, then +4, +5, +6 may be replaced with director 3 and 4 status if needed.

+10 Error Return - This must be a one-word instruction.
Control is returned here if:

- a. A status error was detected while waiting for the condition to be met.
- b. The condition was not met within the time specified.

+11 Normal Return - There were no status errors and the condition was met within the specified time.

EXAMPLE

Device does not have a director 2 status.

	RTJ-	(MNTRST)	
+0	NUM	1000	Wait 1 second for bit 01
+1	NUM	\$0110	to go off in status word 1 (busy)
+2	NUM	\$FDFE	Status 1 mask, ignore bit 09
+3	NUM	\$000D	value, expect ready, busy, data
+4	NUM	\$FFFF	channel status mask
+5	NUM	\$0011	value, expect ready, EOP
+6	NUM	\$FFFF	channel address mask
+7	NUM	\$13A6	value of expected address,
+8	JMP*	ERE000	error return,
+9	JMP*	CKE000	normal return. Check repeat conditions.

3. Recheck Status

This routine checks the status previously copied by another routine. The purpose is to verify the final status copied prior to exiting the Read, Write, Function, or Monitor Status routines. With the exception of not copying new status, this routine is identical to the check status routine.

CALL

	RTJ-	(RECKST)	
+0	NUM	0	Director 1 Status Mask
+1	NUM	0	Director 1 Status Value
+2	NUM	0	Channel Status Mask
+3	NUM	0	Channel Status Value
+4	NUM	0	Channel Address Mask
+5	NUM	0	Channel Address Value
+6	NUM	0	Director 2 Status Mask
+7	NUM	0	Director 2 Status Value
+8			Error Return
+9			Normal Return

PARAMETERS

- +0 +2 Status Mask - Defines those bits which are to be included in each status check.
- +4 +6
- +1 +3 Status Value - Defines the expected status value of each masked status being checked.
- +5 +7
- +8 The monitor return control to the test at +8 if any of the masked status is not exactly as specified by the respective status values. The instruction located at +8 must be a one-word instruction.
- +9 The monitor returns control to the test at +9 if all the masked status values are as specified.

NOTE

For equipments that will not be connected to a 1706/1716, the +2, +3, +4, and +5 parameters may be replaced with status 3 and 4 mask/value fields. Such devices connect directly on DSA.

G. FUNCTIONS I/O

1. The actual exercising of a device is accomplished through the use of the Function routine and Read and Write routines. With status checking and use of this Function routine it is possible to perform all related device operations and check all the responses. Errors are automatically picked up by the monitor. The diagnostic need only attempt all the functions. It is up to the diagnostic programmer to determine any timing considerations, necessary function calls, any peculiar orders of functioning which may cause an error, or any other device manipulation.
2. The function routine issues the specified function code, copies all the available status and branches on the specified response control depending on reject and reply.

3. FN (Function)

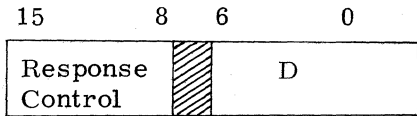
This routine issues the specified function code, copies all the available status and branches on the specified response control depending on reject or reply. The following flow chart should be referenced for more detail on FN operations.

CALL

	RTJ-	(FN)	
+0	NUM	ffff	f = Function Code
+1	NUM	rrdd	r = Response Control/d = Director Bits
+2	NUM	ttt	t = Timer
+3	NUM	bbbb	b = 17X6
+4	Error Return		
+5	Normal Return		

PARAMETERS

- +0 Function code
 The actual function word (16 bits) to be issued to the
 equipment.
- +1 Response Control/D



Response control; the specified action to be taken after a reply or reject.

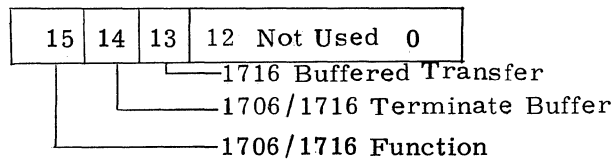
- 00 = CNT
 Continue, regardless of response
- 01 = ERP
 Error on reply
 Continue on reject
- 02 = HRP
 Hang on reply
 Continue on reject
- 10 = ERJ
 Error on reject
 Continue on reply
- 20 = HRJ
 Hang on reject
 Continue on reply
- 22 = HNG
 Hang, regardless of response

NOTE

Bit 15 set in the RRDD word negates copy status and should only be used for special hardware.

D = director bits; appropriate director bits for equipment address to perform desired function.

- +2 Timer
The maximum time (in milliseconds) to hang (HNG, HRP, HRJ).
When the timer has expired, an error will occur.
- +3 17X6
This parameter indicates a function request to a 1706 or 1716.



RETURN

There are two returns:

- +4 Indicates the monitor has sensed an error (see ERR)
- +5 Normal return

TIMING

Assuming no memory cycle stealing due to in process DSA transfers the function requires 400 microseconds on a 1704.

4. The read routine transfers data from the device to the CPU using A/Q or DSA. The routine will branch according to the response (reject, reply) and the specified response control.

Buffered I/O - The response is checked only on the output to the DSA channel. Control is returned to the caller immediately.

A/QI/O - The response is checked after every transfer. Control is returned to the caller after the specified number of words have been transferred.

CALL

	RTJ-	(RD)	
+0	NUM	TTTT	Transfer type
+1	NUM	llll	No. of words
+2	ADC	aaaa	Data buffer FWA
+3	NUM	caca	Continue bit first channel address
+4	NUM	cccc	No. of channels
+5	NUM	RRDD	Response control/director bits
+6	NUM	tttt	Timer (millisecond)
+7			System overload return
+8			Error return
+9			Normal return

PARAMETERS

+0 Transfer type

0 = AQ

1 = DSA

To initiate DSA transfers, the FWA of the buffer (from +2) is output to the device.

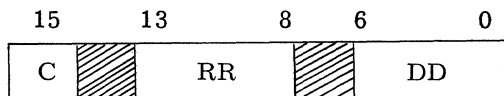
+1 Number of words to be transferred

+2 First word address of data buffer

+3 First extended address (hardware) for device using continue bit addressing (bit 15)

+4 Number of continue bit addresses to increment

+5 Response control



DD = Director (station) bit positions to perform read

RR = Response control

10 = Continue on reply, error on external reject

01 = Continue on external reject, error on reply

20 = Continue on reply, repeat for external reject

C = Copy status after read

0 = Copy status (all specified levels)

1 = Do not copy status. The do not copy feature is for special hardware applications only and is not recommended for standard 1700 peripherals.

- +6 Time in milliseconds to wait for a reply as defined in the response control code RR=20.
- +7 The read routine will return here if this call or another test has made an A/Q read request while a DSA read is in progress. This must be a one-word instruction.
- +8 Control is returned here if:
 - a. A response other than expected occurred.
 - b. The timer expired while waiting for a reply response.The error return must be a one-word instruction.
- +9 Control is returned at the normal return after:
 - a. All A/Q read transfers are complete.
 - b. One output to start DSA transfers

TIMING

<u>From</u>	<u>To</u>	<u>Time</u>
+0 transfer	First transfer	320 microseconds
Nth transfer	N +1 transfer	30 microseconds
Last transfer	First copy status	100 microseconds
Last copy status	+9 return	33 microseconds

NOTE

If DSA transfers were initiated, the caller must use the monitor status call before exiting the I/O cycle for this call.

5. The write routine transfers data from the CPU to the device using A/Q or DSA. The routine will branch according to the response (reply, reject) and the specified response control.

Buffered I/O - The response is checked only on the output to the DSA channel. Control is returned to the caller immediately.

A/Q I/O - The response is checked after every transfer. Control is returned to the caller after the specified number of words have been transferred.

CALL

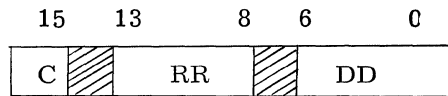
	RTJ-	(WR)	
+0	NUM	TTTT	Transfer type
+1	NUM	llll	No. of words
+2	NUM	aaaa	Data buffer FWA
+3	NUM	caca	Continue bit first channel address
+4	NUM	cccc	No. of channels
+5	NUM	RRDD	Response control/director bits
+6	NUM	ttt	Timer (millisecond)
+7			System overload return
+8			Error return
+9			Normal return

PARAMETERS

+0	Transfer type
	0 = A/Q
	1 = DSA
	To initiate DSA transfers, the FWA of the buffer (from +2) is output to the device.
+1	Number of words to be transferred.
+2	First word address of data buffer.
+3	First extended address (hardware) for device using continue bit addressing (bit 15).

+4 Number of continue bit addresses to increment.

+5 Response control.



DD = Director (station) bit positions to perform write

RR = Response control

10 = Continue on reply, error on external reject

01 = Continue on external reject, error on reply

20 = Continue on reply, repeat for external reject

C = Copy status after read

0 = Copy status (all specified levels)

1 = Do not copy status. The do not copy feature is for special hardware applications only and is not recommended for standard 1700 peripherals.

+6 Time in milliseconds to wait for a reply as defined in the response control code RR=20.

+7 The read routine will return here if this call or another test has made an A/Q read request while a DSA read is in progress. This must be a one-word instruction.

+8 Control is returned here if:

- a. A response other than expected occurred.
- b. The timer expired while waiting for a reply response.

The error return must be a one-word instruction.

+9 Control is returned at the normal return after:

- a. All A/Q read transfers are complete.
- b. One output to start DSA transfers.

TIMING

	<u>From</u>	<u>To</u>	<u>Time</u>
+0		First transfer	320 microseconds
Nth transfer		N +1 transfer	30 microseconds
Last transfer		First copy status	100 microseconds
Last copy status		+9 return	33 microseconds

NOTE

If DSA transfers were initiated, the caller must use the monitor status call before exiting the I/O cycle for this call.

EXAMPLE:

REPEAT

	RTJ-	(RD)	
+0	NUM	\$0000	A/Q transfer
+1	NUM	\$0100	Transfer length = 256
+2	ADC	RDBUF	FWA of read buffer
+3	NUM	\$9400	First continue bit address
+4	NUM	0008	Number of address increments
+5	NUM	\$1001	Continue on reply error on reject, dir = 01
+6	NUM	\$0000	millisecond timer = 0
+7	JMP*	REPEAT	System busy return
+8	JMP*	ERE000	Error return
+9	JMP	CKE000	Normal return

REPEAT

	RTJ-	(RD)	
+0	NUM	\$0000	A/Q transfer
+1	NUM	\$0040	Transfer length = 64 words
+2	ADC	RDBUF	FWA of read buffer
+3	NUM	0	Not used
+4	NUM	0	Not used
+5	NUM	\$2000	Response control, director = 00
+6	NUM	0	Timer = 0
+7	JMP*	REPEAT	System busy return
+8	JMP*	GRE000	Error return
+9	JMP*	CKE000	Normal return

III. ERROR REPORTING

A. ERROR HANDLING

1. When a monitor routine senses an error, control is returned to the caller at the error return address. On the return, the A register will contain an error code describing the type of error encountered.

B. ERROR CODES

1. <u>Subroutine</u>	<u>Error Code</u>
a. CKST	0003
b. MNTRST	0000, 0003
c. RECKST	0003
d. FN	0001 0002
e. RD	0001 0002
f. WR	0001 0002
g. RINT	0003, 0004

2. Error Code Definitions

	<u>MBS Error Code</u>	<u>Definition</u>
a.	0000	Monitor status time out
b.	0001	I/O response time out (reply, reject)
c.	0002	Response was other than predicted
d.	0003	Status error actual \neq expected
e.	0004	Interrupt time out. Interrupt did not occur within specified time

- The error subroutine gets the data from the error file and outputs the error.

CALL

	RTJ-	(ERROR)	
+0	NUM	\$SSEE	Section/error code
+1	ADC	RRRR	Repeat conditions address
+2	NUM	RTN	Address of error caller
+3		Normal return	No repeat requested, continue

PARAMETERS

+0	SS = Section code EE = Error code
+1	Address to repeat conditions if operator requested
+2	Address of the error caller
+3	Control returned here if repeat conditions not requested

EXAMPLE

	RTJ-	(ERROR)	
+0	NUM	\$0630	Section 6, error 30
+1	ADC	RPE000	Repeat conditions address
+2	NUM	125C	Return to caller address
+3	JMP*	EXIT	Return to test section

- Data errors are not handled by MBS. The user must use standard SMM17 error reporting techniques for data errors.
- I/O cycle error reporting is as follows:

The first four pairs of A/Q typeouts have a standardized definition. These definitions are as follows:

A1	Q1	A2	Q2	A3	Q3	A4	Q4
ID	ST/JP	SS/EE	RTN	MBSERR	IO/RP	(A)	(Q)

ID = SMM ID word

ST/JP = SMM Stop/Jump parameter word

SS/EE = Test section number and test error number

RTN = Return address

MBSERR = MBS monitor detected error code

- 0000 - Status time out error
- 0001 - I/O time out error
- 0002 - I/O response error
- 0003 - Status error
- 0004 - Interrupt time out error

IO/RP = Last I/O operations performed and the associated response

- IO - 10 = Write
- 20 = Read
- 30 = Function
- RP - 10 = Reply
- 20 = External Reject
- 30 = Internal Reject

(A), (Q) = Register contents relative to the last I/O operation.

The next three pairs of A/Q typeouts vary with the type of MBS monitor detected error code. The content of these three pairs of typeouts are as follows:

MBS Error

<u>Code</u>	<u>A5</u>	<u>Q5</u>	<u>A6</u>	<u>Q6</u>	<u>A7</u>	<u>Q7</u>
00	ACT STS1	ACT STS2	CLOCK	OCBS	-	-
01	ACT STS1	ACT STS2	CLOCK	-	-	-
02	ACT STS1	ACT STS2	-	-	-	-
03	ACT STS1	EXP STS1	ACT STS2	EXP STS2	-	-
04	ACT STS1	ACT STS2	CLOCK	INT LINE	INT MASK	ASGN LINES

ACT STS1, ACT STS2 - Actual status 1 and status 2 copied from the device.

EXP STS1, EXP STS2 - Expected status 1 and status 2.

CLOCK - The time limit in milliseconds which expired waiting for the specified condition to occur.

OCBS - The control word being used during monitor status (MNTRST).

INT LINE - The line number of the interrupt which occurred.

INT MASK - The interrupt mask being used at the time of the interrupt.

ASGN LINES - A summation of all the interrupt lines assigned to the device, i.e. 0050 = interrupt line 5.

EXAMPLE

A1	Q1	A2	Q2	A3	Q3	A4	Q4	A5	Q5
8068	0241	0630	12D5	0003	3010	0001	0381	0218	0219

Error analysis:

1. Test 80
2. Section 6, error code 30
3. Return address = \$12D5
4. MBS error code = 0003, status error
5. Last operation performed was a function, response was a reply
6. Contents of the A register = 0001
7. Contents of the Q register = 0371
8. Actual status was \$0218
9. Expected status was \$0219

(The device dropped bit 00 (ready) after a clear controller.)

IV. CONTROL

A. MULTIPLEX CONTROL (MPX)

1. The purpose of this routine is to provide multiplexing control between tests. This subroutine is not available to tests, however, control is provided with the hog subroutine.
2. The MPX subroutine is called by all status and I/O routines except recognize interrupt (RINT). If the hog flag was set previous to the call, control is returned to the I/O or status routine in 15.4 microseconds. See Figure 1.
3. The set hog format is as follows:

CALL

	RTJ-	(HOG)
+0	Return	

PARAMETERS

None

RETURN

Hog flags 1 and 2 are set. I/O or status call is guaranteed. Hog is cleared prior to executing the request, therefore, it must be set each time it is required. Interrupts to non MBS tests will be stacked until next call to MPX.

EXAMPLE

To ensure that the MNTRST call is performed in the following I/O cycle:

ENTER

CKST

WR

RECKST

HOG

MNTRST

RECKST

EXIT

Copy and check status.

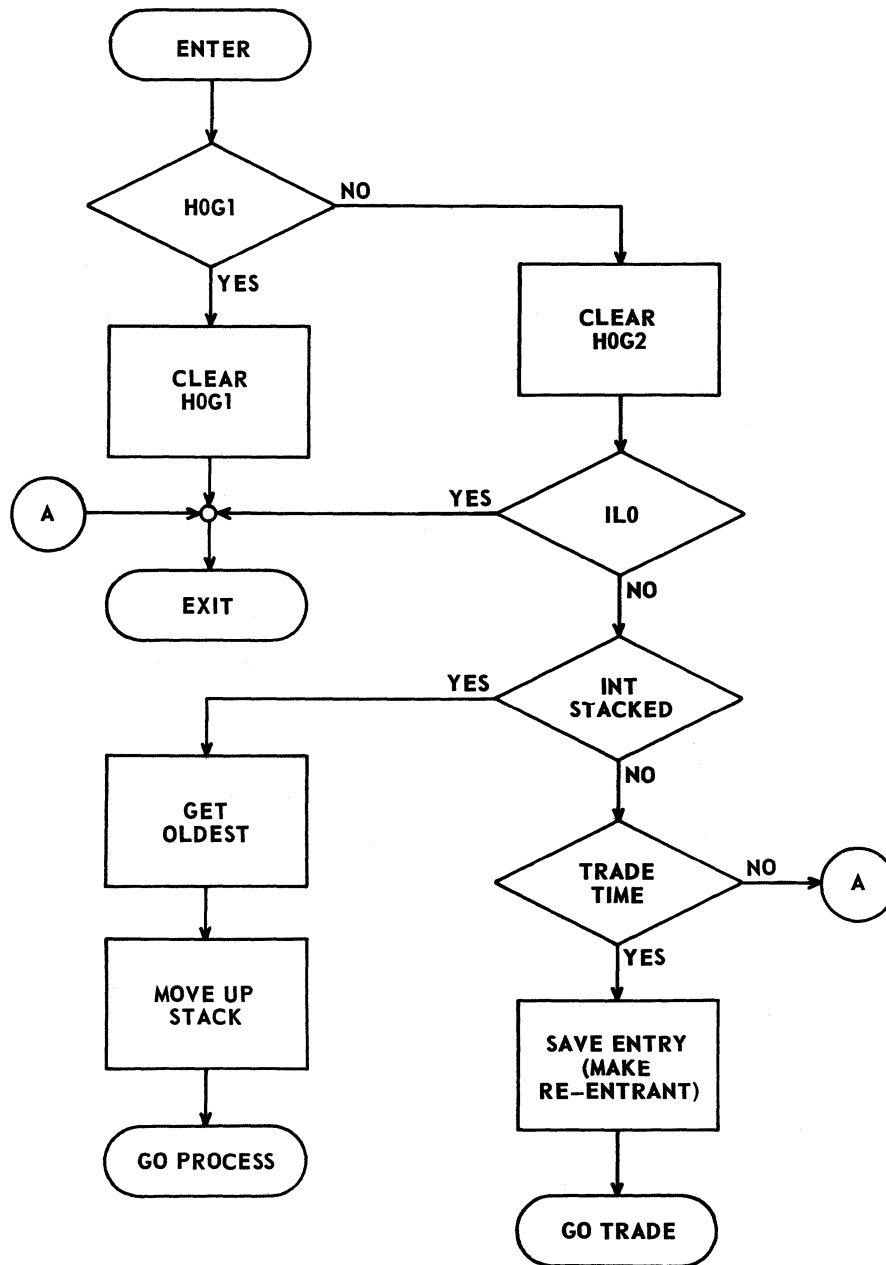
Issue write, copy status.

Check status after write.

Do not trade off control.

Copy and check status wait for specified status bit to change state.

Check last status copied when status bit changed state.



HOG1 Do not release control.
 HOG2 Stack non MBS interrupts until next MPX call.
 IL0 Interrupts in PROCESS mode (Interrupt Lock Out).
 TRADE Release control to monitor every other call.

Figure 1.

B. SET PROTECT BITS

1. The set protect bit routine is used to set memory protect in the callers test area. Care must be used to ensure protect bits are not altered in the monitor.

CALL

	RTJ-	(SETPP)
+0	ADC	fwa
+1	NUM	llll
+2		Return

PARAMETERS

+0	First word address of protect area
+1	Number of words to protect

RETURN

Protect bits are set in callers protect area

C. CLEAR PROTECT BITS

1. The clear protect bit routine is used to clear memory protect bits in the callers test area. Care must be used to ensure protect bits are not altered in the monitor.

CALL

	RTJ-	(CLRPP)
+0	ADC	fwa
+1	NUM	llll
+2		Return

PARAMETERS

+0	First word address of protected area
+1	Number of words to clear

RETURN

Protect bits are cleared in callers protected area.

V. UTILITY

A. DELAY

1. CALL

	RTJ	(FIXDLY)	
+0	NUM	tttt	TIMER
+1	Return		

PARAMETERS

+0	tttt	
	Time in milliseconds.	Range = 0 to 7FFF.
+1	Return after delay.	

B. RANDOM DELAY

1. Minimum and maximum values in milliseconds are given. The delay will range somewhere between these two values.

CALL

	RTJ-	(RDMDLY)	
+0	NUM	xxxx	
+1	NUM	yyyy	
+2	Return		

PARAMETERS

+0	xxxx	
	Minimum value in milliseconds.	
+1	yyyy	
	Maximum value in milliseconds.	
+2	Return after delay.	

SUPPLEMENT F SMM17 LIBRARY FORMAT

Library program data is formatted similar to that of MSOS 1700 Assembler output, RBD (relocatable binary data), except for the first record. This record (absolute binary Quick Look) consists of the actual core image of QL and the attached intermediate loader with the Hex-words broken down as is convenient for a simple bootstrap loading program for a particular device. For example, card, paper, and 9-track mag. tape use a 2x8-bit breakdown while 7-track mag. tape bit sequence is lower 6, upper 6, then middle 4 for each Hex-word.

RBD is blocked data with flags embedded in the block to signal biasing or relocation by SMM's RBD loader according to the load FWA (test IA). Thus the program under load becomes relocatable by modifying the necessary data words and addresses in the program to reflect the program's core position. Block length is one card image or less. Block type is identified by its first word:

NAMBLK (Program name block)	= \$2050
RBDBLK (Program data block)	= \$4050
BZSBLK (Zero data block)	= \$6050
XFRBLK (Transfer, end of program)	= \$C050
OVRBLK (Overlay segment marker)	= \$E050, unique to SMM17
HEXBLK (Program Modification data)	= \$2A48

(Other block types are considered illegal by the loader as an SMM program cannot specify entry points or externals, \$8050, \$A050 respectively. See MSOS Reference Manual.)

Block construction is as follows:

NAMBLK (derived from NAM card of program, and assembly information)

Word #1 = Block ID = \$2050

2 = Zero for SMM, common storage word count for MSOS	
3 = Data area (set=BZS area by pseudo-op to aid SMM DP loader)	
4 = Program length (LWA=1 of last program data word assembled)	
5 = Characters 1, 2	}
6 = Characters 3, 4	
7 = Characters 5, 6	
	Program name in ASCII code

Words 11-29 = Program revision, copyright dates (NAM card columns 25-62)

Word 30 = Blanks or 01 (ASCII) where 1=flag loader to ignore relocation on load
(NAM card columns 63-64)

RBDBLK

Word 1 = Block ID=\$4050

2 = R0, R1, R2, R3* = Relocation Flags for W0-W3

3 = W0 = Starting address of program data in this block

4 = W1

5 = W2

6 = W3

} Program data words

Word 52 = R40, R41, R42, R43 = Relocation Flags for W40-W43

53 = W40

54 = W41

55 = W42

56 = W43

} Program data words

BZSBLK

Word 1 = Block ID = \$6050

This block is ignored by SMM's RBD loader. The program load area is zeroed according to length in NAMBLK before reading data blocks (other block words are unimportant to SMM).

XFRBLK

Word 1 = ID = \$C050

Words 2 through 4 are unimportant to SMM.

OVRBLK

Word 1 = ID = \$E050

2 = 0VXX, where V=overlay segment number, XX=test ID number

3 = Overlay first word address

4 = Unimportant

*Only RX values of 0/1, = no bias/bias, are used in SMM. The upper bit of the Hex-digit is set to flag last program data word of block (for short blocks).

HEXBLK

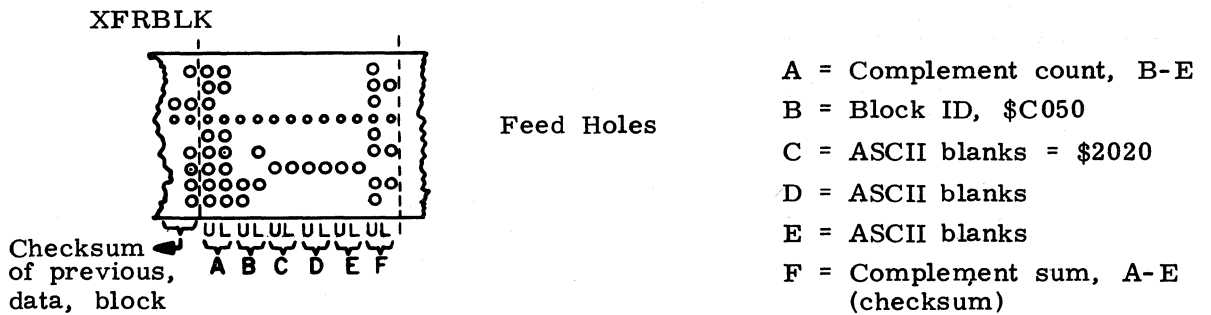
Same as RDBLCK except for word 1.

Word 1 = ID = \$2A48 (* H, ASCII)

Device Format Variance

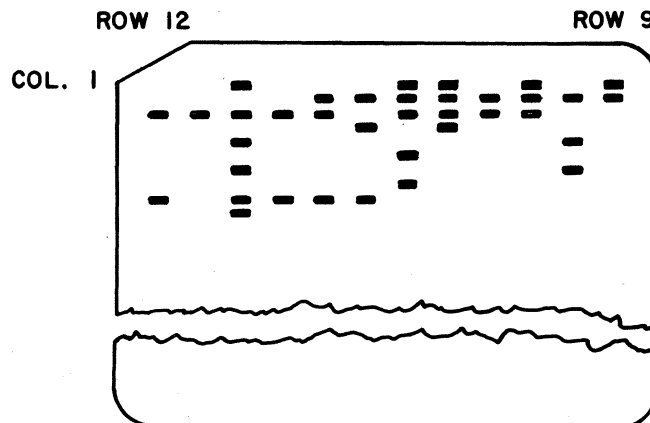
Paper tape RBD records include a prefix word and a suffix word. The prefix word is the complemented Hex-word length of the block, not including the prefix and suffix words. The suffix word is the entire block checksum word, including prefix and suffix.

Paper tape example:



Card image reads from row 12 to row 9, column 1 through column 80, to give Hex-words, upper through lower bits respectively (turn card upside down, column 1 to top). The first Hex-word, however, is broken up into an 8-bit card sequence number (upper bits), a 4-bit Hex-digit signifying binary punched card (7, 9 punched), and a last digit used by MSOS but not SMM. The second Hex-word is the complemented RBD block length and does not include these first two words, nor the last, card checksum word. The checksum word includes the entire card.

Card example:



1st two Hex-digits = Card Sequence = \$23

3rd Hex-digit = Binary (7/9) punch

4th Hex-digit = Zero

5th through 8th Hex-digits = Complement Block word count = \$FFFB

9th through 12th Hex-digits = Block ID = \$C050

13th through 16th Hex-digits = ASCII blanks = \$2020

17th through 20th Hex-digits = ASCII blanks

21st through 24th Hex-digits = ASCII blanks

25th through 28th Hex-digits = Complement sum of all previous Hex-words
(checksum)

OPERATING PROCEDURES

I. LOAD THE BOOTSTRAP

A. EXECUTING QUICK LOOK COMMAND TESTS

There is no difference in initializing SMM, calling of tests, and loading of tests from any of the card, magnetic tape, or paper tape input media. Additional system capability is provided to the operator if the input media is any of the disk units when loading the system monitor. If the TTY input driver is selected, each of its operations are directly equivalent to an operator's panel operation.

To load the two Quick Look tests and the SMM monitor, do the following procedures.

1. Master clear the system.
 - a. If running on a 1774, clear AUTOLOAD PROTECT if Q option for the fourth stop is not exercised.
2. Enter the respective bootstrap for the input media, or autoload the 1733, 1738, 1739 Disk or 8000 Magnetic Tapes. If using the autoload feature of any of these devices, press the AUTOLOAD switch and go to step 4b.
3. Enter the appropriate bootstrap. The suggested address for the bootstraps is P = XFE0. However, it will execute correctly at any address above OFE0. The bootstraps are in Supplement A. If loaded at XFE0, it will be saved during Quick Look Memory testing.
4. Master clear the system. Set P = bootstrap first word address and set the STOP switch; if system parameter changes are desired, set the SKIP switch. Place the RUN/STEP switch to RUN to execute the bootstrap. If the SKIP switch is not set, go to III.
 - a. The bootstrap will now load Quick Look 1. It will terminate when the first full word of zeros is sensed, at which time the bootstrap will jump to the beginning of Quick Look 1 and begin execution.
 - b. Quick Look 1 will terminate with the A register = 0031, the monitor ID word, and the Q register = 20D; the system Stop/Jump parameter. Any other stop is an error. Refer to Supplement B for a description of Quick Look execution and the P address meaning of the instruction that failed. If Quick Look encounters

an error, it will loop on the failing sequence until the error is corrected. To continue execution after an error, set the P register to the error stop displayed, P value +2 and run.

II. SETTING THE SYSTEM PARAMETERS

A. FIRST STOP

All displays after Quick Look 1 execution are in the format of the following first stop.

1. A1 = 0031 ID, the identification word

Q1 = 020D STJP, the Stop/Jump word

a. The ID word is defined as follows:

ID = TTNS,

Where: TT - Test number making this call

N = Number of stops (pairs of A and Q registers),
including this one to the completion of the call

S = Stop type - 1 = Parameter stop

2 = End of section stop

4 = End of test stop

8 = Error stop

b. Stop = the Stop/Jump parameter affecting only this test. However, the monitor Stop (the system Stop) is logically OR'ed with each of the test Stop/Jump parameters when making the decision if a stop should occur.

Stop/Jump Bit Assignment:

0 = Stop to enter parameters	Stop Type 1
1 = Stop at end of test section	Stop Type 2
2 = Stop at end of test	Stop Type 4
3 = Stop on error	Stop Type 8
<u>4</u> = Repeat conditions	Jump Type 1
5 = Repeat section	Jump Type 2
6 = Repeat test	Jump Type 4
<u>7</u> = Repeat Quick Look	
8 = Omit typeouts	
9 = Bias return address	
10 = Reenter test parameters	
<u>11</u> = Preexecution correction stop	
12 - 15 = Defined by individual tests as required at any ID stop.	

2. Make any changes as required to the Stop/Jump parameter word.
Clear the SKIP switch if no further parameter changes are desired;
otherwise, place the RUN/STOP switch to RUN.

B. SECOND STOP

If the SKIP switch is cleared, this and all remaining stops will be omitted.
If the SKIP switch is set, the CPU will proceed to the second stop, with the following display:

1. A2 = XXC0, system parameter
Q2 = XXX1, equipment address for loading tests and monitor overlays
 - a. A2 displays the system parameter bit assignments, defined as follows:
 - 0 = Typeout the library list from the selected loader and equipment address. This capability is available only at the end of Quick Look execution.
 - 1 = Do not load hex corrections.
 - 2,3 = Memory speed selection:
 - Bits 2,3 = 00 = 1704/14 speed - 1.1 microseconds
 - 01 = 1774 speed = 1.5 microseconds
 - 10 = 1784-1 speed = 900 nanoseconds
 - 11 = 1784-2 speed = 600 nanoseconds
 - 4 = Select 713-120 nonimpact printer line feed/carriage return delay.
 - 5 = Typeout in noninterrupt mode (status control).
 - 6 = Select teletype input mode. This package must be selected at the end of Quick Look execution to be loaded into core, but may be enabled or disabled at any time.
 - 7 = Build test list. If set equal to zero, the prestored test list will be executed.
 - 8 = Select the MBS package. This package must be selected at the end of Quick Look execution.
 - 9, 10, 11 = Unit number of the tape or disk drive containing the SMM17 4.0 library.

When a unit number other than the one selected in the bootstrap must be used for loading tests, it will be modified as follows:

- 1) Do not change the unit number (bits 9, 10, and 11) until after the Build Test List Stop has occurred.
- 2) At the Build Test List Stop, set the SELECTIVE SKIP switch, clear A register, and set the RUN/STEP switch to RUN before entering tests to be loaded. This will force another series of monitor stops.
- 3) At the next A2 stop, make the desired unit number change, clear the SELECTIVE SKIP switch, run, and continue through the Build Test List sequence.

12, 13, 14, 15 = Loader type from which the test will be loaded:

15-14-13-12

0001 = Paper Tape (1712, 1713, 1721, 1777)

0010 = Cards (1726, 1728, 1729-2, 1729-3)

0011 = Magnetic Tape (1731, 1732-608/609, 1732-2-6X8/6X9)

0100 = 85X Disk (1738, 1733-2)

0101 = CDD (1733-2)

0110 = Cartridge Disk (1739)

0111 = SMD Disk

1000 = 8000 Magnetic Tapes

The loaders are preselected to the bootstrap input type, but may be changed to any other type at the end of Quick Look only. The contents of the Q register must be changed to the correct Director Status 1 equipment address at this time also.

- b. Q2 displays the loader Director Status 1 equipment address.
2. Make any changes as required.
3. Clear the SKIP switch if no further parameter changes are desired.
4. Place the RUN/STEP switch in RUN.

C. THIRD STOP

1. A3 = 0000 Mask register deselect option
 Q3 = 8000 Line printer option equip address
 - a. The A register display is zero. Setting bits in the A register will permanently clear these bits from the M register, and thus disable the associated interrupt line(s). This is the only opportunity to change the mask register.
 - b. The Q register display is \$8000. If the line printer driver is to be used, enter the Director Status 1 address to be used by the line printer driver. This will cause the line printer driver to be loaded. Once loaded, this driver is enabled by placing the equipment address in Q3; it is disabled with a zero value in Q3.
2. Make desired changes, if any.
3. Clear the SELECTIVE SKIP switch if no further parameter changes are desired.
4. Place the RUN/STEP switch in RUN.

D. FOURTH STOP

If the SKIP switch is cleared, this stop will be omitted. If the SKIP switch is set, the CPU will proceed to a fourth stop.

1. A4 = FE01 Quick Look 2 ID word
 Q4 = FFFF CPU last memory address
 - a. The A register is the ID word for Quick Look 2's parameter stop.
 - b. The Q register contains the last memory address of the CPU. This value may be made smaller by entering the number of the last bank to be tested in bits 15, 14, 13, and 12 of the Q register. The rest of Q is set to FFF by the program; therefore, the minimum is 4K. This allows retention of hand-loaded programs and elimination of the requirement of turning off the AUTO LOAD PROTECT switch on a 1774. For a more permanent change, edit patch QL 2 address \$20 to the desired bank number.

The option is available to check the protect system of the CPU. Clear the STOP switch and SET the PROTECT switch at this stop.

Clear the SKIP switch and RUN; Quick Look 2 will now execute. See QL2 for a description of this test and error codes.

E. FIFTH STOP

If the PROTECT switch has not been SET, this stop will be omitted. If the PROTECT switch has been SET, the OVERFLOW light will begin flashing to indicate completion of Quick Look 2's execution. The operator should clear the PROTECT switch, SET the STOP switch, zero the Q register, and RESTART the CPU.

III. SYSTEM INITIAL MESSAGE

The following initial system message will be presented, either on the teletype or on the line printer:

```
SMM17 VER 4.0 CP2F
COPYRIGHT CONTROL DATA CORP 1974
```

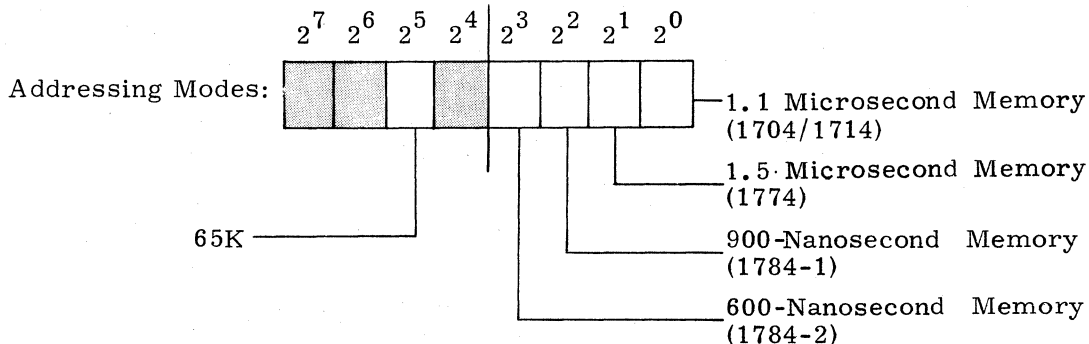
VER. 4.0 is the version of the monitor currently released.

CP2F is the formed number that allows the user to determine the type of configuration for which this module of the diagnostic package is valid.

Consider the number by bit positions, where each bit is:

- 0 = 1.1-microsecond memory speed
- 1 = 1.5-microsecond memory speed
- 2 = 900-nanosecond memory speed
- 3 = 600-nanosecond memory speed
- 4 = Not used
- 5 = 65K mode valid for use

A two-hexadecimal digit derived number accurately reflects the acceptable machine types and memory configurations. Built up, this number will be composed as follows:



Example: CP01 = 32K only, 1704 or 1714 only.

CP2D = 65K acceptable, 1704, 1714, 1784-1 or 1784-2, but not a 1774.

This two-digit number must appear in the typed out monitor heading, all test headings, and in the respective documentation in the reference manual.

If bit 7 of the SMM parameter word is set, the next message is:

BUILD TEST LIST

From this point on, all entries may be made by one of two ways.

1. From the TTY (refer to TTY description)
2. From the operator's panel

A. BUILD TEST LIST

1. Prestored Test List

The monitor may be initialized to contain a preset list of tests for execution in the event bit 7 of the SMM parameter word is not set. If this bit is not set, the monitor will automatically load all the tests in the prestored list and begin test execution.

2. Test Entry

a. A register format = TTRR

Where: TT = Test number
RR = Frequency count

The test number is the two-hexadecimal digit number assigned to each of the tests in the library. Examples are Command Test - 01 and Random Protect Test - 09. RR is the number of times the test will be executed. A minimum value of 01 is required.

- b. Q register format is the equipment address (Direct Function 1) of the peripheral to be tested. A zero entry uses the prestored value assembled into the library.
- c. The operator is to enter as many A/Q register pairs of information as tests to be loaded, each A register defining another test to be loaded and each Q register defining an equipment address to be used by that test. More than one copy of a test may be loaded, but only one test should be defined per peripheral.
- d. The test list can be terminated in one of two ways:
 1. List terminator. Setting the A register to \$0000 will terminate the test list and transfer control to the loader to load all the tests into core.

2. List stringer. Setting the A register to \$0001 will set a flag internal to SMM17 and continue adding tests to the test list until the operator uses the list terminator (\$0000) to finish the test list. Those tests defined prior to the list stringer will be loaded and executed to completion. Those tests defined after the list stringer will then be loaded and executed to completion.
- e. Tests can be loaded anywhere in memory by the following procedure.
1. Set the A register to \$FF00, test number \$FF, a frequency count of \$00. Set the Q register to the desired first-word address of the test.
 2. Set the RUN/STEP switch to RUN.
 3. Set the A and Q registers to the normal test list entry. This test will now be loaded at the specified address. Exceptions are the memory tests, which will relocate themselves adjacent to monitor prior to execution. Monitor error 9 will be reported if an invalid address is requested.
 4. Test execution. After termination of the test list, all tests will be loaded into memory. The monitor transfers control to each test one at a time for test initialization and heading typeouts; control is then given to each test for parameter entry. This program control is then multiplexed among all the list entries to allow complete execution. At the completion of each list entry, this entry will be deleted from the list. When the list is empty, signifying all test list entries have run to completion, the monitor reenters the Build Test List phase.

IV. TEST EXECUTION

A. CONTROLLING TEST EXECUTION

During test execution, the operator may change the controlling parameters (monitor's Stop/Jump word, SMM parameter word, or any individual test's Stop/Jump word) by setting the SKIP switch or pressing MANUAL INTERRUPT. An individual test's Stop/Jump word may also be changed by inputting new values after an error stop, end of section stop, or end of test stop initiated by the test.

1. Changing the Monitor's Parameters

The monitor parameters control all tests currently in execution. A logical OR is performed on the test's parameter value with the monitor's parameter value when making control decisions.

To achieve a change in the monitor's parameters, do the following:

- a. Set the SKIP switch or press MANUAL INTERRUPT.
- b. The first stop will be the identification stop:

A = \$0031 - ID word (monitor)
Q = Stop/Jump value (monitor)

Do not change the ID value. Change the monitor's stop/jump value in the Q register as required (see bit definitions). Place the RUN/STEP switch to RUN.

- c. Second Stop

A = SMM parameter word
Q = Load device director 1 address

Change the monitor's parameter value in the A register as required (see bit definitions); do not change the loader address.

- d. Third Stop

A = System interrupt mask value
Q = Line printer equipment code

Only the Q register may be changed in this stop. A zero value in the Q register will disable the line printer option, if previously enabled. A nonzero value (line printer equipment code) will enable this output option. The line printer driver option must be selected during the initial system loading sequence. Once loaded, it may be enabled and disabled with this parameter.

- e. Clear the SKIP switch if set. Place the RUN/STEP switch to RUN.
- f. The new parameters will be output to the comment device, unless omit typeout is currently selected, and test execution will be continued under control of the new parameters.

2. Changing an Individual Test's Stop/Jump Parameter

A stop/jump value may be defined uniquely for each test in the test list, in addition to the value for the monitor. To change a test's Stop/Jump parameter while testing is in progress, do the following:

- a. Set the SKIP switch or press MANUAL INTERRUPT.
- b. The first stop will be the identification stop:

A = \$0031 - ID word (monitor)
 Q = Stop/Jump value (monitor)

Change the ID value in the A register to identify the test that is to receive a new stop/jump value and place the new stop/jump value in the Q register.

A = \$XX31 - ID word, where XX is the test number
 Q = New stop/jump value for test XX

- c. Clear the SKIP switch, if set. Place the RUN/STEP switch to RUN.
- d. A second and third stop will occur to show the SMM parameter word and loader address; make no changes.
- e. Place the RUN/STEP switch to RUN.
- f. The new parameters will be output to the comment device, unless omit typeout is currently selected, and test execution will be continued under control of the new Stop/Jump word.

3. Changing an Individual Test's Stop/Jump Parameter while at a Stop

The individual test's Stop/Jump word can be changed at the completion of an error stop, end of section stop, or end of test stop. At the completion of any of these stops, one parameter stop is presented.

A = \$XXYZ - ID word, where XX is the test number, Y is the number of stops, and Z is the type of stop
 Q = Test's stop/jump value for test XX

Change the Q register to the desired stop/jump value and place the RUN/STEP switch to RUN. Processing will continue under control of the new parameter.

PRINTER/TELETYPE DUMP

(DMPA3B Test No. 3B)

I. INTRODUCTION

The Dump routine is a service routine for obtaining memory dumps on a line printer or a teletype. This routine is intended for use by customer engineers for dumping SMM test buffer areas during troubleshooting, and also to dump all of memory (if using a printer) when reporting diagnostic problems by DPSR. Snap dump and break-point dump options are also included for diagnostic debugging.

II. REQUIREMENTS

A. HARDWARE REQUIREMENTS

1. A 1700 series computer
2. One of the following:
 - a. 1742 Line Printer
 - b. 1740-501 Line Printer
 - c. 1742-20, -30, -120 Line Printer
 - d. 9322/9323 Line Printer
 - e. 1711/1712/1713 Teletype

B. SOFTWARE REQUIREMENTS

1. Dump loads under control of SMM17.
2. Dump does not take control from the monitor at any time. The operator must give the routine control by starting from its initial address (IA).
3. Dump is less than 200_{16} locations long.

III. OPERATIONAL PROCEDURE

A. LOADING PROCEDURE

The routine must be loaded under SMM17 as test number 3B. The selected equipment address determines which device type will be used. An equipment address of 0091 selects the teletype. Setting bit 15 of the equipment address selects a 1742-20, 1742-30, 1742-120, or 9322/9323 Line Printer. All other equipment addresses select a 1742/1740-501. After being loaded, this routine repeatedly executes test exit monitor calls until the frequency count (FC) equals zero and the routine is removed from the SMM test list. To now execute the dump program, it must be started at its initial address (IA).

B. PARAMETERS

1. Forcing a Parameter Stop

A parameter stop in Dump is not determined by the Stop/Jump word. A parameter stop will always occur when the routine is started at IA by the operator. There are two methods of forcing a parameter stop to start the routine.

- a. If the contents of registers are not important at the time of entering Dump:
 - 1) Master/Clear.
 - 2) Select P register and set P=IA of the routine.
 - 3) Run.
- b. If the contents of the registers are important at the time of entering Dump:
 - 1) Depress STEP key.
 - 2) Continue depressing STEP key until INSTRUCTION light comes on.
 - 3) Select and clear X register.
 - 4) Select and clear P register.
 - 5) Set P=IA of the routine.
 - 6) Run.

2. Selecting Parameters

When the routine is started at IA, the following parameter stops will occur.

A1 = 3B21 ID word

Q1 = 0000 or Snap address (Biased address, not listing)

A1/Q1 are used to select one of three dumping modes:

Panic dump, Snap dump, or Breakpoint dump.

All three modes also contain an A2/Q2 parameter stop (see next page).

Panic dump is selected when Q1 is set to 0000. In this dumping mode the dump is initiated immediately after the parameter stop. When the dump is complete, the parameter stop will again occur. Any dumping mode may then be selected.

Snap dump is selected when Q1 is set to a non-zero snap address and A1 is set to 0000. In this dumping mode a dump is initiated for each time the instruction at the snap address is executed. The dumps will be enabled until a new dump is selected by forcing a dump routine parameter stop. If an illegal snap address is selected, the A1/Q1 stop will be repeated with Q1=0000. Refer to Snap/Breakpoint dump "caution", Section V.

Breakpoint dump is selected when Q1 is set to a non-zero snap (breakpoint) address and A1 is not altered. In this dumping mode a dump is initiated for each time the instruction at the snap address is executed. After each dump, a dump routine parameter stop occurs. Any dumping mode may then be selected. If a panic dump is selected, A1 should be unaltered to enable another parameter stop after the dump is complete. If a snap or breakpoint dump is selected, a new snap address may be selected. A different area of memory to be dumped may be selected with any of the dumping modes. Once a Snap dump is selected, a Breakpoint dump can only be reselected by forcing a dump routine parameter stop. If an illegal Snap address is selected, the A1/Q1 stop will be repeated with Q1=0000. Refer to Snap/Breakpoint dump "caution", Section V.

A2 = First Word address of memory area to dump

Q2 = Last Word address of memory area to dump

If Q2 is set to 0000, no memory dump will occur. Only the contents of the registers A, Q, I, and M will be dumped.

C. SNAP/BREAKPOINT DUMP INFORMATION STOP

This stop occurs to signal the operator to start the test in which the Snap address resides.

A = Instruction at the Snap address

Q = Snap address

This stop will occur only if a Snap dump or a Breakpoint dump has been selected by forcing a dump routine parameter stop (running from the dump routine's IA). The contents of A and Q are not parameters, but rather for information only. When this stop occurs, the operator must start the test in which the Snap address resides. The common method to do this would be to restart that test at its initial address (IA).

IV. OPERATOR COMMUNICATION

A. NORMAL MESSAGES/STOPS

When the routine is loaded, the following will be typed:

DMPA3B PRINTER/TTY DUMP

Q1 = 0 OR SNAP ADR

A2/Q2 = FWA/LWA

IA = XXXX

B. ERROR MESSAGES/STOPS

If the selected Snap address contains an instruction considered illegal to this routine, the A1/Q1 parameter stop occurs again. The Snap address will have been cleared in Q1. No other indication of an illegal request will occur. Refer to Snap/Breakpoint dump "caution", Section V.

V. DESCRIPTION

A. GENERAL

Dump is a service routine which is used for obtaining memory dumps on a line printer or a teletype. The routine does not use any of the monitor routines after it has been loaded and initialized. This is to prevent changes from being made to memory areas that may be dumped.

B. PANIC DUMP

The Panic dump does an immediate dump of the operational registers (A, Q, I, and M) and a selected area of memory. The operational register contents are saved if the dump program is entered by stepping the computer to the start of an instruction cycle and then setting the P register to the dump program's IA. Master clearing the computer and setting P will, of course, clear the operational registers. Under this dumping mode P (Snap address) will always be dumped as 0000.

C. SNAP/BREAKPOINT DUMP

The Snap/Breakpoint dump does conditional dumps of the operational registers (P, A, Q, I, M) and a selected area of memory. The P register dump reflects the selected Snap address. The selected Snap address determines when the dumps will occur. During the parameter stop, the instruction at the Snap address is saved and a return jump to a Snap/Breakpoint dump processor is stored in its place. When the program being snapped reaches the Snap address, the return jump will give control to the Snap/Breakpoint dump processor rather than execute the instruction that was there previously. The Snap/Breakpoint dump processor first inhibits interrupts and saves the operational registers so that they can be restored before returning to the program being snapped. Then the dump of the registers and memory take place. If the selected dump is a Breakpoint dump, a dump routine parameter stop occurs. In either dumping mode, the registers are then restored and the instruction replaced by the return jump is executed. Interrupts are enabled just before exiting the Snap/Breakpoint processor. The exit is to the snapped program at its Snap address plus one. The Snap/Breakpoint dumps will continue until the program being snapped is halted or the return jump at the Snap address is no longer being reached. A

new dump may be selected by forcing a dump routine parameter stop. When a new Snap address is selected, the instruction for the previous Snap address will be restored if that snapped program has not been overlaid with a new program. (The previous Snap address is checked for a return jump to the Snap/Breakpoint dump processor.)

CAUTION

Care must be taken in selecting a Snap address.
The following restrictions must be considered:

1. During the snap/breakpoint dump processing, an interrupt cannot be pending nor can an interrupt be in process. A pending interrupt would be delayed until the Snap/Breakpoint dump was complete. The same would be true of an interrupt being processed.
2. During Snap/Breakpoint dump processing, any timing operation would become grossly inaccurate.
3. The instruction at the Snap address is limited by all of the restrictions as follows:
 - a. Two-word storage reference instructions are illegal (DELTA=0).
 - b. Relative addressing storage reference instructions are illegal (Bit 11=1).
 - c. Return Jump instructions are illegal (F=5).
 - d. Register reference instructions SLS, INP, OUT, EIN, IIN, and all Skip instructions are illegal (F1=0-5).

TAPE TO PRINT ROUTINE

(LST03C Test No. 3C)

INTRODUCTION

This routine accepts BCD magnetic tape information from a 1731 or 1732 Controller and lists it on a 1740/501 or a 1742 Printer.

OPERATIONAL PROCEDURE

A. HARDWARE REQUIREMENTS (Minimum)

1. One 17X4 Computer with 4K of memory.
2. One 1731/601 or one 1732/608-609 tape system.
3. One 1742 printer or one 1740/501 printer.

B. SOFTWARE REQUIREMENTS

1. LST routine must be loaded under control of SMM 1700 System Maintenance Monitor.
2. Upon completion of loading, the following typeouts occur:

\$3C *LIST UNIT 6*

IA= XXX, FC=XX

(A2) = TAPE EQ

(Q2) = PRINT EQUIP

(A3) = # FILES/SKIP

(number of files to skip before listing)

(Q3) = 0 = PROG CONT

listing)

(Q3) ≠ 0 *SINGLE SP*

3. Computer will stop after loading for parameter inputs if STOP switch is set or TTY input is selected.
4. If parameters are not changed, the following will be used.

A1 = 3C51

Q1 = STJP

A2 = 0381

Q2 = 0201

A3 = 0000

Q3 = 0000

A4 = 0000

Q4 = 0000

A5 = 0001

Q5 = FFFF

5. Tape unit 1 of tape equipment 7 will list one file on the printer. After printing one file, the test will check for repeat test. If repeat test is not set, the pass count is incremented. If the selected files have been printed, control will be returned to the monitor. If repeat test is selected, the parameters will be returned to the user for review.
6. Eight-digit part numbers with revision letter will be added to the listing if bit 15 is set in STJP. Part number and page numbers will be added at the bottom of each page. The page number is picked up from the top of the form for format control tapes.
7. If the first or next n number of files on the input tape are not desired, enter the number n in the (A3) parameter. This will cause the list tape to search forward n files before starting to print. If printing more than one file after a skip, be sure to reset the skip count to the desired amount before proceeding to the next file.
8. LST routine can be restarted by starting at IA.
9. All necessary operating instructions are contained in the typeout at load time.

3000 CHANNEL SIMULATOR ASSEMBLER
(SAS03D Test No. 3D)

I. OPERATIONAL PROCEDURE (see SMM Programming specification, Appendix D
for a more detailed procedure.)

A. RESTRICTIONS

1. Runs on a 1700 system with the following minimum configurations.
 - a. Tape units (2 each)
 - 1) Equipment #7
 - 2) Interrupt line #3
 - 3) Converter #0
 - 4) Source unit #0
 - 5) Object unit #1
 - b. Line printer on equipment F
 - c. Teletype
 - d. Core size of 8K
2. Object wrap-around after 256 instructions (W error).
3. Object is assembled in absolute format.
4. Object is intended to be written on 7 track tape.
5. A maximum of 256 symbols may be used.
6. A maximum of 56 remark cards may used (an * in column 1).
7. A maximum of 7 equate cards are allowed for each 40_8 instruction entered.
8. SAS overlays the SMM17 monitor.

B. LOADING PROCEDURE

1. Call as standard SMM17 test #3D.

C. PARAMETERS

1. SKIP switch set inhibits the writing of a file mark after the object.
2. If the SELECTIVE STOP is set, the program stops have the following meaning:
 - a. Stop 1. = Tape assignment with A =

3070 Altering A changes the assignment.

INT. line	—	┌	┌	—	Equipment No.
Conv. No.	—	└	└	—	

- b. Stop 2. A = 0. Set A to move the source tape A files forward before assembly.
- c. Stop 3. A = 0. Set A to move the object tape A files forward before assembly.

II. MESSAGES

A. NORMAL

1. SIMASSEM
3000 CH. SIMULATOR ASSEMBLER
Notes beginning of assembler.
2. Name =
Requests the programmers name. Up to 22 characters may be entered. 22 characters or a carriage return causes the program to continue.

B. ERROR CODES

1. S = Symbol assignment impossible (symbol value forced = 0).
2. D = Symbol in source deck 2 or more times (value forced = 0).
3. A = Address magnitude or assignment is impossible (value forced = 0).
4. O = Illegal Op code. If possible, the address field will be defined and used as if a "VAL" type instruction were given.
5. U = The symbol being referenced is undefined. The address is given the value of 0.
6. W = Simulator core wrap-around. Over 256 instructions have been generated for the object and actual object wrap-around is occurring.
7. I = Ident error. Either the Ident is not present or it is not the first (non REM) card present in the source deck.

III. PROGRAM DESCRIPTION

A. GENERAL

The programmer writes machine language in mnemonic and symbolic form, specific constants and pseudo instructions.

A program is generated between the IDENT pseudo instruction and END pseudo instruction.

Each line of code has the following 5 fields; all instructions are defined in terms of the contents of these fields in the appendix.

1. Location Columns 1-8 (9 is always blank).
2. Operation Begins in column 10 and proceeds through column 18.
3. Address Begins in column 20 and proceeds until first blank.
4. Comments Normally begins in column 41 but may begin after first blank in address field.
5. Sequence No. A unique set of alphanumeric characters used to aid in editing. No rules or restrictions are placed on these characters by the assembler. They must follow the rules set by the edit routine used for updating. Card columns 73-80.

C. CARD IMAGE RULES

1. Columns 1-8 (SYMBOL)
 - a. Up to 8 characters, beginning in column 1, are allowed. Do not include:
 - 1) The symbols +, -, or *.
 - 2) Imbedded spaces.
2. Column 1 = an * (REMARKS CARD)
 - a. These cards are saved from pass 1 for printing on pass 2. A maximum of 56 remark cards are allowed for a full 8 page assembly.
3. Column 20 (ADDRESS)
 - a. With the exception of BCD entries, the address field will terminate on the first blank after column 20. The following symbols may be used.
 - 1) + add this value to the preceding value.
 - 2) - subtract this value from the preceding value.
 - 3) * the value of this address.
** gives twice this address, not zero.
 - b. Any combination of values may be used so long as the computed end result does not exceed the following limitations.

- 1) 255 for addresses and enters.
- 2) 4095 for constants.
- 3) + or - 11 for shifts.
- 4) 63 for increase and decrease "A".

4. Column 41 (COMMENTS)

There are no restrictions as to the content of the comments field.

3000 CHANNEL SIMULATOR INSTRUCTION SET

<u>Instruction</u>	<u>Address</u>	<u>Remarks</u>
CAA	D*	Clear alarm light A.
CAB	D*	Clear alarm light B.
CLA	D*	Clear A.
CLSO	D*	Clear special operation.
CON	D*	Connect with code in A.
CONC	D	Connect continuous, code in A. See D values.
COPY	D*	Copy status to A/.
CPA	M	Skip if (A) = (M).
CPS	M	Skip if (A) Status = M.
CRL	D*	Clear read line.
CRP	D*	Clear rej./par. indications.
CWL	D*	Clear write line.
EINT	D*	Enable interrupt.
ENA	Y	Enter A with Y. Y = 0-256.
ICA	D	Input continuous to A. See D values.
INA+	Y	Increase A +1 Y times. Y = 0-63.
INA-	Y	Increase A -1 Y times. Y = 0-63.
INAW	D*	Input to A.
INCL	D*	Clear interrupt.
INPW	M, N	Input N words to M M+N.
IOCL	D*	Clear I/O.
LAA	D*	Light alert lamp A.
LAB	D*	Light alert lamp B.
LDA	M	Load A from M.
NEG	D*	Negate BCD conversion.
NOP	D*	No operation.

M = Memory location (000 377₈) N = word count to A (000 377₈).

Y = Constant (value varies with instruction using it).

D* = 00 17₈ which will be placed in the instruction but is not used by this instruction.

<u>Instruction</u>	<u>Address</u>	<u>Remarks</u>
OCA	D	Output continuous from A/.
OTAW	D*	Output word from A.
OUTW	M, N	Output from M M+N, N words.
PEJ	M	Jump to M if a parity error.
RTJ	M	Jump to M with jump to P=1 in A/.
SEL	D*	Select with function in A.
SELC	D	Select continuous with function in A. (see D values).
SET	D*	Set A to all 1's.
SHA	+Y	Shift (A) left 1 enough times to accomplish +Y left end around or -Y right end around. Y = 1 11.
SJA	M	Jump to M if JUMP switch A is set.
SJB	M	Jump to M if JUMP switch B is set.
SOF	D*	Turn sound alarm off.
SON	D*	Turn sound alarm on.
SRL	D*	Set read line.
STA	M	Store A at M.
SUP	D*	Suppress assembly/disassembly.
SWL	D*	Set write line.
TCA	D*	A A
TDA	D*	Transfer data switches to A.
TEA	D*	Transfer Equipment switch to A upper.
TPA	D*	Transfer P+1+jump to A.
UCS	D*	Unconditional stop.
UJP	M	Unconditional jump to M.
WMO	D*	Select Word Mark Operation.

Assembler Pseudo Op. codes

BCD	N, CCCC...	Enter N locations with BCD characters C. N = 1 29.
BSS	N	Clear and save N locations. N = 1 256.
END		End of assembly.
EQU	X	Equate the symbol (column 1 8) to X.

<u>Instruction</u>	<u>Address</u>	<u>Remarks</u>
IDENT	NAME	Begin assembly card. Names program (up to 8 characters). Columns 41 through 59 (19 characters for data and version) are also placed with the program name. Columns 73 through 80 (card sequence no.) are placed with the name for printing with the first header only.
PURG	M	Force the program address counter to M. This instruction must force the address forward by at least one location from the current address.
VAL	A, B, C,	For as many values as are present, enter that value into sequential locations.
(* IN COL. 1)		The entire content of this card becomes a remarks card and is placed in the remarks block at the top of the coding from under the following rules: <ul style="list-style-type: none"> 1. An absolute maximum of 56 cards are allowed. 2. Regardless of their position in the source: <ul style="list-style-type: none"> a. They are printed at a rate of 7 per page (starting on page 1) until they are all printed or until 56 are printed or the last page is printed where there are less than 8 pages in the assembly.

D VALUES*

0 = END OF RECORD	8 = STATUS bit 7
1 = STATUS bit 0	9 = STATUS bit 8
2 = STATUS bit 1	10 = STATUS bit 9
3 = STATUS bit 2	11 = STATUS bit 10
4 = STATUS bit 3	12 = STATUS bit 11
5 = STATUS bit 4	13 = PARITY ERROR
6 = STATUS bit 5	14 = REJECT
7 = STATUS bit 6	15 = CONTINUE SW. ONLY

* Used to exit from continuous codes.

SMM17 EDIT ROUTINE
(EDTA3E Test No. 3E)

I. INTRODUCTION

The edit routine is a service program for tailoring an SMM17 library, version 4.0 or later. The basic function of EDT is to copy an SMM17 version 4.0 library from an input device to an output device and to verify that the generated copy is error-free. The positions of tests on the input library should be known when building a new library. If the contents of the input library are unknown, a library list may be selected at system load time by setting a bit 0 of the SMM parameter. Only target tests can be modified when writing them to the new library. All other copied tests are copied as they are. Target test modifications that may be selected include prestored equipment address, prestored parameters, and program patches. A copy to a dummy output media affects skipping of unwanted tests on the input library. Special functions, such as rewind and write file mark, may also be selected. All of the above operations are defined by the operator in a function table. The same function table that defines the copy also defines the verify.

II. RESTRICTIONS/NOTES

- A. EDTA3E will only run with the 4.0 Quick Look and Monitor.
- B. Tape libraries must have file marks on the end of each test unless a special input unit is selected (see Parameter Description). This speeds search for test during load. EDIT does not write this file mark after a program patch operation (bit 11 set in A in the function word). To meet this requirement after a program patch to a test without overlays or the last overlay of a test, the next operation in the function table must be \$X05, where X is the logical unit number of the output tape drive.
- C. The following test sequence must be maintained during library copies:
 - 1. QL1AFF
 - 2. QL2AFE
 - 3. DPC AFC
 - 4. SMM000
 - 5. Remaining library in order of test numbers

- D. Repeating Edit copy/verify has proven to be a good systems test. The following should be done to allow continuous Edit repetition without operator intervention:
1. Enter L.U. table and function table.
 2. Run one pass through the function table.
 3. Set repeat test, disable end of section, end of test, and reenter parameter bits.
 4. Set ST/JP bit 11 to disable the verify attention stop.
 5. Set ST/JP bit 14 to disable the build function table stop.
 6. Run. Edit will now repeat without intervention until an error occurs. (The verify attention ST/JP will still be active for paper tapes and cards since these devices have to be reloaded.)
- E. The actual memory size requirement of Edit is dependent upon the following selectable options:
- Verify (If not selected during Edit load, the space will be used by I/O overlays.)
 - Size of function table
 - Number and type of devices used (Edit's I/O drivers are relocatable overlays and will be attached to the end of Edit after L.U. table selection.)
- F. During Edit's Build L.U. table (parameter entry), the library unit where Edit is loaded from must remain ready until all parameters have been entered. Edit will load its overlays from the library after the parameter entry is finished. Once the build function table typeout appears, the library unit can be used for Edit's I/O. If Reenter parameter is selected, the library unit must be restored for overlay load if any device types are added or deleted in the parameter table.
- G. To reenter parameters, set bit 10 in the ST/JP word upon end of section, test, or error stop. Clear bit 10 upon parameter ID stop. After parameters are changed, ensure library unit is ready for Edit's overlay load in case device types have changed.

III. REQUIREMENTS

A. Hardware

1. A 1700 series computer.
2. One of the following input devices:
 - a. Paper Tape Reader
 - 1721
 - 1722
 - 1777
 - b. Card Reader
 - 1726/405
 - 1728/430
 - 1729-2
 - 1729-3
 - c. Magnetic Tape
 - 1731/601
 - 1732/608-609
 - 1732-2
 - 1732-3
 - ADSE 8000
 - d. Rotating Mass Storage (If one drive is configured on a controller, that same drive cannot be used as both the input and the output device in a function table entry.)
 - FA706/85X
 - 1738/853-854
 - 1739
 - 1733-1/85X
 - 1733-2/856-2, 856-4, 856-12, 856-14
 - 1733-3/CU33/858

3. One of the following output devices:

a. Paper Tape Punch

1723
1724
1777

b. Card Punch

1728/430
1725-1

c. Magnetic Tape

1731/601
1732/608-609
1732-2
1732-3
ADSE 8000

d. Rotating Mass Storage (If one drive is configured on a controller, that same drive cannot be used as both the input and the output device in a function table entry.)

FA706/85X
1738/853-854
1739
1733-1/85X
1733-2/856-2, 856-4, 856-12, 856-14
1733-3/CU33/858

4. The memory requirement is 8K. 12K of memory is required for generation of Binary Quick Look.

B. SOFTWARE

The program operates under control of the SMM17 Version 4.0 monitor.

C. ACCESSORIES

An SMM17 Version 4.0 library in relocatable binary (RBD) format is required for input for copy and/or verify operations.

IV. OPERATIONAL PROCEDURE

A. LOADING PROCEDURE

The program must be loaded under SMM17 as test number 3E. The equipment address placed in the Q register during build test list must be the equipment address for status 1 of the device which will be used to read the new library during the verify portion of the edit. This equipment address is different than the equipment address used by the write device in the case of paper tape or cards. If no verify is desired, the equipment address must be 0000. The verify equipment address may be changed during execution at the time of the verify attention stop.

B. PARAMETER STOP*

First stop (overflow light on).

(A1) = 3E81 - Test ID word

(Q1) = Stop/Jump parameter

Bit 0 - Stop to define logical units

1 - Stop after each function table word pair

2 - Stop after edit complete

3 - Stop on error

4 - Not used

5 - Repeat current function table word pair

6 - Repeat edit

7 - Not used

8 - Omit typeouts

9 - Bias return address display

10 - Stop to redefine logical units after end of test or error stop

11 - Disable verify attention stop (tapes, disks)

12 - Load Edit's I/O routine overlays to even 100 address displacement

13 - Do not load old target test patches (forces end of section stop)

14 - Suppress build function table stop for repeat pass

15 - Verify only - do not copy

NOTE

Initial start of Edit will check bit 0 for enter logical unit table stop.

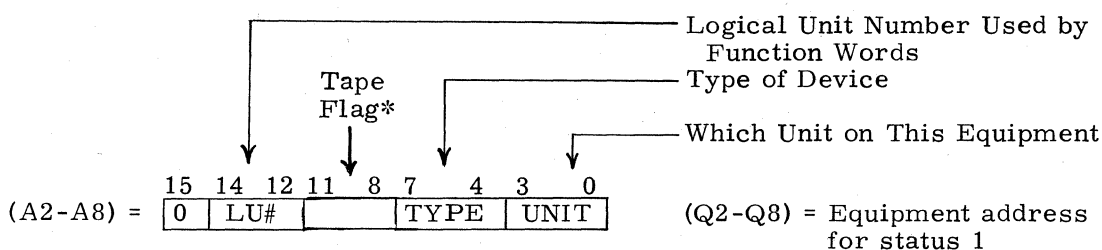
End of test, error stop, and INITIAL ADDRESS RESTART will check bit 10 for reentering the logical unit table.

*Occurs only if bit 0 or bit 10 of the Stop/Jump parameter is set.

Stops 2-8 (logical unit definition)

Stops 2 through 8 enable the operator to define logical units (LU) to be used in function table entries. The logical unit concept used here is similar to 1700 MSOS use. Logical unit definition allows a device to be referred to by a 3-bit number rather than by equipment type, equipment address, and unit number. The parameter stops 2 through 8 define logical units 1-7 respectively. The logical unit entry is terminated by either a zero entry in A or after entering the data for logical unit 7. The logical unit definition word pairs use the format as follows:

LOGICAL UNIT DEFINITION WORD PAIR



Type of Device (same codes as SMM loader types)

- 0001 = Paper Tape
- 0010 = Punched Cards
- 0011 = Magnetic Tape
- 0100 = Rotating Mass Storage (1738 or 1733-1)
- 0101 = Rotating Mass Storage (1733-2)
- 0110 = Rotating Mass Storage (1739)
- 0111 = Rotating Mass Storage (1733-3/858)
- 1000 = 8000 Magnetic Tape

Example 1

Consider a system with a configuration as follows:

Equipment Type	Equipment Number	Unit Number	Channel
1738	3	1	A/Q
1732/608	7	2	1706 #1
1732/609	7	3	1706 #1
1728/430	C	None	A/Q
1721	1	None	A/Q
1723	1	None	A/Q

*Zero if 4.0 tape library with file marks (device type 3 only) and nonzero if library without filemarks.

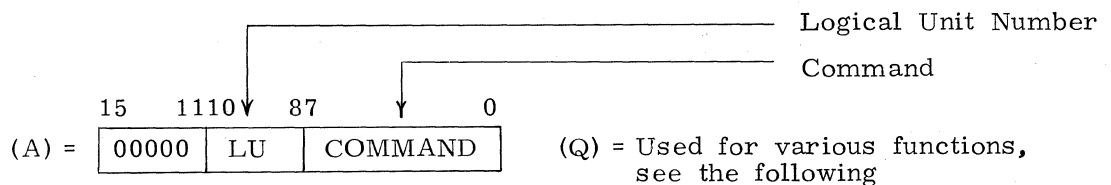
A reasonable logical unit assignment in the parameter stop would be as follows:

A1 = 3E81	Q1 = Stop/Jump parameter	
A2 = 1041	Q2 = 0181	LU#1 1738/853
A3 = 2032	Q3 = 1381	LU#2 1732/608
A4 = 3033	Q4 = 1381	LU#3 1732/609
A5 = 4020	Q5 = 0641	LU#4 1728/430 Punch
A6 = 5020	Q6 = 0621	LU#5 1728/430 Reader
A7 = 6010	Q7 = 00C1	LU#6 1723
A8 = 7010	Q8 = 00A1	LU#7 1721

C. BUILD FUNCTION TABLE

After the last parameter stop "build function table" will be typed and the 1700 will stop for function word entries into the A and Q registers. The operator must enter the function table with some combination of the three types of function word pairs (A and Q registers) to build the desired new library. The three types of function word pairs are as follows: Special function, Skip function, and Copy function. The function table must begin with Special functions to rewind all the magnetic tapes and disks used in the edit. Following the rewind functions, some combination of Copy functions and Skip functions normally occurs. If the new library is to be on magnetic tape or disk, the last function in the table must be a Special function to write a file mark. The function table is terminated by placing 00XX* in both A and Q of a function word pair. A description of the three formats of function word pairs follows:

1. Special Function



The command in bits 0-7 will be executed on the logical unit in bits 8-10. Legal commands (by type of device) are as follows:

Paper Tape (Type 1)

0000 = Feed tape (between target tests)

*xx = not applicable bits.

Magnetic Tape (Type 3)

- 0101 = Write end of file (see Restrictions/Notes, B)
- 0110 = Search file forward (not on 1731)
- 0111 = Search file reverse (not on 1731)
- 1000 = Rewind to load point (start of copy)
- 1100 = Rewind unload
- 1111 = Output binary Quick Look for 8000 tapes (must be done prior to QL-transfer if needed)

Rotating Mass Storage (Types 4, 5, 6, and 7)

- 0000 = Return to zero seek (start of copy)
- 0001 = Write software end of file (done at end of library)
- 0010 = Write address headers on track specified by Q
- 0011 = Write address header from track specified by Q to end of pack
- 0100 = Format write on cylinder in Q (7 only)
- 0101 = Format write on cylinder in Q to end of pack (7 only)
- 0110 = Seek to cylinder in Q (7 only)

NOTE

Functions 2 and 3 restore drive to Sector 0 upon completion. An illegal sector address in Q for functions 2 and 3 will terminate the function.

8000 Magnetic Tapes (Type 8)

Same as device 3 (1732).

- 1111 = Output binary Quick Look in 1732/608/1731/601
Format to be loaded with special bootstrap.

NOTE

If a special function is to be done to generate an 8000 library on a 1731-601 or 1732-608/609 or vice versa, this function must be done prior to Quick Look transfer.

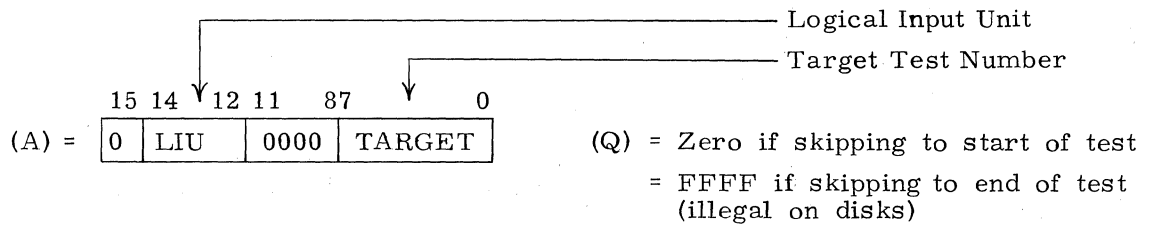
The only difference between 8000- and other 7-track libraries is the format of the binary Quick Look record.

Example 2

Problem: Given the system configuration and logical unit assignments of example 1, rewind both magnetic tapes and disk pack.

<u>Solution:</u>	<u>A</u>	<u>Q</u>	<u>Description</u>
	0208	0000	Rewind the 608
	0308	0000	Rewind the 609
	0100	0000	Return to zero seek on 853

2. Skip Function



The library on the logical input unit in bits 12-14 will be searched to the start or end of the test in bits 0-7. This function must be followed by a copy function from the same input logical unit, if the skip was to the start of the target test. The skip to the end of the target test can be used to add another test to the input library.

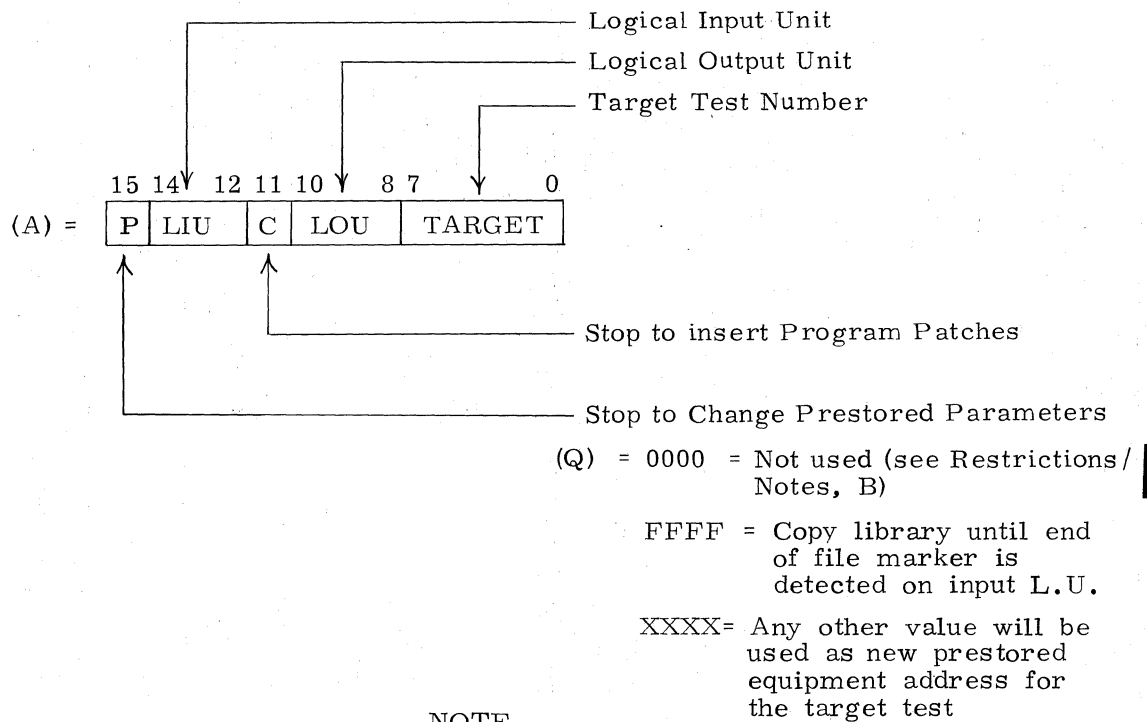
Example 3

Problem: Given the system configuration and logical unit assignments of example 1, search a library on the 608 to test number 15.

Solution:

<u>A</u>	<u>Q</u>	<u>Description</u>
0208	0000	Rewind the 608
2015	0000	Search to test 15

3. Copy Function



NOTE

On copy file operating (Q=FFFF), the target test number has no meaning unless patches or parameter changes are selected. Patches and/or parameter changes will be executed during copy file if the target test number is contained on the input library.

The library on the logical input unit in bits 12-14 will be copied from its current position through the end of the test in bits 0-7 onto the logical output unit in bits 8-10 or to the end of file marker (library end) if Q is set to FFFF.

Example 4

Problem: Given the system configuration and logical unit assignments of example 1, build a disk pack library with all the tests for the configuration in example 1 from a 7-track magnetic tape library containing Quick Look, SMM, and all SMM tests in numerical order.

Solution:	<u>A</u>	<u>Q</u>	<u>Description</u>
	0208	0000	Rewind the 608
	0100	0000	Return to zero seek on 853
	2105	0000	Copy QL, SMM and test 01-05
	2008	0000	Search 608 to test 08
	210A	0000	Copy test 08-0A
	200D	0000	Search to test 0D
	210D	0000	Copy test 0D
	2015	0000	Search to test 15
	2115	0000	Copy test 15
	0101	0000	Write software end of file
	0000	0000	Terminate function table

Example 5

Problem: Configuration of example 1, copy a library to a 1732-608 to be used on an 8000 tape. Input is from 853 Disk.

Solution:	<u>A</u>	<u>Q</u>	<u>Description</u>
	0208	-	Rewind 608
	0100	-	Rewind 853
	020F	-	Switch to 8000 format
	12FF	0460	Transfer Quick Look *
	12FD	FFFF	Transfer rest of library until EOF on disk
	0205	-	Write EOF on tape
	0000	-	Terminate table.

* 0460 (contents of Q) will be prestored as Autoload - Equipment - Address into Quick Look.

a. **Prestored Equipment Address**

If the Q register of a function word pair is not equal to zero or FFFF, the contents of Q will be written onto the logical output unit as the prestored equipment address at the target test's initial address plus six (IA+6).

NOTE

If Quick Look is the target test, the equipment address will be used for autoload devices.

Example 6

Problem: Given the system configuration and logical unit assignments of example 1, punch a paper tape with Quick Look, SMM, and tests 08, 0D, and 15 from a disk library containing Quick Look, SMM, and tests 01, 02, 03, 04, 05, 08, 0D, and 15. Prestore the equipment number into tests 08, 0D, and 15.

<u>Solution:</u>	<u>A</u>	<u>Q</u>	<u>Description</u>
	1600	0000	Punch Quick Look and SMM
	1008	0000	Search to test 08
	1608	0181	Punch test 08
	160D	0601	Punch test 0D
	1615	1381	Punch test 15
	0000	0000	Terminate function table

b. **Stop to Change Prestored Parameters**

If bit 15 of A in a function word is a one, a simulated parameter stop for the test in bits 0-7 will occur when the logical input unit reaches the relocatable binary data block (RBD) in which that test's parameters reside. The parameters selected in the parameter stop will be written onto the logical output unit as the target test's new prestored parameters. In the simulated test parameter stop, the Stop/Jump parameter is not displayed; it is replaced with the new prestored equipment address. The equipment address cannot be changed at this time. SMM prestored parameters are a special case and, therefore, must be handled differently. To change the prestored SMM parameters, the target test must be Quick Look (test number FF). In the simulated SMM parameter stop, Q1 is not used; however, the Stop/Jump parameter is displayed in A2 and may be altered. If SMM (test number 00) is the target test in a function word that requests a stop to define parameters, the operator is allowed to define SMM's prestored test list including the optional entries of equipment address, test load address, and Stop/Jump parameter. In the first

series of stops, the operator must enter the test and frequency into A and may enter the equipment address into Q. A maximum of 10 pairs of entries may be made. The list must be terminated by placing zeros in both A and Q. In the second series of stops, the operator may enter the test load address in A and the Stop/Jump parameter in Q.

Example 7

Problem: Given the system configuration and logical unit assignments of example 1, punch a card deck with Quick Look, SMM, and test 08 from a disk library containing Quick Look, SMM, and tests 01, 02, 03, 04, 05, 08, 0D, and 15. In test 08, prestore the equipment address and change the prestored parameters to run unit 1 instead of unit 0.

Solution:	<u>A</u>	<u>Q</u>	<u>Description</u>
Function			
Table	1400	0000	Punch Quick Look and SMM
Word	1008	0000	Search to test 08
Pairs	9408	0181	Punch test 08
	0000	0000	Terminate function table

After reading the RBD block in which the parameter for test 08 resides, a simulated test 08 parameter stop will occur as follows:

	<u>A</u>	<u>Q</u>	<u>Description</u>
Simulated	0831	0181	ID word/equipment address
Parameter	119F	0063	Sections-unit/cylinders
Stops	0004	0000	Interrupt line/not used

The operator would change A of the second stop from 119F to 319F. This would select unit 1 to be run instead of unit 0.

Special case (target test Quick Look FF) prestored parameter stop is as follows:

Special	<u>A</u>	<u>Q</u>	<u>Description</u>
Case	0031	0000	ID word/not used
Parameter	020D	00C0	STJP/SMM parameter
Stops	FFFF	0201	Mask register deselect option/line printer option equipment address

c. Stop to Insert Program Patches

If bit 11 of the function word in A is a one, stops will occur when the logical input unit reaches the test in bits 0-7 to allow program patches to be entered. These program patch stops will occur at two different positions within the target test. The first stop will occur after the

NAM block of the target test has been read. At this stop, A will contain the current function word and Q will be zero. The operator must enter into Q the number of words to be added to the length of the test. If all patches are to be made to code already within the test's boundaries, Q should remain zero. After the length change stop, the target test will be copied to the logical output unit until the last block of the target test (or overlay) has been read. Additional stops are then made to allow the actual program patches to be entered. The first such stop will occur with A containing the current function word and Q containing zero (or the number of the overlay being changed). The operator must enter the listing address of the program patch into A and the data into Q. If the data in Q is an address that must be biased at load time, bit 15 of A must be set to a one. The program patch stop must be repeated for each patch. When all patches have been made to the test (or overlay), the operator must zero A and Q to terminate patching of the test (or overlay). If the target test contains overlays, an additional stop will occur after A and Q have been cleared. This additional stop allows the operator to bypass all the remaining overlays. When the stop occurs, A contains the current function and Q contains the number of the next overlay to be changed. If the operator does not alter A or Q, the program patch stops (excluding the length change stop) will be repeated for next overlay. If the operator zeros the A register, program patch stops will be bypassed for the remaining overlays.

Example 8

Problem: Given the system configuration and logical unit assignments of example 1, copy to a 9-track magnetic tape library all the tests from a disk library containing Quick Look, SMM, and tests 01, 02, 03, 04, 05, 08, 0D, and 15. In test 08, patch listing address 05DC to contain an NOP instruction.

Solution:	<u>A</u>	<u>Q</u>	<u>Description</u>
Function			
Table	0308	0000	Rewind the 609
Word	1B08	0000	Copy to 08, patch 08
	0305	0000	Write EOF
Pairs	1315	0000	Copy remainder of disk including test 15
	0305	0000	Write end of file
	0000	0000	Terminate function table

When the first record (name block) of test 08 has been read, a length change stop will occur as follows:

	<u>A</u>	<u>Q</u>	<u>Description</u>
Length			
Change	1B08	0000	Active function/0000
Stop			

Since the program patch would not change the length of test 08, the operator would leave Q unaltered and place the RUN/STEP switch into RUN.

When the last record (transfer block) of test 08 has been read, a program patch stop will occur as follows:

	<u>A</u>	<u>Q</u>	<u>Description</u>
Program	1B08	0000	Active function/0000
Patch			
Stop			

To enter the program patch the operator would enter 05DC into the A register and 0B00 (an NOP instruction) into the Q register. On the next stop, the operator would clear both A and Q to terminate the program patch stops. Next item in function table must be a write EOF (see Restrictions/Notes, B).

Example 9

Problem: Given the system of example 1, build a disk pack library including test 15, and insert a program patch to SMM's overlay number 2 (the monitor's final card reader loader, patch location 0713 to 0080).

Solution:	<u>A</u>	<u>Q</u>	<u>Function Table Description</u>
	0208	-	Rewind LU2 (608)
	0100	-	Rewind LU1 (853)
	0103	0000	Write address tags on 853
	2900	-	Copy to SMM and patch SMM
	2115	1381	Copy to test 15, insert EQ. address into test 15
	0101	-	Write software EOF on 853
	0000	-	End of function table, start Verify after first pass

After reading SMM's NAM-block, Edit stops with:

<u>A</u>	<u>Q</u>	<u>Description of Action</u>
2900	0000	No length increase, run
2900	0000	No change to basic monitor, run

After reading the first overlay marker, edit stops with:

	<u>A</u>	<u>Q</u>	<u>Description of Action</u>
	2900	0001	Start overlay 1, no length increase, run
	2900	0001	Overlay 1 end, no patch to this overlay, clear A and Q registers, run
	2900	0002	Start of overlay 2, no length increase, run
	2900	0002	End of overlay 2. Insert
Operator Input	0713	0080	patches, now A = address
	0714	0000	Q = new data. Clear A/Q
Operator Input	0000	0000	if patches are finished.
	2900	0003	Start overlay 3. No changes
Operator Input	0000	0000	to this or the following overlays. Clear A/Q and run.

Edit will now copy the other overlays and continue executing the function table.

NOTES

Overlays are handled as separate target tests during test patch insertion.

Overlay marker blocks must have been inserted by Edit on a previous Edit run (that is, program patch does not work on an MSOS-RBD of a test having overlays).

Since overlay 2 is not the last overlay if output device type was 3, an EOF would not be the next entry unless SMM was the last test to be written on the tape.

d. Verify

Verify can only be enabled during initial Edit load (see III.A.). A verify must only be enabled when a library is to be built on a single logical unit. This is because there are provisions for only one verify device equipment address (entered into Q during build test list). After the copy portion of the edit has completed, a message will be typed as follows: WAITING TO START VERIFY. The program will stop with 3E1F in A and the Stop/Jump parameter in Q. The Stop/

Jump parameter may be changed at this time.* At this stop, the operator must make all paper tapes and card decks ready for the verify. All magnetic tapes and disks will be readied by the program through the rewind special functions at the start of the function table. These special functions must have been entered at build function table time. When all devices have been readied, the operator must place the RUN/STEP switch to RUN. If the TTY-input package is selected, do carriage return. The program then again performs the operations designated in the function table. The only difference between this pass and the copy pass through the function table is that every Write operation is replaced by a read from the verify device and a compare. All prestored parameter entries and program patch entries must be repeated during the verify. Setting bit 15 of the Stop/Jump parameter at the start of an edit causes the copy portion of the edit to be eliminated and only the verify portion is executed.

Example 10

Problem: Given the system configuration and logical unit assignments of example 1, build a disk pack library with all the tests for the configuration in example 1 from a 7-track magnetic tape library containing Quick Look, SMM, and all SMM tests in numerical order. Verify the disk pack library.

Solution:	<u>A</u>	<u>Q</u>	<u>Description</u>
	0208	0000	Rewind the 608
	0100	0000	Return to zero seek on 853
	2105	0000	Copy/verify QL, SMM, and test 01-05
	2008	0000	Search 608 to test 08
	210A	0000	Copy/verify test 08-0A
	200D	0000	Search 608 to test 0D
	210D	0000	Copy/verify test 0D
	2015	0000	Search 608 to test 15
	2115	0000	Copy/verify test 15
	0101	0000	Write software EOF (on copy only)
	0000	0000	Terminate function table

* Bit 15 in Q may be cleared to bypass verify at this time. The A register may be changed to the WESD field to be used for the verify controller, if not inserted correctly during Edit's initial load. The A register may be zeroed out to bypass the verify.

The previous function table would be executed once for a copy and the START VERIFY message would be typed. The computer would stop with 3E1F in A and the Stop/Jump parameter in Q. The operator would place the RUN/STEP switch to RUN. The previous function table would again be executed, but for a verify on this second pass.

E. SEQUENCE OF EDIT EVENTS

Operator action during edit will be required in the sequence as follows:

1. Parameter Entry/Load of Edit's Overlays
2. Build Function Table
3. Program Length Stop (if stop to enter program patches has been selected in the function word)
4. Prestored Parameter Stop (if stop to enter prestored parameters has been selected in the function word)
5. Program Patch Stop (if stop to enter program patches has been selected in the function word)
6. End of Function Word Stop (if stop at end of section has been selected in the Stop/Jump parameter word)
7. Hit Run to Verify Stop (if verify has been selected during build test list)
8. End of Test Stop (if stop at end of test has been selected in the Stop/Jump parameter word)

CAUTION

If at all possible, master clear only at programmed stops.

IV. OPERATOR COMMUNICATION

A. MESSAGES

1. Normal Program Identification Typeout

During initialization, the program identification typeout occurs as follows:

```
EDTA3E          17X4 LIBRARY EDITING ROUTINE
IA = xxxx      CP=2F   VR 4.0   REL.
```

2. Normal Entry Parameter Messages

3E81/STJP? A/Q parameter ID stop

LU/EQ A = Logical unit parameter

1000/0000? Q = Corresponding WESD field

Enter LU/EQ value as required. Terminate entry by zeroing A or after entering value for logical unit 7. Entries may be in any LU sequence.

After the parameters are entered, Edit loads the necessary overlays and reports the new parameter table on the **comment device**.

3. Reenter Parameter Message

Ready Library Unit This is typed if it is necessary to reload Edit's overlays after L.U. table change.

3E1F/0070? Make the library unit (where Edit was initially loaded from) ready and hit RUN/TYPE carriage return.

4. Normal Build Function Table Typeout

After parameter entry, a typeout occurs as follows:

BUILD FUNCTION TABLE

The typeout is followed by a stop with the contents of A and Q zero. At this stop the operator must enter a function word pair into the A and Q registers and place the RUN/STEP switch in RUN. This procedure must be repeated until all the desired function word pairs have been entered. Then, to terminate the function table, the operator must clear both A and Q and place the RUN/STEP switch in RUN. *FOR VERIFY* is typed out if this is a verify only run (STJP bit 15 is set initially).

9. Change Prestored Test Parameters Message
- *PAR. CHANGE A/Q*
- Edit is ready to accept changes to the prestored target test parameters.
10. Change SMM Monitor Parameters Message
- *TFREQ/WESD* 1. TABLE
- Enter up to 10 values in A and Q and terminate with zeros.
- *I.A./STJP* 2. TABLE
11. Patch Stop ID Message
- *CORR. STOP ID*
- Edit is ready to accept a length change in Q if Q is zero. If Q is non-zero, it shows the current overlay number. Hit RUN to get to the actual patch stop or zero A/Q to disable further patches.
12. Patch Stop Message
- *ADDR./DATA*
- End of the target test or one of its overlays and Edit is ready to accept the actual patches. Terminate this patch stop by zeroing A/Q.
13. Test ID Message
- *XXXXXXX*
- Edit has encountered a NAM block on the input unit during copy. Informational typeout only (XXXXXXX is the test ID).
14. Device Not Ready Message
- *LU XX NOT RDY - STATUS = YYYY*
- 3E1F/0X71?
- One of the logical units is not ready/ or not write enabled (XX - L.U. number and YYYY = Error status).
- Make L.U. ready and type (CR) to continue.
15. New Overlay Message
- *NEW OVERLAY XXYY*
- Edit has encountered program address reversal in the current RBD block (XX = overlay number and YY = test number). This should only occur on an MSOS RBD output having address reversal. This is an error indication if the test does not have any programmed overlays (can be caused by reading same RBD block twice).

16. 6000-Type NAM Block Message

3E28/STJP?

0060/TESTNR. ?

Nonstandard 6000 class assembler NAM block encountered. Clear A and Q of the second stop and enter the date and test biasing code in the form as follows:

A = mmdd

Q = yy0x

mm = month (1-9, 10-12 hex)

dd = day (1-9, 10-19, 20-29, 30-31)

yy = year (70-79 hex)

x = 0 for monitor biasing or 1 for test biasing

17. Skipped Cards Message

SKIPPED XXXX CARDS

The card reader loader has read and skipped xxxx non-RBD cards; this should only happen when reading through binary Quick Look.

Criteria for RBD cards:

a. 7/9 punch in column 1

b. \$OFF in column 2 (part of record length)

18. Disk End of File Message

SMM END OF FILE AT (CYL/HD/SEC) = XXYZ

This message was caused by execution of Special Function 1. A software EOF marker is written at CYL XX, Head Y, Sector Z (hardware address).

Edit will automatically write an MSMS end of file on the following sector, so that the first free sector is the reported one + 2.

<u>Code</u>	<u>Description</u>
1A	Reject from read disk or SMD (see 8).
1B	Reject from write disk or SMD (see 8).
1C	Alarm from 1733-3 SMD. A3 = DA status (10 retries if not drive fault)
20	Paper tape punch alarm. Driver will punch a block of equal length and re-punch record when punch ready. Error in an RBD block (4050) is recoverable if past the eighth frame of the block. Error in a NAM block (2050) or an OVL block (E050) may cause load problems. A3 = Status A4 = Block type A5 = Last test number Q3 = Not used Q4 = Data address if block type 4050 Q5 = Current function word
21	Paper tape supply low. Punching continues to gap between tests. A3 = Status A4 = Block type A5 = Last test number Q3 = Not used Q4 = Data address if block type 4050 Q5 = Current function word
22	Punch alarm on 430. Card is repunched. A3 = Status A4 = Block type A5 = Last test number Q3 = Column count if punch error (stop A3 bit 8) status 2 if not punch error Q4 = Data address if block type 4050 Q5 = Current function word
23	Output unit magnetic tape alarm. A3 = Status A4 = Block type A5 = Last test number Q3 = Not used Q4 = Data address if block type 4050 Q5 = Current function word
24	Output unit magnetic tape parity error. A3 = Status A4 = Block type A5 = Last test number Q3 = Block write retry count Q4 = Data address if block type 4050 Q5 = Current function word

<u>Code</u>	<u>Description</u>
25	Disk pack alarm on output or special function. A3 = Status A4 = Block type A5 = Last test number Q3 = Disk address (except function rewind) Q4 = Data address if block type 4050 Q5 = Current function word
26	Missing file mark after XFR block or binary Quick Look. Hit RUN to continue.
27	Reject from magnetic tape, read unit (see 8).
28	Erroneous file mark on input unit. Restart at build function table.
29	Reject from magnetic tape, write unit (see 8).
2A	Tried to skip to XFR block on disk, illegal.
2B	Disk not write enable. Restart.
30	Verify data compare error. A3 = Verify unit data A4 = Block type A5 = Last test number A6 = Failing word number (words numbered from 1) Q3 = Expected data Q4 = Data address if block type 4050 Q5 = Current function word Q6 = Number of words to be verified in this block
31	Error in loading Edit's loader/driver overlays. Reload Edit.
32	Not enough memory, less options or shorter function table. Retry without verify.
33	I/O overlay for device type not loaded. Restart.
34	Illegal L.U. entry. Rebuild L.U. table.
35	Verify has been selected, but verify overlay has not been loaded. Restart.
36	Input and output L.U. are the same. Rebuild function table.
37	Binary Quick Look has a zero cell caused by either coding error, wrong patch, or zero parameter. Restart with build function table. A3 = Address in BIN QL.

NOTES

Data and block ID on output errors (tape, cards) have been disassembled for output and are meaningless.

Card reader/punch will offset NAM block cards. Reader/punch errors will stop with no change to hardware status. Upon reentry, the last I/O will be repeated. Binary Quick Look will be repunched if punch error exists. Error cards will be offset on next feed.

VI. GLOSSARY

LOGICAL UNIT: A system-level software concept (as opposed to the I/O driver level, physical unit). It allows tabulated physical information words (equipment address, device type i. e. mag. tape, etc.) necessary to function and operate a specific physical unit to be referenced by the relative position, or index of that particular set of physical information words within the system's table of physical unit parameters, such table defining those units available to the system at a given installation. For example, logical unit no. 1 refers to the first set of physical parameters in the table, logical unit no. 2 to the second set, etc.

LIU/LOU: Abbreviations for logical input unit and logical output unit. The logical unit number indexes the physical unit table built in EDT's parameter entry routine and is entered in the upper Hex-digit of edit's function word for the desired input unit (LIU) of the operation described by that function word, or in the next-to-upper Hex-digit for the output unit (LOU) of that operation.

FUNCTION WORD: One 16-bit word entered by the operator, via the A register, to edit's function table. It describes a single edit operation and the logical unit(s) involved. The table, consisting of at least one, but usually more, function words, is executed one word at a time to perform the desired, operator defined, SMM17 library editing operations. Possible operations are:

1. Skip (LOU = zero; expects to be followed by a copy from the same LIU)
2. Copy (neither LIU nor LOU = zero)
3. Special function (LIU = zero) i. e., rewind library on mag. tape or disk pack, write end-of-library (file mark) on mag. tape or disk pack, punch leader on paper tape, etc.

TARGET TEST: An 8-bit hexadecimal test identification number (i.e. COM=\$01, MEM=\$02, etc.) placed in the lower byte of edit's function word for a Skip or Copy operation. It is used by edit to flag the end of that Skip or Copy operation and the subsequent read-up and execution of the next function word in the function table. Any program containing overlays must be designated as a target test when editing the "raw" output from the assembler. Edit adds overlay blocks (see error code 4).

SPECIAL FUNCTION: If the LIU of the function word is zero, there is no input unit defined for that operation; hence, it is neither a Skip nor a Copy operation. The lower 8-bits of the function word, in this case, will determine the actual operation, according to the output device type. For example, if the LOU points to physical information for a disk pack, and the lower 8-bits = \$01, a special data sector is written which will be decoded when read by the DP input routine as "end-of-file" and result in rewinding the library, i.e., positioning the DP, to the beginning.

3000 CHANNEL SIMULATOR PROGRAM UPDATE

(UD302D Test No. 2D)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. Runs on an SC/1700 in the following configuration:
 - a. (2) Tape Units
 - b. (1) Teletype
 - c. (1) 1729 Card Reader (Optional) *
 - d. A minimum of 8K of core
 - e. The number of edit entries are limited by:
 - 1) Core size. A maximum of 9600 characters (including spaces) can be entered into 8K, and an additional 8182 characters for each additional 4K of core. Trailing spaces not included.
 - 2) Or a maximum of 1000 entries, whichever comes first.
2. UD3 overlays SMM17.

B. LOADING PROCEDURE

1. Call as standard SMM17 test No. 2D

C. PARAMETERS

1. If the SKIP switch is set, typeout is suppressed.

II. CALLS

A. EDIT (.)

1. .DAAAA,BBBB
Delete AAAA0 (if only entry) or AAA0 through BBBB0.
2. .EAAAA
End update after transferring card AAAA0 to Unit 1, an EOF mark is then written to Unit 1.
Tape handling begins after this entry.
3. .FAAAA,BBBB
Fetch card AAAA0 (if only entry) or AAAA0 through BBBB0. These cards, from Unit 2, are typed and given an entry number. They are now data entries as if entered via TTY.

* If the card reader is available, has power on, and is loaded with cards, the cards will be read and handled as if they were teletype entries. The card columns must be correct. Control is returned to the TTY when the card reader faults depress rub-out to continue.

4. .IAAAA
Insert the following entries after card AAAA0.
5. .KAAA, BBB
Correct previous entry AAA (if only number entered) or AAA through BBB. After the corrections are complete, entry numbers jump back to normal sequencing.
6. - (minus sign)
A minus sign will delete an edit entry, of which the entry number is showing. This entry can be used in conjunction with the .K option to strike a bad edit entry.
7. .T
Terminate EDIT and begin tape handling. An EOF mark is written to Unit 1 after the last EDIT entry.

B. CONTROL (/)

1. /UX (X = Units 0 through 2 only)
Unload unit X.
2. /MX
Mark (Write an EOF on) unit X.
3. /SFFX
Search file forward on unit X.
4. /SFBX
Search file mark backward unit X. (1732 only)
5. /RX
Rewind unit X.

III. OPERATION

A. REQUIRED ENTRIES

1. In response to message:
"MTO = ICE, MT1 = ICE, MT2 = ICE"

Enter the interrupt line, converter, and equipment numbers for units 0, 1, and 2 respectively.
2. In response to message:
"ID = "

Enter the three characters desired in columns 73 through 75 of the sequence number on the output (Unit 1).

NOTE

New source is automatically sequence numbered by tens.

3. Enter any of the commands in II needed to perform the update. Any entry, other than a command, is considered to be a data entry and the following rules apply.
 - a. If the entries are from TTY, tabs are generated when a space is entered. Cards are required to carry the proper tabs but column skipping is done when spaces are encountered to reduce false data entries.
 - b. Auto tabbing is suppressed when:
 - 1) An * is in column 1.
 - 2) After column 20 on a BCD entry.
 - 3) After column 41.
 - c. The tabs (columns) for this program are 10, 20, and 41.
4. Tape assignments are:
 - Unit 0 = Old Source
 - Unit 1 = New Source
 - Unit 2 = Fetch (Card Images)
5. With the exception of Unit 2, no tape operations are performed as the result of an EDIT entry before an .E or .T entry is made.
6. When the EDIT is complete, the TTY will output Message III.A.2. and the program is ready to start the next job.

1700 SOURCE/6000 TVC UPDATE

(UD1A57 Test No. 57)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. Does not run with SMM17 (uses for loading only).
2. Requires two tape units and one teletype.
3. Requires the 1700 source to be at load point and end with an EOF mark.
4. Requires the 6000 external listing tape from a "TVC" assembly to be at load point on its tape. (Stacked programs or blocked tape are not acceptable.)
5. Requires special control cards in the 1700 source prior to a run with this program. (See Figure 1, types 5 and 6.)
6. Updates only one program at a time.
7. Sets P=2 and runs to reload SMM17.

B. LOADING PROCEDURES

1. Standard SMM17 call.
2. Test number 57.

C. PARAMETERS

1. Entry via teletype. (See II B.)

D. OPERATING INSTRUCTIONS

1. Load UPD via SMM17 operation instructions.
2. Mount a scratch tape on TU1.
3. Mount 1700 source tape on TU0.
4. Respond to teletype requests. (See II B.)
5. When TU0 rewinds, unload (if using a 2-tape system) and mount the 6000 listing tape on TU0. Change TU0 to TU2.
6. Depress and release the SKIP switch.
7. When TU2 rewinds, unload (if using a 2-tape system) and remount the original 1700 Source Tape. Change TU2 to TU0.
8. Depress and release the SKIP switch.
9. When the teletype goes back to request entries again, the job is complete and TU1 is holding the new source tape.

II. MESSAGES

A. NORMAL MESSAGES

1. Start UD1A57 IA=0000

B. COMMAND MESSAGES

1. MTO=ICE, MT1=ICE, MT2=ICE
 XXX YYY ZZZ

This message is requesting the interrupt line, converter number, and equipment number respectively for tape units (TU)0, 1, and 2 respectively.

2. BC NAME = AAAAAAAAA

This message is requesting a name of up to eight characters (less with a carriage return) which exactly matches the name in the 1700 source special control cards. (See Figure 1, types 5 and 6)

C. ERROR MESSAGES

1. None

	0	0	0	1	1	11	2	3	3	4	7
	1	3	7	1	3	78	6	0	5	1	2
1	*	REMARKS CARDS									
2		NUM	\$HHHH	AAA	SSSSSSS		OOO	RRRRRRRRRR		COMMENTS	
3				NLST							
		NUM	\$0000								
		NUM	\$10000								
		NUM	\$00000								
4				LS T							
				EJ T							
	*			NNNNNNNN				PAGE	NNNN		
5*	*			STRT	NNNNNNNN						
6*	*			END	NNNNNNNN						

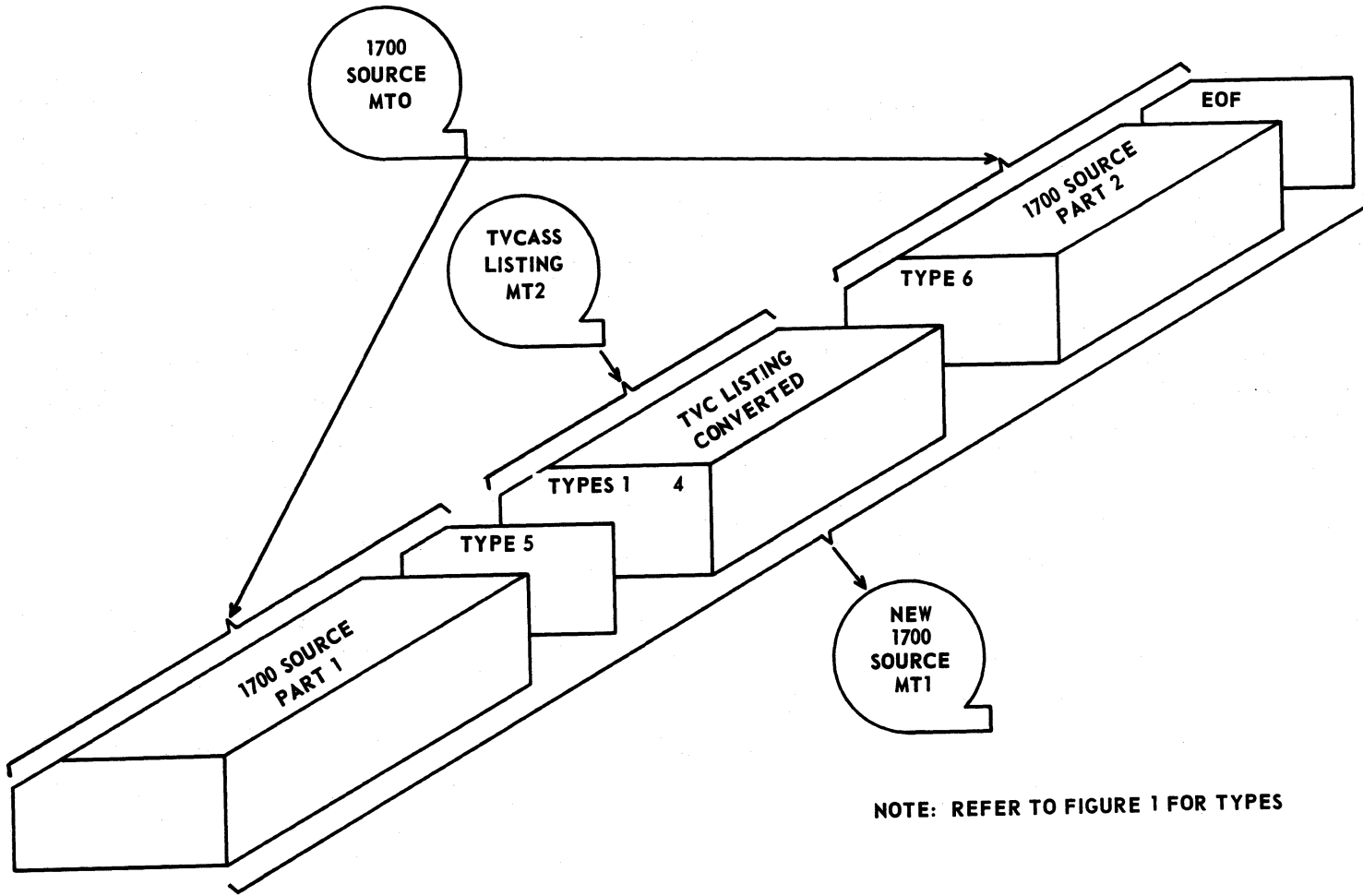
*These types are not made by the Edit program (in 1700 source).

The above shows the types of cards generated by the BC2 Edit program.

The program is extracted from a 6000 External Listing tape of the TVC assembly and inserted between type 5 and 6 cards bearing the same name as on those cards and specified by teletype. Up to an 8-character name is permitted. Types 5 and 6 are 1700 rem cards and inserted in the 1700 source some time prior to the use of BC2 Edit. The following is a list of the types generated for the respective TVC code:

TVC	GENERATES TYPE
Page eject	4
Rem card	1
BSS card/ORG card	1 for the statement, 3 with as many NUM \$0000 as necessary to fill the block requirement.
INST. card	2
END card	1
TIME and all other non instruction types	
CON/code types	2

Figure 1.



NOTE: REFER TO FIGURE 1 FOR TYPES

Figure 2

1700 SOURCE/6000 BUCAL UPDATE

(UD2A58 Test No. 58)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. Does not run with SMM17 (uses for loading only).
2. Requires two tape units and one teletype.
3. Requires the 1700 source to be at load point and end with an EOF mark.
4. Requires special control cards in the 1700 source prior to a run with this program. (See Figure 1, types 5 and 6.)
5. Requires that the 6000 external listing tape from a BUCAL assembly carry the same name in its identification card in the 1700 source. (See Figure 1, types 5 and 6.)
6. Updates only one program at a time.
7. Sets P=2 and runs to reload SMM17.

B. LOADING PROCEDURES

1. Standard SMM17 call.
2. Test number 58.

C. PARAMETERS

1. Entry via teletype. (See II B.)

D. OPERATING INSTRUCTIONS

1. Load UPD via SMM17 operation instructions.
2. In response to message II B 3, mount a scratch tape on TU1 and 1700 source tape on TU0.
3. Respond to teletype requests. (See II B.)
4. In response to message II B 4, unload (if using a 2-tape system) and mount the 6000 listing tape on TU0. Change TU0 to TU2.
5. Depress and release the SKIP switch.
6. In response to message II B 5, unload (if using a 2 tape system) and mount the original 1700 source tape on TU2. Change TU2 to TU0.
7. Toggle SKIP switch.

8. When the teletype goes back to request entries again, the job is complete and TU1 is holding the new source tape.

II. MESSAGES

A. NORMAL MESSAGES

1. Start UD2A58 IA=0000

B. COMMAND MESSAGES

1. MTO=ICE, MT1=ICE, MT2=ICE
 XXX YYY ZZZ

This message is requesting the interrupt line, converter number, and equipment number respectively for tape units (TU)0, 1, and 2 respectively.

2. BC NAME = AAAAAAAAA

This message is requesting a name of up to eight characters (less with a carriage return) which exactly matches the name in the 1700 source special control cards and the identification card for the Bucal Assembly. (See Figure 1, types 5 and 6)

3. MTO=BCX, MT1=SCRATCH (self explanatory).
4. Load BUCAL list tape on MT2 and toggle SKIP switch (self explanatory).
5. Reload original BCX source on MTO and toggle SKIP switch (self explanatory).

	0	0	0	1	1	11	2	3	3	4	7
	1	3	7	1	3	78	6	0	5	1	2
1	*	REMARKS		CARDS							
2		NUM	\$HHHH	AAA	S	SSSSSSS	OOO	RRRRRRRRR	COMMENTS		
3				NLST							
4				ORG		PRGBUF +	\$NNNN				
5*	*			LST							
6*	*			EJT							
				STRT		NNNNNNNN					
				END		NNNNNNNN					

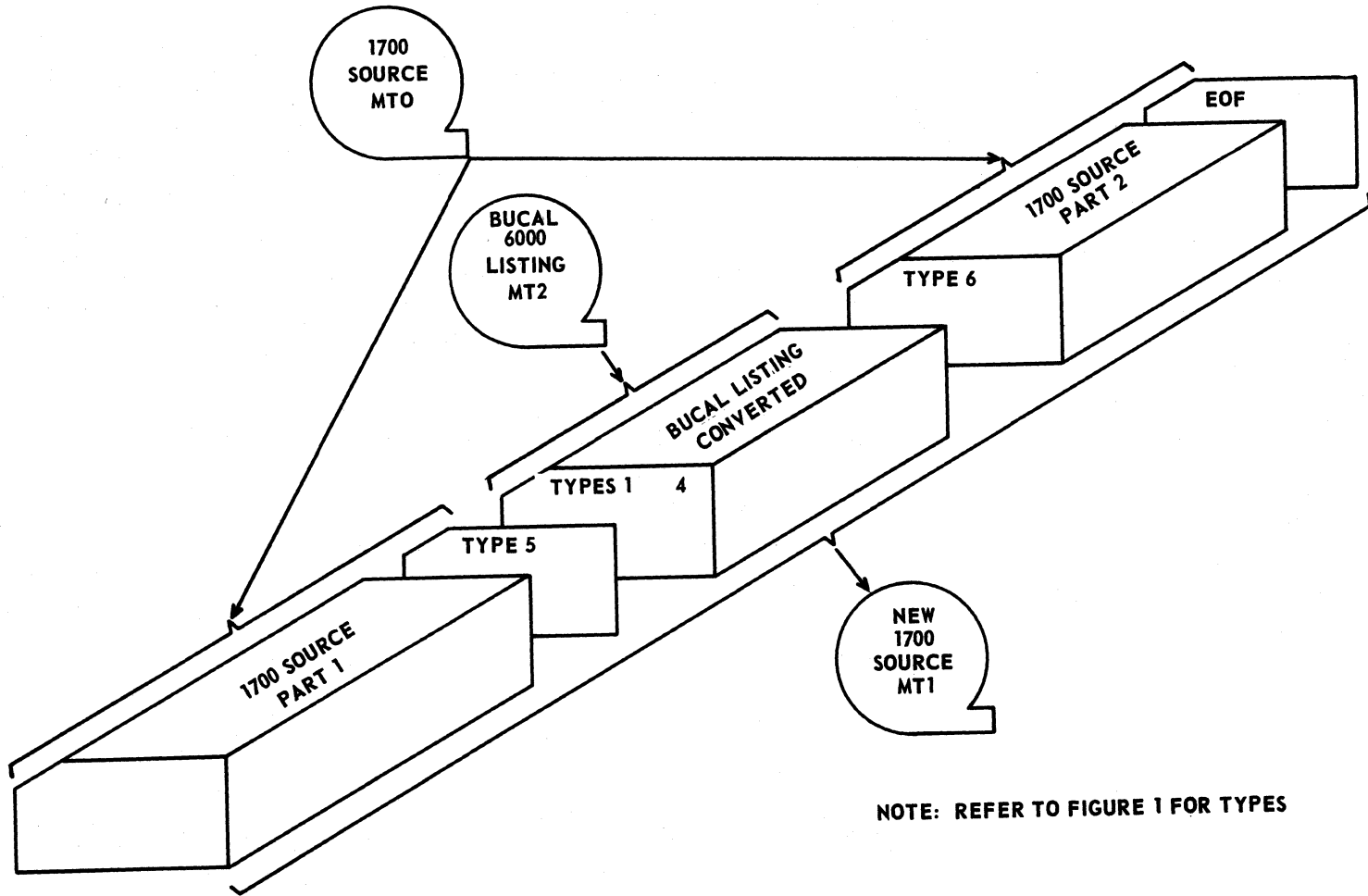
*These types are not made by the Edit program (in 1700 source).

The above shows the types of cards generated by the BC2 Edit program.

The program is extracted from a 6000 External Listing tape of the BUCAL assy. and inserted between type 5 and 6 cards bearing the same number as on those cards and specified by teletype. Up to an 8 character name is permitted. Types 5 and 6 are 1700 rem cards and inserted in the 1700 source some time prior to the use of BC2 Edit. The following is a list of the types generated for the respective BUCAL code:

<u>BUCAL</u>	<u>GENERATES TYPE</u>
Page eject	4
Rem card	1
BSS card	1 for the statement and 3
/ORG card	
INST. card	2
END card	1
TIME and all other non instruction types	
CON/code types	2

Figure 1



NOTE: REFER TO FIGURE 1 FOR TYPES

Figure 2

1700 MASS STORAGE MAINTENANCE SYSTEM INSTALLATION PROGRAM
(INSAFD Test No. FD)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. INSAFD needs a minimum of 8K of memory to run.
2. INSAFD runs only from a disk library.
3. The supplied parameters must be valid for proper operation and should be prestored on initial edit copy to disk.
4. All disk addresses are in the cylinder-head-sector format (actual hardware address).
5. INSAFD is a run-alone test.
6. Do not Master Clear during header typeout.
7. Sector addresses for CONH/CONS routines must be legal.

B. LOADING PROCEDURE

Standard SMM test loading, test number FD. Equipment number for the disk during the Build Test List phase must be valid or zero if correctly prestored.

C. PARAMETERS

1. Use the normal SMM parameter entry procedure, if required. These parameters should get prestored via Edit. To reeneter parameters for initial address restart, set bit 10 and bit 0 in the Stop/Jump word.
2. Parameter Stops.
 - A1 = ID word (FD21)
 - Q1 = Stop/Jump word
 - A2 = Disk type (4)
 - Q2 = Disk unit number (0)

Remarks:

A2 - Disk type is prestored as 4:

FA706	Use 4
1733-1	Use 4
1739	Use 6
1738	Use 4
1733-2	Use 5

D. SKIP AND STOP SETTING

1. The STOP switch is used only for SMM operation, and is not used in INSAFD.
2. SKIP switch set during I/O error reporting of disk driver or recovery of magnetic tape driver will terminate recovery.

Other SKIP switch actions will be requested on TTY.

E. MESSAGES

All communication between the operator and the program is done via TTY input/output. Output messages are typed via SMM's typeout routine. Input is done via INSAFD's driver. In the Idle loop, INSAFD rings the bell, types REQUEST, and waits for an operator input.

Input is interpreted as either hexadecimal data or ASCII characters, and may be of any length. Only the last four hexadecimal characters or six ASCII characters are used. A line feed is ignored. Rubout characters will repeat the input after receiving a terminator. The terminator for the input operation is a carriage return.

F. ERROR CODES

There is only clear-text error reporting. (For input driver errors, refer to I.H.4.) The INSAFD disk driver reports on error file (refer to appendix disk error) and retries I/O until the SKIP switch is set temporarily.

G. FORMATS OF TESTS ON THE DISK

1. SMM Tests

RBD format, one test RBD record = one disk sector (except QL binary).
Installed with SMM's Edit routine (test 3E).

Word 1 of a NAM block-sector holds the sector-address of the next test's NAM block.

Word \$39 of a sector being zero means SMM EOF.

2. MSMS Tests

Absolute format. The first disk sector holds the test header as follows:

Word 0 = Sector of next test

1 = Length (zero is MSMS EOF)

2

3 } 6-character ASCII name

4 }

5 = Load address

6 = Go address (FFFF is load only)

7 = Actual first data word of an MSMS test and beyond

This test header will automatically be attached to a test by INSAFD during an Install operation.

H. SECTIONS

Section selection is done by typing one of the following control statements when the program is in the Idle loop.

INST _{CR}	Install an MSMS test on disk
HEAD _{CR}	Change an MSMS test header
CORR _{CR}	Correct (update) an MSMS test
LUIN _{CR}	Select input logical unit
DMP _{CR}	Dump core on list device
DMPD _{CR}	Dump disk on list device
SAVE _{CR}	Save MSMS tests on disk scratch area
REST _{CR}	Restore MSMS tests from scratch
LHXC _{CR}	Load hexadecimal data into core
LHXD _{CR}	Load hexadecimal data on disk
WEOF _{CR}	Write end of file on disk
KILT _{CR}	Kill (remove) MSMS test on disk

EXIT _{CR}	Exit to SMM (end of test)
CONH _{CR}	Convert hardware disk address
CONS _{CR}	Convert sequential disk address
FUNC _{CR}	Function to input unit
GOTO	Give control to P
CPAR	Change INSAFD and run parameters)

1. INST - Install an MSMS Test on Disk

The MSMS test will first be read into memory from the selected input device. Then the MSMS test header data information is requested via TTY.

LOAD = A load address for later loading into memory by DPCAFC.

GOAD = An address in memory within a loaded test where control is to be given to the test. A GOAD of FFFF (negative zero) will only load the test, and not execute it.

Next, INSAFD asks for the name of the MSMS test to be installed. The name may be any ASCII input from one to six characters (except line feed or rubout). After the header is complete, INSAFD searches for the MSMS EOF on the disk and writes the new test, including header in place of this EOF. A new MSMS EOF is then written behind this new test.

2. HEAD - Change MSMS Test Header

This instruction allows the operator to change the LOAD, GOAD, and name of an installed MSMS test (refer to I.H.1).

3. CORR - Correct/Update an MSMS Test on Disk

This statement allows the operator to modify hexadecimal data within a test on the disk, thus giving the possibility to debug and modify an MSMS test.

Upon execution of CORR, INSAFD asks for the name of the MSMS test to be modified. It then tries to locate the test on disk and loads it into memory, if present. The TTY asks for an address within the test. (Do not add any bias, use the test listing address.) After checking the address for legality, INSAFD reports the old data and asks for the new data to be stored at that address. This continues until either an address or data input is S which will terminate the CORR operation and rewrite the modified test on disk. A carriage return input for data will leave data the same. Use caution.

4. LUIN - Select Input Unit by Device Type to be used for INST:

Current legal logical unit device types are:

- 0 Input from memory (FWA/LWA)
- 1 Paper tape (1721, 1777) - Olympus/Utopia format
- 2 Card input (1729-1, 1729-2, 1728, 1726) - 8-bit cards with checksum
- 3 Magnetic tape input (1731/1732-608-609) - 7-track = three frames unformatted and 9-track = two frames assembly mode
- 4 Disk input (1738/1733-1/1739/1733-2/FA 706) - sector, length
- 5 Card input (1729-1, 1729-2, 1728, 1726) - 8-bit cards without checksum (binary 8-bit core dump to cards, no length and checksum)
- 6/7 Not used (illegal)
- 8 8000 magnetic tapes (refer to Device 3) - 7- or 9-track
- 9 Illegal
and
above

The input routine for device zero (memory input) is included in INSAFD. All other devices need drivers. These drivers are overlays located at the end of INSAFD's RBD records on the Disk library. The overlays are called from disk via SMM's overlay loader.

Magnetic Tapes (Device 3, 8)

The magnetic tape drivers will read the record positioned under the read head of the selected tape unit.

Paper Tapes (Device 1)

Paper tapes are formatted, two frames per 16-bit word. (First word = length, last word = checksum.)

8-Bit Cards Device Type 2

One computer word = two 8-bit columns on card.

Card column 1 and 2 on first card = complement length of deck. Last two columns of deck = complement checksum.

No EOF card required.

8-Bit Cards Device Type 5

Data format is the same as in Device Type 2, but no length or checksum on cards.

Column 1 and 2 of first card = first computer word.

Card decks of this format must be terminated with an EOF card. The EOF EOF card has any punch in column 1 upper unused rows and must be added by the CE.

Driver Errors

Core	Addresses in error
Paper tape	Test too long, checksum error, alarm
Cards	Test too long, alarm, checksum error
Mag tape	Test too long, alarm
Disk	Alarm, too long, illegal addresses

Error Codes

- 13 = Checksum Error
- 14 = Paper tape alarm
- 15 = Card reader alarm
- 16 = EOP not on column 80 (cards)
- 17 = Magnetic tape alarm
- 32 = Not enough memory
- 33 = Illegal tape unit number

WESD and Unit Selection

INSAFD requests the device WESD field and unit number after the driver is loaded.

WESD for 1728/1729-2: Set bit 5 for reader (\$05A1 for EQ.#B)

1729-1: Use zero as WESD (assumes \$E1)

8000 tapes: Use 046X (station always zero)

5. DMPC - Dump Core to the TTY

This statement will dump core to the TTY. The FWA and LWA are requested via the TTY. The program then dumps core until the SKIP switch is set, or the current core address and last word address compare.

6. DMPD - Dump Disk to the TTY

The starting sector number will be requested, and sector after sector will be dumped on the TTY until the SKIP switch is set.

7. SAVE - Save MSMS Tests on Scratch

This statement is used to move all the installed MSMS tests out into the disk scratch area. This will be necessary when a new SMM library is to be installed on a disk. The scratch sector address is requested via the TTY and checked for legality. This usually should be \$ 7000 for an 854 and 1739, \$ 4000 for an 853. No testing should be done on the scratch area until the MSMS library is reestablished on the new SMM disk library. The selected scratch sector must be outside of the library area to ensure proper operation.

8. REST - Restore MSMS from Scratch

This statement will be used after a new SMM library is installed on disk via Edit in order to attach the MSMS library from the disk scratch area to the new SMM library. The starting scratch sector (previous Save operation) is requested via TTY. INSAFD moves all MSMS tests from the disk scratch to the EOF of the new SMM disk library.

9. LHXC - Load Hexadecimal Data in Core

This statement allows the operator to enter hexadecimal data into core via the TTY. The core address is requested. The old contents is reported, and then the TTY asks for new data. INSAFD updates the core address by one and waits for TTY data input. This repeats until S is input instead of hexadecimal data for a normal exit or an illegal hexadecimal character is found for an error exit. Input of carriage return only will advance the address but leave data the same. Use caution.

10. LHXD - Load Hexadecimal Data on Disk

This statement allows the operator to load hexadecimal data on any legal disk sector. The sector address is requested via TTY. That sector is then read into core. The word address within that sector is requested, legal is 0 to \$5F for 60 words in one sector. Data input is the same way as during a LHXC operation. An S input or a current word address of \$ 60 will terminate LHXD. The updated sector will then be rewritten on the disk. Changes can be verified with the DMPD operation. (Refer to I.H.6) Use caution.

11. WEOF - Write EOF on the Disk

This statement will write an MSMS EOF on the disk sector specified by the operator via TTY input.

A purpose for this capability is to recover a portion of a destroyed MSMS library. If the MSMS library is overwritten, the good part can be recovered and used by writing the EOF on the last good sector, then restoring the MSMS system to this new EOF. Use caution.

12. KILT - Kill a Single MSMS Test

This statement allows the removal of an MSMS test. (Must be used before installation of a new version of the same MSMS test.) The name of the test to be killed is input through the TTY and that test is located on disk. The rest of the MSMS system is then moved down, overwriting the test to be killed. Any new test will always get installed at the end of the MSMS system, so KILT/INST must be used to replace a test with a newer version.

13. EXIT - Exit to SMM Monitor

This will generate an End of Test Stop and then give control to SMM. There is no return to INSAFD after this statement, except by reloading.

14. CONH - Convert Hardware Disk Address to Software (MSOS) Address

This converts a disk sector address into a sequential sector address (cylinder, head, sector) into sectors. The sector address input via the TTY must be valid. The converted address will be typed out on the TTY (not valid for fixed platters on cartridge disks).

15. CONS - Convert a Software Disk Address to Hardware (Cyl, Head, Sector) Address

This converts a disk sector address into a hardware sector address (sectors into cylinder,head,sector). The requested sector address should be valid. The converted sector address will be typed out on TTY (not valid for fixed platters on cartridge disks.

16. FUNC - Function to the Input Logical Unit

This statement asks for a function code and transfers control to the currently selected input driver's function entry, if any. This function is legal only for 601/608/609/8000 Magnetic Tapes.

Code 3 = Backspace record

Code 6 = Advance file

Code 7 = Backspace file

Code 8 = Rewind

Code C = Rewind unload

17. GOTO - Give Control to P

The TTY will request the address and control will then be given to that location.

18. CPAR - Change INSAFD Run Parameters CPAR will cause a normal parameter entry stop using the SMM monitor.

II. DESCRIPTION

A. SCOPE

INSAFD is part of the 1700 Mass Storage Maintenance System. This system was designed to enable CEs to have, in addition to the standard SMM library, the ability to install and maintain non-SMM test routines, programs, utilities, bootstraps, and so on, on a maintenance disk pack so that all these routines are quick and easy to install, load, and execute. Another part of MSMS17, test number FC DPCAFC, is the call-up program enabling a selection between SMM and non-SMM tests after autoload and Quick Look execution.

INSAFD is the installation program for the MSMS tests. It also includes debugging routines to enable easy maintenance of the MSMS library on disk. INSAFD has to be called as a run-alone SMM Test (test number FD) with normal SMM loading procedure. The Q register at Build Test List time must be the valid WES code for the SMM library disk. Director bits are generally NOT APPLICABLE within INSAFD and DPCAFC.

The INSAFD parameters are handled the normal SMM way and should be prestored by Edit.

INSAFD requires at least 8K to run. It runs only from the disk library to make sure the CE is not using the customer's disk pack.

After the header typeout, INSAFD waits in the Idle loop and expects the CE to input one of the control statements.

INSAFD uses input drivers to read the MSMS tests into core. These drivers are overlays located at the end of the INSAFD RBDs on the SMM library. These overlays are called in after the LUIN selection via SMM's overlay loader.

INSAFD can be restarted at IA via master clear restart any time, except after EXIT-TYPEIN. EXIT-TYPEIN will generate an End of Test Stop and give control to SMM's multiplex routine.

B. SAMPLE INSTALLATION PROCEDURE

1. LUIIN

Select input device and load input driver overlay from SMM library unit.

2. INST

Reads test from input, requests header data, and installs this test in place of current MSMS EOF. Writes new MSMS EOF behind the installed test.

Repeat step 2 until all tests are installed. Use step 1 as required.

To replace an MSMS test with a newer version, use KILT, then INST. Also, use KILT to delete an MSMS test from the MSMS library.

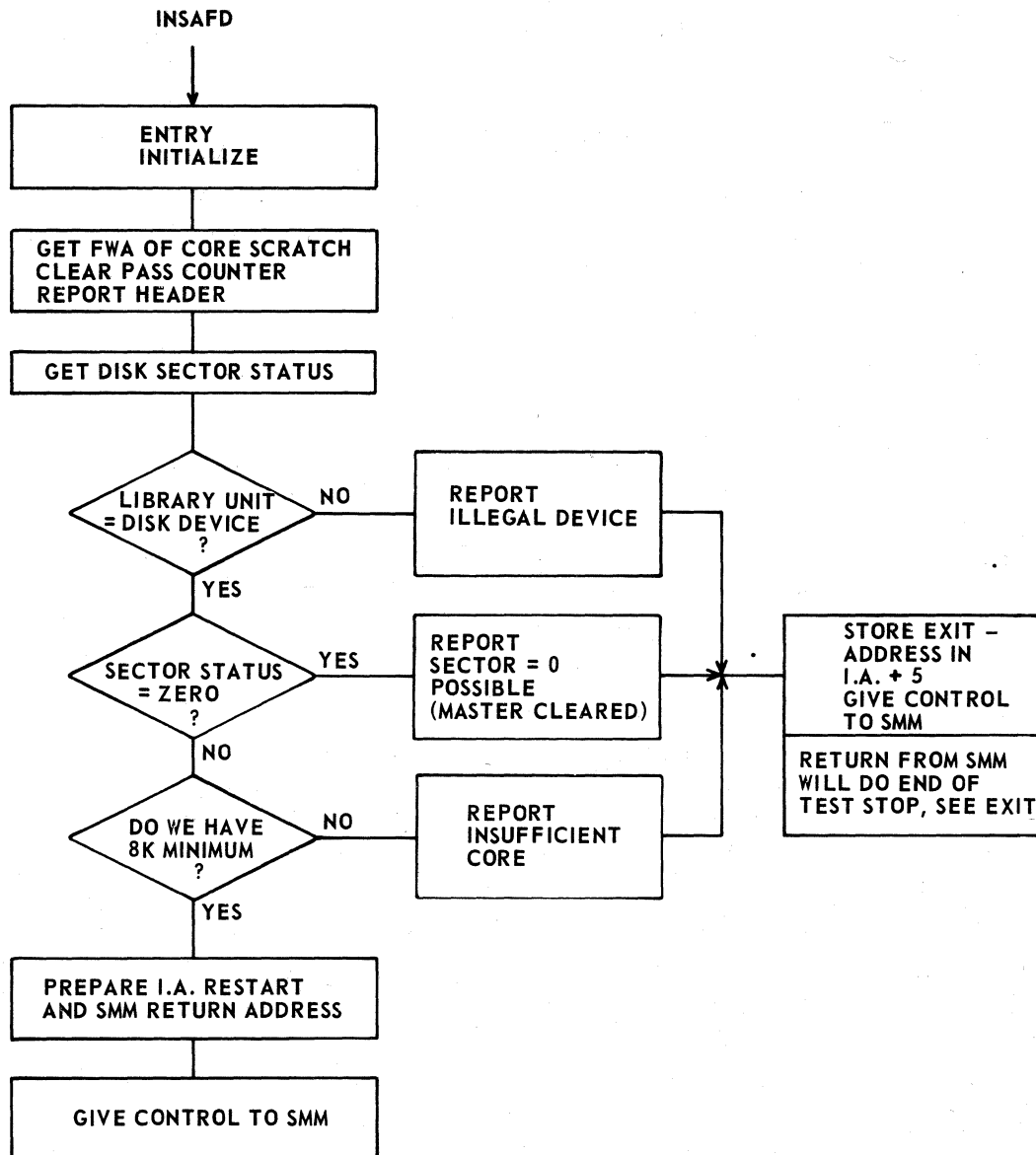
To get a new standard SMM test library installed, move the whole MSMS library out on scratch by using SAVE.

Do a normal Edit run on disk.

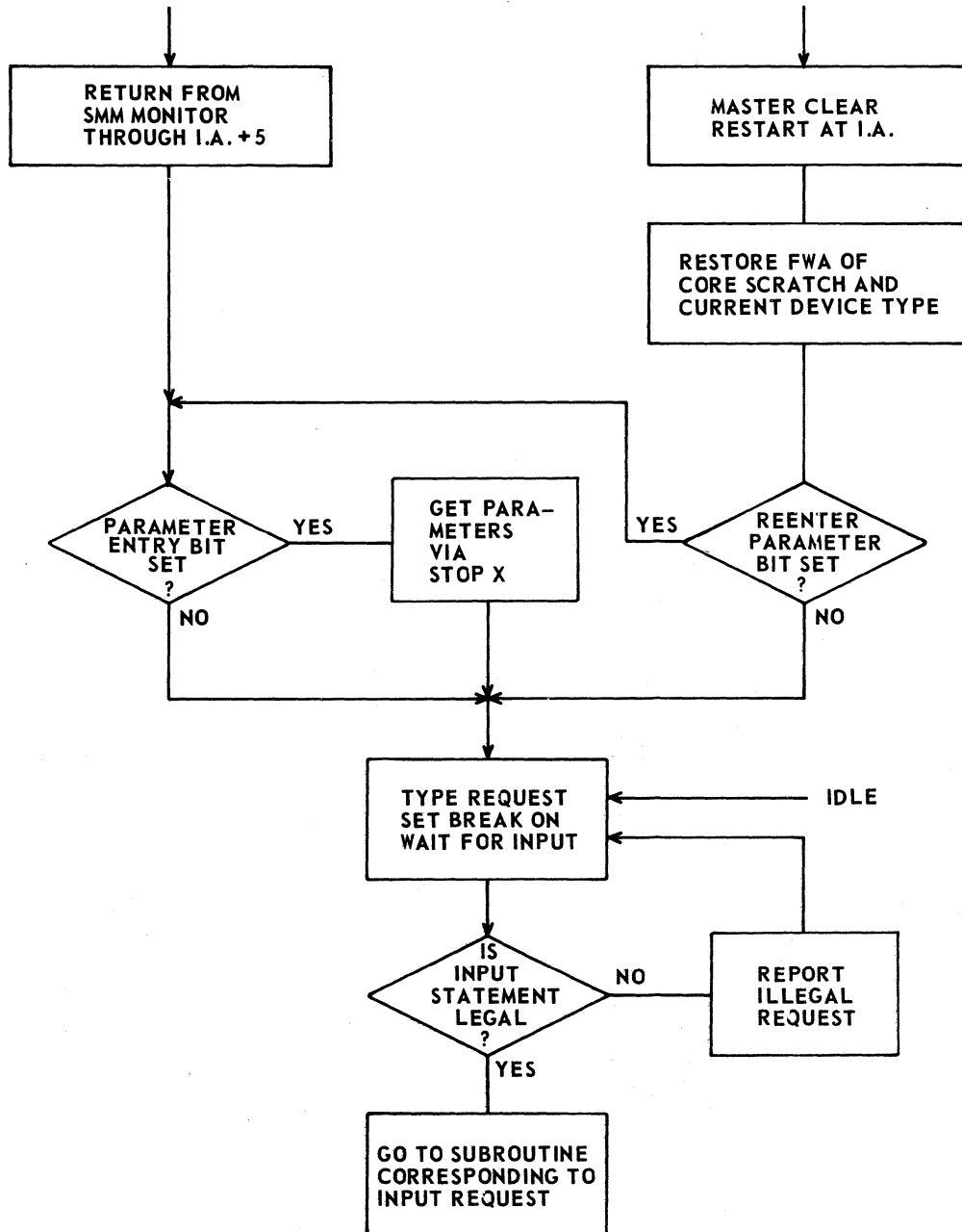
Reload INSAFD from the new SMM library. Restore the whole MSMS library to the new SMM EOF by typing REST.

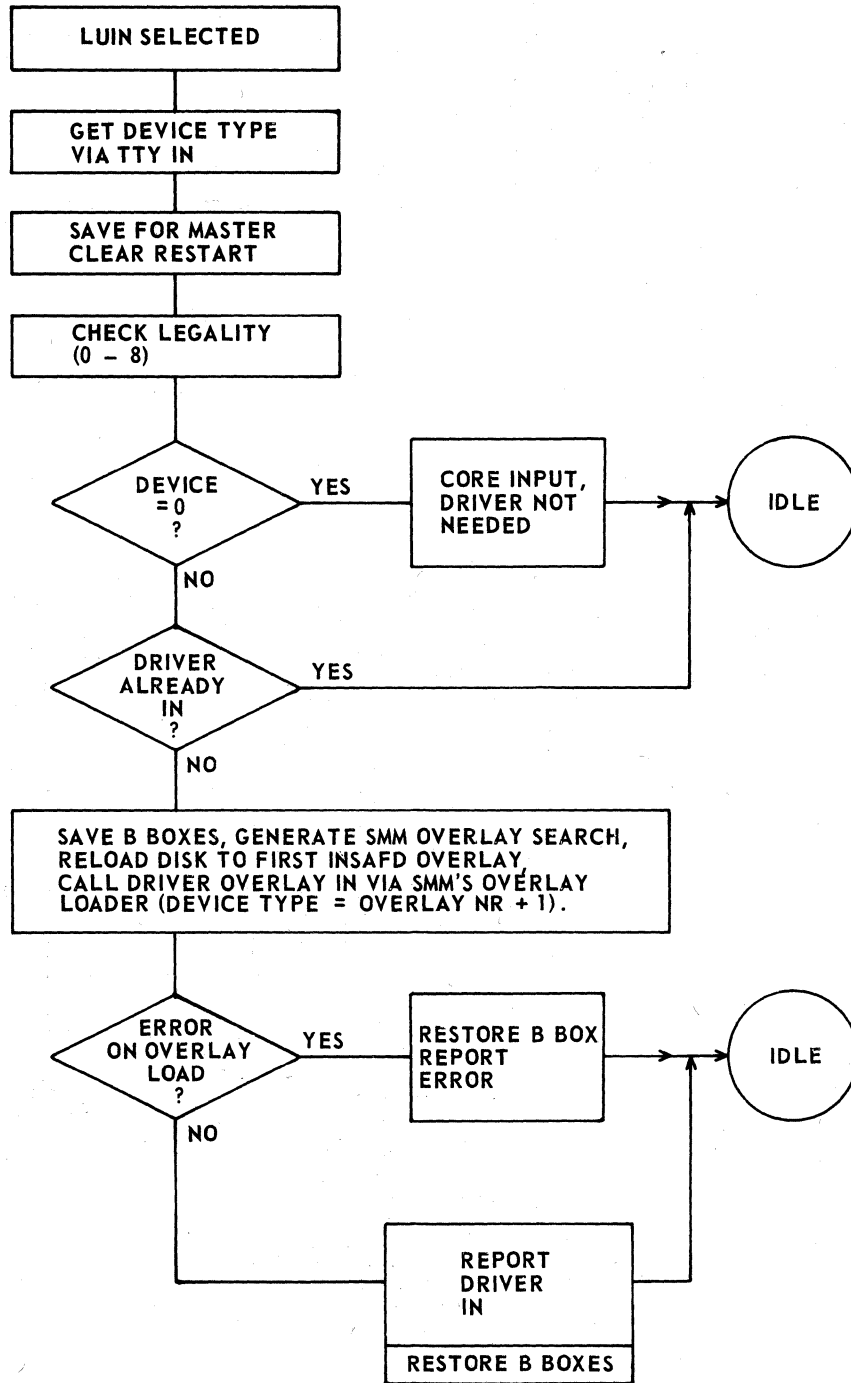
This operation requires that the starting scratch sector be kept in mind.

The initial MSMS EOF is written behind the SMM EOF by EDTA3E version 4.0, SMM editing routine.



INSAFD FLOW





APPENDIX A - DISK ERROR

I. DISK I/O ERROR MESSAGE

A. STATUS ERROR

Example:

```
OLDSTA = 0000    NEWSTA = 0000    OLDSEC = 0000    NEWSEC = 5000
DSKFWA = 17D9    DSKLEN = 0002    DSKDIR = 0183    DSKFUN = FFFF
Driver called from P = XXXX
```

OLDSTA	Status before operation
NEWSTA	Status after operation
OLDSEC	Sector address before operation
NEWSEC	Sector address after operation
DSKFWA	FWA of transfer
DSKLEN	Length of transfer (minus = function only)
DSKDIR	WESD field
DSKFUN	Function code (zero = input)
XXXX	Absolute address of call to disk driver

The above error was caused by status = zero (not ready). The DISKIO is retried until the SKIP switch is set.

B. OPERATION EXAMPLES (Operators input underlined.)

1. Select device type for INST operation:

```
17X4 MSMS INSTALLATION PROGRAM
INSAFD VR 4.0 CP 2F, I.A. = XXXX
```

Request - <u>LUIN</u> CR	Select input loader
Device - <u>2</u> CR	Two = 8-bit cards with checksum
Driver In	Overlay is loaded
WESD = <u>05A1</u> CR	Select equipment address
Request -	Wait for statement

2. Write an EOF on disk sector 4000:

Request- WEOFCR Select WEOF operation
Sector = 4000CR Write EOF on sector 4000
Request- (Hardware address)

3. Save the MSMS system on scratch area of disk:

Request - SAVECR Select save operation
Scratch Sector = 4000CR Save MSMS at sector 4000 and up
Last used sector is 4606

Last Scratch = 4606
Request-

4. Rewind tape unit:

Request- LUINCR Select new input device, tape
Device = 3CR
Driver In
WESD = 0381CR Select WESD
Unit = 0CR Select unit number
Request- FUNCCR Function request, 8 = rewind
Function = 8CR
Request-

5. Restore MSMS system from scratch:

Request- RESTCR Restore MSMS from scratch into system
Scratch Sector = 4000CR (Attach to end of SMM library)
Request-

6. Convert a sequential disk address to a hardware address:

Request- CONSCR Sequential disk address 1234 = cylinder 1D,
Sector = 1234CR head 1, sector 4

1D14
Request-

7. Install all MSMS test from tape:

Request- INSTCR

Load= 200CR

Goad= FFFFCR

Name= CRBOOTCR

Request-

Select installation. Reads tape in and requests load and go-address. Requests name for DPCAFC-CALLUP

8. Change name of MSMS test CRBOOT to CARDBT. Change load and go address to 3000:

Request- HEADCR

Name= CRBOOTCR

Load= 3000CR

Goad= 3000CR

Name- CARDBTCR

Request-

Change header

Reads old test header. New load go address and name will be inserted into the header and rewritten on disk

NOTE

INSAFD reads in the record under the read head of the selected tape unit in example 7, regardless what that record actually contains. As long as there is no hardware alarm and the record is not too long, it will get installed as an MSMS test.

1700 DISK CALL-UP PROGRAM
(DPCAFC Test No. FC)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. DPCAFC executes only from disk library.
2. The memory size parameter must be correct for proper operation and should be prestored via Edit when building the disk library.
3. The disk pack library must be installed by EDTA3E Version 4.0/INSAFD for proper MSMS list or loading operations.
4. All disk addresses are in the cylinder-head-sector format (hardware address).
5. DPCAFC needs a minimum of 8K of core, a TTY, and a disk to operate.
6. DPCAFC is a run-alone test.

B. LOADING PROCEDURE

The following two ways are used to load DPCAFC.

1. Load Via the Preloader

DPCAFC must be placed between the QL2-RBD and the SMM monitor-RBD on the disk library. After autoloading and QL execution, the preloader will load DPCAFC-RBD's instead of SMM monitor RBD's, beginning at location 0000. DPC will only execute if selected via bit 14 in QL1 Stop/Jump parameter.

2. Load as an SMM Test Via Monitor

Use the standard SMM loading procedure as outlined in the reference manual. The test number is FC and the equipment address has no meaning.

C. PARAMETERS

The following two ways are used to enter DPCAFC parameters, depending on how DPCAFC was loaded.

1. Loaded by the Preloader After QL Execution

Setting bit 15 in the QL Stop/Jump parameter will force one DPCAFC parameter stop if the STOP switch is on. (Bit 15 in the QL Stop/Jump parameter will be cleared after the parameter entry.) There is no possibility to reenter parameters in this case.

2. Loaded as a Standard SMM Library Test

Use standard test parameter stop techniques.

3. Parameter Stops

A1 = ID word = FC21 }
Q1 = Stop/Jump word } Not set if autoloaded

A2 = Highest memory location (prestored 7FFF)

Q2 = Sector of SMM monitors NAM block (displayed for information only)

Q2 is determined via Sector Address status and is displayed for information only. The bank select bits of A2 are used for DPC's final initial address and must match the hardware memory size.

D. SKIP AND STOP SWITCH SETTINGS

1. STOP Switch

This switch is needed only during QL load and parameter entry times.

2. SKIP Switch

Setting the SKIP switch during a LIST or LSMM operation will terminate the listing.

Setting the SKIP switch during disk error reporting will terminate the disk I/O retry.

E. MESSAGES

1. All communications between the program and operator are done via TTY input and output. After being loaded and the parameters entered, DPCAFC will move itself into the highest memory bank (usually \$7000) and report the following header:

SMM/MSMS 17 VR. 4.0

DPCAFC CP2F IA = X000

MSMS 17 stands for Mass Storage Maintenance System.

4.0 points to the version of DPCAFC

IA (initial address) is based on the parameter entry A2.

2. After all initialization has been done, DPCAFC will type RQ. Ring the bell, and wait for the operator's response; this is the Idle loop. This step will also be executed upon restart at the test's initial address.
3. The operator has the following possible responses:

LSMM_{CR}

LIST_{CR}

SM_{CR}

XXXXXX_{CR}

These responses result in the following.

a. LSMM

Lists the SMM library residing on the disk. (Refer to Appendix A.) This list is terminated by the operator setting the SKIP switch or the program locating the SMM end of file record.

b. LIST

Lists the non-SMM tests, their load address, and their execution address, residing on the disk. (Refer to Appendix B.) This list is terminated by the operator setting the SKIP switch or the program locating the MSMS end of file record.

c. SM

If DPCAFC was loaded by QL, control will be returned to the disk pre-loader to load the SMM monitor. If DPCAFC was loaded as a test by the SMM monitor, an end of test stop is executed.

d. Any Other Input

Interpreted as a name of an MSMS (non-SMM) test, DPCAFC tries to find and load that test.

4. If DPCAFC cannot find that test on the disk, the following will be typed.

TEST NOT IN LIBRARY, RESELECT
RQ-

5. If test XXXXXX is found, it will be loaded into core at the load address established during installation. Execution is dependent on the Execute flag contained within the test header and established during installation.

6. To have LSMM or List operations using the SMM printer for listing, load DPCAFC as SMM test.

F. ERROR CODES

1. Should any I/O error happen during I/O to the disk, an error file will be reported (Appendix C) and this I/O will be retried until the SKIP switch is set. Setting the SKIP switch returns control to the idle loop.
2. If DPCAFC is loaded as an SMM test and the SMM Final Loader device type is not 4, 5, or 6 (disk), ILLEGAL DEVICE TYPE message will be reported, and a normal end of test operation will be performed to exit to SMM.

If DPCAFC is loaded by QL and the device type is not a disk, an immediate return to the preloader is done to load the SMM monitor.

G. FORMAT FOR TTY INPUT

The terminator for teletype operations is either a / (slash) or a CR (carriage return).

Example:

TTY Output: RQ -

Operator Response: S_{CR} = load an MSMS test with a name starting with S

H. FORMAT OF TESTS ON DISK

1. All SMM tests are installed with SMM's Edit routine.
 - a. Every NAM block contains the sector address of the next NAM block in word 1.
 - b. The SMM EOF sector is the last SMM library sector on the disk. Word \$39 of that sector is zero.
 - c. DPCAFC should be placed between QL2 and SMM.
2. The MSMS tests are installed with the program INSAFD, test number FD, and are located on the disk, beyond SMM's EOF sector. The first sector of an MSMS test includes a 7-word header as follows:

Word 0 = Sector address of the next test

Word 1 = Length of this test (zero is MSMS EOF)

Word 2 = }
Word 3 = } 6-character ASCII name
Word 4 = }

Word 5 = Load address (test is loaded beginning at this core address)

Word 6 = Go address (control is given to this core address)

With a go address of negative zero (FFFF), DPCAFC will only load the test in core, and not execute. This can be used for the loading of data files as required by other MSMS test.

MSMS tests must be in absolute form (non-RBD). When installed on the disk, the first test data starts at word 7 of the first sector.

I. SAMPLE INSTALLATION OF SMM ONTO A DISK PACK

Edit is used to locate DPCAFC between Quick Look and SMM.

NOTE

This is not necessary if 4.0 release is used, since DPC is already located in front of the monitor.

1. L.U. - Table

1030/1381	LU1 = Tape
2040/181	LU2 = 854 Disk
.	
.	
.	
.	
.	

2. Function Table

0108	Rewind tape
0200	Rewind disk
12FE	Copy Quick Look I and II to disk
10FC	Skip tape to DPCAFC
12FC	Copy DPCAFC to disk
0108	Rewind tape
1000	Skip tape to SMM monitor
12XX	Copy rest of library to disk (XX = last test)
0201	Write disk EOF
0108	Rewind tape
0200	Rewind disk
0000	End of function table

II. DESCRIPTION

A. SCOPE

DPCAFC is one of three programs of the mass storage maintenance system MSMS 17. This system was designed to enable customer engineers to have, in addition to the standard SMM library, the possibility of installing and maintaining other useful test routines, programs, bootstraps, etc., on a disk pack so that all these routines are easy to load and execute. The MSMS installation program, INSAFD, is used to install and maintain the MSMS test library. It also includes debugging routines. REPAFB is used to maintain the SMM portion of the library.

DPCAFC is the call-up program used to call either the SMM monitor or any other non-SMM test residing on the maintenance disk pack. It will be executed either after disk autoloading and QL execution or after loading via SMM monitor as a standard SMM test, test number FC.

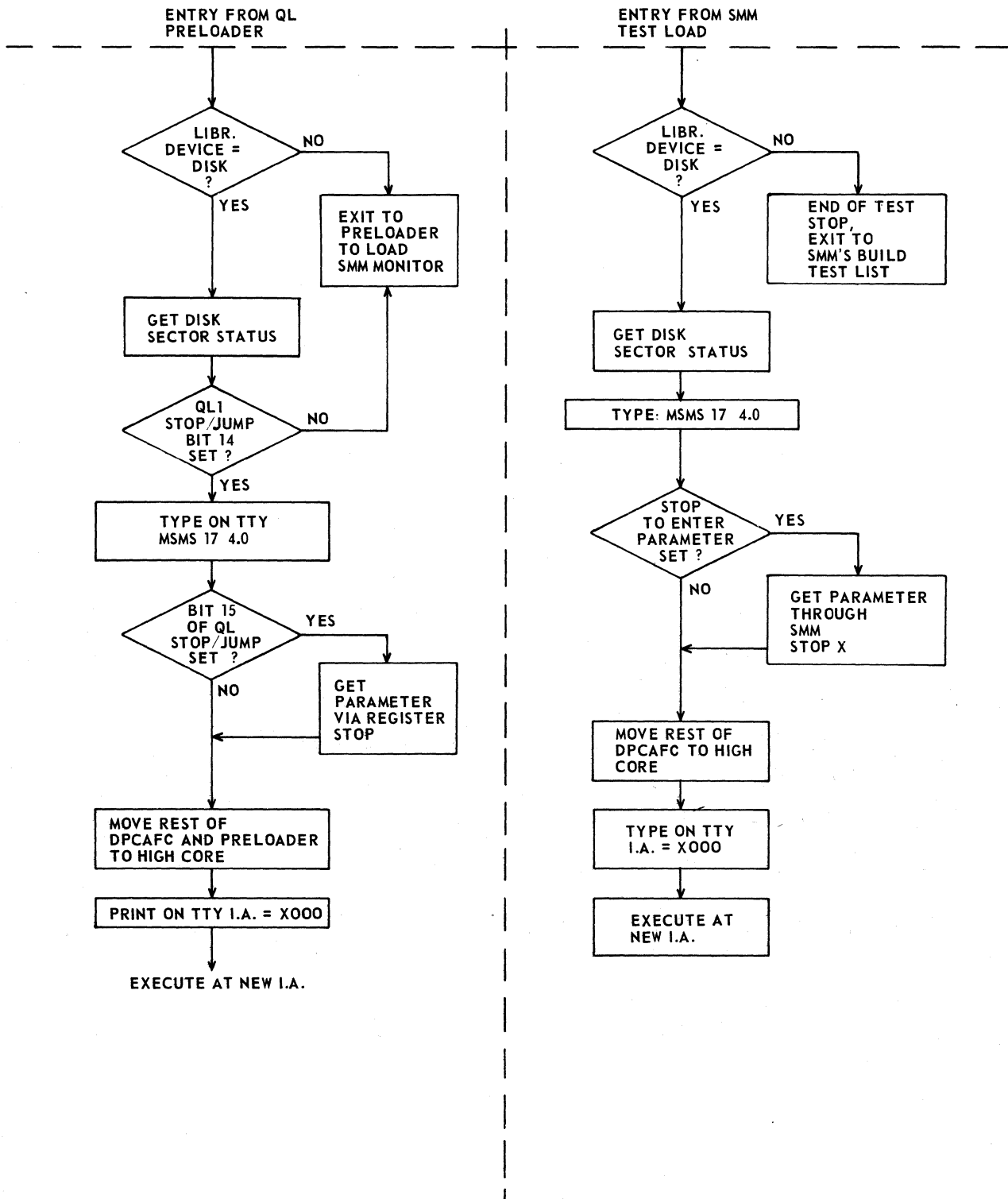
In both cases, the DPCAFC parameters may be changed, as required.

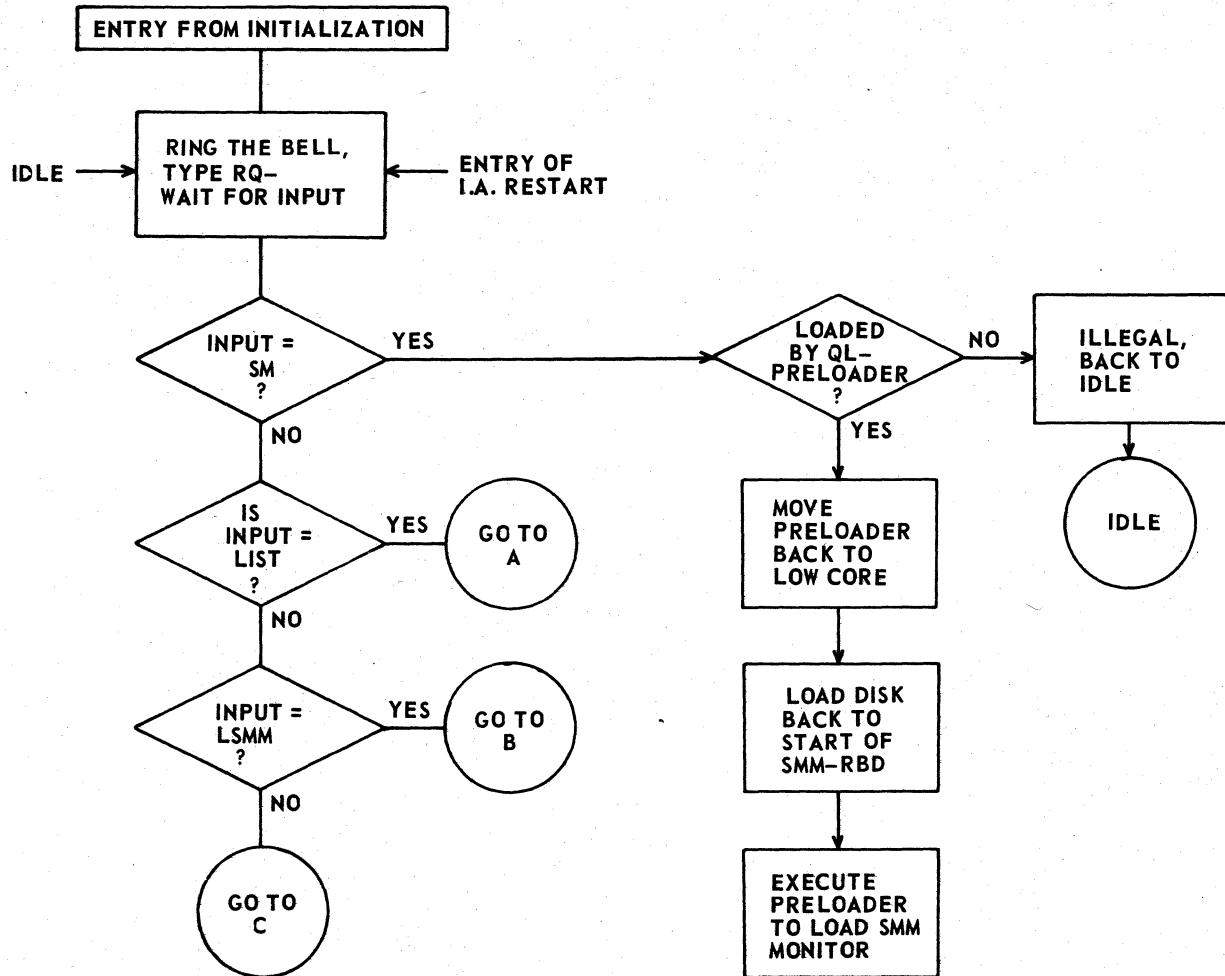
After the header typeout, the program waits for an input from the operator via the TTY. At this time, the CE may select library listing (SMM library on disk = LSMM, MSMS library on disk = LIST), select to run SMM (SM typein), or select to run any MSMS test by typing in the name of that MSMS test.

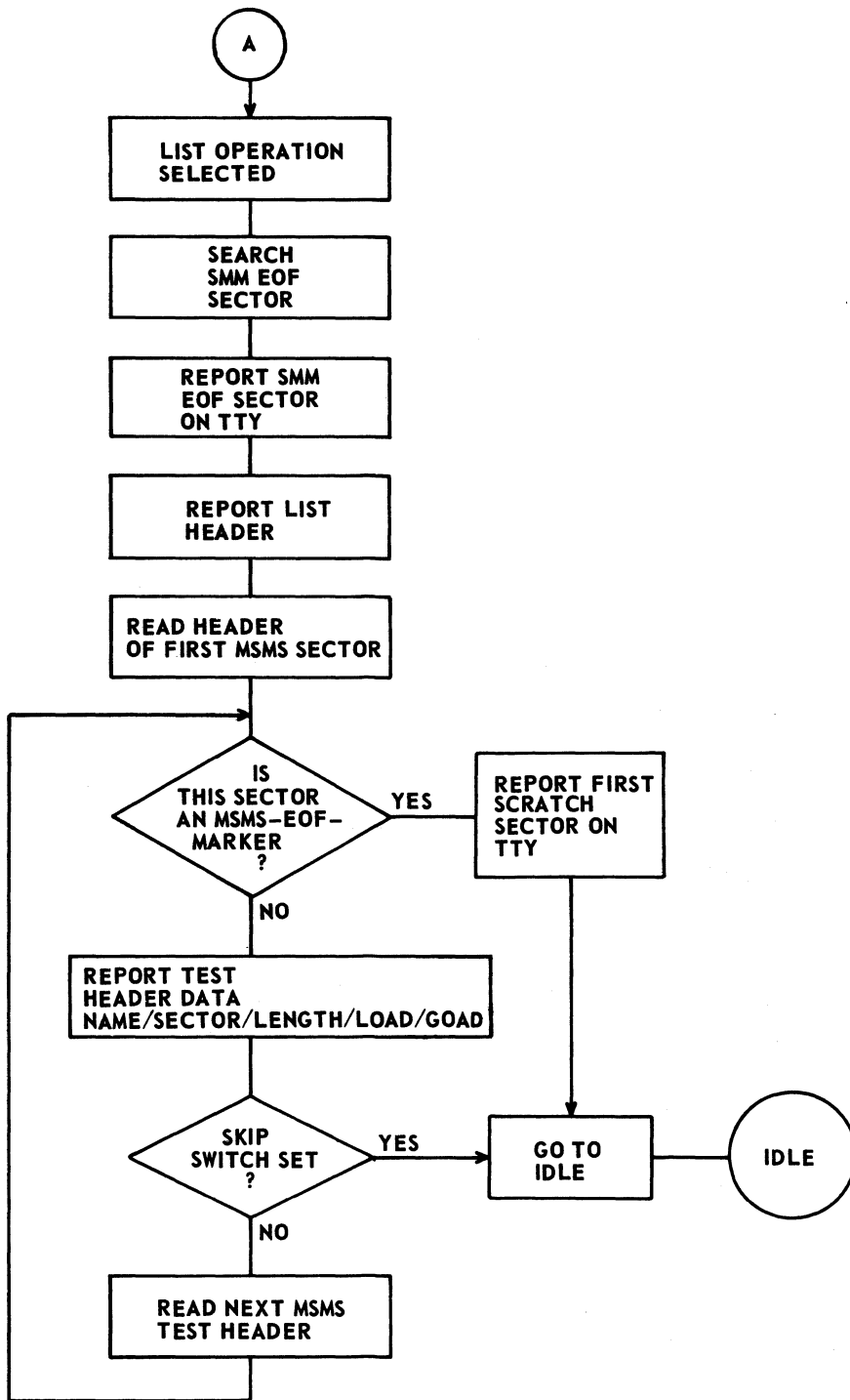
DPCAFC can be restarted at any time at its initial address (refer to header typeout), except when it has been overwritten by execution of another test.

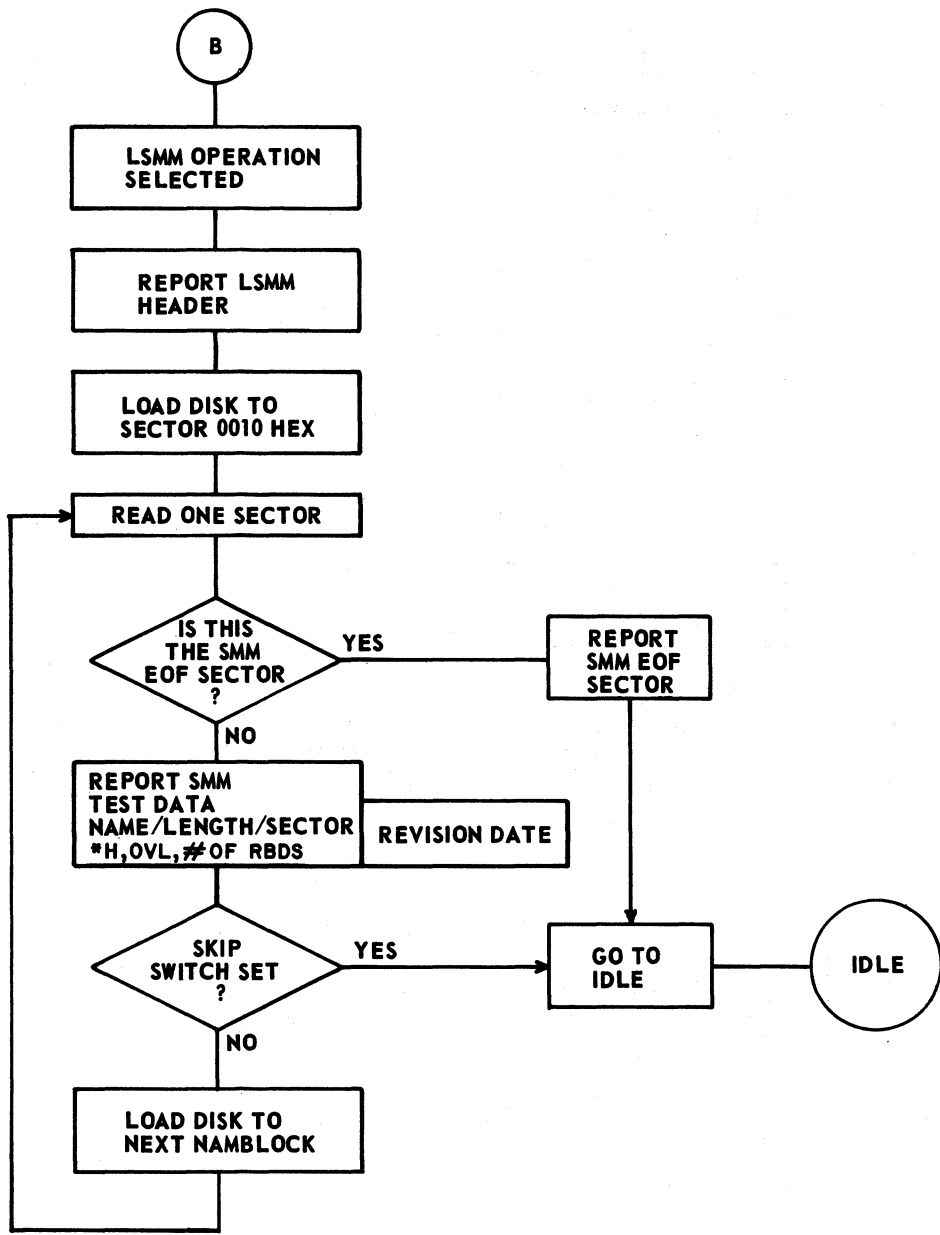
Either an MSMS EOF mark or MSMS-test must be installed on the disk pack by INSAFD, test NR.FD, in order to be able to do LIST and MSMS Load operations with DPCAFC. The MSMS EOF marker is also written by EDTA3E on the initial copy to disk (function 1).

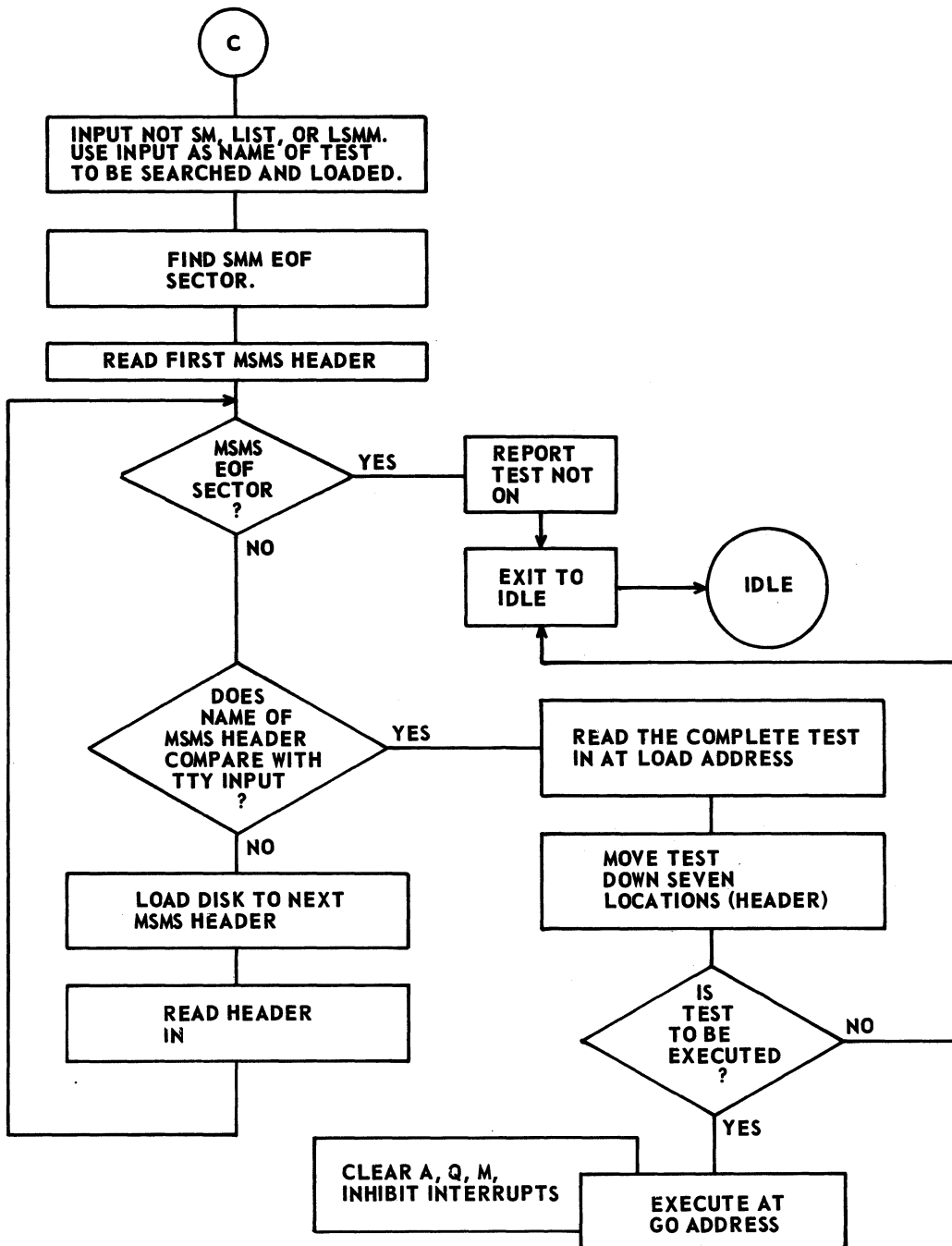












APPENDIX A LSMM LIST OF SMM TESTS ON DISK

NAME	LENGTH	SEC- TOR	REV DATE			
QL0AFF	0585	0010	08294	0022	RBDS	
DPCAFC	061D	0034	07224	0026	RBDS	
SMM000	0EEC	005C	08184	0044	RBDS	
OVERLAY,	ID=0100		SECT.=0101	FWA=06A1	0003	RBDS
OVERLAY,	ID=0200		SECT.=0105	FWA=06A1	0003	RBDS
OVERLAY,	ID=0300		SECT.=0109	FWA=06A1	0003	RBDS
OVERLAY,	ID=0400		SECT.=010D	FWA=06A1	0004	RBDS
OVERLAY,	ID=0500		SECT.=0112	FWA=06A2	0001	RBDS
OVERLAY,	ID=0600		SECT.=0114	FWA=06A2	0001	RBDS
OVERLAY,	ID=0700		SECT.=0116	FWA=06A1	0004	RBDS
OVERLAY,	ID=0800		SECT.=011B	FWA=00CA	0022	RBDS
COM001	096D	013F	06074	003E	RBDS	
MY2002	079A	017F	09044	002E	RBDS	
PTP003	05B9	020F	08054	0023	RBDS	
PTR004	03BD	0234	08044	0017	RBDS	
TTY005	08EA	024D	06074	0030	RBDS	

SMM-EOF AT SECTOR XXXX

APPENDIX B LIST OF MSMS TESTS ON DISK

SMM-EOF AT SECTOR = 184B

NAME	SEC- TOR	LENGTH	LOAD	GOAD
OLSILY	184C	24DC	4000	4000
RTMDRM	190F	20B6	0000	0000
VLOLYM	1967	2670	4000	4000

FIRST SCRATCH SECTOR=XXXX

APPENDIX C DISK ERROR

I. DISK I/O ERROR MESSAGE

A. STATUS ERROR - 10 RETRIES

OLDSTA = 0000 NEWSTA = 0000 OLDSEC = 0000 NEWSEC = 5000
DSKFWA = 17D9 DSKLEN = 0002 DSKDIR = 0183 DSKFUN = FFFF
Driver called from P= XXXX

OLDSTA = Status before operation
NEWSTA = Status after operation
OLDSEC = Sector address before operation
NEWSEC = Sector address after operation
DSKFWA = FWA of transfer
DSKLEN = Length of transfer (minus = function only)
DSKDIR = WESD field
DSKFUN = Function code (zero = input)
XXXX = Absolute address of call to disk driver

The above error was caused by status = zero (not ready).

The disk I/O is retried until the SKIP switch is set.

REPLACE/UPDATE 17X4 SMM DISK LIBRARY
(REPAFB Test No. FB)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. REPAFB needs minimum 8K of memory.
2. To ensure error-free operation, the disk pack must be prepared with address tags and surface test also in the scratch area.
3. The supplied parameters must be valid for proper operation. They should be prestored during initial edit-copy to disk. The valid disk-equipment-address must be supplied at load time (if not prestored).
4. All disk addresses are in the cylinder-head-sector format (hardware address). Cylinder addresses for COPP and COPY options are cylinders only, right-justified.
5. REPAFB works with the following disk controllers/drives:
 - a. 1738/85X (type 4)
 - b. 1739 - 1 (type 6)
 - c. 1733 - 1 (type 4)
 - d. FA706/85X (type 4)
 - e. 1733 - 2/856 - X (type 5)

B. REQUIREMENTS

1. Hardware
 - One 17X4 CPU with 8K of memory
 - One 171X TTY (or CRT)
 - One disk-subsystem (one drive)
 - One RBD-input device, if a test is to be replaced, either:
 - Paper tape input (type 1)
 - Card input (type 2), 173X Mag Tape (type 3)
 - 8000 Mag Tape (type 8)

2. Software

- One disk pack with the SMM library initialized by EDTA3E, Version 4.0
- One input media if a test is to be replaced
- The SMM 17 monitor version 4.0 or above with the TTY-input package selected
- To dump large amount of memory or disk SMM's line printer driver

C. LOADING PROCEDURE

Standard SMM test loading, test NR FB. The proper disk controller WES-field must be prestored in the test or supplied at Build Test time. Load REPAFB as a single test.

Example:

Build Test List

0101/000? FB01/181/CR

0201/0000? 0 CR

D. PARAMETERS

Parameters should be prestored via the edit routine during initial copy to the disk. The parameters must be valid for proper operation.

1. Parameter Stops (prestored values are in parentheses)

A1 = Test ID word (FB31)

Q1 = Stop/Jump parameter

A2 = Disk type, prestored as 4

4 = 1738, 1733-1

5 = 1733-2

6 = 1739

Q2 = Physical disk unit number, prestored as zero

A3 = Starting scratch sector address on disk, prestored as 4000
(hardware address)

Q3 = Library check flag, prestored as zero

0 = Check disk library

Nonzero = Do not check disk library unless there are known problems on the disk. This flag should be left zero.

E. SKIP AND STOP SWITCH SETTING

1. The STOP switch should be set at all times during SMM operations. There is no SLS instruction in REPAFB.
2. The SKIP switch will trigger the following.
 - a. I/O error retry on disk will be terminated.
 - b. Dump operations will terminate when the SKIP switch is set.

F. MESSAGES

1. General

All communication to the program is done via TTY (CRT) input/output. Output messages are typed via SMM's type out routine. Dumps are printed via SMM's Printer if selected in the SMM parameter word. Input is done via REPAFB's TTY input driver. In the Idle loop, REPAFB sets the break light, types REQUEST, and waits for operator input.

The input data is either interpreted as a hexadecimal number or an ASCII character and may be of unlimited length. Only the last four hexadecimal digits or six characters (ASCII) are used by the program. Line feed is ignored. Rubout will repeat the input operation after receiving the input terminator. The terminator is a carriage return.

2. Message Directory

a. Initialization

CHECKING LIBRARY - This message appears after parameter entry and indicates that REPAFB is checking the SMM and MSMS library strings. If the disk library is OK, REPAFB reports OK.

ILLEGAL DEVICE TYPE - This message is reported if the selected disk type is not 4, 5, or 6.

DISK NOT WRITE ENABLE - TYPE CR WHEN READY OR S CR IF TO USE AS IS - This message is reported if the disk type is 5, and REPAFB was unable to modify word \$5F on sector \$10.

b. During Operation

TEST NOT IN LIBRARY - Perform KILT or UPDA operation, where REPAFB cannot locate the selected test in the SMM library.

DRIVER IN - REPAFB has successfully loaded the selected I/O overlay.

FOUND TEST XXXX - REPAFB has read the NAM block of the selected test on the old disk library or input media.

RESELECT INPUT UNIT IF REQUIRED (LUIN) - REPAFB has executed a non-I/O overlay, and the previously loaded I/O overlay must be reloaded via LUIN if more REPL operations will be done.

SMM LIBRARY STRING BROKEN LAST GOOD SECTOR = XXXX - REPAFB detected a problem in the SMM library; XXXX is the last good NAM block sector. Recover the good library portion by writing an EOF on that sector (WEOF).

MSMS LIBRARY STRING BROKEN LAST GOOD SECTOR = XXXX - REPAFB detected a problem in the MSMS library; XXXX is the last good MSMS start sector. Recover with WEOF on that sector.

***COPYING TEST ON SCRATCH - REPAFB is moving an SMM test out on the scratch area during REPL or CORR operations.

***MOVING SMM LIBRARY FORWARD

***MOVING SMM LIBRARY REVERSE - REPAFB is moving the SMM/MSMS library on the disk to adjust for the different test length.

***ADJUSTING LIBRARY STRING - REPAFB is updating the NAM block sector string in the SMM library to accommodate different test length.

***MOVING NEW TEST INTO LIBRARY - REPAFB moves the new test from scratch into the prepared disk library.

***CALLING OVERLAY XX - REPAFB is calling (loading) its overlay number XX via the SMM monitor from the SMM library (LUIN or HELP, COPP, COPY, BIN QL update).

XXX SEARCHING TEST IN DISK LIBRARY - REPAFB is trying to locate the selected test on the old disk library.

TEST NOT INSTALLED, SCRATCH CONFLICT - The new test is too large for the space allocated between MSMS library end and the beginning of the scratch. Reselect scratch parameters.

c. Overlay Messages

SKIPPED XXXX CARDS - The card loader has read and skipped XXXX non-RBD cards (no 7/9 punch in column and/or no \$OFF in column 2).

ZERO DATA IN BIN. QL AT P = XXXX BIN. QL NOT GENERATED - Overlay 5 was called to generate a new binary Quick Look autoloading track. The binary data generated contains a zero cell at BIN QL address XXXX that would cause autoloading problems. Old BIN QL is left on disk. (This is caused by parameter change or illegal hex correction.)

ILLEGAL OPERATION - The COPP overlay received illegal parameters (start/end cylinder, unit number).

G. ERROR CODES

There is only clear text error reporting.

1. Disk Driver Errors

There are three types of errors as follows:

- a. Status error (usually alarm or not ready) - 10 retries before error is reported
- b. Disk reject
- c. Request error (illegal address)

All the above disk error messages are supported by the following error file.

Example: Request Error

OLDSTA = 0000 NEWSTA = 0000 OLDSEC = 0000 NEWSEC = CECE
DSKFWA = 7586 DSKLEN = 0060 DSKDIR = 0184 DSKFUN = 0000
DRIVER CALLED FROM P = YYYY

OLDSTA Status before error occurred
NEWSTA Status upon error
OLDSEC Sector address to go
NEWSEC Sector address to go
DSKFWA First word address of transfer
DSKLEN Length of transfer (minus=function only)
DSKDIR WESD field
DSKFUN Function bits (zero=input operation)
YYYY Absolute address of call to the disk driver

The above request error was caused by trying to access an illegal sector address on a 1733-2/856-4.

This error printout and the operation will be repeated until the SKIP switch is set.

2. Loader Errors

Loader errors are reported in the following form.

Input error, code = xxxx and status = YYYY

The code is similar to that used in the edit routine, that is:

07 = 1 Card reader sequence error
08 = Reject from input unit
09 = Input unit not ready
13 = Checksum error
14 = Paper tape alarm
15 = Card reader alarm
16 = Card reader EOP not on 80th column
17 = Magnetic tape parity error
26 = Missing file mark on tape unit (behind BIN.QL or XFR.block)
11 = Illegal RBD block in test
12 = No transfer block at end of test

The status is the device status at time of error or (A) if reject.

If the device uses recovery (magnetic tapes) and the reread was successful before the eighth attempt, the operation will continue; otherwise, an error exit to idle is done.

H. SECTIONS

Section selection is done by inputting one of the following control-statements on the TTY while REPAFB is in the Idle loop.

REPL	Replace an SMM test on the disk library
UPDA	Update an SMM test on the disk library
PARA	Set program flag to change the test parameters on the next REPL or UPDA
CORR	Set flag to do hexadecimal corrections on the next REPL or UPDA
LUIIN	Select the input unit by device type for REPL operations
DMPC	Dump core on list device
DMPD	Dump disk on list device
KILT	Delete (KILL) test from the disk library
EXIT	Exit to SMM's Build Test List = End of REPAFB
FUNC	Do function to input device
LHXC	Load hexadecimal data in core
GOTO	Give control to A
COPP	Coppy CDD parameters
COPY	Copy disk packs
HELP	List options
WEOF	Write software EOF
LHXD	Load hexadecimal data on disk
CPAR	Change run parameters
CONS	Convert software to hardware address
CONH	Convert hardware to software address

Each input must be terminated with an input-terminator (carriage return) to be recognized.

1. REPL Operation

Before this operation is selected, the input device should be made ready, and the input loader must be in memory. If parameters are to be changed and/or hexadecimal corrections to be done on the new test, the PARA and/or CORR mnemonics must be entered before REPL is started. This operation is illegal if no input device is selected.

After REPL input on the TTY, the program asks for the Test Number=. At this time, input the two-digit test number of the test to be replaced. The program now searches for that test on the input device. When the test is located, a typeout of FOUND TEST 00XX takes place.

This test now gets transferred to the scratch area on disk. After copy of the test's transfer-block, the old SMM library is searched for that test -NR.

If the test is in the old library, the new version on scratch will take the place of the old one within the library. The SMM library, as well as the MSMS library, will get moved in or out, as necessary.

If the test is not in the old library, it will be transferred into the sequential place within the library (for example, test 03 would be put between test 02 and test 04). Also, here the SMM library and the SMM tests will get moved on disk.

NOTE

If QL1AFF is being updated, REPAFB will call it's overlay 5 and produce a new binary Quick Look for autoloader. DPCAFC, test FC, must be placed between QL RBD and SMM monitor RBD for proper MSMS autoloader-operation. This must be done on the initial edit copy. REPAFB maintains that position if test FC is to be replaced.

2. UPDA - Update an Existing SMM Test

This operation uses much the same subroutines as REPL does. The major difference is that the old disk library is used as input rather than an input unit.

If the test is not found in the old disk library, REPAFB types out TEST NOT IN LIBRARY and exits to idle.

The CORR input must be done prior to UPDA input when hexadecimal corrections are to be done to the test being updated. In this case, the SMM library will get moved out because hexadecimal corrections will require extra sectors. Parameter changes alone will not increase the amount of sectors needed by the test. Therefore, no movement on the disk will take place when a PARA/UPDA only run takes place.

3. PARA - Change Parameters on the Next REPL

This input statement will set a flag in REPAFB to generate parameter changes on the next REPL or UPDA run. (Refer to examples for further explanation, Section III.)

4. CORR - Set Hexadecimal Correction Flag

This input statement will set a flag in REPAFB to generate correction-stops on the next REPL or UPDA run. (Refer to examples for further explanation.) Hexadecimal corrections will generate additional *H blocks and require more space on disk.

NOTE

If a test on the disk library has hexadecimal corrections and the CE wants to eliminate those, do an UPDA run with CORR selected for that test and have bit 13 set in REPAFB's Stop/Jump parameter word. This way the old *H blocks will not be copied. This applies to the REPL option also.

5. LUIN - Select Input Unit

This statement is used to select the input device for REPL operations. LUIN can be done any time REPAFB is in the idle state. The RBD loaders are overlays in the SMM library. These overlays are loaded via the monitor's overlay loader. If the input device is magnetic tape, REPAFB will also ask for the unit number.

Example:	<u>Explanation</u>
Request ? LUIN CR	Select device
Device = 2 CR	2=Card reader
Driver in	Overlay is loaded
EQ. address=5A1 CR	Equipment ADDR
Request?	For 430, status 1

If an error happens during overlay load, the SMM monitor will return an error to REPAFB. REPAFB will report input error.

Legal Device Types:

1 = Paper Tape	(1721/22 1777, 1713)
2 = Cards	(1729-2, 430, 1726-405)
3 = Mag tape	(1731-601, 1732-608/605, 1732-3)
8 = 8000 Mag tape	

The RBD loaders are similar to Edit's loaders.

6. DMPC - Dump Core on List Device

REPAFB asks for FWA and LWA (hexadecimal) and then dumps that portion of memory until CWA=LWA or the SKIP switch is set.

7. DMPD - Dump Disk on List Device

REPAFB asks for the hardware sector address and reads that sector into memory. The sector is then dumped on the list device, the sector address updated, and the dump continues until the SKIP switch is set or disk I/O error occurs.

NOTE

For DMPC and DMPD, select the SMM monitors line printer if available (monitor parameter).

8. KILT - Delete a Test from the Disk Library

KILT is used to delete unused tests from the disk library to tailor the library and make room on the disk. KILT asks for the test NR., locates the test on the library, and moves the rest of the library down, hereby overwriting the test to be deleted. If the selected test is not in the library, a TEST NOT IN LIBRARY message is typed before return to idle.

9. EXIT - Exit to SMM

EXIT will delete REPAFB from the monitor test list by making an end-of-the test stop and return to exit. There is no return to REPAFB after that, REPAFB must be reloaded. (During idle, REPAFB may be master cleared and restarted any time at initial address.)

The main purpose of EXIT is to restore the disk to SMM operation. This exit should be taken rather than a master clear/run.

10. FUNC - Do a Function to the Input Device

FUNC allows to function the input tape unit.

Example:

		<u>Explanation</u>
Request?	FUNC (CR)	Select function option
Function=	8 (CR)	Code 8=rewind
Request?		

Available function codes:

- 3 = Backspace record
- 6 = Advance file
- 7 = Backspace file
- 8 = Rewind
- C = Rewind unload

This applies only to 173X and 8000 tape. The driver overlay must be loaded for FUNC to operate.

11. LHXC - Load Hexadecimal Data in Core

LHXC allows entry of hexadecimal data into memory. It first asks for the address where the data is to be loaded. It then types the old contents and asks for the new data.

Example:

	<u>Explanation</u>
Request ? LHXC CR	Select LHXC option
Address = 4000 CR	Select core address
Old = 0000 New = 1234 CR	New data = 1234
Old = 0000 New = S CR	Terminate with S=stop
Request?	

An input of S instead of the new data will terminate the LHXC operation. Otherwise, the address will be incremented by one and again a request for new data will come up. An input of carriage return only instead of new data will leave the old data in that location and just update the address.

12. GOTO - Give Control to P

GOTO requests a memory address via the Comment device and gives control to that address.

13. COPP - Copy Fixed to Cartridge to Fixed Disk (Disk Types 5 and 6)

Requests option:

- 1 = Cartridge to fixed platter
- 2 = Fixed to cartridge platter
- 3 = Cartridge to fixed to cartridge platter
- 0 = Exit

Request start cylinder (right-justified), end cylinder, and disk unit number.

After copy returns to select another option, it types alert message before writing on fixed disk and before changing cartridge packs.

14. COPY - Copy Disk Packs

Copies disk pack from one drive to another drive. Requests start cylinder (right-justified), end cylinder, read, and write unit number. Types ALERT message before starting copy.

15. HELP - List Options, Error Codes, REPAFB Operations

This option can be selected any time REPAFB is in the Idle loop. The generated typeout/printouts explain basic REPAFB actions and operations.

16. WEOF - Write SMM/MSMS End of File on Disk

WEOF requests the hardware sector address and will write a double software end of file mark on that and the following sector. This can be used to properly terminate a library in case the NAM sector address string is broken.

17. LHXD - Load (Store) Hexadecimal Data on the Disk Pack

LHXD requests the sector address and the address within that sector (0-\$5F). The old data is reported, and new data is requested. The modified data is written on the disk after S is typed as a change terminator (see LHXC).

18. CPAR - Change REPAFB's Run Parameter

This entry will cause a parameter entry stop in the monitor to change REPAFB's run parameters. It may also be used to access the monitor parameters by setting the SKIP switch before CPAR typein.

19. CONS - Convert Software (MSOS) Sector Address to Hardware Sector Address (Cyl/Head/Sector)

This option is not legal for fixed platter addresses on cartridge disks.

20. CONH - Convert Hardware Sector Address (Cylinder/Head/Sector) to Software (MSOS) Sector Address

This option is not legal for fixed platter addresses on cartridge disks.

II. DESCRIPTION

REPAFB is part of the MSMS 17 17X4 Mass Storage Maintenance System. This is a subsystem of SMM 17 and runs under the 4.0 and above SMM versions.

REPAFB is a utility program designed to support the SMM 17 Edit routine on a disk-based system, making use of the advantages of these read/write random access devices.

This program basically allows two different operations. One operation is to update an existing test on the disk library, and the other is to replace an existing (or nonexisting) test by a new one. On the second operation, all SMM input devices except disks are utilized. During both operations the test parameters may be changed and/or hexadecimal corrections may be inserted.

During the library manipulation on the disk, the NAM block-sector reference between tests will be updated.

REPAFB is a run-alone test and requires a minimum of 8K of memory. The program will use track-size (600 words-hexadecimal) transfers.

At Build-Test-List time the WESD-field for the disk controller used by REPAFB must be valid or zero if prestored in the test. This address may be changed by manually changing location 6 in REPAFB.

Upon load, REPAFB types out the following header.

```
REPLACE/UPDATE 17X4 SMM DISK LIBRARY  
REPAFB VR 4.0 CP2F, I.A. = XXXX.  
XXXX = Initial address of REPAFB
```

After header typeout, REPAFB does the parameter entry stop if ST/JP bit 1 is selected.

A1/Q1 = Disk type/disk unit number

A2/Q2 = Scratch sector/check library flag

After parameter entry, REPAFB checks disk type. If it is not 4, 5, or 6, a return to parameter is done.

If the library check flag is zero, REPAFB checks the SMM and MSMS disk library.

If OK, REPAFB reports OK. REPAFB then ensures that the selected scratch address is outside of the library. If it is not, REPAFB returns to the parameter entry routine.

If disk type is 5 (1733-2 Cartridge Disk), REPAFB checks if the disk is a write-
enable by reading, complementing, rewriting, and restoring word \$5F in sector \$10.
If the disk is not write-enable, REPAFB reports this and allows for operator reaction.

REPAFD has the following programmed overlays:

<u>OVL NR</u>	<u>Purpose</u>
1	Paper tape loader device 1
2	Card loader device 2
3	Mag tape loader device 3
4	8000 tape loader device 8
5	Binary Quick Look generator
6	Copy cartridge disk platters
7	Copy disk packs
8	Help

III. EXAMPLES OF USAGE

Replace test 7A with a newer version on mag tape.

Example:

	<u>Comments</u>
Request? <u>LUIN</u> (CR)	Select input device
Device = <u>3</u> (CR)	3 = magtape
Driver in	Overlay loaded
EQ. address = <u>1381</u> (CR)	Address (WESD) for 1731
Unit NR = <u>0</u> (CR)	Tape unit number
Request? <u>REPL</u> (CR)	Select replace option
Test number = <u>7A</u> (CR)	Test to be replaced
Found test 7A	Test found on input device
Request?	All done

Replace test FB with a newer version from the 430 Card Reader. Change parameters on the new test.

Request ? <u>PARA</u> (CR)	Select parameter change
Request ? <u>LUI</u> N (CR)	Select new input
Device = <u>2</u> (CR)	Device 2 = 430 card reader
Driver in	Overlay loaded
EQ. address = <u>5A1</u> (CR)	Equipment address for 430
Request ? <u>REPL</u> (CR)	Select replace option
Test number = <u>FB</u> (CR)	Test FB to be replaced
Found test FB	Test is found in the card library
EQ address = <u>0</u> (CR)	Change the prestored EQ
Parameter change A/Q	Address for test FB (0=NO)
FB31/0000 ? <u>CR</u>	Tests parameters
0004/0000 ? (CR)	
4000/0000 ? <u>5000</u> (CR)	Change scratch sector to 5000
Request ?	All done

Update test FC on the disk library. Change location 455 in test FC to an NOP instruction and change params for memory size.

Request ? <u>PARA</u> (CR)	Select parameter change
Request ? <u>CORR</u> (CR)	Select correction stop
Request ? <u>UPDA</u> (CR)	Select update run
Test number = <u>FC</u> (CR)	For test FC
Found test FC	Located FC on disk library
Length increase = <u>0</u> (CR)	No length increase (zero)
EQ. address = <u>181</u> (CR)	Prestored EQ. address
Parameter change A/Q	For test FC
FC21/0000 ? (CR)	Leave as is
7FFF/XXXX ? <u>5FFF</u> (CR)	Change to 24K memory

Correction stop, A = address/Q=data

Correction stop

00FC/0000 ? <u>455/0B00</u> (CR)	At end of test. Clear
0456/0000 ? <u>0/0</u> (CR)	A/Q to terminate
Request ?	All done

NOTE

Parameter change and correction stops work similar to the Edit routine except for the ID/length change stop.

If the correction stop data is to be biased on load, set bit 15 in the address field. In the previous example, the change would be 8455/0B00 if the value was to be biased on later load.

Normally, do not add any BIAS to the listing address when doing a hexadecimal correction.

Special attention has to be given when the test to be corrected has overlays.

Do a hexadecimal correction to test FB's second overlay, the card reader RBD driver. Change location ACE to an NOP (load from cards).

Request ? <u>CORR</u> (CR)	Set correction flag
Request ? <u>REPL</u> (CR)	Replace
Test Number? <u>FB</u> (CR)	Test FB
Found test FB	Located in card reader
Length increase = <u>0</u> (CR)	No length change
Correction stop, A = address/ Q = data 00FB/0000 ? <u>0/0</u> (CR)	Correction stop for The basic test, no
Correction stop, A = address/ Q = data 00FB/0001 ? <u>0/0</u> (CR)	Change Overlay 1, no change
Correction stop, A = address/ Q = data 00FB/0002 ? <u>ACE/B00</u> (CR)	Overlay 2, change Location ACE to 0B00
0ACF/0000 ? <u>8000/0</u> (CR)	No more changes, set bit 15 to terminate

NOTE

If there are no more changes to this and the future overlays (3 to 8 in this case), set the address to 8000 to eliminate future stops.

The first correction stop in Q register contains the overlay number. Zero means this is the basic test.

REPAFB will generate overlay blocks by itself; therefore, a 1700 assembler binary output may be taken as input deck for REPAFB even if the test has overlays.

1700 PUNCH A DECK

(PADA69 Test No. 69)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. This test runs on a 1728/430 Card Punch only.
2. This test requires 8K of core to execute.

B. LOADING PROCEDURE

1. This test operates as a stand-alone subtest under control of the SMM17 diagnostic monitor.
2. The calling sequence is that specified by SMM17.
3. The test can be restarted by Master Clearing, setting the P register to the tests initial address, and setting the RUN/STOP switch to RUN.

C. PARAMETERS

If a parameter stop is selected, the following test parameters may be altered.

1. First Stop

A1: 6951 ID word
Q1: Stop/Jump word

2. Second Stop

A2: \$0064 Hollerith card count
Q2: \$0064 25 card count, columns of 5 and A

3. Third Stop

A3: \$0064 52 card count, columns of A and 5
Q3: \$0064 6789 card count

4. Fourth Stop

A4: \$0064 Blank card count (1 to 255)
Q4: XXXX alarm interrupt line assignment

5. Fifth Stop

A5: Unused

Q5: \$0581 equipment code

All card count values must be positive nonzero values. A zero or negative value will force a card count of 1 (100 if blank cards). The prestored count is 100. Maximum count is \$7FFF (32767). Blank card count is limited to \$FF (255).

II. MESSAGES

A. INITIAL MESSAGE

PUNCH-A-DECK

TEST 69 IA-XXXX

THIS ROUTINE PUNCHES THE 1726/405 TEST DECK.

B. ERROR MESSAGES

1. First Stop

A1: \$6938 ID word

Q1: \$XXXX Stop/Jump word

2. Second Stop

A2: \$01XX XX=01 Internal reject
=02 External reject
=03 Punch alarm

Q2: Error routine return address

3. Third Stop

For A2 = 0101 A3 = A register

Q3 = Q register

For A2 = 0103 A3 = Status 1

Q3 = Status 2

C. END OF TEST

A1 = \$6924 A1 = STJP

A2 = Pass count

III. DESCRIPTION

This test contains a core image of all card types to be punched, along with the card count of each that can be changed. Images of the random cards are not stored. Card image of random cards is built just like test CR2 builds an image.

Punch-a-Deck contains the contents of columns 10, 12, 20, and 22 from a group of random cards generated in the past.

Each card to be punched is represented by a card count and an 80-column image of that card or a special character (FFFF) and a 4-column image (columns 10, 12, 20, and 22).



CARD PUNCH VERIFY ROUTINE
(CPVAF0 Test No. F0)

I. INTRODUCTION

This routine will read and verify the cards punched during the execution of test 88, CPC088.

II. REQUIREMENTS

A. HARDWARE

1. 17x4 CPU with 8K of memory
2. Standard SMM17 card reader input device:

1726/405

1728/430

1729-2

1729-3

B. SOFTWARE

The last card of the punched deck (a 5, 6, 7, 8, and 9 level punches in column 1) must be duplicated. Place a blank card and this separator card at the end of the deck to terminate the verify program.

III. OPERATIONAL PROCEDURE

This test is a stand-alone test under SMM17, test number \$F0. The correct director 1 equipment address must be specified when the test is loaded.

PARAMETERS

There are no operator-supplied parameters for this test. The test is controlled by the contents of the first column of each card and by the separator cards generated by the punch test.

IV. OPERATOR COMMUNICATION

A. INITIAL MESSAGE

CARD PUNCH VERIFY UTILITY
IA = XXXX, FC = XXXX, CP = 0C

B. ERROR MESSAGES

Error messages are presented in the following format.

A1 = ID word

Q1 = Stop/jump word

A2 = Pattern type/error code

Q2 = Return address

A3 = Actual data or sequence number

Q3 = Expected data or sequence number

A4 = Column count if data error

Q4 = Total errors

Error Codes:

01 = Sequence error

02 = Data miscompare

03 = Pattern field not zero

04 = Pattern field is zero

4. SELECTIVE JUMP and SELECTIVE STOP key setting

- a. SELECTIVE STOP must be set.
- b. Use SELECTIVE SKIP switch as outlined in the SMM17 Reference Manual.

D. MESSAGES

1. Typeouts or Alarms

a. Normal Program Typeouts

- 1) Command test identification at start of test
COM001 COMMAND TEST
IA = XXXX, FC = XX CP2F, VER. 4.0
- 2) End of Command test

A	Q	A	Q
0104	STOP	PASS NO.	RETURN ADDRESS

b. Error Alarms

- 1) The following is typed out:
 - a) Identification word
 - b) Stop/Jump parameter
 - c) Section and error number
 - d) Return address
 - e) Correct data or simulated results
 - f) Incorrect data
- 2) See each individual section write up for its specific error alarm.

c. Error Codes - See individual section write up.

E. ERROR STOPS

- 1. The same data typeout for error display will be displayed in A and Q.
- 2. See each individual section write in for its specific error.

II. DESCRIPTION

A. PROGRAM DESCRIPTION

This test checks internal 1700 instructions with fixed and random operands. The instructions are tested beginning with the simpler ones and ending with the more complex.

The check of actual results is made by either a simulation or a comparison to anticipated results. The writeups of the individual tests follow and are in the same order as they appear in the program.

Section 0 REGISTER BIT CHECK

Error Number 1

A register bits would not clear to zero. A3 contains all bits which would not clear.

Error Number 2

Q register bits would not clear to zero. A3 contains all bits which would not clear.

Error Number 3

M register bits would not clear to zero. A3 contains all bits which would not clear.

Error Number 4

I register bits would not clear to zero. A3 contains all bits which would not clear.

Error Number 5

A register bits would not set to one. A3 contains the bit which would not set.

Error Number 6

Q register bits would not set to one. A3 contains the bit which would not set.

Error Number 7

M register bits would not set to one. A3 contains the bit which would not set.

Error Number 8

I register bits would not set to one. A3 contains the bit which would not set.

Section 1 JUMP

Error Number 0

Test if Jump instruction is executed and if destination is correct. If destination is ± 1 instruction, a Stop should occur.

Error Display: Standard error format not used.

A = 108 and OVERFLOW light will not be lit.

Section 2 RETURN JUMP

Error Number 1

Incorrect address was stored in return address.

Error Display: A3 = correct address, Q3 = incorrect address

Section 3 LOAD A

Error Number 1

An operand was loaded into A. The same operand and the contents of A are then compared via an EXCLUSIVE OR.

Error Display: A3 = correct data, Q3 = incorrect data

Section 4 STORE A

Error Number 1

An operand was loaded into A and stored. A was then compared with the operand stored via an EXCLUSIVE OR.

Section 5 LOAD Q

Error Number 1

An operand was loaded into Q and transferred to A. The same operand and the contents of Q were then compared via EXCLUSIVE OR.

Error Display: A3 = correct data, Q3 = incorrect data

Section 6 STORE Q

Error Number 1

An operand was loaded into Q and A. Q was then stored and the stored operand was compared to (A).

Error Display: A3 = correct data, Q3 = incorrect data

Section 7 SKIP IF A = +0

Error Number 1

No Skip occurred when positive zero was loaded into A and then tested.

Error Display: No data display

Error Number 2

A Skip occurred when negative zero was tested in A.

Error Display: No data display

Error Number 3

Positive zero Skip did not occur for contents of A, but when tested in Q, a Skip occurred.

Error Display: A3 = contents of A, Q3 = contents of Q

Error Number 4

Positive zero Skip occurred when tested in A, but did not occur when tested in Q.

Error Display: A3 = contents of A, Q3 = contents of Q

Section 8 SKIP IF A ≠ 0

Error Number 1

No Skip occurred when negative zero was loaded into A and tested.

Error Display: No data display

Error Number 2

A Skip occurred when zero was loaded into A and tested.

Error Display: No data display

Error Number 3

A was tested with SAN instruction and no Skip took place, but when tested by SAZ instruction it was found to be non-zero.

Error Display: A3 = contents of A

Error Number 4

A was tested with SAN instruction and a Skip took place, but when tested by SAZ instruction it was found to be zero.

Error Display: A3 = contents of A

Section 9 SKIP IF A = + (positive)

Error Number 1

When A was loaded with an operand and tested for positive value a Skip did not occur. When complemented and tested again the Skip did not occur.

Error Display: A3 = operand tested first, Q3 = complement of operand

Error Number 2

When A was loaded with an operand and tested for positive value, a Skip occurred. When complemented and tested again, the Skip occurred.

Error Display: A3 = operand tested first, Q3 = complement of operand

Section A SKIP IF A = - (negative)

Error Number 1

When A was loaded with an operand and tested for negative value, a Skip did not occur. Then A was tested for positive and found to be negative.

Error Display: A3 = operand tested

Error Number 2

When A was loaded with an operand and tested for negative value, a Skip occurred. Then A was tested for positive and found to be positive.

Error Display: A3 = operand tested

Section B SKIP IF Q = +0

Error Number 1

Q and A were loaded with the same operand, Q was then tested with SQZ, and no Skip occurred. But when A was tested with SAZ, a Skip occurred.

Error Display: A3 = operand tested

Error Number 2

Q and A were loaded with the same operand, Q was then tested with SQZ, and a Skip occurred. But when A was tested with SAZ, a Skip did not occur.

Error Display: A3 = operand tested

Section C SKIP IF Q \neq + 0

Error Number 1

Q and A were loaded with the same operand; Q was then tested with SQN and no Skip occurred. But when A was tested with SAN, a Skip occurred.

Error Display: A3 = operand tested

Error Number 2

Q and A were loaded with the same operand, Q was then tested with SQN and a Skip occurred. But when A was tested with SAN, a Skip did not occur.

Error Display: A3 = operand tested

Section D SKIP IF Q = + (positive)

Error Number 1

Q and A were loaded with the same operand, Q was then tested with SQP, and no Skip occurred. But when A was tested with SAP, a Skip occurred.

Error Display: A3 = operand tested

Error Number 2

Q and A were loaded with the same operand, Q was then tested with SQP, and a Skip occurred. But when A was tested with SAP, a Skip did not occur.

Error Display: A3 = operand tested

Section E SKIP IF Q = - (negative)

Error Number 1

Q and A were loaded with the same operand, Q was then tested with SQM, and no Skip occurred. But when A was tested with SAM, a Skip occurred.

Error Display: A3 = operand tested

Error Number 2

Q and A were loaded with the same operand; Q was then tested with SQM and a Skip occurred. But when A was tested with SAM, a Skip did not occur.

Error Display: A3 = operand tested

Section F A LEFT SHIFT AND TEST FOR OVERFLOW

An operand was loaded into A and shifted left a predetermined number of times. A check is made for overflow after shifting.

Error Number 1

The end results of the shifting did not equal the starting value.

Error Display: A3 = correct data, Q3 = incorrect data

Error Number 2

Overflow occurred.

Error Display: A3 = operand, Q3 = not applicable

Section 10 Q LEFT SHIFT AND TEST FOR OVERFLOW

Error Number and displays are the same as Section F.

Section 11 A RIGHT SHIFT

Error Number 1

A was loaded with 4000, then shifted to the right once and compared to the known value. This is repeated 15 times for each iteration of the test section.

Error Display: A3 = correct data, Q3 = incorrect data

Section 12 A RIGHT SHIFT

Error Number 1

Same as Section 11 with the exception that A = 800 instead of 4000 when starting the shifting.

Section 13 A RIGHT, Q RIGHT

Error Number 1

A and Q were loaded with the same operand, then shifted and compared.

Error Display: A3 = content of shifted A, Q3 = content of shifted Q

Section 14 LONG LEFT SHIFT AND CHECK FOR OVERFLOW

A and Q were loaded with operands and shifted 1, 2, . . . 31 places then compared to the original value loaded into A and Q and a check is made for overflow.

Error Number 1

Shifted results did not equal original values in A and Q.

Error Display: A3 = original contents of A, Q3 = original contents of Q
A4 = shifted contents of A, Q4 = shifted contents of Q

Error Number 2

Overflow occurred

Error Display: Inapplicable.

Section 15 LONG RIGHT SHIFT

Error Number 1

A and Q were loaded with 8000 and 4000, then shifted one place and compared to the known result. The shifting continues until the bits have traversed the registers.

Error Display: A3 = correct contents of A, Q3 = correct contents of Q
A4 = incorrect contents of A, Q4 = incorrect contents of Q

Error Number 2

A and Q were loaded with 8000, then shifted one place and compared to the known results. The shifting continues until the bits have traversed the registers.

Error Display: Same as Error Number 1

Section 16 ENTER A

Error Number 1

A known quantity is masked into an ENA instruction; then the instruction is executed and compared to the known quantity.

Error Display: A3 = correct data, Q3 = incorrect data

Section 17 ENTER Q

Error Number 1

The same as Section 16 with the exception that Q is used instead of A.

Section 18 AND WITH A

Error Number 1

A is set to all one's; then an AND is executed with a known operand. The contents of A and the operand are then compared.

Error Display: A3 = correct data, Q3 = incorrect data

Section 19 ADD TO A

Error Number 1

The ADD instruction was simulated by shifting and bit comparison. The simulated results were then compared to the instruction results.

Error Display: A3 = simulated results, Q3 = instruction results

Section 1A EXCLUSIVE OR WITH A

Error Number 1

The EOR instruction was simulated by shifting and bit comparison. Then the simulated results were compared to the instruction results.

Error Display: A3 = simulated results, Q3 = instruction results

Section 1B ADD TO Q

Error Number 1

An ADD instruction was used to simulate the ADQ. Then the ADQ instruction is executed and compared to the simulated results.

Error Display: A3 = simulated results, Q3 = instruction results

Section 1C INCREASE A

Error Number 1

An ADD instruction was used to simulate the INA instruction. Then the INA instruction was executed and compared to the simulated results.

Error Display: A3 = simulated results, Q3 = instruction results

Section 1D INCREASE Q

Error Number 1

An INA instruction was used to simulate the INQ instruction. Then the INQ instruction was executed and compared to the simulated results.

Error Display: A3 = simulated results, Q3 = instruction results

Section 1E REPLACE ADD ONE IN STORAGE AND CHECK FOR OVERFLOW

Error Number 1

An ADD instruction was used to simulate the RAO instruction. Then RAO instruction was executed and compared to the simulated results.

Error Display: A3 = simulated results, Q3 = instruction results

Error Number 2

Overflow occurred

Error Display: A3 = simulated results, Q3 = instruction results

Section 1F SUBTRACT

Error Number 1

An ADD instruction with a complemented operand is used to simulate the SUB instruction. Then the SUB instruction is executed and compared to the simulated results.

Error Display: A3 = simulated results, Q3 = instruction results

Section 20 SKIP ON OVERFLOW, SKIP ON NO OVERFLOW

Error Number 1

An overflow was forced (+ to -), but when tested with SOV the Skip did not occur.

Error Display: No data display

Error Number 2

The execution of a SOV instruction failed to clear an overflow.

Error Display: No data display

Error Number 3

An overflow was forced (- to +), but when tested with SOV the Skip did not occur.

Error Display: No data display

Error Number 4

An overflow was forced (+ to -), but when tested with SNO the Skip occurred.

Error Display: No data display

Error Number 5

After executing SNO instruction, another one was executed and no Skip occurred.

Error Display: No data display

Error Number 6

An overflow was forced (- to +), but when tested with SNO the Skip occurred.

Error Display: No data display

Section 21 MULTIPLY INTEGER

Error Number 1

The simulation of the MUI instruction was accomplished by adding and shifting.

Error Display: A3 = simulated results, Q3 = simulated results (most significant)

A4 = instruction results, Q4 = instruction results (most significant)

Section 22 DIVIDE INTEGER

Error Number 1

The simulation of the DIV instruction was accomplished by subtracting and shifting.

Error Display: A3 = simulated quotient, Q3 = simulated remainder

A4 = instruction quotient, Q4 = instruction remainder

Error Number 2

An overflow occurred but the OVERFLOW indicator was not set.

Error Display: Same as Error Number 1

Error Number 3

No overflow occurred but the OVERFLOW indicator was set.

Error Display: Same as Error Number 1

Section 23 STORE PARITY TO A

Errors Number 1 and 2

The SPA instruction is simulated and the results of the simulation are compared to the instruction results.

Error Display: A3 = parity of simulation, Q3 = parity of instruction
A4 = contents of A when parity was determined
Q4 = data stored by SPA

Section 24 SET TO ONES A, Q

Error Number 1

The SET instruction is executed and then A and Q are checked for all bits being set.

Error Display: A3 = correct data = A and Q, Q3 = contents of A after
SET execution
A4 = contents of Q after SET execution

Section 25 CLEAR TO ZERO A, Q

Error Number 1

The CLR instruction is executed and then checked for all bits being cleared in AQ.

Error Display: A3 = correct data = A and Q, Q3 = contents of A after CLR
execution
A4 = contents of Q after CLR execution

Section 26 TRANSFER A TO A, Q

Error Number 1

The TRA instruction is executed and then the registers are checked for equality with the original contents of A.

Error Display: A3 = correct data, Q3 = contents of A after TRA execution
A4 = contents of Q after TRA execution

Section 27 TRANSFER Q TO A, Q

Error Number 1

The TRQ instruction is executed and then the destination register is checked for equality with the original contents of Q.

Error Display: A3 = correct data, Q3 = contents of Q after TRQ execution

Section 28 TRANSFER COMPLEMENT A TO A, Q

Error Number 1

The TCA instruction is simulated by complementing the operand with an EOR instruction. The results of the simulation are then compared to the destination registers of the TCA instruction.

Error Display: A3 = simulated data, Q3 = contents of A after TCA
A4 = contents of Q after TCA

Section 29 TRANSFER COMPLEMENT Q TO A, Q

Error Number 1

The TCQ instruction is simulated by complementing the operand with an EOR instruction. The results of the simulation is then compared to the destination registers of the TCQ instruction.

Error Display: A3 = simulated results, Q3 = contents of A after TCQ
A4 = contents of Q after TCQ execution

Section 2A TRANSFER THE ARITHMETIC SUM A, Q TO A, Q

Error Number 1

The ADD instruction is used to simulate the AAQ instruction.

Error Display: A3 = simulated results, Q3 = contents of A after AAQ execution
A4 = contents of Q after AAQ execution

Section 2B TRANSFER EXCLUSIVE OR A, Q TO A, Q

Error Number 1

The EOR instruction is used to simulate the EAQ instruction.

Error Display: A3 = simulated results, Q3 = contents of A after EAQ execution
A4 = contents of Q after EAQ execution

Section 2C TRANSFER THE LOGICAL PRODUCT OF A, Q, TO A, Q

Error Number 1

The AND instruction is used to simulate the LAQ instruction.

Error Display: A3 = simulated results, Q3 = contents of A after LAQ execution
A4 = contents of Q after LAQ execution

Section 2D TRANSFER COMPLEMENT LOGICAL PRODUCT A, Q, TO A, Q

Error Number 1

The AND and FOR instructions are used to simulate a CAQ instruction.

Error Display: A3 = simulated results, Q3 = contents of A after CAQ execution
A4 = contents of Q after CAQ execution

Section 2E SET TO ONES - M

Error Number 1

The SET instruction is executed and then M is checked for all bits being set.

Error Display: A3 = simulated results, Q3 = contents of M after SET execution

Section 2F TRANSFER A TO M

Error Number 1

Simulation accomplished the same as Section 26.

Error Display: A3 = correct data, Q3 = contents of M after TRA

Section 30 CLEAR TO ZERO - M

Error Number 1

The CLR instruction is executed and then M is checked for all bits to be cleared.

Error Display: A3 = correct data, Q3 = contents of M after SET

Section 31 TRANSFER Q TO M

Error Number 1

Simulation accomplished the same as Section 27.

Error Display: A3 = correct data, Q3 = contents of M after TRQ

Section 32 TRANSFER COMPLEMENT A TO M

Error Number 1

Simulation accomplished the same as Section 28.

Error Display: A3 = correct data, Q3 = contents of M after TCA

Section 33 TRANSFER COMPLEMENT Q TO M

Error Number 1

Simulation accomplished the same as Section 29.

Error Display: A3 = correct data, Q3 = contents of M after TCQ

Section 34 TRANSFER ARITHMETIC SUM A, Q TO M

Error Number 1

Simulation accomplished the same as Section 2A.

Error Display: A3 = simulated results, Q3 = contents of M after AAQ

Section 35 TRANSFER EXCLUSIVE OR A, Q TO M

Error Number 1

Simulation accomplished the same as Section 2B.

Error Display: A3 = simulated results, Q3 = contents of M after EAQ

Section 36 TRANSFER THE LOGICAL PRODUCT OF A, Q TO M

Error Number 1

Simulation accomplished the same as Section 2C.

Error Display: A3 = simulated results, Q3 = contents of M after LAQ

Section 37 TRANSFER COMPLEMENT LOGICAL PRODUCT A, Q TO M

Error Number 1

Simulation accomplished the same as Section 2D.

Error Display: A3 = simulated results, Q3 = contents of M after CAQ

Section 38 TRANSFER M TO A, Q, M

Error Number 1

The TRM instruction is executed and then the registers are checked for equality with original contents of M.

Error Display: A3 = simulated results, Q3 = contents of A after TRM
A4 = contents of Q after TRM, Q4 = contents of M after TRM

Section 3A TRANSFER COMPLEMENT M TO A, Q, M

Error Number 1

The TCM instruction is simulated by complementing the operand with an EOR instruction.

Error Display: A3 = simulated results, Q3 = contents of A after TCM
A4 = contents of Q after TCM, Q4 = contents of M after TCM

Section 3B TRANSFER COMPLEMENT Q + M TO A, Q, M

Error Number 1

The TCB instruction is simulated by using AND and EOR instructions.

Error Display: A3 = simulated results, Q3 = contents of A after TCB
A4 = contents of Q after TCB, Q4 = contents of M after TCB

Section 3C TRANSFER ARITHMETIC SUM A, M TO A, Q, M

Error Number 1

The AAM instruction is simulated by using the ADD instruction.

Error Display: A3 = simulated results, Q3 = contents of A after AAM
A4 = contents of Q after AAM, Q4 = contents of M after AAM

Section 3D TRANSFER ARITHMETIC SUM A, Q + M TO A, Q, M

Error Number 1

The AAB instruction is simulated by using ADD, EOR, and AND instructions.

Error Display: A3 = simulated results, Q3 = contents of A after AAM
A4 = contents of A after AAB, Q4 = contents of M after AAB.

Section 3E TRANSFER EXCLUSIVE OR A, M TO A, Q, M

Error Number 1

The EAM instruction is simulated by using the ROR instruction.

Error Display: A3 = simulated results, Q3 = contents of A after EAM
A4 = contents of Q after EAM, Q4 = contents of M after EAM

Section 3F TRANSFER EXCLUSIVE OR A, Q + M TO A, Q, M

Error Number 1

The EAB instruction is simulated by using the AND, ADD, and EOR instruction.

Error Display: A3 = simulated results, Q3 = contents of A after EAB
A4 = contents of Q after EAB, Q4 = contents of M after EAB

Section 40 TRANSFER LOGICAL PRODUCT A, M TO A, Q, M

Error Number 1

The LAM instruction is simulated by using the AND instruction.

Error Display: A3 = simulated results, Q3 = contents of A after LAM
A4 = contents of Q after LAM, Q4 = contents of M after LAM

Section 41 TRANSFER LOGICAL PRODUCT A, Q + M TO A, Q, M

Error Number 1

The LAB instruction is simulated by using the AND, ADD, and EOR instructions.

Error Display: A3 = simulated results, Q3 = contents of M after LAB
A4 = contents of Q after LAB, Q4 = contents of M after LAB

Section 42 TRANSFER COMPLEMENT LOGICAL PRODUCT A, M TO A, Q, M

Error Number 1

The CAM instruction is simulated by using AND and EOR instructions.

Error Display: A3 = simulated results, Q3 = contents of A after CAM
A4 = contents of Q after CAM, Q4 = contents of M after CAM

Section 43 TRANSFER COMPLEMENT LOGICAL PRODUCT A, Q+M TO A, Q, M

Error Number 1

The CAB instruction is simulated by using AND, ADD, and EOR instructions.

Error Display: A3 = simulated results, Q3 = contents of A after CAB
A4 = contents of Q after CAB, Q4 = contents of A after CAB

Section 44 ADDRESSING MODE CHECK

<u>Error Number</u>	<u>Addressing Mode Which Failed</u>
0	Direct, Delta \neq 0
1	Direct, Delta = 0
2	Indirect, Delta \neq 0
3	Indirect, Delta = 0
4	Q Index, Delta \neq 0
5	Q Index, Delta = 0
6	I Index, Delta \neq 0
7	I Index, Delta = 0
8	Relative Indirect, Delta = 0
9	Indirect, Q Index, Delta \neq 0
A	Indirect, Q Index, Delta = 0

<u>Error Number</u>	<u>Address Mode Which Failed</u>
B	Q Index, I Index, Delta \neq 0
C	Q Index, I Index, Delta = 0
D	Relative, I Index, Delta \neq 0
E	Relative, I Index, Delta = 0
F	Indirect, I Index, Delta \neq 0
10	Indirect, Index, Delta = 0
11	Relative, Q Index Delta \neq 0
12	Relative, Q Index Delta = 0
13	Relative, Indirect, Q Index, Delta \neq 0
14	Relative, Indirect, Q Index, Delta = 0
15	Indirect, Q and I Index, Delta \neq 0
16	Indirect, Q and I Index, Delta = 0
17	Relative, Indirect, I Index, Delta \neq 0
18	Relative, Indirect, I Index, Delta = 0
19	Relative, Q and I Index, Delta \neq 0
1A	Relative, Q and I Index, Delta = 0
1B	Relative, Indirect, Q and I Index, Delta \neq 0
1C	Relative, Indirect, Q and I Index, Delta = 0
1D	Relative, Q Index, Delta = 0
1E	Immediate Operand or Relative, Negative Delta
1F	Two-word, Negative Operand, Indirect or Relative Positive Delta
20	Two-word Negative Operand or Two-word Positive Operand
21	Relative Negative Delta, I Index or Relative Positive Delta, Q Index

III. PHYSICAL REQUIREMENTS

- A. STORAGE REQUIREMENT - About 2500₁₀ locations.
- B. TIMING - 1 min. 25 sec.
- C. EQUIPMENT CONFIGURATION
 1. 1704 with a minimum of 4K of memory.
 2. A device for loading the program.

1700 MEMORY TEST
MEM014 Test No. 14
(CP = 23)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. Bit 9 of the A register on the second parameter stop must be set to test a specific area of core by entering the test block addresses in the third parameter stop.
2. The operator should never restart SMM while the Memory Test is testing bank zero, since SMM and the Memory Test have been moved to the last memory stack.
3. This test is valid in core memory only. Bits 2 and 3 of the SMM parameter word must specify the correct machine type.
4. This test cannot be loaded concurrently with MY1, MY2, or RPT.
5. This test cannot be loaded concurrently with the TTY input package, the line printer driver, or the MSB package; error 10 will occur.

B. LOADING PROCEDURE

1. The Memory Test operates under control of the 1700 System Maintenance Monitor (SMM17).
2. The calling sequence is that specified by SMM17.
3. The test automatically relocates itself adjacent to the monitor before typing the initial address (IA).

C. PARAMETERS

1. Normal operation requires no parameters. Preselected test sections to be run are Sections 02 to 40.
2. To alter or review parameters, use the procedure explained in SMM17 Description.
 - a. FIRST STOP
 - A = Stop ID words = \$1441
 - Q = Tests Stop/Jump word

d. FOURTH STOP

A = 15	0
Number of Reads	
Q = 15	0
Not Used	

Enter in A the number of times each test cell in the Worst Pattern section will be read before the data is checked.

D. SELECTIVE SKIP and STOP SETTINGS

1. Stop - for information stops.
2. Skip - forces SMM Parameter stop.
3. Stop/Jump Parameter
 - a. Bits 0 to 10 - refer to SMM17 for standard usage.
 - b. Bit 11 - repeats all test sections selected in the same test block (fixed test block on the same memory stack).
 - c. Bit 12 - output errors only at end of each section pass with Error stop CXXX and Omit Error stops 8XXX and 4XXX.

E. MESSAGES

1. Typeouts
 - a. Normal Program Typeouts
 - 1) If additional tests are loaded in the test list, the following message will be printed:
MEM014 MEMORY TEST LOADED
WAITING FOR OTHER TESTS TO COMPLETE
 - 2) Memory Test identification at start of test.

MEM014	MEMORY TEST.
IA = ----, FC = --	CP23, VER. X.X
 - 3) "CLEAR PROTECT SWITCH. DISABLE AUTOLOAD PROTECT"
followed by programmed stop. Run when action complete.
(If key up, monitor error stop for Protect fault will occur instead.)
 - 4) End of Test.

A	Q	A	Q
1424	S/J	PASS NO.	RETURN ADDRESS
 - b. Error typeouts are in the standard SMM17 format.

2. Error Codes

- a. Error 1 - A parity error was sensed after the cell under test was read three times. If the pattern read equals the pattern stored in the test cell, the error occurred in the set up of the pattern and not in cell under test.
- b. Error 2 - The pattern of the third read does not equal the pattern of the second read.
- c. Error 3 - The pattern of the third read does not equal the pattern of the first read.
- d. Error 4 - The pattern read from the test cell does not equal the pattern stored in the cell.
- e. Error 0 - The pattern stored in the test cell of Section 01, single cell inspection, does not equal the pattern read out.

F. ERROR STOPS

Three different Memory Test error formats can be identified on the first stop by the number of stops designated in the identification word (see SMM17 Description, SMM/Operator Interface). A fourth error format is identified only with the single cell inspection section.

1. First Error in Group (Five Stops)

This error stop is in the Common Compare subroutine used by sections 02-40. This format displays the first error of a group of errors that have occurred in consecutive core locations.

a. FIRST STOP

A = 1458 Identification Word.
Q = XXX8 Stop/Jump parameter.

b. SECOND STOP

A = section number/error code
Q = return address of where the stop originated.

c. THIRD STOP

A = expected pattern (pattern stored in the test cell)
Q = actual pattern read out

d. FOURTH STOP

A = address of the failing test cell
Q = number of failing cells so far detected in this section

e. FIFTH STOP

A = common address bits of all failing cells so far detected in this section that are set ("1's")

Q = common address bits of all failing cells so far detected in this section that are reset ("0's")

2. Last Error of Group (Three Stops)

This error stop is in the Common Compare subroutine used by sections 02-40. This format will occur after the five-stop format (see Error Stops preceding) has occurred and two or more errors in consecutive core locations have been detected.

a. FIRST STOP

A = 1438 Identification Word.

Q = XXX8 Stop/Jump parameter.

b. SECOND STOP

A = section number/error code

Q = return address of where the stop originated.

c. THIRD STOP

A = address of the first test cell that was read correctly after two or more errors were detected in consecutive core locations.

Q = not used.

3. Section Error Summary (Four Stops)

This error stop is in the Common Compare subroutine used by sections 02-40. This format will occur at the end of sections 02-40 whenever any errors are detected within the test section. There may be two four-stop formats in the same section since some sections used one pattern on the first pass and its complement pattern on the second pass. (Example: All "1's" the first pass, all "0's" the second pass.)

a. FIRST STOP

A = 1448 Identification Word.

Q = XXX8 Stop/Jump parameter.

b. SECOND STOP

A = section number/error code

Q = return address of where the stop originated.

c. THIRD STOP

A = common address bits of all failing cells detected within this section that were set "0's".

Q = common address bits of all failing cells detected within this section that were reset ("0's").

d. FOURTH STOP

A = number of failing cells detected in this section.

Q = 0000 this is the first pass of the section.

Q = FFFF this is the second pass of the section using the complement pattern of the first.

Example of Previous Three Error Formats

The Memory Test detects parity errors in the test cells located at addresses 1031_{16} , 1032, 1033, and 1034 in the test block from locations 1000_{16} to 2000_{16} . The errors were found after the memory worst pattern in Section 08 was stored in the test block and then read back.

The Memory Test will display the following information:

1) After the first parity error at location 1031_{16} is detected:

A = 1458	Q = XXX8
A = 0301	Q = XXXX
A = 0000	Q = XXXX
A = 1031	Q = 0001
A = 1031	Q = EFCE

2) After location 1035_{16} is read out correctly:

A = 1438	Q = XXX8	(Three stops identifies last error of group.)
A = 0301	Q = XXXX	
A = 1035	Q = FFFF	(Address 1034 was the last failure in the group; 1035 was the first correct memory address.)

3) After this pass of the section is completed:

A = 1448	Q = XXX8	(Four stops identifies the section error summary.)
A = 0301	Q = XXXX	
A = 1030	Q = EFC8	
A = 0004	Q = 0000	

4. Single Cell Inspection Error

This error stop is only used on the Run-Alone Single Cell Inspection Test Section. If this test section has been selected, only the following error format can occur:

a. FIRST STOP

A = 1448 Identification Word.
Q = XXX8 Stop/Jump parameter.

b. SECOND STOP

A = section number 01/error code 00
Q = return address of where the stop originated.

c. THIRD STOP

A = expected pattern (pattern stored in test cell)
Q = actual pattern read out

d. FOURTH STOP

A = address of the test cell
Q = not used

5. Load Error

Error code 10 indicates that the LWA of the test has exceeded one memory bank, 4096 words.

II. DESCRIPTION

The test determines the memory size and checks memory in 1000_{16} word blocks (one stack). The first test block will be the last memory stack and testing will continue from higher to lower stacks until only stack zero is left to be tested. The test is moved to the last stack and stack zero is then tested. If the machine was only 4K of memory only the upper half of the stack will be tested.

The test consists of six main body sections and one optional section. The first two sections consist of a simple check of memory for the ability to hold all ones, all zeros and to hold its own cell address. The third section, worst pattern, will test memory for the ability to hold worst pattern and complement worst pattern.

Sections 10, 20, and 40 test memory to hold $AAAA_{16}$ 5555_{16} pattern, random pattern, and special pattern; the operator has the option to change the special pattern at parameter input time. In all sections except Section 4, the program protect plane is set equal to bit 15 of each word of the test block.

Section 2 - ALL "1's", ALL "0's"

A. Set the test block to all "1's".

- B. Each test cell is excited by reading it three times.
- C. Data is checked from read three.
- D. Complement the pattern in the test block and repeat steps B and C.

Section 4 - S REGISTER TEST

- A. Set each cell within the test block to its own address.
- B. Each test cell is excited by reading it three times.
- C. Data is checked from read three.

Section 8 - WORST PATTERN

- A. Set the test block to worst pattern.
- B. The pattern of the test cell is determined by the cell address and this pattern is complemented and stored in the test cell.
- C. The test cell is then excited in two ways:
 1. An all-zeros cell within the inhibit group of the test cell is found and then read X number of times, which may be set at parameter input time.
 2. The test cell is excited by reading it three times.
- D. Data is checked from read three.
- E. Recomplement pattern in test cell.
- F. Set the test block to the complement worst pattern and repeat steps B, C, D and E.

Section 10 - AAAA₁₆, 5555₁₆ PATTERNS

- A. Set the test block to AAAA₁₆, 5555₁₆.
- B. Each test cell is excited by reading it three times.
- C. Data is checked from read three.
- D. Complement the pattern in the test block and repeat steps B and C.

Section 20 - RANDOM PATTERN

- A. Generate eight random patterns.
- B. Set the test block to these patterns.
- C. Determine test cell pattern by the cell address.
- D. Each test cell is excited by reading it three times.
- E. Data is checked from read three.
- F. Complement the pattern in the test block and repeat steps C, D and E.

Section 40 - SPECIAL PATTERN

- A. Set the test block to the pattern entered on stop 2 in the Q register at parameter input time.
- B. Each test cell is excited by reading it three times.
- C. Data is checked from the read three.
- D. Complement the pattern in the test block and repeat steps B and C.

Section 1 - SINGLE CELL INSPECTION

This section is an optional section and may only be manually selected by the operator at parameter input time. At that time the address of the cell to be checked must be entered in the A register on the third stop.

- A. Store pattern in test cell.
- B. Read test cell.
- C. Check data.

NOTES

Testing specific core areas: At parameter input time the operator may select any specific core area by setting bit 9 in the A register on the second stop and entering the first and last word address + 1 in the A and Q registers on the third stop. If the block of core to be tested is greater than the core available, the block will be tested in two passes.

After the Memory Test has run all sections selected, it will determine which device SMM17 is loading from, and then move a hand bootstrap loader for that device to the end of core. The Memory Test will also clear all the memory protect bits and then protect only the hand loader. The starting address for all hand bootstrap loaders is \$XFE0, where X represents the last core bank.

Worst Pattern is defined as follows:

- A. Locations 00_{16} to 03_{16} are set to $AAAA_{16}$, 0000_{16} , 5555_{16} , $FFFF_{16}$ pattern which is repeated to location $3F_{16}$.
- B. Locations 40_{16} to 43_{16} are set to $FFFF_{16}$, 5555_{16} , 0000_{16} , $AAAA_{16}$ pattern which is repeated to location $7F_{16}$.
- C. Locations 80_{16} to BF_{16} are set equal to the complement of the first 40_{16} locations.
- D. Locations $C0_{16}$ to FF_{16} are set equal to the complement of location 40_{16} to $7F_{16}$.
- E. This pattern is repeated every 100_{16} locations throughout the stack. Hence, storage is set to worst pattern (see Figure 1).

III. PHYSICAL REQUIREMENTS

- A. MEMORY REQUIREMENT about 3800_{10} locations and all remaining memory.
- B. TIMING - 1 min. 30 sec.
- C. EQUIPMENT CONFIGURATION - 1704 Computer with 4K memory and device for loading Memory Test and SMM.

	00	01	02	03	04	3C	3D	3E	3F
000	AAAA	0000	5555	FFFF	AAAA	AAAA	0000	5555	FFFF
040	FFFF	5555	0000	AAAA	FFFF		5555	0000	AAAA
080	5555	FFFF	AAAA	0000	5555			AAAA	0000
0C0	0000	AAAA	FFFF	5555	0000				5555
100	AAAA	0000	5555	FFFF					
140	FFFF	5555	0000						
180	5555	FFFF							
1C0	0000								
F00	AAAA								FFFF
F40	FFFF	5555						0000	AAAA
F80	5555	FFFF	AAAA			FFFF	AAAA	0000	
FC0	0000	AAAA	FFFF	FFFF	5555	000	AAAA	FFFF	5555

Figure 1. Schematic Representation of 4096 Words
1700 Memory's Worst Pattern

1700 PROTECT TEST
(RPT009 Test No. 09)
(CP = 2F)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. A flashing OVERFLOW light is used as a signal to the operator.
 - a. When the test is being initialized, the flashing light means the operator should clear the STOP switch and set the PROGRAM PROTECT switch. A message will be typed out at this time if a teletype is available.
 - b. While the test is running, the flashing light means the operator should clear protect switch, set the STOP switch, clear the Q register, and run the computer. An End of Section, Error-message, or an End of Test stop will then occur.
 - c. If the Repeat Section bit is set, an End of Section stop will occur each time the test is repeated.
 - d. This test cannot be loaded concurrently with MY1, MY2, or MEM.
 - e. When restarting at initial address after RUN, the test will hang in a loop flashing the OVERFLOW light until the operator clears the PROTECT switch, sets the STOP switch, clears the Q register, and sets the computer to RUN to continue.
2. There is no signal at the end of test if the corresponding stop bit is not set in the test's Stop/Jump parameter. Thus, system automation is an operator option; he can set the bit and wait completion to set the STOP switch or use the ensuing load, test heading typeout, or "Build Test List" typeout (a function or RPT's test list position, see B.3) to key operator action.

B. LOADING PROCEDURE

1. The Protect Test operates under control of the 1700 System Maintenance Monitor (SMM17).
2. The calling sequence is that specified by SMM17.
3. The RPT acts as a test list load stringer (see SMM17, Description) and awaits the completion of those loaded before it, then runs alone.

C. PARAMETERS

1. Normal operation requires no parameters.
2. To alter or review parameters, use the procedure explained in SMM17 (see SMM17 Description.)

a. First Stop

A = 15 0

IDENTIFICATION WORD

Q = 15 0

STOP/JUMP PARAMETER

b. Second Stop

A = 15 0

SELECTABLE SECTIONS

Bit	Section
0	0
1	1
2	2

Q = Inapplicable

D. SELECTIVE SKIP AND STOP SETTINGS

1. SKIP switch is cleared.
2. STOP switch is cleared except when needed for End of Section or End of Test stops.

E. MESSAGES

Typeouts

1. Normal Program Typeouts.
 - a. If additional tests are loaded in the test list, the following message will be printed:

RPT009 RANDOM PROTECT TEST LOADED
WAITING FOR OTHER TESTS TO COMPLETE

- b. Test identification at start.

RPT009 RANDOM PROTECT TEST. CP2F, VER 4.0 IA = XXXX, FC = XX
--

(XXXX = Starting address of test and XX = frequency.)

- c. CLEAR STOP SWITCH, SET PROTECT SWITCH
(The test will hang up with the OVERFLOW light flashing until the PROTECT switch is set.)
 - d. End of Test

A	Q	A	Q
0924	ST/JP	PASS COUNT	RETURN ADDRESS

2. Error Stops

1st Stop	<u>A</u> ID	<u>Q</u> ST/JP		
2nd Stop	A <u>SECTION NUMBER/ ERROR CODE</u>	Q <u>RETURN ADDRESS</u>		
3rd Stop	A <u>FAILING INSTRUCTION</u>	Q <u>FAILING ADDRESS (SECTION 3)</u>		
4th Stop	A <table border="1" style="width: 100%; text-align: center;"> <tr> <td>NUMBER OR ERRORS</td> </tr> </table>	NUMBER OR ERRORS	Q <table border="1" style="width: 100%; text-align: center;"> <tr> <td>ZERO</td> </tr> </table>	ZERO
NUMBER OR ERRORS				
ZERO				

F. ERROR CODES

<u>Code</u>	<u>Description</u>
1	Memory parity error.
F	Protect fault failed to occur.

II. DESCRIPTION

This test checks the protect hardware of the 1704 Computer. The test protects all locations in memory, performs the Parameter stop, and then waits, with the OVERFLOW light flashing, until the PROTECT switch is set. The STOP switch is also to be cleared at this time. If it is not cleared, a stop will occur on each protect fault.

A. INITIALIZATION (INIT1)

1. Multiplex until other tests in core are finished.
2. Type out test heading, initial address (IA), and frequency count (FC).
3. Set protect bits of all memory locations.

4. Enter parameters if selected.
5. Type out message to operator:

CLEAR STOP SWITCH, SET PROGRAM PROTECT SWITCH

6. Flash OVERFLOW light and wait until PROTECT switch is set.
7. Begin execution.

B. SECTION ZERO (SECO)

This section checks for protect faults caused by the execution of non-protected EIN, IIN, SPB, CPB, EXI, and all mask register instructions.

1. Store section number.
2. Set instruction counter.
3. Clear Protect bit of execution address and address following.
4. Set mask for internal interrupt and enable interrupts.
5. Set counter. This is the number of times each instruction will be executed ($F000_{16}$).
6. Clear Error Counter.
7. Load instruction and store it in the execution address and in the error routine.
8. Execute instruction.
9. Check for a protect fault. If it did not occur, add one to error count.
10. Check loop counter. If not zero, repeat from item 8.
11. If loop count is zero, check error count. If not zero, make an error stop and output the failing instruction and the number of errors.
12. If no errors occurred, update instruction counter and loop to item 11 until all instructions have been set.
13. Return to control routine (CNT).

C. SECTION ONE (SEC1)

This section checks for protect faults caused by trying to execute a protected instruction following the execution of a non-protected instruction.

1. Store section number.
2. Set Protect bits in all memory locations.

3. Set instruction counter.
4. Clear Protect bit on execution address -1.
5. Set mask for internal interrupt and enable interrupts.
6. Set loop counter. The number of times each instruction will be executed ($F000_{16}$).
7. Load instruction and store it in execution address and in the error routine.
8. Execute a non-protected no-op instruction and then the instruction being checked.
9. Check for protect fault interrupt.
10. If protect fault did not occur, add one to error count.
11. Loop to item 7 until instruction has been executed $F000_{16}$ times.
12. Check for errors (protect fault did not occur); if present, make an error stop and display the failing instruction and the number of errors.
13. Loop to item 6 until all instructions have been checked.
14. Return to control routine (CNT).

D. SECTION TWO (SEC2)

This section generates protect faults by attempting to store in all available memory locations (protected) with a non-protected instruction.

1. Store section number.
2. Set protect bits in all available memory locations.
3. Set instruction counter.
4. Clear Protect bit of instruction execution address.
5. Set mask for internal interrupt and enable interrupts.
6. Pick up last word address of available memory.
7. Form the instruction to be executed and store it in the execution address and in the error routine.
8. Execute the instruction.
9. Check for protect fault. If it did not occur, make an error stop and display the failing instruction and the failing address.
10. If protect fault occurred, subtract one from the memory storage address.

11. Loop to item 8 until all available memory has been checked.
12. Update instruction counter. If not zero, loop to item 6.
13. Return to control routine (CNT).

E. CONTROL ROUTINE (CNT)

This routine checks for section selection, re-entry of parameters, End of Section and End of Test stop, and for Repeat Section or Repeat Test.

1. Check for re-entry of parameters.
2. Check for section 0 selection. If not selected, go to item 6.
3. Run section 0.
4. Check for End of Section stop.
5. Check for Repeat Section.
6. Check for section 1 selection. If not selected, go to item 10.
7. Run section 1.
8. Check for End of Section stop.
9. Check for Repeat Section.
10. Check for section 2 selection. If not selected, go to item 14.
11. Run section 2.
12. Check for End of Section stop.
13. Check for Repeat Section.
14. Check for End of Test stop.
15. Check for Repeat Test.
16. Return to SMM. If pass count is not zero, the test will be repeated.

III. PHYSICAL REQUIREMENTS

A. STORAGE REQUIREMENTS

Approximately 500 locations.

B. TIMING - 0 min. 45 sec.

C. EQUIPMENT CONFIGURATION

1. 17X4 Computer with 4K memory.
2. A device for loading the program.

1700 QUICK LOOK MEMORY AND PROTECT TEST
(QL2IFE Test No. FE)
(CP = 2F)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. QL2 must reside on the input medium as the test immediately after QL and only executes when loaded before the SMM Monitor loads.
2. When loaded as a test by SMM, QL2 issues the message:

QL2IFE IS NOT DESIGNED TO BE CALLED AS A TEST IT SHOULD BE LOADED BEFORE SMM INITIALIZES.

B. LOADING PROCEDURE

The test is loaded by the same loading procedure that the SMM Monitor is loaded.

C. PARAMETERS

QL2 does not have parameters.

D. MESSAGES

1. Normal Program Typeouts

- a. Typeout routine checks the SMM Stop/Jump word to see if bit 8 (omit typeouts) is set. When it is set, the typeout is bypassed.
- b. The following messages are used:

1) CLEAR PROGRAM PROTECT

This message is output at the end of QL2. The message will be repeated every 10 seconds as long as the CPU is running and PROGRAM PROTECT is set.

2) PROGRAM PROTECT NOT SET PROTECTED TEST NOT RUN

This message is output at the end of QL2 for information purposes only. The message is output only if the PP switch was not set when the protected test was due to start.

3) QL2IFE IS NOT DESIGNED TO BE CALLED AS A TEST. IT SHOULD BE LOADED BEFORE SMM INITIALIZED.

E. STOP PROCEDURE

1. QL2 issues a stop request when it wants to communicate with the operator. The stop request flashes the OVERFLOW light on and off. When the STOP switch is off, the routine continues to flash. When the STOP switch is on, the routine stops with A and Q equal to one of the following:
 - a. A = FE01 when the test is initiated. This stop will occur only if STOP and SKIP switches are set; it may appear as a fourth stop for QL. When the CPU stops, set PROGRAM PROTECT switch and clear the STOP switch to run protected.

Q = FFFF. This is the last address of memory to be tested. This parameter may be changed by entering the number of the last bank to be tested in bits 15, 14, 13, and 12 of the Q register. For a more permanent change, edit patch address \$20 to the desired bank number.
 - b. A = FE14 at the end of test. A flash routine flashes OVERFLOW on and off every second. This interface occurs after outputting the message CLEAR PROGRAM PROTECT. This interface times out in 10 seconds; however, an early exit may be accomplished by stopping and clearing Q. Upon exit, the program protect condition is checked to determine if the program protect has been cleared. The STOP switch and the preceding message will be repeated when the program protect has not been cleared.

Q = FFFF or prestored value. Q is the last memory address to be tested, prestored to \$FFFF. To reduce test memory size, Q entry format is BNNN, where B is the last block to test and NNN is replaced by FFF in the program (minimum is 4K). No attempt is made to save hand-loaded boot straps. To prescale this value to memory size, edit patch address \$20 to last available memory address for test purposes during library build.

NOTE

If entry is correct, AUTO LOAD PROTECT switch can be left on.

c. A = FEX8 when an error is encountered. The test may be run with the STOP switch cleared or set. Perform the following steps as appropriate:

1) STOP switch cleared

The OVERFLOW light will flash on and off at a rate of 10 per second when an error is encountered. Set the STOP switch and follow the procedure indicated for STOP switch set.

2) STOP switch set

The first stop will occur with the A register as indicated previously and the Q register containing the error code. After observing the error code, clear Q and press GO to display the next error A/Q pair. Press GO to display each remaining A/Q pair.

Error numbers and data meanings are defined in the individual section descriptions.

Q = Error code



II. DESCRIPTION

A. PROGRAM DESCRIPTION

The following tests check the CPU's ability to discretely address each memory location, use various modes of memory addressing, detect and process system interrupts, utilize system program protect, and hold certain data patterns in memory.

Verification of the results of these operations is done by comparison to anticipated results. The following writeups of the individual tests are in the same order as they appear in the program.

1. S Register Test

This test determines how much memory is available for testing purposes. The next operation sets the program protect bits for core. The SKIP switch is tested and if it is set, the program will halt with Q = FFFF and A = FE01. The operator now clears the STOP switch and sets the PROGRAM PROTECT switch if the protect section is run.

This test moves the preloaded SMM adjacent to itself to save it and does the same to the bootstrap at XFE0.

It then proceeds with each address from \$C00 to the end of memory which is written with its own address. After writing each cell, parity error is tested, and any write parity error is reported as error code 21. After writing all of memory, it is read back in; the parity error encountered while reading is reported as code 22. Actual cell contents are compared with its expected value; any difference is reported as error code 20.

Errors are reported in the following format:

A1 = FE28	Q1 = 002X
A2 = Memory contents	Q2 = Memory address

X = 0 Data mis-compare was detected

1 Write parity error detected

2 Read parity error detected

After checkout, the mode (32K or 65K) in which the processor is operating is determined.

2. Addressing Modes Test

This test uses the 16-bit and the 16-complement bit masks as operands or operand address modifiers to create one- and two-word load A instructions. The true operand addresses are calculated and used to create a two-word indirect address Load Q instruction. The results are compared and differences are reported as errors in the following format:

A1 = FE58	Q1 = 0010
A2 = Actual A	Q2 = Expected A (actual Q)
A3 = I index value	Q3 = Q index value (for load A)
A4 = Actual instruction	Q4 = OP address if delta - 0
A4 = Calculated OP address	Q5 = N/A

3. Protect Test

This test checks the PROGRAM PROTECT switch to see if protected operation can be undertaken. If the switch is not set, a flag is set and control is transferred to the interrupt section. If the switch is set, the following protect sections are executed.

- a. Protect section 0 checks for protect faults caused by the execution of non-protected EIN, IIN, SPB, CPB, EXI, and all mask register instructions. Errors are reported in the following format:

A1 = FE38	Q1 = 000D
A2 = Failing instruction	Q2 = Failing address
A3 = Number of errors	Q3 = N/A

- b. Protect section 1 generates protect faults executing a protected instruction following a non-protected instruction. Failure to get a program protect fault generates the following error:

A1 = FE38	Q1 = 000E
A2 = Failing instruction	Q2 = Failing address
A3 = Number of errors	Q3 = N/A

- c. Protect section 2 generates protect faults by trying to store in protected locations with non-protected storage modification instructions. Failure generates the following error code:

A1 = FE38	Q1 = 000F
A2 = Failing instruction	Q2 = Failing address
A3 = Number of errors	Q3 = N/A

4. Interrupt Forcing Test

This test uses the TTY interface to generate interrupts. It reports an error for unexpected interrupts and interrupts expected but not generated. If functioning of the controller is rejected, this test section is skipped without error. Errors are reported as follows:

A1 = FE18

Q1 = 000X

- X = 1 Interrupt with mask clear and interrupts disabled
- 2 Interrupt enabled but mask bit not set
- 3 Mask bit enabled but interrupts disabled
- 4 No interrupt when expected (interrupt enabled and mask bit set)

5. Memory Test

This test ensures that all cells of the memory can hold patterns 0000, FFFF, AAAA, and 5555. Error codes used to report memory errors are as follows:

A1 = FE38

Q1 = 002X

A2 = Actual memory content Q2 = Expected memory contents

A3 = Failing address Q3 = N/A

- X = 5 Data read did not compare to data written
- 6 Parity error was detected during memory write
- 7 Parity error was detected during memory read

A parity error may occur at other times. When this happens, the address may not be known. This will generate an error code as follows:

A1 = FE18

Q1 = 0030

6. After all test sections have been run, the bootstrap and the intermediate loader will be restored to their original positions. The program ensures that the PROTECT switch is set. If it is set, the message

CLEAR PROGRAM PROTECT

will be issued. The program will toggle the OVERFLOW light and attempt to stop. This is followed by a Q Zero sense for 10 seconds. This will be repeated until the PROGRAM PROTECT switch is cleared. The protect test run flag is tested and if the test was not run, the message PROGRAM PROTECT NOT SET PROTECTED TEST NOT RUN is issued. Control is then transferred to the intermediate loader which loads SMM.

1774 SYSTEM CONTROLLER COMMAND TEST
(CAR01B Test No. 1B)
(CP = 02)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. If the operator wishes to run section 5 or 6, he must call the system controller command test alone. The test overlays the normal interrupt processor with its own special processor in these sections.
2. Section 5 may be run only if the system controller has an AQ channel. A special interrupt generator, which is attached to the AQ channel, is used to generate interrupts on the lines corresponding to bits set in Q.
 - a. Special Interrupt Generator
The INT special cables consist of a male 61-pin connector and 15 female 3-pin interrupt jacks. Each Q bit (bit 1 to bit 15) is wired via 3-wire interrupt cable from the 61-pin connector to the 3-pin interrupt jacks. The special interrupt cable allows each Q bit (excluding bit 0) to generate the corresponding interrupt level.
3. If sections 5 or 6 are selected, the test must be executed to completion. This is to allow restoring of the normal interrupt processor for additional tests that are to be executed.

B. LOADING PROCEDURE

This test is called as test number 1B under SMM17. It may be run with other tests if (and only if) section 5 is not chosen and section 6 is not chosen.

C. PARAMETERS

1. Normal operation runs test sections 1 through 4 and 6 and requires no parameters.
2. To alter parameters, use the procedures outlined in SMM17. The identification word and Stop/Jump parameter are displayed on the first stop. On the second stop, set bit 0 to run section 1, etc., according to the table below:

<u>Bit in A</u>	<u>Test Section</u>
0	Section 1 -- LDA, Character mode, with character designation bit = 0.
1	Section 2 -- LDA, Character mode, with character designation bit = 1. Indexing used here.
2	Section 3 -- STA, Character mode, with character designation bit = 0. Indexing used here.

<u>Bit in A</u>	<u>Test Section</u>
3	Section 4 -- STA, Character mode, with character designation bit = 1.
4	Section 5 -- Character mode enable/disable with interrupts.
5	Section 6 -- powerfail interrupt on line 0 and auto restart test.

II. MESSAGES

A. NORMAL

1. Test identification at beginning of test:
CAR01B, SYSTEM CONTROLLER TEST
IA = XXXX FC = YY
2. End of test:

A	Q	A	Q
1B04	STOP	PASS NO.	RETURN ADDRESS
3. Message to operator to hook up the interrupt generator to the AQ channel (section 5):
INTP. GEN. ON AQ CHAN/
4. Message to operator that he may restore the AQ channel as it originally was (section 5):
RESTORE AQ CHAN.
5. Message to operator to set AUTO RESTART switch:
SET AUTO RESTART SWITCH
6. Message to operator to drop mainframe power, then bring it up again, checking Auto Restart operation:
DROP POWER, THEN RESTORE
7. Message to operator that he may clear the AUTO RESTART switch:
CLR AUTO RESTART SWITCH

B. ERROR

The following is typed out:

1. Identification word
2. Stop/Jump parameter
3. Section and error number
4. Return address
5. Actual results
6. Expected results

C. ERROR STOPS

<u>Error Code</u>	<u>Description</u>
01	LDA, character failed with character designator bit = 0. (A) = actual contents of A upon the LDA character instruction (Q) = expected contents of A upon the LDA character instruction
02	Same as 01 above, except the character designator bit here = 1.
03	STA, character failed with character designator bit = 0. (A) = actual data stored in memory by the STA character instruction (Q) = expected data stored in memory by the STA character instruction
04	Same as 03 above, except the character designator bit here = 1.
05	Level 0 - No interrupt occurred when power was dropped on the mainframe.
06	Level 0 - An interrupt occurred when power was dropped, but skip on parity error indicated a parity error caused the interrupt.
07	Level 0 - An interrupt occurred when power was dropped, but skip on protect fault indicated a protect fault caused the interrupt.

<u>Error Code</u>	<u>Description</u>
11 - 1F	Character mode was not properly re-enabled upon exit from interrupt state. Error 11 corresponds to interrupt state 1, 12 corresponds to state 2, etc.
21 - 2F	Character mode was not properly disabled when an interrupt occurred on the line indicated. Error 21 corresponds to state 1, 22 corresponds to state 2, etc.

III. DESCRIPTION

- A. Section 1 - LDA, Character mode is checked with the character designator bit = 0 (upper half of memory location referenced). The EOR instruction is used in testing proper operation.
- B. Section 2 - LDA, Character mode is checked with the character designator bit = 1. Indexing is used, with the index originally set = FFFF. The index is then decremented by twos, checking proper right shift and sign extension of the address in I.
- C. Section 3 - STA, Character mode is checked with the character designator bit = 0. Indexing is used, with the index originally set = FFFE. The index is then decremented by twos, checking proper right shift and sign extension of the address in I.
- D. Section 4 - STA, Character mode is checked with the character designator bit = 1.
- E. Section 5 - The automatic disabling of Character mode is checked upon an interrupt on each interrupt line (except 0). The automatic re-enabling of Character mode is checked upon execution of exit state for each interrupt state. This requires the operator to hook up the AQ interrupt generator, and, upon completion, to remove it.
- F. Section 6 - Test the auto restart feature, checking that the proper level of interrupt is generated when power is dropped on the mainframe. The operator is instructed to set the AUTO RESTART switch, turn off power and turn it back on again, and clear the AUTO RESTART switch. Unlike lines 1-F tested in section 5, this is tested only once (unless Repeat Condition is set).

NOTE

The system controller test must be called to run alone if section 5 is to be run. Section 5 can only be run:

- a) If the system controller has an AQ channel.
- b) If a special interrupt generator is available which is attached to the AQ channel, and which generates an interrupt on each line corresponding to a bit set in Q.

The system controller test must be called to run alone if section 6 is to be run.

IV. PHYSICAL REQUIREMENTS

A. STORAGE REQUIREMENTS

Approximately 900_{10} locations.

B. TIMING -

C. EQUIPMENT CONFIGURATION

1. 1774 SC with 4K memory.
2. A device for loading the program.

1700 MEMORY TEST
(MY2002 Test No. 02)
(CP = 2F)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. The test requires a machine with at least 8K (two stacks) of memory.
2. The operator never should restart SMM while the test is testing stack zero, since SMM and the test have been moved to the stack specified by the test parameters (stack one is standard).
3. To test 65K, SMM must have been loaded in 65K mode.
4. Bits 2 and 3 of the SMM parameter word must specify the correct machine type.
5. This test cannot be loaded concurrently with MY1, MEM, RPT, TTY input package, line printer driver, or the MBS package.

B. LOADING PROCEDURE

1. The test is loaded under SMM. When the test is given control to initialize, it does not return control to SMM until the test is completed. When control is returned to SMM, the next test is loaded and this test is destroyed.
2. The test automatically relocates itself adjacent to the monitor before before typing the initial address (IA).

C. PARAMETERS

1. Normal operation requires no parameters. Preselected test sections to be run are Sections 00 to 05.
2. To alter or review parameters, use the procedure explained in SMM17.

First Stop

A1 = ID Word (\$0231)

Q1 = Stop/Jump Parameter

Second Stop

A2 = Sections to be Run (Prestored as \$003F)

Bit 0 - Section 00 - Zeros

Bit 1 - Section 01 - Ones

Bit 2 - Section 02 - Address Test

Bit 3 - Section 03 - Parity Plane Test

Bit 4 - Section 04 - Worst Pattern

Bit 5 - Section 05 - First Pass Test

Bits 6 to 15 - Not used

Q2 = Stacks to be Tested (Bits 0-15)

Bit 0 - Stack 0	Bit 8 - Stack 8
Bit 1 - Stack 1	Bit 9 - Stack 9
Bit 2 - Stack 2	Bit 10 - Stack 10
Bit 3 - Stack 3	Bit 11 - Stack 11
Bit 4 - Stack 4	Bit 12 - Stack 12
Bit 5 - Stack 5	Bit 13 - Stack 13
Bit 6 - Stack 6	Bit 14 - Stack 14
Bit 7 - Stack 7	Bit 15 - Stack 15

(Q2 is determined by the test - All Available Core)

Third Stop

A3 = time between successive Memory References in Section 05 of test (prestored as \$0101) only one bit 0-3 should be set at a time (not used in 65K mode).

Bit 0 - 2.2 μ sec
Bit 1 - 3.3 μ sec
Bit 2 - 4.4 μ sec
Bit 3 - 5.5 μ sec
Bit 4 to 7 - Not used

A3 also designates the stack to which the test and SMM are moved when testing stack zero. Only one bit in bits 8-15 should be set at a time.

Bit 8 - Stack 1	Bit 12 - Stack 5
Bit 9 - Stack 2	Bit 13 - Stack 6
Bit 10 - Stack 3	Bit 14 - Stack 7
Bit 11 - Stack 4	Bit 15 - Not used

Q3 = X X X X

XXXX = Number of passes to be made with each pattern in Section 05 (prestored as \$0001).

D. SELECTIVE SKIP AND STOP SETTINGS

1. STOP switch - Normal SMM usage
2. SKIP switch - Brings up test Stop/Jump parameter stop
3. Stop/Jump Parameter
 - a. Bits 0 to 10 - Standard SMM17 usage
 - b. Bit 11 - Isolation Bit. Loops on test cell, storing expected value and making compare check.
 - c. Bit 12 - Exit Section Bit. Aborts the present test section, begins operation of the next selected section. The bit is cleared by the program.

E. PROGRAM PROTECT SWITCH

The PROGRAM PROTECT switch should remain in the OFF position.

F. MESSAGES

1. Typeouts

a. Normal Program Typeouts

- 1) If additional tests are loaded in the test list, the following message will be printed:

MY2002 MEMORY TEST LOADED
WAITING FOR OTHER TESTS TO COMPLETE

- 2) Identification at start of test.

MY2002, MEMORY TEST
IA = XXXX, FC=XX CP2F, VER. X.X

- 3) End of Section

First Stop

A = 0232 Ident Word
Q = XXXX Stop/Jump Word

Second Stop

A = XXXX Section
Q = XXXX Stack Just Tested

Third Stop

A = XXXX Number of Errors in this Section
Q = XXXX Return Address

- 4) End of Test

First Stop

A = 0224 Ident Word
Q = XXXX Stop/Jump Word

Second Stop

A = XXXX Pass Count
Q = XXXX Number of Errors in Test

b. Error Typeouts

Error typeouts are in the standard SMM17 format.

NOTE: All error stops consist of four stops.

First Stop

A = 0248 Identification Word
Q = XXXX Stop/Jump Parameter

Second Stop

A = XXYY Section/Error Code
Q = XXXX Return address for errors 01 through 08.
Location of error cell for errors 09 through 0B.

2. Error Codes

Error 1 - Compare error, no parity error was found while making the compare check.

Third Stop

A = XXXX Actual value
Q = XXXX Expected value

Fourth Stop

A = XXXX Location of error cell
Q = 0000 Not used

Error 2 - Parity error occurred on a load instruction, but no compare error was found.

Third Stop

A = XXXX Actual value
Q = XXXX Expected value

Fourth Stop

A = XXXX Location of error cell
Q = 0000 Not used

Error 3 - A parity error occurs while performing a load instruction. There is also a compare error.

Third Stop

A = XXXX Actual value
Q = XXXX Expected value

Fourth Stop

A = XXXX Location of error cell
Q = 0000 Not used

Error 4 - A parity error occurs while doing a store instruction whose effective address is in the test stack. This indicates that bits were dropped in this location before the store instruction may have been properly executed. If not, a compare error will occur later.

Third Stop

A = XXXX Value being stored
Q = XXXX Location of error cell

Fourth Stop

A = 0000 Not used
Q = 0000 Not used

Error 5 - A parity error occurs on a load instruction which references the stack being tested. The exact error cell could not be found.

Third Stop

A = XXXX Address of instruction being executed when error occurred or address of the check that revealed the error.
Q = 0000 Not used.

Fourth Stop

A = 0000 Not used
Q = 0000 Not used

Error 6 - A parity occurs in the Memory Stack which contains SMM and the test. This indicates that bits have been dropped within the test itself. If the operator attempts to continue the running of the test, the section in which the error occurred is aborted and the test attempts to run the next section.

Third Stop

A = XXXX Normally gives the address stored in trapped location \$100. But when stack zero is being tested, the level zero interrupt is not enabled. This stop gives the address of the parity check which revealed that there had been a parity error in the program sometime since the last parity error check.
Q = 0000 Not used

Fourth Stop

A = 0000 Not used
Q = 0000 Not used

Error 7 - Unexpected program protect fault.

Third Stop

A = XXXX Address of the location in the test stack which was being tested at the time of the error.
Q = 0000 Not used

Fourth Stop

A = 0000 Not used
Q = 0000 Not used

Error 9 - A parity error occurred while making a first pass compare check (Section 05), but no compare error is found.

Third Stop

A = XXXX Actual value of the previous pattern.
Q = XXXX Expected value of the previous pattern.

Fourth Stop

A = XXXX Actual value of the new pattern.
Q = XXXX Expected value of the new pattern

Error B - A compare error occurred while making a first pass compare check (Section 05), but no parity occurred.

Third Stop

A = XXXX Actual value of the previous pattern.
Q = XXXX Expected value of the previous pattern.

Fourth Stop

A = XXXX Actual value of the new pattern.
Q = XXXX Expected value of the new pattern.

Error 10 - Load error. The last-word address of the test has exceeded 4096 words.

II. DESCRIPTION

The test determines the size of memory and checks the memory in 1000_{16} word blocks (one stack). Stack one is the first to be tested. After the last stack has been tested the test and SMM are moved to the stack specified by the test parameters (normally stack one), and stack zero is then tested. The test consists of six standard sections 00 to 05. Testing is done by testing one stack with all sections before going on to the next stack.

SECTION 00 - ZEROS TEST

Tests the ability of the test stack to hold zero in every location.

- A. Fill the test stack with all zeros (0000).
- B. Read each location and check its contents.

SECTION 01 - ONES TEST

Tests the ability of the test stack to hold all ones in every location.

- A. Fill the test stack with all ones (FFFF).
- B. Read each location and check its contents.

SECTION 02 - ADDRESS TEST

Tests the S register and the ability of each cell, in the test stack, to hold its own address.

- A. Starting with the first location of the test stack and continuing through the last, fill each location of the test stack with its own address.
- B. Read each location and check its contents.
- C. Starting with the last location of the test stack, fill each location with its own address.
- D. Read each location and check its contents.

SECTION 03 - PARITY PLANE TEST

Tests the ability of each core, in the parity plane of the test stack, to hold zero and one while the rest of the plane holds worst pattern and while the rest of the plane holds complement worst pattern.

- A. Fill the test stack with complement worst pattern except for plane zero which is masked to zeros. This causes worst pattern to be generated in the parity plane.
- B. Disturb each location (X) and check its contents.
- C. Set bit zero of location (X), disturb the location, check its contents, then replace its original contents.
- D. Fill the test stack with worst pattern except for plane zero which is masked to zeros. This causes complement worst pattern to be stored in the parity plane.
- E. Repeat steps B and C.

NOTE: Disturbs are done by reading a combination of locations (Y1) and (Y2), where (Y1) is a location in the pattern which contains all zeros and is in the same inhibit group as (X), but is not on a common drive line with (X), and (Y2) has the same specifications as (Y1) except that it is on a common drive line with (X). When (Y1) is read, (X) are disturbed in the Write cycle. When (Y2) is read, (X) are disturbed in the Read cycle.

SECTION 04 - WORST PATTERN TEST

Tests the ability of each location in the test stack to hold worst pattern and complement worst pattern while the remainder of the test stack holds worst pattern. Then tests the ability of each location in the test stack to hold complement worst pattern and worst pattern while the remainder of the test stack holds complement worst pattern.

- A. Fill the test stack with worst pattern.
- B. Disturb each location (X) and check its contents.
- C. Replace each location (X) with the complement of its original contents, disturb the location, check its contents, then replace its original contents.
- D. Fill the test stack with complement worst pattern and repeat steps A, B, and C.

NOTE: Disturbs are accomplished in the same manner as those in Section 03.

SECTION 05 - FIRST PASS TEST

This test burns a pattern into the test stack, and then writes a new pattern into the stack only once, and makes a compare check.

- A. Make a number of passes in the test stack, storing the first pattern. On each pass the pattern is stored in each location 32 times. The number of passes is specified by the input parameters.
- B. On the last pass the 32nd reference of each location, stores the second pattern in that location.
- C. After this last pass, which stores a new pattern in the test stack, is completed, a compare check of each location is made.
- D. Now the second pattern is burned into the test stack, the last store of the last pass stores the third pattern and another compare check is made.
- E. This procedure continues until all of the patterns listed below have been used.

PATTERN 1 - All Zeros
PATTERN 2 - Worst Pattern
PATTERN 3 - All Ones
PATTERN 4 - Complement Worst Pattern
PATTERN 5 - All Zeros
PATTERN 6 - Complement Worst Pattern
PATTERN 7 - All Ones
PATTERN 8 - Worst Pattern
PATTERN 9 - All Zeros

NOTE: The time between successive references of the same location, may be varied from 2.2 microseconds to 5.5 microseconds by varying the input parameters.

III. PHYSICAL REQUIREMENTS

A. Storage Requirements

Approximately 2500₁₀ locations plus SMM monitor

B. Timing

Section 0 = 1 second per stack

Section 1 = 1 second per stack

Section 2 = 2 seconds per stack

Section 3 = 3 seconds per stack

Section 4 = 3 seconds per stack

Section 5 = 20 seconds per stack

Total = 30 seconds per stack

C. Equipment Configuration

1700 Series Computer with 8K memory

1700 MEMORY TEST
(MY1012 Test No. 12)
(CP = 2F)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. The test requires a machine with at least 8K (two stacks) of memory.
2. The operator never should restart SMM while the test is testing stack zero, since SMM and the test have been moved to the stack specified by the test parameters (stack one is standard).
3. To test 65K SMM must have been loaded in 65K mode.
4. Bits 2 and 3 of the SMM parameter word must specify the correct machine type.
5. This test cannot be loaded concurrently with MY2, MEM, RPT, TTY input package, line printer driver, or the MBS package.

B. LOADING PROCEDURE

1. The test is loaded under SMM. When the test is given control to initialize, it does not return control to SMM until the test is completed. When control is returned to SMM, the next test is loaded and this test is destroyed.
2. The test automatically relocates itself adjacent to the monitor before typing the initial address (IA).

C. PARAMETERS

1. Normal operation requires no parameters. Preselected test sections to be run are Sections 00 and 01.
2. To alter or review parameters, use the procedure explained in SMM17.

First Stop

A1 = ID Word (\$1231).

Q1 = Stop/Jump Parameter

Second Stop

A2 = Sections to be Run (Prestored as \$0003)

Bit 0 - Section 00 - Worst Pattern with Multiple Indirect Referencing
Disturb

Bit 1 - Section 01 - Optimal Worst Pattern Test

Bit 2 - Section 2 - Program Protect Test

Q2 = Stacks to be Tested (Bits 0-15)

Bit 0 - Stack 0	Bit 8 - Stack 8
Bit 1 - Stack 1	Bit 9 - Stack 9
Bit 2 - Stack 2	Bit 10 - Stack 10
Bit 3 - Stack 3	Bit 11 - Stack 11
Bit 4 - Stack 4	Bit 12 - Stack 12
Bit 5 - Stack 5	Bit 13 - Stack 13
Bit 6 - Stack 6	Bit 14 - Stack 14
Bit 7 - Stack 7	Bit 15 - Stack 15

(Q2 is determined by test - All Available Core)

Third Stop

A3 designates the stack to which the test and SMM are moved when testing stack 0. Only one bit in bits 8-15 should be set at a time.

Bit 8 - Stack 1
Bit 9 - Stack 2
Bit 10 - Stack 3
Bit 11 - Stack 4
Bit 12 - Stack 5
Bit 13 - Stack 6
Bit 14 - Stack 7
Bit 15 - Not used

(A3 is prestored as \$0100) (Q3 is not used)

D. SELECTIVE SKIP AND STOP SETTINGS

1. STOP switch - Normal SMM usage
2. SKIP switch - Brings up test Stop/Jump parameter stop.
3. Stop/Jump Parameter
 - a. Bits 0 to 10 - Standard SMM17 usage
 - b. Bit 11 - Isolation Bit. Loops on test cell, storing expected value and making compare check.
 - c. Bit 12 - Exit Section Bit. Aborts the present test section, begins operation of the next selected section. The bit is cleared by the program.

E. PROGRAM PROTECT SWITCH

Normally the PROGRAM PROTECT switch should remain in the OFF position. If Section 02 is being run, the operator must set and clear the PROGRAM PROTECT switch according to the description Section 02.

F. MESSAGES

1. Typeouts

a. Normal Program Typeouts

- 1) If additional tests are loaded in the test list, the following message will be printed:

```
MY1012 MEMORY TEST LOADED
WAITING FOR OTHER TESTS TO COMPLETE
```

- 2) Identification at start of test.

```
MY1012, MEMORY TEST
IA = XXXX, FC=XX, CP2F, VER. X.X
```

- 3) End of Section

First Stop

```
A = 1232   Ident Word
Q = XXXX   Stop/Jump Word
```

Second Stop

```
A = XXXX   Section
Q = XXXX   Stack Just Tested
```

Third Stop

```
A = XXXX   Number of Errors in this Section
Q = XXXX   Return Address
```

- 4) End of Test

First Stop

```
A = 1224   Ident Word
Q = XXXX   Stop/Jump Word
```

Second Stop

```
A = XXXX   Pass Count
Q = XXXX   Number of Errors in Test
```

b. Error Typeouts

Error typeouts are in the standard SMM17 format.

NOTE: All error stops consist of four stops.

First Stop

A = 1248 Identification Word
Q = XXXX Stop/Jump Parameter

Second Stop

A = XXYX Section/Error Code
Q = XXXX Return address for errors 01 through 08.

2. Error Codes

Error 1 - Compare error, no parity error was found while making the compare check.

Third Stop

A = XXXX Actual value
Q = XXXX Expected value

Fourth Stop

A = XXXX Location of error cell
Q = 0000 Not used

Error 2 - Parity error occurred on a load instruction, but no compare error was found.

Third Stop

A = XXXX Actual value
Q = XXXX Expected value

Fourth Stop

A = XXXX Location of error cell
Q = 0000 Not used

Error 3 - A parity error occurs while performing a load instruction.
There is also a compare error.

Third Stop

A = XXXX Actual value
Q = XXXX Expected value

Fourth Stop

A = XXXX Location of error cell
Q = 0000 Not used

Error 4 - A parity error occurs while doing a store instruction whose effective address is in the test stack. This indicates that bits were dropped in this location before the store instruction may have been properly executed, if not a compare error will occur later.

Third Stop

A = XXXX Value being stored
Q = XXXX Location of error cell

Fourth Stop

A = 0000 Not used
Q = 0000 Not used

Error 5 - A parity error occurs on a load instruction which references the stack being tested. The exact error cell could not be found.

Third Stop

A = XXXX Address of instruction being executed when error occurred or address of the check that revealed the error.
Q = 0000 Not used

Fourth Stop

A = 0000 Not used

Q = 0000 Not used

Error 6 - A parity occurs in the Memory Stack which contains SMM and the test. This indicates that bits have been dropped within the test itself. If the operator attempts to continue the running of the test, the section in which the error occurred is aborted and the test attempts to run the next section.

Third Stop

A = XXXX Normally gives the address stored in trapped location \$100. But when stack zero is being tested, the level zero interrupt is not enabled, and this stop gives the address of the parity check which revealed that there had been a parity error in the program sometime since the last parity error check.

Q = 0000 Not used

Fourth Stop

A = 0000 Not used

Q = 0000 Not used

Error 7 - Unexpected program protect fault. This error should only occur during Section 02, since this is the only time that the PROGRAM PROTECT switch should be set.

Third Stop

A = XXXX Address of the location in the test stack which was being tested at the time of the error.

Q = 0000 Not used

Fourth Stop

A = 0000 Not used

Q = 0000 Not used

Error 8 - A program protect fault did not occur when expected.

Third Stop

A = XXXX Location being tested

Q = 0000 Not used

Fourth Stop

A = 0000 Not used

Q = 0000 Not used

Error 10 - Load error. The LWA of the test has exceeded one memory bank, 4096 words (A3 = actual LWA).

II. DESCRIPTION

The test determines the size of memory and checks the memory in 1000_{16} word blocks (one stack). Stack one is the first to be tested. After the last stack has been tested the test and SMM are moved to the stack specified by the test parameters (normally stack one), and stack zero is then tested. The test consists of two standard sections (00 and 01) and one optional section (02).

SECTION 00 - WORST PATTERN TEST WITH MULTIPLE INDIRECT REFERENCING DISTURB

Tests the ability of each core, in the parity plane of the test stack, to hold zero and one while the rest of the plane holds worst pattern and while the rest of the plane holds complement worst pattern.

- A. Fill the test stack with complement worst pattern except for plane zero which is masked to zeros. This causes worst pattern to be generated in the parity plane. Skip to end of section if 65K mode.
- B. Disturb each location (X) and check its contents.
- C. Set bit zero of location (X), disturb the location, check its contents, then replace its original contents.
- D. Fill the test stack with worst pattern except for plane zero which is masked to zeros. This causes complement worst pattern to be stored in the parity plane.
- E. Repeat steps B and C.

NOTE: The disturbing of each location (X) is accomplished by setting up and initiating a multiple indirect referencing chain within the inhibit group which contains the location (X).

SECTION 01 - OPTIMAL WORST PATTERN TEST

Tests the ability of each core, in the parity plane of the test stack, to hold zero and one while the rest of the plane holds worst pattern and while the rest of the plane holds complement worst pattern.

- A. Fill the test stack with complement worst pattern except for plane zero which is masked to zeros. This causes worst pattern to be generated in the parity plane.
- B. Disturb each location (X) and check its contents.
- C. Set bit zero of location (X), disturb the location, check its contents, then replace its original contents.
- D. Fill the test stack with worst pattern except for plane zero which is masked to zeros. This causes complement worst pattern to be stored in the parity plane.
- E. Repeat steps B and C.

NOTE: The disturbing of each location (X) is accomplished by reading a word containing all zeros in each inhibit group of the test stack and then by reading one of four noise producing diagonals of ones. Each location (X) in the test stack is checked with the first diagonal before the next diagonal is used. When all four diagonals have been used, complement worst pattern is stored in the test stack and the procedure is repeated.

SECTION 02 - PROGRAM PROTECT PLANE TEST

This section requires the operator to set and clear the PROGRAM PROTECT switch at specified times in the section.

- A. Set the program protect bit in each location of the test stack.
- B. Stop. Operator must set the PROGRAM PROTECT switch and run.
- C. An attempt is made to write into each location of the test stack.
- D. A check is made for expected and unexpected protect fault.
- E. Stop. Operator must clear the PROGRAM PROTECT switch and run.
- F. Store worst pattern in the program protect plane.
- G. Repeat steps B, C, D, and E.
- H. Store complement worst pattern in the program protect plane.
- I. Repeat steps B, C, D, and E.
- J. Clear the program protect bit in each location of the test stack.
- K. Repeat steps B, C, D, and E.

III. PHYSICAL REQUIREMENTS

A. Storage Requirements

Approximately 2500₁₀ locations plus SMM monitor

B. Timing

Section 0 = 9 minutes per stack in 32K mode

Section 1 = 3 minutes per stack

Section 2 = Manual operation required

Total = 12 minutes per stack

C. Equipment Configuration

1700 Series Computer with 8K memory

4. SELECTIVE JUMP and SELECTIVE STOP key setting

- a. SELECTIVE STOP must be set.
- b. Use SELECTIVE SKIP switch as outlined in the SMM17 Reference Manual.

D. MESSAGES

1. Typeouts or Alarms

a. Normal Program Typeouts

- 1) Command test identification at start of test
CMD01C 4K COMMAND TEST
IA = XXXX, FC = XX CP2F, VER. 3.1
- 2) End of Command test

A	Q	A	Q
1C04	STOP	PASS NO.	RETURN ADDRESS

b. Error Alarms

- 1) The following is typed out:
 - a) Identification word
 - b) Stop/Jump parameter
 - c) Section and error number
 - d) Return address
 - e) Correct data or simulated results
 - f) Incorrect data
- 2) See each individual section write up for its specific error alarm.

c. Error Codes - See individual section write up.

E. ERROR STOPS

- 1. The same data typeout for error display will be displayed in A and Q.
- 2. See each individual section write in for its specific error.

II. DESCRIPTION

A. PROGRAM DESCRIPTION

This test checks internal 1700 instructions with fixed and random operands. The instructions are tested beginning with the simpler ones and ending with the more complex.

The check of actual results is made by either a simulation or a comparison to anticipated results. The writeups of the individual tests follow and are in the same order as they appear in the program.

Section 1 JUMP

Error Number 0

Test if Jump instruction is executed and if destination is correct. If destination is ± 1 instruction, a Stop should occur.

Error Display: Standard error format not used.

A = 108 and OVERFLOW light will not be lit.

Section 2 RETURN JUMP

Error Number 1

Incorrect address was stored in return address.

Error Display: A3 = correct address, Q3 = incorrect address

Section 3 LOAD A

Error Number 1

An operand was loaded into A. The same operand and the contents of A are then compared via an EXCLUSIVE OR.

Error Display: A3 = correct data, Q3 = incorrect data

Section 4 STORE A

Error Number 1

An operand was loaded into A and stored. A was then compared with the operand stored via an EXCLUSIVE OR.

Section 5 LOAD Q

Error Number 1

An operand was loaded into Q and transferred to A. The same operand and the contents of Q were then compared via EXCLUSIVE OR.

Error Display: A3 = correct data, Q3 = incorrect data

Section 6 STORE Q

Error Number 1

An operand was loaded into Q and A. Q was then stored and the stored operand was compared to (A).

Error Display: A3 = correct data, Q3 = incorrect data

Section 7 SKIP IF A = +0

Error Number 1

No Skip occurred when positive zero was loaded into A and then tested.

Error Display: No data display

Error Number 2

A Skip occurred when negative zero was tested in A.

Error Display: No data display

Error Number 3

Positive zero Skip did not occur for contents of A, but when tested in Q, a Skip occurred.

Error Display: A3 = contents of A, Q3 = contents of Q

Error Number 4

Positive zero Skip occurred when tested in A, but did not occur when tested in Q.

Error Display: A3 = contents of A, Q3 = contents of Q

Section 8 SKIP IF A ≠ 0

Error Number 1

No Skip occurred when negative zero was loaded into A and tested.

Error Display: No data display

Error Number 2

A Skip occurred when zero was loaded into A and tested.

Error Display: No data display

Error Number 3

A was tested with SAN instruction and no Skip took place, but when tested by SAZ instruction it was found to be non-zero.

Error Display: A3 = contents of A

Error Number 4

A was tested with SAN instruction and a Skip took place, but when tested by SAZ instruction it was found to be zero.

Error Display: A3 = contents of A

Section 9 SKIP IF A = + (positive)

Error Number 1

When A was loaded with an operand and tested for positive value a Skip did not occur. When complemented and tested again the Skip did not occur.

Error Display: A3 = operand tested first, Q3 = complement of operand

Error Number 2

When A was loaded with an operand and tested for positive value, a Skip occurred. When complemented and tested again, the Skip occurred.

Error Display: A3 = operand tested first, Q3 = complement of operand

Section A SKIP IF A = - (negative)

Error Number 1

When A was loaded with an operand and tested for negative value, a Skip did not occur. Then A was tested for positive and found to be negative.

Error Display: A3 = operand tested

Error Number 2

When A was loaded with an operand and tested for negative value, a Skip occurred. Then A was tested for positive and found to be positive.

Error Display: A3 = operand tested

Section B SKIP IF Q = +0

Error Number 1

Q and A were loaded with the same operand, Q was then tested with SQZ, and no Skip occurred. But when A was tested with SAZ, a Skip occurred.

Error Display: A3 = operand tested

Error Number 2

Q and A were loaded with the same operand, Q was then tested with SQZ, and a Skip occurred. But when A was tested with SAZ, a Skip did not occur.

Error Display: A3 = operand tested

Section C SKIP IF Q \neq + 0

Error Number 1

Q and A were loaded with the same operand; Q was then tested with SQN and no Skip occurred. But when A was tested with SAN, a Skip occurred.

Error Display: A3 = operand tested

Error Number 2

Q and A were loaded with the same operand, Q was then tested with SQN and a Skip occurred. But when A was tested with SAN, a Skip did not occur.

Error Display: A3 = operand tested

Section D SKIP IF Q = + (positive)

Error Number 1

Q and A were loaded with the same operand, Q was then tested with SQP, and no Skip occurred. But when A was tested with SAP, a Skip occurred.

Error Display: A3 = operand tested

Error Number 2

Q and A were loaded with the same operand, Q was then tested with SQP, and a Skip occurred. But when A was tested with SAP, a Skip did not occur.

Error Display: A3 = operand tested

Section E SKIP IF Q = - (negative)

Error Number 1

Q and A were loaded with the same operand, Q was then tested with SQM, and no Skip occurred. But when A was tested with SAM, a Skip occurred.

Error Display: A3 = operand tested

Error Number 2

Q and A were loaded with the same operand; Q was then tested with SQM and a Skip occurred. But when A was tested with SAM, a Skip did not occur.

Error Display: A3 = operand tested

Section F A LEFT SHIFT AND TEST FOR OVERFLOW

An operand was loaded into A and shifted left a predetermined number of times. A check is made for overflow after shifting.

Error Number 1

The end results of the shifting did not equal the starting value.

Error Display: A3 = correct data, Q3 = incorrect data

Error Number 2

Overflow occurred.

Error Display: A3 = operand, Q3 = not applicable

Section 10 Q LEFT SHIFT AND TEST FOR OVERFLOW

Error Number and displays are the same as Section F.

Section 11 A RIGHT SHIFT

Error Number 1

A was loaded with 4000, then shifted to the right once and compared to the known value. This is repeated 15 times for each iteration of the test section.

Error Display: A3 = correct data, Q3 = incorrect data

Section 12 A RIGHT SHIFT

Error Number 1

Same as Section 11 with the exception that A = 800 instead of 4000 when starting the shifting.

Section 13 A RIGHT, Q RIGHT

Error Number 1

A and Q were loaded with the same operand, then shifted and compared.

Error Display: A3 = content of shifted A, Q3 = content of shifted Q

Section 14 LONG LEFT SHIFT AND CHECK FOR OVERFLOW

A and Q were loaded with operands and shifted 1, 2, . . . 31 places then compared to the original value loaded into A and Q and a check is made for overflow.

Error Number 1

Shifted results did not equal original values in A and Q.

Error Display: A3 = original contents of A, Q3 = original contents of Q
A4 = shifted contents of A, Q4 = shifted contents of Q

Error Number 2

Overflow occurred

Error Display: Inapplicable.

Section 15 LONG RIGHT SHIFT

Error Number 1

A and Q were loaded with 8000 and 4000, then shifted one place and compared to the known result. The shifting continues until the bits have traversed the registers.

Error Display: A3 = correct contents of A, Q3 = correct contents of Q
A4 = incorrect contents of A, Q4 = incorrect contents of Q

Error Number 2

A and Q were loaded with 8000, then shifted one place and compared to the known results. The shifting continues until the bits have traversed the registers.

Error Display: Same as Error Number 1

Section 16 ENTER A

Error Number 1

A known quantity is masked into an ENA instruction; then the instruction is executed and compared to the known quantity.

Error Display: A3 = correct data, Q3 = incorrect data

Section 17 ENTER Q

Error Number 1

The same as Section 16 with the exception that Q is used instead of A.

Section 18 AND WITH A

Error Number 1

A is set to all one's; then an AND is executed with a known operand. The contents of A and the operand are then compared.

Error Display: A3 = correct data, Q3 = incorrect data

Section 19 ADD TO A

Error Number 1

The ADD instruction was simulated by shifting and bit comparison. The simulated results were then compared to the instruction results.

Error Display: A3 = simulated results, Q3 = instruction results

Section 1A EXCLUSIVE OR WITH A

Error Number 1

The EOR instruction was simulated by shifting and bit comparison. Then the simulated results were compared to the instruction results.

Error Display: A3 = simulated results, Q3 = instruction results

Section 1B ADD TO Q

Error Number 1

An ADD instruction was used to simulate the ADQ. Then the ADQ instruction is executed and compared to the simulated results.

Error Display: A3 = simulated results, Q3 = instruction results

Section 1C INCREASE A

Error Number 1

An ADD instruction was used to simulate the INA instruction. Then the INA instruction was executed and compared to the simulated results.

Error Display: A3 = simulated results, Q3 = instruction results

Section 1D INCREASE Q

Error Number 1

An INA instruction was used to simulate the INQ instruction. Then the INQ instruction was executed and compared to the simulated results.

Error Display: A3 = simulated results, Q3 = instruction results

Section 1E REPLACE ADD ONE IN STORAGE AND CHECK FOR OVERFLOW

Error Number 1

An ADD instruction was used to simulate the RAO instruction. Then RAO instruction was executed and compared to the simulated results.

Error Display: A3 = simulated results, Q3 = instruction results

Error Number 2

Overflow occurred

Error Display: A3 = simulated results, Q3 = instruction results

Section 1F SUBTRACT

Error Number 1

An ADD instruction with a complemented operand is used to simulate the SUB instruction. Then the SUB instruction is executed and compared to the simulated results.

Error Display: A3 = simulated results, Q3 = instruction results

Section 20 SKIP ON OVERFLOW, SKIP ON NO OVERFLOW

Error Number 1

An overflow was forced (+ to -), but when tested with SOV the Skip did not occur.

Error Display: No data display

Error Number 2

The execution of a SOV instruction failed to clear an overflow.

Error Display: No data display

Error Number 3

An overflow was forced (- to +), but when tested with SOV the Skip did not occur.

Error Display: No data display

Error Number 4

An overflow was forced (+ to -), but when tested with SNO the Skip occurred.

Error Display: No data display

Error Number 5

After executing SNO instruction, another one was executed and no Skip occurred.

Error Display: No data display

Error Number 6

An overflow was forced (- to +), but when tested with SNO the Skip occurred.

Error Display: No data display

Section 21 MULTIPLY INTEGER

Error Number 1

The simulation of the MUI instruction was accomplished by adding and shifting.

Error Display: A3 = simulated results, Q3 = simulated results (most significant)

A4 = instruction results, Q4 = instruction results (most significant)

Section 22 DIVIDE INTEGER

Error Number 1

The simulation of the DIV instruction was accomplished by subtracting and shifting.

Error Display: A3 = simulated quotient, Q3 = simulated remainder

A4 = instruction quotient, Q4 = instruction remainder

Error Number 2

An overflow occurred but the OVERFLOW indicator was not set.

Error Display: Same as Error Number 1

Error Number 3

No overflow occurred but the OVERFLOW indicator was set.

Error Display: Same as Error Number 1

Section 23 STORE PARITY TO A

Errors Number 1 and 2

The SPA instruction is simulated and the results of the simulation are compared to the instruction results.

Error Display: A3 = parity of simulation, Q3 = parity of instruction
A4 = contents of A when parity was determined
Q4 = data stored by SPA

Section 24 SET TO ONES A, Q

Error Number 1

The SET instruction is executed and then A and Q are checked for all bits being set.

Error Display: A3 = correct data = A and Q, Q3 = contents of A after
SET execution
A4 = contents of Q after SET execution

Section 25 CLEAR TO ZERO A, Q

Error Number 1

The CLR instruction is executed and then checked for all bits being cleared in AQ.

Error Display: A3 = correct data = A and Q, Q3 = contents of A after CLR
execution
A4 = contents of Q after CLR execution

Section 26 TRANSFER A TO A, Q

Error Number 1

The TRA instruction is executed and then the registers are checked for equality with the original contents of A.

Error Display: A3 = correct data, Q3 = contents of A after TRA execution
A4 = contents of Q after TRA execution

Section 27 TRANSFER Q TO A, Q

Error Number 1

The TRQ instruction is executed and then the destination register is checked for equality with the original contents of Q.

Error Display: A3 = correct data, Q3 = contents of Q after TRQ execution

Section 28 TRANSFER COMPLEMENT A TO A, Q

Error Number 1

The TCA instruction is simulated by complementing the operand with an EOR instruction. The results of the simulation are then compared to the destination registers of the TCA instruction.

Error Display: A3 = simulated data, Q3 = contents of A after TCA
A4 = contents of Q after TCA

Section 29 TRANSFER COMPLEMENT Q TO A, Q

Error Number 1

The TCQ instruction is simulated by complementing the operand with an EOR instruction. The results of the simulation is then compared to the destination registers of the TCQ instruction.

Error Display: A3 = simulated results, Q3 = contents of A after TCQ
A4 = contents of Q after TCQ execution

Section 2A TRANSFER THE ARITHMETIC SUM A, Q TO A, Q

Error Number 1

The ADD instruction is used to simulate the AAQ instruction.

Error Display: A3 = simulated results, Q3 = contents of A after AAQ execution
A4 = contents of Q after AAQ execution

Section 2B TRANSFER EXCLUSIVE OR A, Q TO A, Q

Error Number 1

The EOR instruction is used to simulate the EAQ instruction.

Error Display: A3 = simulated results, Q3 = contents of A after EAQ execution
A4 = contents of Q after EAQ execution

Section 2C TRANSFER THE LOGICAL PRODUCT OF A, Q, TO A, Q

Error Number 1

The AND instruction is used to simulate the LAQ instruction.

Error Display: A3 = simulated results, Q3 = contents of A after LAQ execution
A4 = contents of Q after LAQ execution

Section 2D TRANSFER COMPLEMENT LOGICAL PRODUCT A, Q, TO A, Q

Error Number 1

The AND and FOR instructions are used to simulate a CAQ instruction.

Error Display: A3 = simulated results, Q3 = contents of A after CAQ execution
A4 = contents of Q after CAQ execution

Section 2E SET TO ONES - M

Error Number 1

The SET instruction is executed and then M is checked for all bits being set.

Error Display: A3 = simulated results, Q3 = contents of M after SET execution

Section 2F TRANSFER A TO M

Error Number 1

Simulation accomplished the same as Section 26.

Error Display: A3 = correct data, Q3 = contents of M after TRA

Section 30 CLEAR TO ZERO - M

Error Number 1

The CLR instruction is executed and then M is checked for all bits to be cleared.

Error Display: A3 = correct data, Q3 = contents of M after SET

Section 31 TRANSFER Q TO M

Error Number 1

Simulation accomplished the same as Section 27.

Error Display: A3 = correct data, Q3 = contents of M after TRQ

Section 32 TRANSFER COMPLEMENT A TO M

Error Number 1

Simulation accomplished the same as Section 28.

Error Display: A3 = correct data, Q3 = contents of M after TCA

Section 33 TRANSFER COMPLEMENT Q TO M

Error Number 1

Simulation accomplished the same as Section 29.

Error Display: A3 = correct data, Q3 = contents of M after TCQ

Section 34 TRANSFER ARITHMETIC SUM A, Q TO M

Error Number 1

Simulation accomplished the same as Section 2A.

Error Display: A3 = simulated results, Q3 = contents of M after AAQ

Section 35 TRANSFER EXCLUSIVE OR A, Q TO M

Error Number 1

Simulation accomplished the same as Section 2B.

Error Display: A3 = simulated results, Q3 = contents of M after EAQ

Section 36 TRANSFER THE LOGICAL PRODUCT OF A, Q TO M

Error Number 1

Simulation accomplished the same as Section 2C.

Error Display: A3 = simulated results, Q3 = contents of M after LAQ

Section 37 TRANSFER COMPLEMENT LOGICAL PRODUCT A, Q TO M

Error Number 1

Simulation accomplished the same as Section 2D.

Error Display: A3 = simulated results, Q3 = contents of M after CAQ

Section 38 TRANSFER M TO A, Q, M

Error Number 1

The TRM instruction is executed and then the registers are checked for equality with original contents of M.

Error Display: A3 = simulated results, Q3 = contents of A after TRM
A4 = contents of Q after TRM, Q4 = contents of M after TRM

Section 3A TRANSFER COMPLEMENT M TO A, Q, M

Error Number 1

The TCM instruction is simulated by complementing the operand with an EOR instruction.

Error Display: A3 = simulated results, Q3 = contents of A after TCM
A4 = contents of Q after TCM, Q4 = contents of M after TCM

Section 3B TRANSFER COMPLEMENT Q + M TO A, Q, M

Error Number 1

The TCB instruction is simulated by using AND and EOR instructions.

Error Display: A3 = simulated results, Q3 = contents of A after TCB
A4 = contents of Q after TCB, Q4 = contents of M after TCB

Section 3C TRANSFER ARITHMETIC SUM A, M TO A, Q, M

Error Number 1

The AAM instruction is simulated by using the ADD instruction.

Error Display: A3 = simulated results, Q3 = contents of A after AAM
A4 = contents of Q after AAM, Q4 = contents of M after AAM

Section 3D TRANSFER ARITHMETIC SUM A, Q + M TO A, Q, M

Error Number 1

The AAB instruction is simulated by using ADD, EOR, and AND instructions.

Error Display: A3 = simulated results, Q3 = contents of A after AAM
A4 = contents of A after AAB, Q4 = contents of M after AAB.

Section 3E TRANSFER EXCLUSIVE OR A, M TO A, Q, M

Error Number 1

The EAM instruction is simulated by using the ROR instruction.

Error Display: A3 = simulated results, Q3 = contents of A after EAM
A4 = contents of Q after EAM, Q4 = contents of M after EAM

Section 3F TRANSFER EXCLUSIVE OR A, Q + M TO A, Q, M

Error Number 1

The EAB instruction is simulated by using the AND, ADD, and EOR instruction.

Error Display: A3 = simulated results, Q3 = contents of A after EAB
A4 = contents of Q after EAB, Q4 = contents of M after EAB

Section 40 TRANSFER LOGICAL PRODUCT A, M TO A, Q, M

Error Number 1

The LAM instruction is simulated by using the AND instruction.

Error Display: A3 = simulated results, Q3 = contents of A after LAM
A4 = contents of Q after LAM, Q4 = contents of M after LAM

Section 41 TRANSFER LOGICAL PRODUCT A, Q + M TO A, Q, M

Error Number 1

The LAB instruction is simulated by using the AND, ADD, and EOR instructions.

Error Display: A3 = simulated results, Q3 = contents of M after LAB
A4 = contents of Q after LAB, Q4 = contents of M after LAB

Section 42 TRANSFER COMPLEMENT LOGICAL PRODUCT A, M TO A, Q, M

Error Number 1

The CAM instruction is simulated by using AND and EOR instructions.

Error Display: A3 = simulated results, Q3 = contents of A after CAM
A4 = contents of Q after CAM, Q4 = contents of M after CAM

Section 43 TRANSFER COMPLEMENT LOGICAL PRODUCT A, Q+M TO A, Q, M

Error Number 1

The CAB instruction is simulated by using AND, ADD, and EOR instructions.

Error Display: A3 = simulated results, Q3 = contents of A after CAB
A4 = contents of Q after CAB, Q4 = contents of A after CAB

III. PHYSICAL REQUIREMENTS

- A. STORAGE REQUIREMENT - About 2500_{10} locations.
- B. TIMING - 1 min. 15 sec.
- C. EQUIPMENT CONFIGURATION
 1. 1704 with a minimum of 4K of memory.
 2. A device for loading the program.

1784 MOS MEMORY TEST
(MOS01E Test No. 1E)
(CP=2C)

I. OPERATIONAL PROCEDURES

A. RESTRICTIONS

This test cannot be loaded concurrently with MY1, MY2, MEM, RPT, TTY input package, line printer driver, or the MSB package.

B. REQUIREMENTS

1784-1 or -2 with 8 to 65K of memory
SMM standard input device
TTY or equivalent as an optional comment device
Loaded in Stack 0 with the SMM monitor

C. LOADING PROCEDURE

This test will load and execute correctly under SMM17, Version 4.0. The calling sequence is that specified by the current SMM17 system. The test uses normal SMM17 routines for all operator communications.

D. TESTING PROCEDURE

The program starts its test with stack 1 or first selected stack greater than zero. It executes all selected sections before testing the next selected stack. After the highest selected stack has been checked and stack zero is requested, bits 8-15 of the test section parameters are checked to determine which stack the contents of stack zero is to be moved to. Upon completion of testing, stack zero is restored (SMM and TEST) and end of test occurs.

E. PARAMETERS

During initialization the test will determine the stacks available for test from SMM (monitor) contained parameters, this is offered for change in test parameters.

TSTJP - Test stop/jump parameter bit assignments. (This parameter may be changed anytime it is displayed at a stop.)

Bit 0 = Stop to enter parameters.

Bit 1 = Stop at end of section within each bank.

Bit 2 = Stop at end of test.

Bit 3 = Stop on error.

Bit 4 = Repeat conditions, irrespective of errors, in same bank.

Bit 5 = Repeat current section in current bank.

Bit 6 = Repeat test, all banks

Bit 7 = Repeat test, all sections in current bank.

Bit 8 = Omit typeouts.

Bit 9 = Bias return address display.

1. Stop to Enter Parameters (Type 1)

Stop 1:

A1 = 1E21 Test ID (1E), (2) stops, parameter code (1)
Q1 = TSTJP Test unique Stop/Jump parameters.

Stop 2:

A2 = STKS Memory stacks to test (bit 0 = stack 0, bit 8 =
stack 8). Any bit not set is not tested.
Q2 = MMOS Section select; S = section by bit position 0, 1,
2, 3, MM = stack in which to move SMM when
testing stack zero, bit 8 = stack 1, bit 9 =
stack 2 (if zero, test will use highest stack 7
or less).

2. Stop at End of Section (Type 2) - occurs only if TSTJP bit 1 is set.

A1 = 1E32 Test number (1E), stops (3), type (2) end of
section
Q1 = TSTJP See TSTJP.
A2 = BOOS B = current bank, S = section just completed
Q2 = TRTA Test return address
A3 = SEERS Number of errors that occurred this section.
Q3 = PERRS Number of errors that occurred so far this pass.

3. Stop at end of Test (Type 4)

A1 = 1E34 Test number (1E), stops (3), type (4) end of test
Q1 = TSTJP See TSTJP.
A2 = 00PC Pass count starting from 1
Q2 = TRTA Test return address
A3 = PERRS Number of errors this pass
Q3 = TERRS Total number of errors since the test was loaded

4. Stop on Error (Type 8)

A1 = 1E48	Test number (1E), stops (4), type (8) error stop.
Q1 = TSTJP	See TSTJP.
A2 = BSEE	B = stack in error, S = section detecting error, EE = error code
Q2 = TRTA	Test return address, usually to area that de- tected the error
A3 =	Failing data, data read on or after failure
Q3 =	Expected data, data written
A4 =	Failing address
Q4 =	Address of location under test if non-zero; otherwise, this is not used.

a. Word Messages

Initialization of the test only.

MOS01E 1784 MOS Memory Test.

VRS 4.0 IA = XXXX CP = 2C

II. ERROR CODE LOGIC

The error codes are formed-numbers by bit position using the following logic.

Bit 0 = PE detected during store operation.
Bit 1 = PE detected during read operation.
Bit 2 = Data compare error
Bit 3 = No level 0 (PE) interrupt on PE
Bit 4 = PE during execution of SPE sequence
Bit 5 = PE during RTJ sequence
Bit 6 = PE within SMM or test, should be considered fatal

The following third and fourth standard stops are indicated for each error message.

A3 = Failing data, data read on or after failure
Q3 = Expected data, data written
A4 = Failing address
Q4 = Address of location under test if non-zero; otherwise, this is not used.

A. ERROR CODES

- 01 PE occurred during a store operation. This can be caused by a previous PE in this location. No load PE or data compare error.
- 02 PE occurred during a load operation. No store PE or data compare error was detected (except Section 0). This is caused by memory decay or another location disturbing either the PE bit or Protect bit.
- 03 Write parity error and read PE, no data compare error. This can be caused by a solid failure in either the Protect bit or parity bit circuitry.
- 04 Data compare error with neither write PE nor read PE.
- 05 Write parity error and data compare error with no read parity error.
- 06 Read parity error and data compare error with no write parity error.
- 07 Write parity error, read parity error, and data compare error. Normal indication of a solid failure.
- 08 Illogical
- 09 No level 0 interrupt on PE with a write parity error.
- 0A No level 0 interrupt on PE with read parity error and no data compare error or write parity error.
- 0B No level 0 interrupt on PE with write PE and read PE. No data compare error.
- 0C Illogical
- 0D No level 0 interrupt on PE with write parity PE; no read PE but with data compare error.
- 0E No level 0 interrupt on PE without write PE but with read PE and data compare error.
- 0F No level 0 interrupt on PE with write PE, read PE and data compare error.
- 10 Parity error occurred while executing SPB or CPB instruction without write PE, read PE, or data compare error.
- 11 Illogical
- 12 Illogical
- 13 Illogical

- 14 Parity error occurred while executing SPB or CPB instruction. Data compare error occurred.
- 15 Illogical
- 16 Illogical
- 17 Illogical
- 18 No level 0 interrupt occurred due to PE when PE occurred due to SPB or CPB sequence.
- 19 Illogical
- 1A Illogical
- 1B Illogical
- 1C No level 0 interrupt occurred due to PE when PE occurred due to SPB or CPB sequence and data error.
- 1D Illogical
- 1E Illogical
- 1F Illogical
- 20 Parity error occurred while performing RTJ sequence (Section 3). No data compare error or load PE found.
- 21 Illogical
- 22 Parity error occurred while performing RTJ sequence (Section 3). Load PE also occurred with no data error.
- 23 Illogical
- 24 PE occurred while performing RTJ sequence (Section 3). No load PE found but data error did occur.
- 25 Illogical
- 26 PE occurred while performing RTJ sequence (Section 3). Load PE found with data compare error.
- 27 Illogical
- 28 PE occurred while performing RTJ sequence (Section 3). No level 0 interrupt occurred, no data compare error or load parity error.
- 29 Illogical

- 2A PE occurred while performing RTJ sequence (Section 3). No level 0 interrupt occurred, although a load PE did occur but there was no data compare error.
- 2B Illogical
- 2C PE occurred while performing RTJ sequence (Section 3). No level 0 interrupt occurred, but there was a data compare error.
- 2D Illogical
- 2E PE occurred while performing an RTJ sequence (Section 3). No level 0 interrupt occurred. A load PE and a data compare error also existed.
- 2F Illogical
- 30-3F Illogical
- 40 Parity error within SMM or test. This error should be considered fatal because the integrity of the testing procedure cannot be ensured.
- 41-7F Illogical
- 80 Parameter entry error. After this error stop the test will stop again for parameters.
- 8F Length of monitor and test exceeds 4K. Test may not execute correctly; delete other tests or monitor options to reduce memory requirements

III. SECTION DESCRIPTIONS

A. SECTION 0 - READ ONLY

This is a non-destructive section that is intended to scan the stack under test for PE. It is this section that is intended to multiplex with other tests (DSA-type devices) to help isolate DSA-generated errors.

1. Initialize test address to FWA of the test bank.
2. Initialize control count to 64.
3. Load contents of test address to expected data, (Q3) clear PE.
4. Load contents of test address to A register 10 times.
5. If no parity error, go to 6.
 - a. Store A register at actual data (A3).
 - b. Report PE on load.
6. Increment address +1.
7. Decrement control count 1.

8. If control count is not equal to zero, go to 4.
9. If test address \neq LWA+1 of bank, go to 10.
 - a. Exit section.
10. Return control to SMM and to other tests.
11. Go to 2.

B. SECTION 1 - BASIC OPENS AND SHORTS TEST

This section is to detect simple data errors and addressing failures by storing test (Q3) data into a test address (Q4). Then storing complement data into the respective location in each of the three other rows of chips on the memory board end-around and finally within the same row all other locations end-around. The test cell is then examined for PE and data accuracy. The test cell is then moved +1 location until all locations within the bank have been tested. Finally the test data is complemented and the procedure is repeated.

1. Set test data to zero.
2. Set test address to stack FWA.
3. Write test data into test address.
4. Write complement test data into test address +1, 2, and 3 end-around.
5. Write complement test data into test address +4, 8, C, 10, etc. end-around to test address, staying within row of chips on board.
6. Read test data and check for PE and data compare errors.
7. If test address \neq LWA of bank, go to 8.
 - a. If test data \neq \$FFFF, end section.
 - b. Complement test data, go to 2.
8. Move test address +1.
9. Go to 3.

C. SECTION 2 - CHIP COLUMN AND ROW REFRESH SENSITIVITY AND DISTURB TEST

This section attempts to check a data chip's sensitivity to refresh an addressing by addressing the X and Y axis of a chip at the maximum speed possible, every three cycles. It is the chip vendor's opinion that this is the worst case test for these MOS chips.

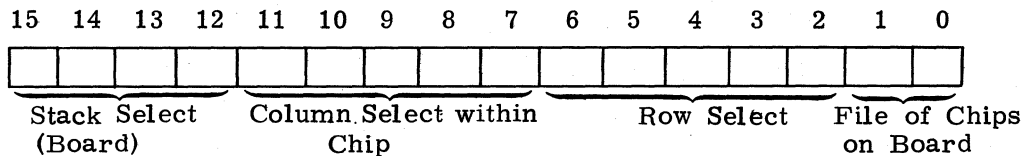
1. Initialize test data to zero.
2. Initialize address = bank FWA + file + 0.

3. Call SUBCON. (See SUBCON.)
4. Increment file + 1.
5. If file = 4:
 - a. If test data \neq FFFF, exit section.
 - b. Test data = FFFF, go to 2.
6. Go to 3.

SUBCON:

7. Initialize current address and test address = bank + current file.
8. Write test data at test address (ROW).
9. If current address \neq 32:
 - a. Increment current cell +4.
 - b. Write disturb complement data at current address.
 - c. Go to 9.
10. Make current address = 0 + file + bank.
11. If current address \neq test address:
 - a. Write complement data at current address.
 - b. Current address = +4.
 - c. Go to 11.
12. Read data from current address (test address). Examine for accuracy and PE.
13. If test address = 32, go to 15.
14. If current address and test address = +4, go to 2.
15. Repeat from 8, testing columns this time and changing increment to \$80.

1784 MEMORY ADDRESSING



D. SECTION 3 - 1784 WORST CASE TEST

This section is an execution test of the memory under test. The stack is first filled with RTJ* *-0 (\$58FF) instructions. The last two locations are JMP (-4095) instructions. The stack is executed; by execution, the contents of the stack should be replaced with the address +1 of each instruction. On return, the data is checked in each location and PEs are also checked. This test will execute an RNI from P on one cycle and an STO at P on the next cycle (i.e., maximum speed). This is also an addressing test. The second part of this section first fills the stack with JMP* (*-65) (\$1CBE) instructions. It then starts from bank address +1 and fills 64 locations with SPB 0 (\$0600) instructions. The 0 register is entered with the address of the last SPB instruction and the sequence is executed. On return, PEs are checked along with the data in the block. The block is then moved +1 location and the sequence is repeated until the LWA of the block is equal to BANK LWA -1.

E. HINTS TO USER

1. The stack select bits (parameter A2) select each bank irrespective of each other. In other words, the stacks tested do not need to be contiguous.
2. Section 0 is designed to multiplex with other tests. The purpose of this is to help pinpoint DSA-generated errors.
3. This test can be loaded into and executed from any other stack than zero, but stack zero or the stack the test is residing in cannot be tested.
4. If the test cannot find a known intermittent PE, the following procedure will help detect the problem.
 - a. Enter 0600 throughout memory.
 - b. Enter OD01 in location XFFE of memory (X = last bank).
 - c. Clear PROGRAM PROTECT switch.
 - d. Set STOP on PE switch.
 - e. Run.
 - f. When CPU stops, PE is at the contents of the P register or Q register. This procedure takes a long time to check (i.e., Q incrementing from 0 to LWA).
5. To eliminate the possibility of an intermittent problem, it is recommended to run the test twice.

SECTION 1 - TEST BANK MEMORY MAP

First Pass

	0 0	0 1	0 2	0 3
X 0 0 0	F F F F	N N N N	N N N N	N N N N
X 0 0 4	<u>0 0 0 0</u>	F F F F	F F F F	F F F F
X 0 0 8	F F F F	N N N N	N N N N	N N N N
X 0 0 C	F F F F	N N N N	N N N N	N N N N
X 0 1 0	F F F F	N N N N	N N N N	N N N N
X F F 0	F F F F	N N N N	N N N N	N N N N
X F F 4	F F F F	N N N N	N N N N	N N N N
X F F 8	F F F F	N N N N	N N N N	N N N N
X F F C	F F F F	N N N N	N N N N	N N N N

Location X004 is
under test
(NNNN = unused/
unwritten locations)

Second Pass

	0 0	0 1	0 2	0 3
X 0 0 0	N N N N	0 0 0 0	N N N N	N N N N
X 0 0 4	N N N N	0 0 0 0	N N N N	N N N N
X 0 0 8	0 0 0 0	<u>F F F F</u>	0 0 0 0	0 0 0 0
X 0 0 C	N N N N	0 0 0 0	N N N N	N N N N
X 0 1 0	N N N N	0 0 0 0	N N N N	N N N N
X F F 0	N N N N	0 0 0 0	N N N N	N N N N
X F F 4	N N N N	0 0 0 0	N N N N	N N N N
X F F 8	N N N N	0 0 0 0	N N N N	N N N N
X F F C	N N N N	0 0 0 0	N N N N	N N N N

Location X009 is
under test
(NNNN = unused/
unwritten locations)

SECTION 2 - TEST STACK MEMORY MAP

	0 0	0 1	0 2	0 3
X 0 0 0	N N N N	0 0 0 0	N N N N	N N N N
X 0 0 1	N N N N	F F F F	N N N N	N N N N
X 0 9 C	N N N N	F F F F	N N N N	N N N N
X 0 8 0	N N N N	<u>0 0 0 0</u>	N N N N	N N N N
X 0 8 4	N N N N	F F F F	N N N N	N N N N
X 0 F C	N N N N	F F F F	N N N N	N N N N
X 1 0 0	N N N N	<u>0 0 0 0</u>	N N N N	N N N N
X 1 0 4	N N N N	F F F F	N N N N	N N N N
X 1 7 C	N N N N	F F F F	N N N N	N N N N
X 1 8 0	N N N N	<u>0 0 0 0</u>	N N N N	N N N N
X 1 8 4	N N N N	F F F F	N N N N	N N N N
X 2 F C	N N N N	F F F F	N N N N	N N N N
X 2 0 0	N N N N	<u>0 0 0 0</u>	N N N N	N N N N
X 2 0 4	N N N N	F F F F	N N N N	N N N N
X F 7 C	N N N N	F F F F	N N N N	N N N N
X F 8 0	N N N N	<u>0 0 0 0</u>	N N N N	N N N N
X F 8 4	N N N N	F F F F	N N N N	N N N N
X F F C	N N N N	F F F F	N N N N	N N N N

Row Test

Row 0 of chips in address file 1 are under test.

Complement data is also used.

SECTION 2 - TEST STACK MEMORY MAP (COLUMN TEST)

	0 0	0 1	0 2	0 3
X 0 0 0	N N N N	N N N N	0 0 0 0	N N N N
X 0 0 4	N N N N	N N N N	0 0 0 0	N N N N
X 0 0 8	N N N N	N N N N	0 0 0 0	N N N N
X 0 1 0	N N N N	N N N N	0 0 0 0	N N N N
X 0 1 4	N N N N	N N N N	0 0 0 0	N N N N
X 0 7 8	N N N N	N N N N	0 0 0 0	N N N N
X 0 7 C	N N N N	N N N N	0 0 0 0	N N N N
X 0 8 0	N N N N	N N N N	<u>F F F F</u>	N N N N
X 0 8 4	N N N N	N N N N	<u>F F F F</u>	N N N N
X 0 8 8	N N N N	N N N N	<u>F F F F</u>	N N N N
X F F 4	N N N N	N N N N	<u>F F F F</u>	N N N N
X F F 8	N N N N	N N N N	<u>F F F F</u>	N N N N
X F F C	N N N N	N N N N	<u>F F F F</u>	N N N N

Column Test

Column 0 of chips in address file 2 are under test.

Complement data is also used.

SECTION 3 TEST BANK MEMORY MAP

RTJ Test - Pass 1

	0 0	0 1	0 2	0 3
X 0 0 0	N N N N	5 8 F F	5 8 F F	5 8 F F
X 0 0 4	5 8 F F	5 8 F F	5 8 F F	5 8 F F
X 0 0 8	5 8 F F	5 8 F F	5 8 F F	5 8 F F
X 0 0 C	5 8 F F	5 8 F F	5 8 F F	5 8 F F
X 0 1 0	5 8 F F	5 8 F F	5 8 F F	5 8 F F
X F F 0	5 8 F F	5 8 F F	5 8 F F	5 8 F F
X F F 4	5 8 F F	5 8 F F	5 8 F F	5 8 F F
X F F 8	5 8 F F	5 8 F F	5 8 F F	5 8 F F
X F F C	5 8 F F	5 8 F F	1 C 0 0	F 0 0 0

After Execution

X 0 0 0	R T A	X 0 0 2	X 0 0 3	X 0 0 4
X 0 0 4	X 0 0 5	X 0 0 6	X 0 0 7	X 0 0 8
X 0 0 8	X 0 0 9	X 0 0 A	X 0 0 B	X 0 0 C
X 0 0 C	X 0 0 D	X 0 0 E	X 0 0 F	X 0 1 0
X 0 1 0	X 0 1 1	X 0 1 2	X 0 1 3	X 0 1 4
X F F 0	X F F 1	X F F 2	X F F 3	X F F 4
X F F 4	X F F 5	X F F 6	X F F 7	X F F 8
X F F 8	X F F 9	X F F A	X F F B	X F F C
X F F C	X F F D	X F F E	1 C 0 0	F 0 0 0

SECTION 3 - TEST BANK MEMORY MAP (SPB INSTRUCTION)

	0 0	0 1	0 2	0 3
X 0 0 0	R T A	R T A	0 6 0 0	0 6 0 0
X 0 0 4	0 6 0 0	0 6 0 0	0 6 0 0	0 6 0 0
X 0 0 8	0 6 0 0	0 6 0 0	0 6 0 0	0 6 0 0
X 0 0 C	0 6 0 0	0 6 0 0	0 6 0 0	0 6 0 0
X 0 1 0	0 6 0 0	0 6 0 0	0 6 0 0	0 6 0 0
X 0 1 4	0 6 0 0	0 6 0 0	0 6 0 0	0 6 0 0
X 0 1 8	0 6 0 0	0 6 0 0	0 6 0 0	0 6 0 0
X 0 3 C	0 6 0 0	0 6 0 0	0 6 0 0	0 6 0 0
X 0 4 0	0 6 0 0	<u>0 6 0 0</u>	1 C B E	1 C B E
X 0 4 4	1 C B E	1 C B E	1 C B E	1 C B E
X 0 4 8	1 C B E	1 C B E	1 C B E	1 C B E
X 0 4 C	1 C B E	1 C B E	1 C B E	1 C B E
X F F C	1 C B E	1 C B E	1 C B E	1 C B E

Location X041 is under test. On entry, the Q register = X041.

1723 PAPER TAPE PUNCH TEST
(PTP003 Test No. 3)
(CP = 2F)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

Any time the punch Ready drops, the test will display an error diagnostic and then wait until the punch is made Ready.

Bits 2 and 3 of the SMM parameter word must specify the correct machine type.

B. LOADING PROCEDURE

1. Call as external test under SMM17.
2. Restart test after loading by Master Clear, set P = IA and RUN.

C. PARAMETERS (in order in which they are entered)

Selected parameters are typed out.

1. CNTWRD - Control word

Bits

- | | |
|------|---|
| 14 | Run test with 8-level patterns. |
| 13 | Run test with 7-level patterns. |
| 12 | Run test with 5-level patterns.
(Only one bit may be set in this group.) |
| 11=1 | Run test completely in Interrupt mode. |
| =0 | Run test in both Interrupt and Character mode. |
| 10=1 | Omit delay in Character mode. |

Sections to be run if bit is set.

- | | |
|---|--|
| 7 | No section. |
| 6 | No section |
| 5 | Section 20 - C9, 36 pattern |
| 4 | Section 10 - all ones, all zeros pattern |
| 3 | Section 8 - complement pyramid pattern |
| 2 | Section 4 - pyramid pattern |
| 1 | Section 2 - complement zigzag pattern |
| 0 | Section 1 - zigzag pattern |

2. REPEAT - Number of times each pattern is to be repeated in each section
(prestored as $1E_{16}$)

3. DELAYA - Delay constant. Increasing this number increases the switching time between Interrupt and Character mode and decreases the delay between punches. (Prestored as $0C00_{16}$)
4. DELAYB - Delay constant. Increasing this number increases the total delay before the delay is reset. (Prestored as $0E00_{16}$)
5. DELAYC - Delay constant. Increasing this number decreases the switching time between Interrupt and Character mode. (Prestored as 0400_{16})
6. INT11 - Interrupt line mask.

D. MESSAGES

1. Test Title

PTP003, 1723/77 PUNCH TEST.

CP2F, VER. 3.1

IA = XXXX, FC = XX

(XXXX = starting address of test, XX = Frequency Count)

2. End of Test Message

A	Q	A	Q
0324	S/J	PASS NO.	RETURN ADDRESS

3. Other Messages

Comments to operator

- a. PUNCH PROTECT OFF
- b. PUNCH PROTECTED
- c. PUNCH TAPE LOW

4. Error Messages

- a. All error messages are in the format specified by SMM17.
- b. Description of Error Codes

<u>Error Code</u>	<u>Description</u>
0	(Not used)
1	External reject on a Clear Equipment function. Check equipment address for correct code.
2	Internal reject on Equipment Clear function. See error 1.
3	External reject on status. See error 1.
4	Internal reject on status. See error 1.
5	After an Equipment Clear, only Power On and Ready status bits should be set.
6	External reject on a Data Interrupt Request function.

<u>Error Code</u>	<u>Description</u>
7	Internal reject on a Data Interrupt Request function.
8	External reject on a Start Motion function.
9	Internal reject on a Start Motion function.
A	External reject on status after outputting a Start Motion function.
B	Internal reject on status after outputting a Start Motion function.
C	After outputting an Equipment Clear and Start Motion function, only Busy, Power on, Ready, and data status bits should have been set. Check status

ON ERRORS 1 TO C

The test will repeat in the same loop until the error condition is corrected. On errors 5 and C, Tape Low, Alarm, and Protect status bits are not checked. The checking of these bits is done in the Punch Frame Driver (see listing tag, PPT).

D	External reject on a Clear or Request Interrupt function.
E	Internal reject on a Clear or Request Interrupt function. This error will repeat on the same loop until error condition is corrected.
F	External reject on a Stop Motion and Clear Interrupt function.
10	Internal reject on a Stop Motion and Clear Interrupt function.
11, 12	Not used.
13	External reject on status after outputting a Clear Interrupt and Stop Motion function
14	Internal reject on status after a Clear Interrupt and Stop Motion function.
15	After outputting a stop motion and Clear Interrupt function, only Power On and Ready status should have been set.

Before errors F to 15, the test is delayed 10 ms to allow the last frame to be punched and Busy status to drop. The test will repeat in the same loop until the error condition is corrected. Error 15 does not check Tape Low, Alarm, and Protect status bits.

16	External reject on a Clear Interrupt function.
17	Internal reject on a Clear Interrupt function.
18	External reject on a Data Interrupt Request function.
19	Internal reject on a Data Interrupt Request function.
1A to 1F	Not used.
20	External reject on a Clear Interrupt or Data Interrupt Request function.
21	Internal reject on a Clear Interrupt or Data Interrupt Request function.
22 and 23	Not used.
24	External reject on status just before outputting the next frame of data.
25	Internal reject on status just before outputting the next frame of data.
26	External reject on output of frame of data.
27	Interrupt status (bit 2) is set; only Data Interrupt requested, but Data Ready status is not set.
28	Interrupt bit is set. Data interrupt and alarm interrupt bits are clear.
29	Interrupt and Data Ready bits set, but no interrupts selected by program.
2A	Alarm status and Tape Break status set. The test will hang up here until the break is repaired and the punch made ready.
2B	Alarm status is set, but no alarm conditions can be found. Power on is set. Tape break and tape low are reset.

2C	Start Motion has been outputted and Ready status set, but Busy status not set.
2D	Busy bit not set.
2E	The punch has dropped Ready and will hang up here until it is made Ready again by the operator.
2F	Unidentified interrupt.
30	Alarm status is set but Power On reset. Check status.
31	External reject on status.
32	Internal reject on status.
33	External reject on Start Motion function.
34	Internal reject on Start Motion function.
35	External reject on status when entering interrupt processor, or after making punch ready and starting motion following a tape break.
36	Internal reject on status when entering interrupt processor, or after making punch ready and starting motion following a tape break.

E. ERROR STOPS

1. FIRST STOP

(A) = ID word, (Q) = Stop/ Jump parameter

2. SECOND STOP

(A) = section number/
error code (Q) = return address

3. THIRD STOP

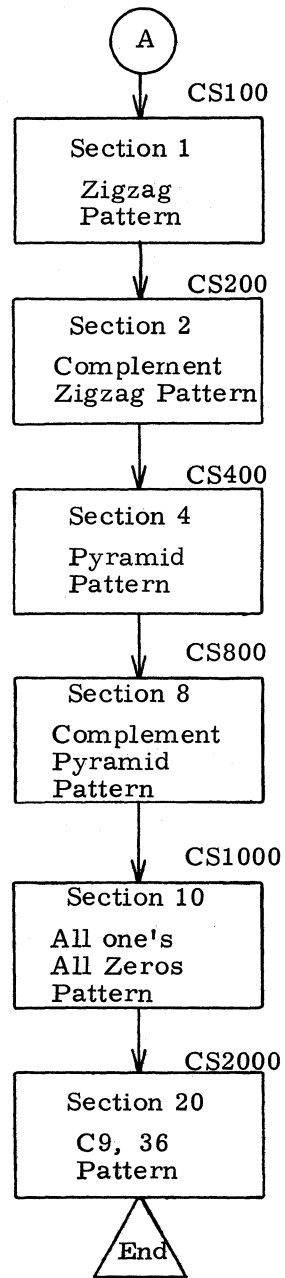
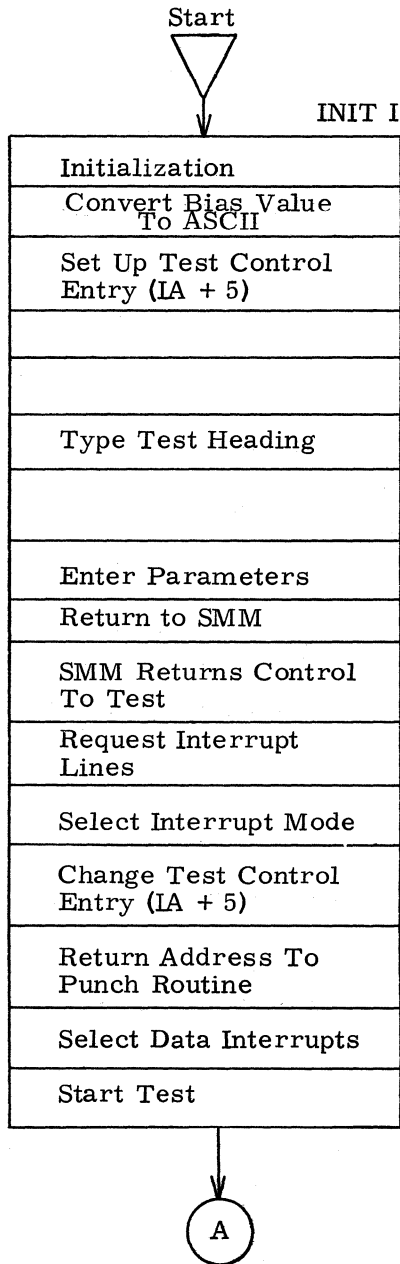
(A) = punch status at
time of error (Q) = punch equipment address

4. FOURTH STOP

(A) = punch character (Q) = test mode at time of error
FFFF = Interrupt mode
0000 = Character mode

II. DESCRIPTION

A. BLOCK DIAGRAM



II. B. SECTION DESCRIPTION

(Cont'd)

1. Initialization (INIT1)
 - a. Convert bias value and frequency count to ASCII and store in typeout message.
 - b. Set up test return control entry (RETURN).
 - c. Type test heading (INIT1B).
 - d. Return to SMM.
 - e. Enter parameters if selected.
 - f. Return to SMM.
 - g. SMM returns control to test (CNTRL).
 - h. Request interrupt lines.
 - i. Start punch in Interrupt mode.
 - j. Change test control entry (RETURN = CHECK).
 - k. Store return address (CS00A) in paper tape punch routine (PPT).
 - l. Select data interrupt and exit to SMM.
 - m. SMM returns control to (INTENTRY) when interrupt occurs, test ignores first interrupt and returns to (CS00A).
 - n. If frequency number is greater than one, punch 10 blank frames.
 - o. If not, punch one frame of all ones.
 - p. Check if Section 1 is to be executed. If so, go to (CS100); if not, go to Section 2 (CS200).
2. Section 1 (CS100) - Zigzag. Pattern
 - a. Set up section exit address.
 - b. Check for re-entry of parameters.
 - c. Determine punch level to be used (8, 7, or 5).
 - d. Store words used to generate pattern.
 - e. Punch section identification ten times.
 - f. Set up cycle counter.

- g. Generate pattern.
 - h. Punch generated pattern.
 - j. Delay (in Character mode only).
 - k. Repeat from g until cycle is complete.
 - l. Repeat from g until required number of cycles has been punched.
 - m. Check for End of Section stop.
 - n. Return to Interrupt mode if test was in this mode before stop.
 - p. Check for repeat section.
 - q. Go to next section.
3. Section 2 (CS200) - Complement Zigzag Pattern
- a. If Section 2 is not selected, go to Section 4 (CS400).
 - b. If selected, set up section exit address.
 - c. Go to (CS100). Section 2 is a repeat of Section 1 except the complement pattern is used.
4. Section 4 (CS400) - Pyramid Pattern
- a. If Section 4 is not selected, go to Section 8 (CS800).
 - b. If selected, set up section exit address.
 - c. Check for re-entry of parameters.
 - d. Determine punch level to be used (8, 7, or 5).
 - e. Store words used to generate pattern.
 - f. Punch section identification ten times.
 - g. Set up cycle counter.
 - h. Generate pattern.
 - j. Punch pattern.
 - k. Delay (Character mode only).
 - l. Repeat from h until cycle is complete.
 - m. Repeat cycle g the number of times specified by parameters.
 - n. Check for End of Section stop.
 - p. Return to Interrupt mode if test was in this mode before stop.
 - q. Check for Repeat Section.
 - r. Go to next section.

5. Section 8 (CS800) - Complement Pyramid Pattern
 - a. If Section 8 is not selected, go to Section 10 (CS1000).
 - b. If selected, set up section exit address.
 - c. Go to (CS401). Section 8 is a repeat of Section 4 except the complement pattern is used.
6. Section 10 (CS1000) - 10 all ones and 3 all zeros
 - a. If Section 10 is not selected, go to Section 20 (SC2000).
 - b. If selected, omit delay between punches.
 - c. Punch section identification ten times.
 - d. Set counter for number of cycles.
 - e. Generate pattern.
 - f. Punch pattern.
 - g. Repeat from e until cycle is complete.
 - h. Update cycle count.
 - j. Repeat from e until requested number of cycles has been punched.
 - k. Check for End of Section stop.
 - l. Return to Interrupt mode if test was in this mode before stop.
 - m. Check for Repeat Section.
 - n. Go to Section 20 (SC2000).
7. Section 20 (SC2000) - C9, 36 Pattern
 - a. If Section 20 is not selected, go to End of Test routine (CS4000).
 - b. If selected, omit delay between punches.
 - c. Store word used to generate pattern.
 - d. Punch section identification ten times.
 - e. Set cycle counter.
 - f. Generate pattern.
 - g. Punch pattern.
 - h. Repeat from f until cycle is complete.
 - j. Update cycle count.

- k. Repeat from f for required number of cycles.
 - l. Check for End of Section stop.
 - m. Return to Interrupt mode if test was in this mode before stop.
 - n. Check for Repeat Section.
 - p. Go to End of Test routine (CS4000).
8. Punch Routine (PPT).
- a. Store (A), character to be punched.
 - b. Check if test is in Interrupt mode; if so, return to SMM and wait for interrupt.
 - c. If test is in Character mode, return to monitor.
 - d. When an interrupt occurs, control goes to location (INTENTR).
 - e. If this is the first interrupt after start motion, do not punch the character. Exit punch routine.
 - f. If not, check status for Ready, Data, Busy, and Power On.
 - g. If other than these bits are set, go to 1.
 - h. If status is correct, punch the character.
 - j. Check for repeat conditions.
 - k. Exit punch routine.
 - l. If data bit is missing, try again to punch if in Character mode (error if in Interrupt mode).
 - m. If data bit is present, go to CH BT and determine incorrect status.

III. PHYSICAL REQUIREMENTS

- A. STORAGE REQUIREMENTS - approximately 1400₁₀ locations.
- B. TIMING - 0 min. 45 sec.
- C. EQUIPMENT CONFIGURATION
 - 1. 17X4 with 4K of memory.
 - 2. 1723/24/77 Paper Tape Punch.
 - 3. A device for loading the program.

1721 PAPER TAPE READER TEST

(PTR004 Test No. 4)

(CP = 2F)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

The control parameters of the reader test must correlate with the data contained within the punched paper tape. If repeat conditions, repeat section or repeat test is selected; the punch test must have previously punched a tape to provide this type of data sequence.

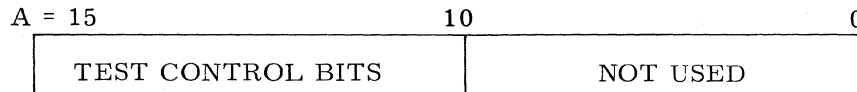
Bits 2 and 3 of the SMM parameter word must specify the correct machine type.

B. LOADING PROCEDURE

1. The test operates under control of the 1700 System Maintenance Monitor (SMM17).
2. The calling sequence is that specified by SMM17.
3. Restart test after loading by Master Clear, set P = IA and RUN.

C. PARAMETERS (The selected parameters are typed out)

1. Normal operation requires no parameters.
2. To alter parameters, enter desired parameters in A and Q registers at Parameter Stops.
3. The Identification Word (A) and Stop/Jump parameter (Q) are displayed first. Next parameter displayed is the test control information (A) and Interrupt Line Mask bit (Q). This is the line that the paper tape reader interrupt is cabled to.



Bits 0-9 (Unused): The 1721 Paper Tape Reader test cannot select the sections to be read. These sections are selected in the 1723 Punch test, and this information is relayed to the Reader test via the punched paper tape output. (See 1723 Paper Tape Punch test).

- Bits 10-15 Test control bits.
- Bit 14 Run test with 8-level patterns.
- Bit 13 Run test with 7-level patterns.
- Bit 12 Run test with 5-level patterns.
- Bit 11=1 Run test in Interrupt mode.
- Bit 11=0 Run test in Character mode.
- Bit 10=1 Delay between frames (parameter DELRD) - character mode only - Bit 11 of control word must be zero.
- Bit 10=0 No delay between reading frames.

Q = 15

INTERRUPT LINE MASK BIT

The interrupt cable may only be changed at the start of the test or at parameter input time.

The last parameter displayed (DELRD) causes the test to delay the specified number of milliseconds between reading each frame (A) - Bit 10 of Control Word = 1, Bit 11 = 0 - preset to $000F_{16}$ (Q) inapplicable.

D. MESSAGES

1. Test title

PTR004, 1721/77 READER TEST
 CP2F, VER. 4.0
 IA = XXXX, FC = XX

(XXXX = Starting address of test XX = Frequency Count).

2. End of Test Message

A	Q	A	Q
0424	S/J	PASS NO.	RETURN ADDRESS

3. Error Messages

- a. All error messages are in the format specified by SMM17.
- b. Description of error codes:

<u>Error Code</u>	<u>Description</u>
1	External reject on status request.
2	Internal reject on status request.
3	External reject on data input.
4	Internal reject on data input.
5	(Not used)
6	More than one bit punched in control frame.
7	All 10-level control punches not equal.

<u>Error Code</u>	<u>Description</u>
8	Control punch did not have even parity.
9	No pattern could be found for this control punch.
A	Interrupt occurred but Interrupt Status bit not set.
B	Improper status should be Ready, Busy, Power On, and Existence.
C	Alarm and/or Lost Data.
D	Unidentified interrupt.
E	Data read was not expected data.
F	First frame was not all ones.
10	Internal reject on Start Motion function.
11	External reject on Start Motion function.
20	Timeout waiting for Data RDY status.
21	Timeout waiting for Data status.

E. ERROR STOPS

1. First Stop

(A) = ID word (Q) = Stop/Jump parameter

2. Second Stop

(A) = Section number / (Q) = Return address
Error code

3. Third Stop (where applicable)

(A) = Expected data (Q) = Actual data

II. DESCRIPTION

A. BLOCK DIAGRAM

See diagram for Punch test.

B. SECTION DESCRIPTION

1. Initialization (INIT)

- a. Convert bias value and frequency count to ASCII and store in typeout message.

- b. Set up return address (INITB).
 - c. Type out test heading.
 - d. Return to SMM.
 - e. Enter parameters if selected in Stop/Jump word.
 - f. Set up for manual start if selected or if test was loaded from 1721 Tape Reader.
 - g. Set up Return Address (IA+5).
 - h. Return to SMM.
2. Control (CNTRL)
- SMM returns control to test through Return (IA+5).
- a. Start paper tape motion.
 - b. Read until first non-zero frame.
 - c. First non-zero frame must be all ones - error stop if it is not.
 - d. Go to first section of test (LC010).
3. Section 1 (LC010) - Zigzag pattern
- a. Check if section 1 ID is punched; if not, go to section 2 (LC020).
 - b. If punched, store section number.
 - c. Determine tape level to be read (8, 7, or 5).
 - d. Space over remaining level control punches.
 - e. Generate zigzag pattern.
 - f. Read and check one frame.
 - g. Repeat from (e) until section 1 is read and checked; then exit to next section.
4. Section 2 (LC020) - Complement Zigzag Pattern
- a. Check if section 2 ID is punched; if not, go to section 3 (LC040).
 - b. If punched, use the section 1 routine, but read the complement zigzag pattern.
5. Section 3 (LC040) - Pyramid Pattern
- a. Check if section 3 ID is punched; if not, go to section 4 (LC080).
 - b. If punched, store section number.
 - c. Determine tape level to be read (8, 7, or 5).

- d. Space over remaining level control punches.
 - e. Generate pyramid pattern.
 - f. Read and check one frame.
 - g. Repeat from (e) until section 3 is read, then exit to section 4.
6. Section 4 (LC080) - Complement Pyramid Pattern
- a. Check if section 4 ID punched; if not, go to section 5 (LC100).
 - b. If punched, use the section 3 routine, but read the complement pyramid pattern.
7. Section 5 (LC100) - All Ones, All Zeros Pattern
- a. Check if section 5 ID punched; if not, go to section 6 (LC200).
 - b. If punched, store section number.
 - c. Determine level of tape to be read (8, 7, or 5).
 - d. Generate one frame of pattern.
 - e. Read and check one frame.
 - f. Repeat from (d) until section 5 is read, then exit to next section.
8. Section 6 (LC200) - C9, 36 Pattern
- a. Check if section 6 ID punched; if not, check for end of tape punch (LC800).
 - b. If punched, store section number.
 - c. Determine level of tape to be read (8, 7, or 5).
 - d. Generate one frame of pattern.
 - e. Read and check one frame.
 - f. Repeat from (d) until section 6 is read, then exit to end of tape routine.
9. End of Tape Routine (LC800)
- a. Check for end of tape punch; error stop if not present.
 - b. Space over remaining end of tape punches.
 - c. Update pass count.
 - d. Go to SMM for end of test stop.
 - e. Load bias.
 - f. Check for repeat test and parameter re-entry.

- g. SMM returns control here if frequency number is not zero.
- h. Check for re-entry of parameters.
- j. Go to (CNTRL) to repeat test.

III. PHYSICAL REQUIREMENTS

A. STORAGE REQUIREMENTS

Approximately 850₁₀ locations.

B. TIMING - 0 min. 25 sec.

C. EQUIPMENT CONFIGURATION

1. 17X4 Computer with 4K memory.
2. 1721/22/77 per Tape Reader.
3. A device for loading test.

1720-1 PAPER TAPE PUNCH TEST
(PT1092 Test No. 92)
(CP-2C)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

Sections 1 through 8 should execute with the device unprotected but can be executed with the device protected. Section 9 operates in protected mode. Before executing test for Section 9, operator must insert jumper on controller board to protect device. Bits 2 and 3 of the SMM parameter word must specify the correct machine type.

The Static test is best for troubleshooting the device. Clear interrupts, punch in character, interrupt (set interrupts) mode, and backspacing is performed in this section; repeat conditions are available after each subsection (see SMM Reference Manual to set the repeat condition after an error or non-error condition).

Section 9 will require operator control. This test will display in A and Q or on the comment device a \$9210/\$00FF. The operator must set the PROGRAM PROTECT and SKIP switches. The operator must then type a **CR** or push GO. Upon completion of the section, the test will flash the OVERFLOW light, and the operator must take down the PROTECT switch first and the SKIP switch second; if any errors occurred, they will be printed out at that time. Considerable care must be taken when executing Section 9 and multiplexing with other tests.

B. LOADING PROCEDURE

1. The test operates under control of the 1700 System Maintenance Monitor (SMM17).
2. The calling sequence is that specified by SMM17.
3. Restart test after loading by Master Clear, set P=IA, and RUN.

C. PARAMETERS

1. Normal operation requires no parameters.
2. To alter parameters, enter desired parameters in A and Q registers at Parameter Stops.
3. The Identification Word (A) and Stop/Jump (Q) are displayed first.
 - a. CNTWRD - Control Word (A register)

Bits

- | | | | |
|------|---|---|--|
| 15 | Run test with 8 level pattern | } | (Only one bit may be set in this group.) |
| 14 | Run test with 7 level pattern | | |
| 13 | Run test with 6 level pattern | | |
| 12 | Run test with 5 level pattern | | |
| 11=1 | Run test completely in interrupt mode | | |
| =0 | Run test in both interrupt and character mode | | |

Selection:

- 9 Section 9 Protect test
 - 8 Section 8 Backstepping test
 - 7 Section 7 \$C9, \$36 pattern
 - 6 Section 6 All ones, all zero patterns
 - 5 Section 5 Complement pyramid pattern
 - 4 Section 4 Pyramid pattern
 - 3 Section 3 Complement zigzag pattern
 - 2 Section 2 Zigzag pattern
 - 1 Section 1 Static test
- b. Repeat - Number of times each pattern is to be repeated in each section (Q register)
 - c. DELAYA - Delay constant; decreasing this number decreases the switching time between interrupt and character mode (A register)
 - d. INT11 - Interrupt line mask, prestored as 2, line 1 (Q register)
 - e. Device Protected - 1 = yes and 0 = no (a register)
 - f. Unused (Q register)

D. MESSAGES

1. Test Title

PT1092, 1720-1 PUNCH TEST
CP2C, VER. 4.0
IA = XXXX, FC = XX
(XXXX = starting address of test and XX = frequency count)

2. End of Test Message

A	Q	A	Q
9224	S/J	PASS NO.	RETURN ADDRESS

3. Error Messages

a. All error messages are in the format specified by SMM17.

b. Description of error codes:

<u>Error Code</u>	<u>Description</u>
1	External reject while trying to clear punch
2	Internal reject while trying to clear punch
3	External reject while trying to read status after a clear punch
4	Internal reject while trying to read status after a clear punch
5	External reject while trying to read status before doing an output
6	Internal reject while trying to read status before doing an output
7	External reject while trying to read status after a character function has been outputted
8	Internal reject while trying to read status after a character function has been outputted
9	External reject while trying to do a device message function output
10	Internal reject while trying to do a device message function output

<u>Error Code</u>	<u>Description</u>
11	External reject while trying to read status after a device message function output
12	Internal reject while trying to read status after a device message function output
13	External reject while trying to clear interrupts
14	Internal reject while trying to clear interrupts
15	External reject while trying to enable data/function, alarm, device message interrupts
16	Internal reject while trying to enable data/function, alarm, device message interrupts
17	External reject while waiting for busy to drop after a character output
18	Internal reject while waiting for busy to drop after a character output
19	External reject trying to read status after a select interrupt
20	Internal reject trying to read status after a select interrupt
21	External reject trying to read status after an acknowledge interrupt
22	Internal reject trying to read status after an acknowledge interrupt
23	Device is still busy after a character has been outputted, character mode timeout
24	Timeout occurred waiting for an interrupt
25	Controller seems to hang busy, tried 10 times, status input
34	Interrupt was received after status was taken while in interrupt mode.
35	After equipment clear, only ready should be set and data/function
36	Alarm condition exists, device is not ready

<u>Error Code</u>	<u>Description</u>
37	Data/function not set after a character output
38	Device is still busy after a character output was found in status check routine, probably has already been flagged
39	Interrupt set in status, but test not in interrupt mode
40	Test is in interrupt mode, but interrupt bit was not set in status
41	Protect bit set in status
42	Parity error on last character punched
43	Tape low
44	Echo error, echo error status is not present
45	Device message and data/function set, but no message was sent
46	External reject while trying to output a character
47	Internal reject while trying to output a character
48	External reject while trying to read status after a character output
49	Internal reject while trying to read status after a character output
50	Expected device message to be set because test backstepped one character or tried to backstep
51	Caused a device message error, but device message status was not set
52	Caused an echo error but did not receive any indication of an echo error in status
53	External reject trying to output a director function to backstep
54	Internal reject trying to output a director function to backstep
55	Acknowledged interrupts but interrupt bit still set
56	Backstepping error, punch did not backstep correctly, operator can visually inspect tape

<u>Error Code</u>	<u>Description</u>
57	Interrupt was received after status was taken while in interrupt mode
61	Internal reject doing an input status (unprotected to a protected device)
62	External reject doing an input status (unprotected to a protected device)
63	Should have gotten an external reject when an unprotected program outputs to a protected device
64	Internal reject doing a character output (unprotected) to a protected device
65	Continuous interrupts, hardware problem, abort test

E. ERROR STOPS

1. First Stop

(A) = ID word	(Q) = Stop/Jump parameter
---------------	---------------------------
2. Section Stop

(A) = Section number/error code	(Q) = Return address
---------------------------------	----------------------
3. Third Stop

(A) = Equipment code	(Q) = Test mode at time of error
	FFFF = Character mode
	0000 = Interrupt mode
4. Fourth Stop

(A) = Function or character used when error occurred	(Q) = Status, if applicable
--	-----------------------------
5. Fifth Stop

(A) = Address of original call, if level 2 call	(Q) = Unused
---	--------------

II. DESCRIPTION

A. SECTION DESCRIPTION

1. Initialization

- a. Convert bias value and frequency count to ASCII and store in typeout message.
- b. Set up test return control entry.
- c. Type test heading.
- d. Return to SMM.
- e. Enter parameters if selected.
- f. Request interrupt line.
- g. Return to SMM.
- h. Start test in interrupt mode.
- i. Clear punch.
- j. Check status; if error, report it.
- k. Set up control entry to Section 1.
- l. Return to SMM.

2. Section 1 - Static Test

- a. Check if Section 1 ID is set; if not, go to Section 2.
- b. Pick up number of times to execute section and set repeat condition.
- c. Punch section number 10 times.
- d. Set up repeat condition and clear interrupts.
- e. Check to see if interrupt bit cleared.
- f. Set up repeat condition.
- g. Output a character.
- h. Check status.
- i. Set up repeat condition and backstep one character.
- j. Check status.
- k. Set up repeat condition and punch one character.
- l. Check status.

- m. Finished 1 through 10 pattern; if not go to f.
 - n. If number of cycles completed, do an end of section stop; otherwise, return to d.
 - o. Check for re-entry of parameters.
 - p. Check for repeat section if set and go to d; otherwise, go to Section 2.
3. Section 2 - Zigzag Pattern
- a. Check if Section 2 ID is set; if not, go to Section 3.
 - b. Set up section exit address.
 - c. Check for re-entry of parameters.
 - d. Determine punch level to be used (8, 7, 6, or 5).
 - e. Store words used to generate pattern.
 - f. Punch section number (1 or 2) 10 times.
 - g. Set up cycle counter and repeat condition.
 - h. Generate pattern.
 - i. Punch generated pattern.
 - j. Repeat from h until cycle is completed.
 - k. Check for end of section stop.
 - l. Check for repeat; if set, go to g.
 - m. Go to next section.
4. Section 3 - Complement Zigzag Pattern
- a. Check if Section 3 ID is set; if not, go to Section 4.
 - b. Set up section exit address.
 - c. Use the Section 2 routine but punch the complement zigzag pattern.
5. Section 4 - Pyramid Pattern
- a. If Section 4 is not selected, go to Section 5.
 - b. Set up section exit address.
 - c. Determine punch level to be used (8, 7, 6, or 5).
 - d. Punch section ID 10 times.
 - e. Set up cycle count and repeat condition.

- f. Generate pattern.
 - g. Punch pattern.
 - h. Repeat from f until cycle is completed.
 - i. Repeat cycle f the specified number of times.
 - j. Check for end of section stop.
 - k. Check for repeat section; if set, go to e.
 - l. Go to next section.
6. Section 5 - Complement Pyramid Pattern
- a. Check if Section 5 ID is set; if not, go to Section 6.
 - b. Set up section.
 - c. Use Section 4 routine but punch the complement pyramid pattern.
7. Section 6 - All Ones, All Zeros
- a. If Section 6 is not selected, go to Section 7.
 - b. Set up section ID.
 - c. Punch section ID 10 times.
 - d. Set up counter for number of cycles and repeat condition.
 - e. Generate pattern.
 - f. Punch pattern.
 - g. Repeat from e until cycle is complete.
 - h. Update cycle count.
 - i. Repeat from e until requested number of cycles has been punched.
 - j. Check for end of section stop.
 - k. Check for repeat section; if set, go to d.
 - l. Go to Section 7.

8. Section 7 - \$C9, \$36 Pattern

- a. If Section 7 is not selected, go to Section 8.
- b. Set up section ID.
- c. Punch section ID 10 times.
- d. Set up cycle count and repeat condition.
- e. Generate pattern.
- f. Punch pattern.
- g. Repeat from e until cycle is complete.
- h. Update cycle count.
- i. Repeat from f for required number of cycles.
- j. Check for end of section stop.
- k. Check for repeat section; if set, go to d.
- l. Go to next section.

9. Section 8 - Backstepping Test

- a. If Section 8 is not selected, go to Section 9.
- b. Punch section ID 10 times.
- c. Set up repeat condition.
- d. Punch pattern.
- e. Backstep one frame.
- f. Re-punch pattern and visually inspect tape for correct operation of section.
- g. Increment pattern to be punched.
- h. Finished 1 through 10 pattern; if not go to c.
- i. Update cycle count.
- j. Repeat from c until desired number of cycles completed.
- k. Check for end of section stop.
- l. Check for repeat section; if set, go to c.
- m. Go to next section.

10. Section 9 - Protect Test

- a. If Section 9 is not selected, go to end of test routine.
- b. Punch section ID 10 times and set repeat condition.
- c. Clear protect bit on input and output instructions.
- d. Tell operator to set protect bit and SKIP switch (A=\$9210, Q=\$00FF).
- e. Do an unprotected status request, should not reject.
- f. Do a function output, should get an external reject.
- g. Tell operator to turn off PROTECT and SKIP switches by flashing OVERFLOW light.
- h. Check for end of section stop.
- i. Check for repeat section; if set, go to c.
- j. Go to end of test.

11. End of Test Routine

- a. Punch end of tape code.
- b. Check for end of test stop.
- c. Check for re-entry of parameters.
- d. Go to SMM to update frequency count; if completed, do not return.
- e. Repeat test.

III. PHYSICAL REQUIREMENTS

A. STORAGE REQUIREMENTS - 1483₁₀

B. TIMING - 3 minutes

C. EQUIPMENT CONFIGURATION

1. 1784-1/2 with 4K of memory
2. 1720-1 Paper Tape Punch
3. A device for loading the program

1720-1 PAPER TAPE READER TEST
(PT2093 Test No. 93)
(CP 2C)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

The frequency number of the reader test must be set to the same value used when the punch test was run, or the Repeat Test option must be used.

Bits 2 and 3 of the SMM parameter word must specify the correct machine type.

The test should run in unprotect mode for Sections 1 through 8 (unless Protect Flag is set, which enables user to execute all sections in protected mode, see Section C). Section 9 will test the device in protected mode and should be run as a stand-alone section. Before executing test for Section 9, the operator must insert jumper on controller board to protect device.

Test 9 is the only section that can be manually repeated unless there was an error. Sections 1 through 8 cannot be repeated, because they are dependent upon what is on the tape and will automatically repeat themselves until a new section ID is read. Control is then transferred to that section.

The operator can repeat on a particular function by setting the SKIP switch after the error has occurred or if the repeat condition bit was already set. The test will loop on that particular function until released by the operator.

Section 1 is best for troubleshooting since it clears interrupts, reads a frame in character or interrupt mode (selects interrupts), and backspaces.

Sections 1 and 8 use the backspace capability (see the SMM Hardware Reference Manual for a detailed explanation on its operation). The reader, controller, and interface have a total of four registers. When reading a frame, the reader is either three or four frames beyond with the information stored in registers. When backspacing over the frame just read, N+1, N+2, or N+3, N, and N-1 are read. The test (Sections 1 and 8) reads eight frames, backspaces three frames, reads one frame, and expects that N+1, N, N-1, N-2, and any other condition are considered an error.

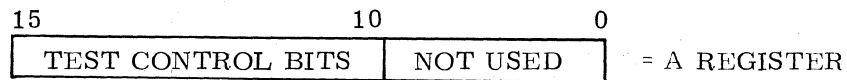
Section 9 will require operator control. This test will display in A and Q or on the comment device a \$9310/\$00FF. The operator must set the PROGRAM PROTECT and SKIP switches. The operator must then type a **CR** or push GO. Upon completion of the section, the test will flash the OVERFLOW light, and the operator must take down the PROTECT switch first and the SKIP switch second; if any errors occurred, they will be printed out at that time.

B. LOADING PROCEDURE

1. The test operates under control of the 1700 System Maintenance Monitor (SMM17).
2. The calling sequence is that specified by SMM17.
3. Restart test after loading by Master Clear, set P=IA, and RUN.

C. PARAMETERS

1. Normal operation requires no parameters.
2. To alter parameters, enter desired parameters in A and Q at parameter stops.
3. The identification word (A) and Stop/Jump parameter (Q) are displayed first. The next parameters to be displayed are the Test Control Information (A) and the Interrupt Line Mask Bit (Q); this is the line that the paper tape reader interrupt is cabled to. The last parameter to be displayed is the Delay Constant (A) which is used while delaying between frames in character mode. Q is used to signify that device is protected.



A_2 = Bits 0-9 The 1720-1 Paper Tape Reader cannot select the sections to be read. These selections are selected in the 1720-1 Punch Test, and this information is relayed to the reader test via the punched paper tape output (see 1720-1 Paper Tape Punch test).

- | | |
|------|----------------------------------|
| 10=1 | Delay between frames. |
| 10=0 | No delay between reading frames. |
| 11=1 | Run test in Interrupt mode. |
| 11=0 | Run test in Character mode. |
| 12 | 5-level tape |
| 13 | 6-level tape |
| 14 | 7-level tape |
| 15 | 8-level tape |

Q₂ = INTERRUPT LINE MASK BIT = Q REGISTER

The interrupt cable may only be changed at the start of the test or at parameter input time.

A₃ = DELAY CONSTANT = A REGISTER

This parameter causes the test to delay the specified number of milliseconds between reading each frame.

Q₃ = DEVICE IS PROTECTED (SECTION 9) = Q REGISTER

This parameter when equal to one signifies that the reader is protected; if it is zero, the device is unprotected.

D. MESSAGES

1. Test Title
PT2093, 1720-1 READER TEST
CP2V, VER. 4.0
IA=XXXX, FC=XX
(XXXX = starting address of test and XX = test frequency)

2. End of Test Message

A Q A Q
9324 S/J PASS NO. RETURN ADDRESS

3. Error Messages

- a. All error messages are in the format specified by SMM17
- b. Description of error codes:

<u>Error Code</u>	<u>Description</u>
1	External reject on status request
2	Internal reject on status request
3	External reject on data input
4	Internal reject on data input
5	Incorrect equipment address, restart test
6	Punch dropped from control frame
7	All 10-level control punches are not equal
8	Control punch did not have even parity
9	No pattern could be found for this control punch
A	Interrupt mode, but interrupt status bit not set

<u>Error Code</u>	<u>Description</u>
B	Improper status, should be ready
C	Alarm
D	Unidentified interrupt
E	Data read was not data expected
F	First frame was not all ones
21	Timeout in interrupt mode waiting for an interrupt
22	External reject on select interrupt
23	Internal reject on select interrupt
24	External reject on clear interrupt
25	Internal reject on clear interrupt
26	Controller seems to hang busy
27	Controller is busy but should not be, device should be ready to read a frame
28	Received an illegal interrupt while in character mode
29	No data interrupt received while in interrupt mode
30	Data not set, device should be ready to read
31	Alarm set, device not ready or parity error
32	Device is protected, it should be unprotected for Sections 1 through 8
33	Parity error
34	Backward motion was detected
35	Internal reject on selecting backward motion
36	External reject on selecting backward motion
37	Backward motion was selected but no backward motion detected
38	Tape level was not specified, bits 12 through 15
39	Continuous interrupts, hardware, abort test
40	Internal reject on a status input (unprotected) to a protected device

<u>Error Code</u>	<u>Description</u>
41	External reject on a status input (unprotected) to a protected device
42	Should have received an external reject doing an output (unprotected) to a protected device
43	Internal reject doing an unprotected output to a protected device
44	Not ready, tape too tight, no tape in reader, feed is too slow, or no feed holes
45	Reader is moving more than one frame per read
46	Reader is not backspacing correctly, look at expected frame and actual frame in error message to find difference (the test expects N+1, N, N-1, N-2)

E. ERROR STOPS

1. First Stop

(A) = ID word

(Q) = Stop/Jump parameter

2. Second Stop

(A) = Section number/error code

(Q) = Return address

3. Third Stop

(A) = Expected frame
(if applicable)

(Q) = Character read (if applicable)

4. Fourth Stop

(A) = Status of equipment
(if applicable)

(Q) = EQ code

5. Fifth Stop

(A) = Function (if applicable)

(Q) = Address of original call, if
level 2 call

II. DESCRIPTION

A. SECTION DESCRIPTION

1. Initialization

- a. Convert bias value and frequency count to ASCII and store in typeout message.
- b. Set up return address.
- c. Type out test heading.
- d. Return to SMM.
- e. Enter parameters if selected in Stop/Jump word.
- f. Set up return address (IA+5) and determine tape level to be read (8, 7, 6, or 5).
- g. Return to SMM.

2. Start

SMM returns control to test through return (IA+5)

- a. Request interrupt line from SMM if in interrupt mode.
- b. Read until first non-zero frame.
- c. First non-zero frame must be all ones, error stop if it is not.
- d. Go to Section 1.

3. Section 1 - Static Test

- a. Check if Section 1 ID is punched; if not, go to Section 2.
- b. If Section 1 ID is punched, store section number.
- c. Set tape level to be read (5).
- d. Space over remaining level control punches.
- e. Clear interrupts.
- f. Select interrupts if only in interrupt mode; otherwise, skip f and g.
- g. Interrupt response routine will input a data frame, input status and clear interrupts and go to i.
- h. Read a character in character mode.
- i. If eight frames have not been read, go to g or h.

- j. Selects backward motion in non-interrupt mode.
 - k. Input status to verify backward motion was selected.
 - l. Backspace over three frames.
 - m. Read over character in interrupt or character mode, check to see if expected pattern.
 - n. Repeat from g until Section 1 is read and checked; exit to next section.
4. Section 2 - Zigzag Pattern
- a. Check if Section 2 ID is punched; if not, go to Section 3.
 - b. If Section 2 ID is punched, store section number.
 - c. Space over remaining level control punches.
 - d. Generate zigzag pattern.
 - e. Read and check one frame.
 - f. Repeat from d until Section 2 is read and checked; exit to next section.
5. Section 3 - Complement Zigzag Pattern
- a. Check if Section 3 ID is punched; if not, go to Section 4.
 - b. If Section 3 ID is punched, use the Section 2 routine, but read the complement zigzag pattern.
6. Section 4 - Pyramid Pattern
- a. Check if Section 4 ID is punched; if not, go to Section 5.
 - b. If Section 4 ID is punched, store section number.
 - c. Space over remaining level control punches.
 - d. Generate pyramid pattern.
 - e. Read and check on frame.
 - f. Repeat from d until Section 4 is read and checked; exit to next section.
7. Section 5 - Complement Pyramid Pattern
- a. Check if Section 5 is punched; if not, go to Section 6.
 - b. If Section 5 ID is punched, use the Section 4 routine, but read the complement pyramid pattern.

8. Section 6 - All Ones, All Zeros Pattern
 - a. Check if Section 6 ID is punched; if not, go to Section 7.
 - b. If Section 6 ID is punched, store section number and space over control punches.
 - c. Generate one frame of pattern.
 - d. Read and check one frame.
 - e. Repeat from c until Section 6 is read and checked; exit to next section.
9. Section 7 - \$C9, \$36 Pattern
 - a. Check if Section 7 ID is punched; if not, go to Section 8.
 - b. If Section 7 ID is punched, store section number and space over control punches.
 - c. Generate one frame of data.
 - d. Read and check one frame.
 - e. Repeat from c until Section 7 is read and checked; exit to next section.
10. Section 8 - Backward Motion Test
 - a. Check if Section 8 ID punched; if not, go to Section 9.
 - b. If Section 8 ID is punched, store section number.
 - c. Set level of tape to be read (5).
 - d. Read eight frames.
 - e. Select backward motion.
 - f. Backspace over three frames.
 - g. Read and check one frame; see if it is expected pattern.
 - h. Repeat from d until Section 8 is read and checked; exit to next section.
11. Section 9 - Protect Test (Stand-Alone Test)
 - a. Device must be protected and ready.
 - b. This section will not read tape unless an error occurs other than Section ID.
 - c. Tell operator to set PROTECT and SKIP switches by displaying A=\$9310, Q=00FF. (Device must be already protected before test was started.)

- d. Clear protect bits on status input and director function.
 - e. Do an unprotected status input to a protected device, should be accepted.
 - f. Do an unprotected director function, should receive an external reject.
 - g. Flash OVERFLOW light to signal operator to take down PROTECT and SKIP switches.
 - h. Print any errors that might have occurred.
 - i. See if repeat section is set; if it is, go to c.
 - j. Exit to End of Tape routine.
12. End of Tape Routine
- a. Check for end of tape punch; error stop if not present.
 - b. Space over remaining end of tape punches.
 - c. Update pass count.
 - d. Go to SMM for end of test stop.
 - e. Load bias.
 - f. Check for repeat test and parameter re-entry.
 - g. SMM returns control if frequency number is not zero.
 - h. Check for re-entry of parameter.
 - i. Go to (CNTRL) to repeat test.

III. PHYSICAL REQUIREMENTS

A. STORAGE REQUIREMENTS

Approximately 1200₁₀ locations

B. TIMING - 45 seconds

C. EQUIPMENT CONFIGURATION

- 1. 1784-1/2 with 4K memory
- 2. 1720-1 Tape Reader
- 3. A device for loading test



1711/12/13 TELETYPE TEST
(TTY005 Test No. 5)
(CP = 2F)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. Sections 4, 5, 6, 7, and 8 are not normally run since they require operator intervention. If these sections are selected, the Teletype Test should be run alone.
2. If the computer stops with OVERFLOW light flashing when running Sections 5, 6, 7, or 8, the operator must select K mode and press Manual Interrupt to continue.
3. A minimum of four horizontal tabs must be set for Section 3 to operate correctly.
4. The paper tapes punched in Sections 5 and 7 are used as input in Sections 6 and 8, respectively. In the case of a 1713 making mode changes by computer function (parameter MOD \neq 0), the paper tape must be loaded in the reader before Sections 6 or 8 are started.
5. Parameter MOD 1 must be non-zero if teletype is a 1712.
6. Section 6 should be run in both T and KT modes (manual mode selection).
7. Bits 2 and 3 of the SMM parameter word must specify the correct machine type.
8. MANUAL INTERRUPT button must not be used to emulate setting of the SKIP switch while the TTY test is loaded.
9. REPEAT key should not be used to enter data for Section 4, because it will cause an error 16 and will require reload to recover.

B. LOADING PROCEDURE

1. Call as external test under SMM17.
2. The test may be restarted by MC, set P = IA and RUN.

C. PARAMETERS (in order in which they are entered)

1. A2, SECTNS - Section Select parameter. Prestored as 0E0F₁₆.
2. Q2, COUNTR - Set to number of lines to be typed in Section 1 and/or number of times bell is to be run in Section 11. Prestored as 0040₁₆.
3. A3, MOD - For 1712 operation or a 1713 and manual selection of modes, this parameter must be zero. For 1713 operation with Program mode selection, set to any non-zero value (prestored as zero).
4. Q3, MOD 1 - must be non-zero for 1712.
5. A4 PARMOD - Parity mode of character transmission (prestored as \$8000.
0000 = Odd parity
8000 = No parity
0001 = Even parity
6. Q4, Not used - Prestored as \$0000

D. MESSAGES

1. Initial Message

TTY005, 1711/12/13. TELETYPE TEST.

CP2F, VER. 4.0

IA = XXXX, FC = XX

(XXXX = starting address of test, XX = frequency count)

2. Error Messages

- a. All error messages are in the format specified by SMM17.
- b. When a Data Compare error occurs in Sections 6 or 8, the actual and expected data are included in the error typeout.
- c. Description of individual error codes.

<u>Error Code</u>	<u>Description</u>
1	Teletype Not Ready
2	Reply on input - reject expected.
3	Internal reject on status request
4	Internal reject on data output

<u>Error Code</u>	<u>Description</u>
7	Internal reject on function select
8	Improper status (other than Ready, Read Mode, Motor On, and EOP)
9	Data interrupt (reject on data output)
10	No Read Mode status
11	Write mode selected, Read mode status up
12	Data interrupt (reject on data input)
13	Busy status not up after data output
15	Data compare error
16	Interrupt status did not clear
18	No Busy status after an output
20	No interrupt status
21	Non-requested interrupt
22	Interrupt generated by ringing bell
24	Alarm interrupt but no alarm conditions
27	Lost data
29	EOP interrupt did not occur
30	Delay adjustment error (adjust delay on bottom even side of Z20-4 card located at C-15 so that controller never goes Not Busy when reading continuously from paper tape)
31	Parity error in even parity mode
32	Parity error in odd parity mode
33	Parity present in no parity mode

E. ERROR STOPS

1. First Stop

(A) = ID word

(Q) = Stop/Jump parameter

2. Second Stop

(A) = section number/error code

(Q) = return address

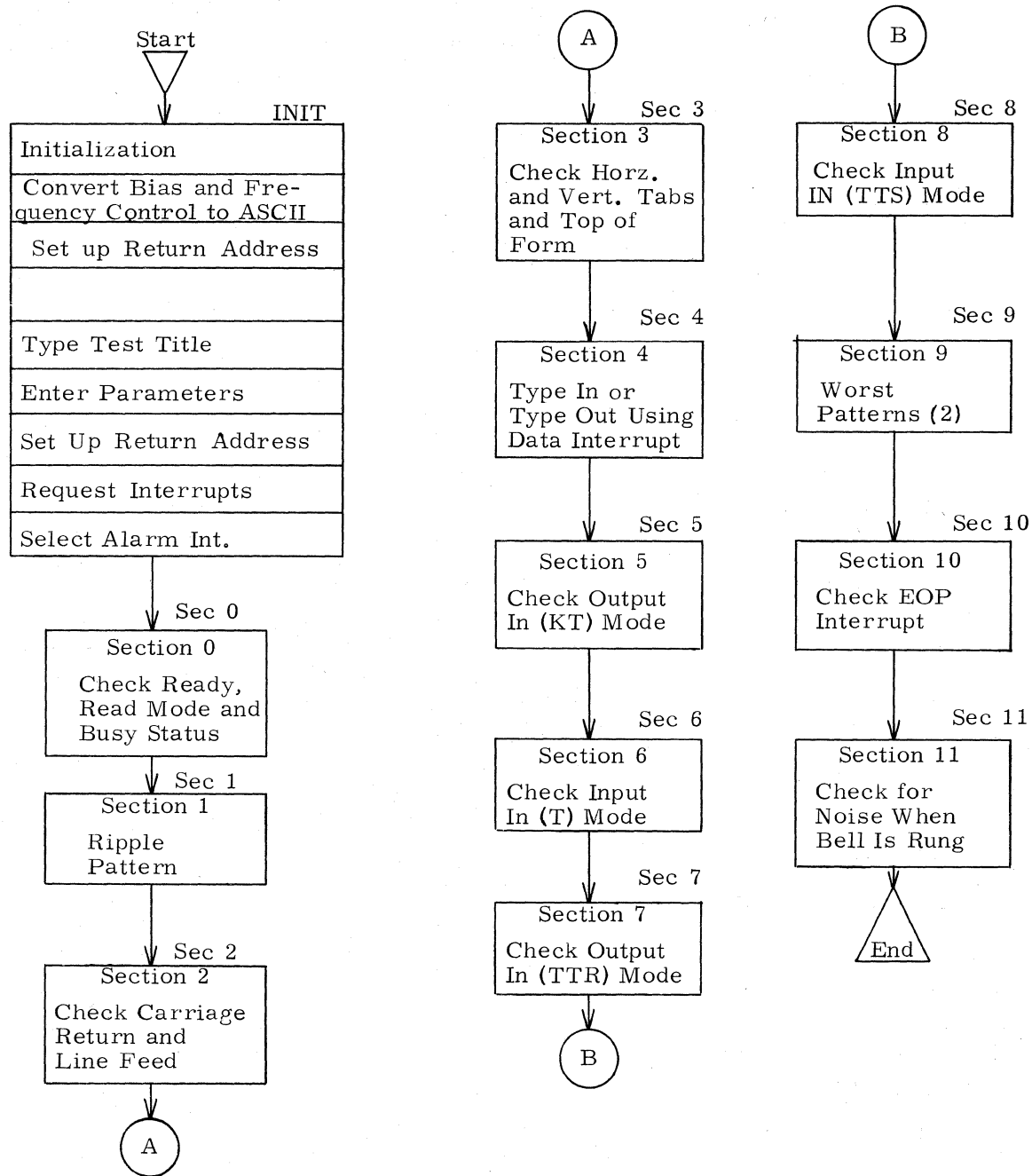
3. Third Stop (used in Sections 6 and 8)

(A) = actual data

(Q) = expected data

II. DESCRIPTION

A. BLOCK DIAGRAM



B. SECTION DESCRIPTION

1. Initialization (INIT 0)
 - a. Convert bias address and frequency count to ASCII and store in typeout routine.
 - b. Set up return address.
 - c. Type out test title.
 - d. Return control to SMM.
 - e. Enter parameters if selected.
 - f. Set up return address.
 - g. Request interrupts.
 - h. Select alarm interrupt.
 - j. Return control to SMM.
2. Section 0; Status Check (Sec 0)
 - a. Clear controller.
 - b. Check status. Error if status other than Read Mode, Ready, and End of Operation.
 - c. Check Ready and Read Mode status bits. Error if not present.
 - d. Select Write Mode, check for Read Mode status. Error if present.
 - e. Select Read Mode and check for Read Mode status. Error if not present.
 - f. Repeat from step b 500 times.
 - g. Output a character and check for Busy 10 times. Error if Not Busy.
 - h. Repeat step g six times.
 - j. Check for Repeat Section.
 - k. Go to next section.
3. Section 1; Ripple Pattern (Sec 1)
 - a. Check parameter for number of lines requested and set up line counter.
 - b. Output one line of ASCII code table.
 - c. Shift line left one character and output next line.
 - d. Repeat step c for the requested number of lines.
 - e. Check for Repeat Section.
 - f. Go to next section.

4. Section 2; Check Carriage Return and Line Feed (Sec 2)
 - a. Typeout two characters (CR).
 - b. Carriage return and line feed.
 - c. Space to end of previous typeout.
 - d. Repeat from step a until one line is typed.
 - e. Check for Repeat Section.
 - f. Go to next section.

5. Section 3; Check Horizontal and Vertical Tab and Top of Form Functions (Sec 3)
 - a. Output horizontal tab code.
 - b. Typeout: HTAB.
 - c. Repeat from step a three times.
 - d. Carriage return, line feed.
 - e. Repeat from step a three times.
 - f. Output vertical tab code.
 - g. Typeout: VTAB.
 - h. Repeat from f three times.
 - i. Carriage return, line feed.
 - j. Repeat from f.
 - k. Output Top of Form code.
 - l. Typeout: FORM.*
 - m. Repeat from k.
 - n. Two carriage return, line feed combinations.
 - o. Check for Repeat Section
 - p. Go to next section.

6. Section 4; Type Out Random Data Typed In by Operator (Sec 4)
 - a. Type out instruction to operator: TYPE IN RANDOM DATA, (200 Character Maximum) PRESS MANUAL INTERRUPT (bell rings twice).
 - b. Read in and store data as typed using data interrupt.**
 - c. Check for Manual Interrupt.

*Typeout may not be on one line since Busy status does not remain up until paper motion is completed.

**REPEAT key should not be used for data entry; it will cause an error 16 and will require reload to recover.

- d. When Manual Interrupt occurs, type out stored data also using data interrupts.
 - e. Check for Repeat Section.
 - f. Go to next section.
7. Section 5; Check Output in KT Mode (Sec 5)
- a. Output instruction to operator: SELECT (KT) MODE. PRESS MANUAL INTERRUPT.*
 - b. Punch out paper tape leader.
 - c. Generate and store 434 random characters with carriage return and line feed placed at appropriate places.
 - d. Punch and type out stored data.
 - e. Punch out paper tape leader.
 - f. Check for Repeat Section.
 - g. Go to next section.
8. Section 6; Check Input in T or KT Mode (Sec 6)
- a. Output instruction typed to operator: LOAD TAPE PUNCHED IN SECTION 5, SELECT (T) MODE, RUN PAPER TAPE READER.*
 - b. Wait for data.
 - c. Read in data from paper tape reader (data typed at same time).
 - d. Compare data read into actual data. If Compare error occurs, display actual and expected data during Error stop.
 - e. Check for Repeat Section.
 - f. Go to next section.
9. Section 7; Check Output in TTR Mode (Sec 7)
- a. Output instruction typed to operator: SELECT (TTR) MODE. PRESS MANUAL INTERRUPT).*
 - b. Wait for Manual Interrupt.
 - c. Punch out paper tape leader.
 - d. Output and store a 217-word block of the Teletype Test from memory.

*Omitted in the case of a 1713 selecting modes by computer function.

- e. Check for Repeat Section.
 - f. Go to next section.
10. Section 8; Check Input in TTS Mode (Sec 8)
- a. Output instruction typed to operator: LOAD TAPE PUNCHED IN SECTION 7, SELECT (TTS) MODE, RUN PAPER TAPE READER. *
 - b. Wait for data.
 - c. Input and store data from paper tape reader.
 - d. Compare data read into actual data stored in section 7. If data does not compare, display actual and expected data during Error stop.
 - e. Check for Repeat Section.
 - f. Go to next section.
11. Section 9; Worst Patterns (2) (Sec 9)
- a. Output five lines of first pattern (*U).
 - b. Output five lines of second pattern (=F).
 - c. Check for Repeat Section.
 - d. Go to next section.
12. Section 10; Check End of Operation Interrupt (Sec 10)
- a. Clear interrupts.
 - b. Select Write mode.
 - c. Output a character (E), Error stop if internal reject.
 - d. If reply received, select End of Operation interrupt.
 - e. Exit to SMM.
 - f. Check for EOP interrupt occurrence, Error stop if interrupt did not occur.
 - g. Repeat from step c 50 times.
 - h. Check for Repeat Section.
 - j. Go to next section.
13. Section 11; Check for Interrupt Generated by Ringing Bell (Sec 11)
- a. Load counter with parameter (COUNTR), 40_{16} unless changed during parameter stop.

*Omitted in the case of a 1713 selecting modes by computer function.

- b. Select Alarm interrupt.
 - c. Ring bell.
 - d. Check status for interrupt, error if present.
 - e. Update counter.
 - f. Repeat from step c until counter goes to zero.
 - g. Check for Repeat Section.
 - h. Go to End of Test routine.
14. Interrupt Routine (INT)
- a. Control is transferred to location INT when an interrupt occurs.
 - b. Store (Q), exit interrupt state value.
 - c. Check status for interrupt bit (error if not set).
 - d. Check whether section 10 is being executed; if so, go to step s.
 - e. Check sections 6 or 8 are being executed; if so, go to step h.
 - f. Check for Alarm interrupt; if not set, go to step h.
 - g. If Alarm interrupt, check for alarm conditions (Not Ready, Lost Data, or End of Tape*) error if none is present.
 - h. Check if interrupt was one requested (Data, Manual, or EOT), error if not requested.
 - j. Check for Manual interrupt; if not present, go to m.
 - k. Was Manual interrupt requested: If so, set occurrence flag; if not, return control to SMM.
 - l. Is Section 4 being executed? If not, clear interrupts and exit Interrupt state.
 - m. Was it a data interrupt? If not, go to step s.
 - n. Is an output requested? If not, go to step y.
 - p. Output data, update counter; if counter is zero, clear controller, clear interrupts, and exit Interrupt state.
 - q. If counter is not zero, exit Interrupt state.

*End of Tape Status is the Alarm Bit set, Ready Bit Set, and Lost Data Bit not set.

- r. Input and store data, update counter, clear interrupt status, and exit interrupt State.
- s. Check for End of Operation interrupt, set flag. If present, clear interrupts and exit Interrupt state.
- t. If EOP is not present, exit Interrupt state.

III. PHYSICAL REQUIREMENTS

- A. STORAGE REQUIREMENTS - approximately 2500₁₀ locations.
- B. TIMING - 3 min. 45 sec. with 10 lines typed in section 1.
- C. EQUIPMENT CONFIGURATION
 - 1. 1704 with 4K memory.
 - 2. 1711/12/13 Teletype
 - 3. A device for loading program if configuration is a 1711.

1740/501, 1742, 1742-20/30 LINE PRINTER TEST
(LP100C Test No. 0C)
(CP = 2F)

I. OPERATING INSTRUCTIONS

A. RESTRICTIONS

1. The three line printer interrupts, Data, End of Operation, and Alarm, may be on separate interrupt lines or on a common line to run this test.
2. Section 10, an optional section, requires manual intervention. A stop occurs near the start of the section for the operator to cause an Alarm condition on the line printer, for example, by opening the interlock. He must clear the Alarm condition at the end of the section for further testing.
3. Section 11, an optional section, requires manual intervention. A stop occurs near the start of the section for the operator to set the Protect switches on the 1704 console and on the line printer. He must clear the Protect switch on the console at the end of the section for further testing.
4. Section 12, an optional section, assumes that all the format tape levels may be selected sequentially without the paper tearing.
5. Section 13, an optional section, assumes that a change from 6 to 8 lines per inch may be made when 1742 status bit 9 is set. Do not run section 13 on the 1740/501.
6. This test will run on the 1742 and on the 1740. The 1740 does not have 6/8 line per inch select so do not run section 13 on this line printer. The 1740/501 has only eight format function codes while the 1742 has 12 format function codes. The function codes that select format channels 8 through 11 are not used in the 1740/501. There may be a print out for these levels, but they can be ignored.
7. Bits 2 and 3 of the SMM parameter word must specify the correct machine type.
8. Test sections 2, 4, 10, and 11 cannot run on a 1742-30 or -120.

B. LOADING PROCEDURE

1. Call as external test number C under SMM17.
2. Test may be restarted by MC, set P = IA, and RUN.

C. PARAMETERS

If bit 0 of the SMM Stop/Jump parameter is set at the start of the test, or if bits 10 and 0 are set at the start of succeeding passes through the test, a parameter stop occurs.

1. First Stop, A = 0C41, Q = Stop/Jump parameter. The Stop/Jump parameter may be changed if desired.
2. Second Stop, A = 00FE, Q = 0004. The bits in the A register specify the sections to be tested; bit 1 implies section 1, bit 2 implies section 2, etc. The section selection may be changed as desired. The bit in the Q register specifies the Data interrupt line. Interrupt line 0 is internal and must not be chosen for the 1742 or 1740/501.
3. Third Stop, A = 0004, Q = 0004. The bit in the A register specifies the End of Operation interrupt line. The bit in the Q register specifies the Alarm interrupt line. These interrupt line selections must be identical with the physical interrupt line connections. The interrupt line is assumed to be common and is prestored as line 2.
4. Fourth Stop, A = 004D. The A register contains the character code for the character to be used in section 8 (an optional section), the Clarify Test. This character code may be changed as desired to any legal character code.
Q = 0 = 1742 Q = 1 = 1742-20/30 LINE PRINTER FLAG.

D. MESSAGES

No timeouts occur if bit 8 of the Stop/Jump parameter is set.

1. Test title and initial address timeout:

LP100C, 1740/42/93X2 LINE PRINTER TEST CP2F, VER. 3.1 IA = XXXX, FC = XX
--

XXXX is the initial address of the test.

2. Status bit 7 set at start of test.

PROTECT STATUS SET

3. Start of Section 10

CAUSE ALARM ON LINE PRINTER

This message is also printed on the line printer, preceded and followed by a page eject. This message instructs the operator to cause an Alarm condition on the line printer (for example, by opening the interlock).

4. Start of Section 11

SET PROTECT SWITCHES ON 1704 CONSOLE AND LINE PRINTER

This message is also printed on the line printer, preceded and followed by a page eject. This message instructs the operator to set the Protect switch on the computer console and to set the Protect switch on the printer.

5. End of 1740/501, 1742 test.

A	Q	A	Q
0C24	S/J parameter	Pass Number	Return Address

6. Error Messages

a. All error messages are in the format specified by SMM17.

A	Q	A	Q
0CX8	S/J parameter	0YZZ	Return Address

X = Number of stops (if any) or number of pairs of words typed (if any)

Y = Section number

ZZ = Error code

Additional information is given, depending on the type of error, if X (number of stops) is greater than 2.

b. Description of individual error codes (See below).

E. ERROR STOPS

Stops occur upon errors if bit 3 of the Stop/Jump parameter word is set. At least two stops occur. Additional stops may occur depending on the type of error.

DESCRIPTION OF INDIVIDUAL ERROR CODES

<u>Error Code</u>	<u>Description</u>
01	Equipment address error (operator error). Test must be called again.
02	Parameter error (operator error). Parameters must be selected again.
03	Internal reject of function.
	A = contents of A when reject occurred
	Q = contents of Q when reject occurred

DESCRIPTION OF INDIVIDUAL ERROR CODES (Cont'd)

<u>Error Code</u>	<u>Description</u>
04	External reject of function. A = contents of A when reject occurred Q = contents of Q when reject occurred
05	Internal reject of status input A = 0 Q = contents of Q when reject occurred
06	External reject of status input A = 0 Q = contents of Q when reject occurred
07	Incorrect status A = actual status Q = expected status (Additional information in Section 2, loop 1A) A = word count Q = current word count
08	No data interrupt when expected after selecting data interrupt (Section 1, loop 4 or 5). A = status Q = 0
09	No data interrupt when expected after selecting data interrupt and interrupt clear simultaneously (Section 1, loop 6). A = status Q = 0
0A	No data interrupt when expected after selecting data interrupt and clear printer simultaneously (Section 1, loop 7) A = status Q = 0
0B	No data interrupt when expected after selecting data interrupt or after selecting all interrupts. A = status Q = 0
0C	No alarm interrupt when expected after output of illegal character or after cause alarm on line printer. A = status Q = 0
0D	No end of operation interrupt when expected (Section 4, loop 2 and 5). A = status Q = 0
0E	No data interrupt when expected (Section 3, loop 5 and Section 4, loops 4 and 5) A = status Q = 0

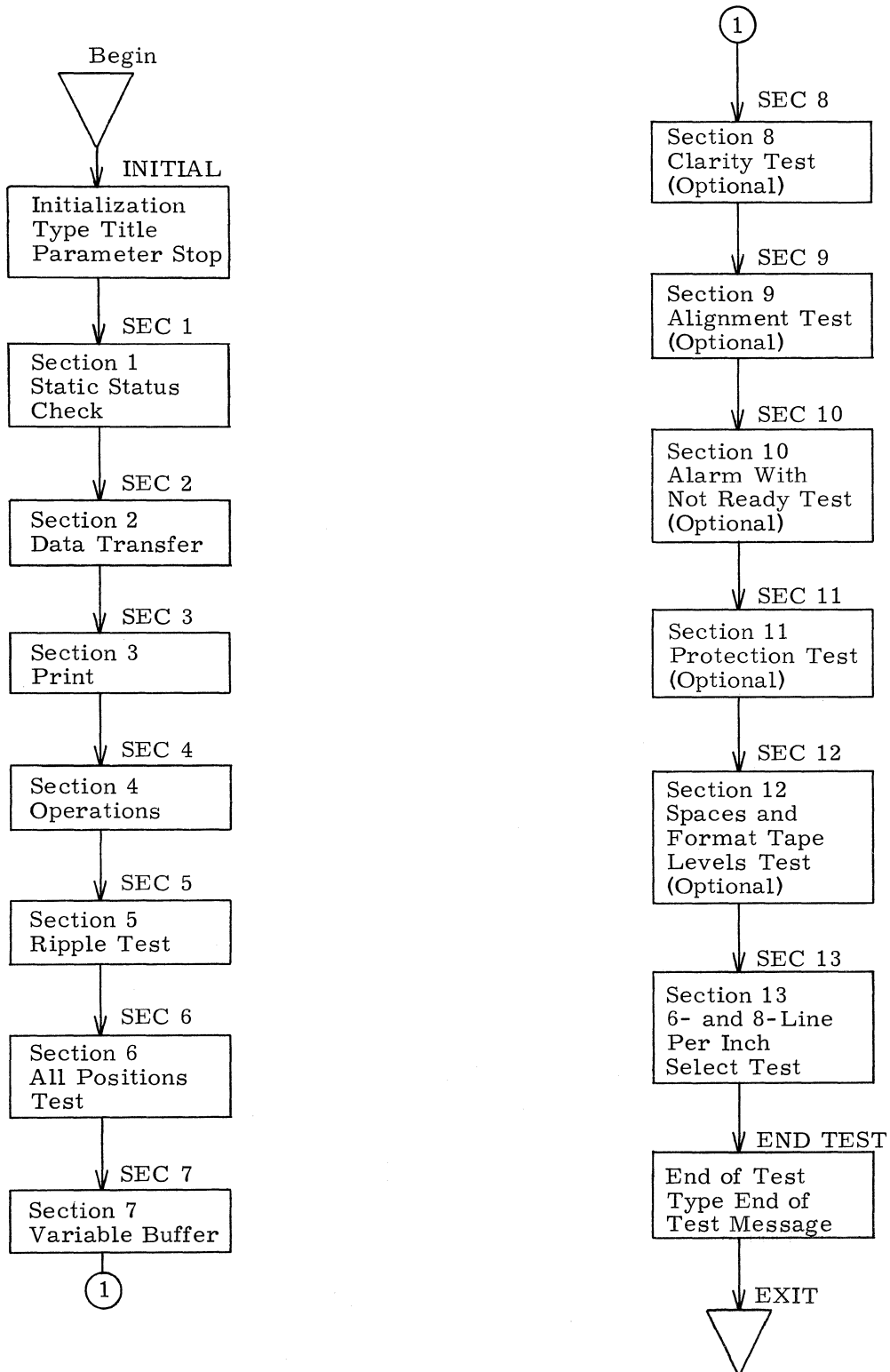
DESCRIPTION OF INDIVIDUAL ERROR CODES (Cont'd)

<u>Error Code</u>	<u>Description</u>
0F	Interrupt but no interrupt status A = status Q = 0
10	Unrequested interrupt A = status Q = 0
11	No external reject of data when expected after exceeding memory capacity by one word (69 words) A = status Q = 0
12	External data reject A = contents of A when reject occurred Q = contents of Q when reject occurred
13	Internal data reject A = contents of A when reject occurred Q = contents of Q when reject occurred
14	Wrong word count A = status Q = 0 A = actual word count Q = expected word count
15	Interrupt status failed to clear A = status Q = 0
16	Internal reject when expecting an external reject A = status Q = 0
17	No external reject when expected (clear, space, data, or during print) A = status Q = 0
18	No End of Operation status occurred on buffer data channel after buffered output of 69 or 68 words. A = line printer status Q = BDC status before termination A = current word address Q = expected word address

DESCRIPTION OF INDIVIDUAL ERROR CODES (Cont'd)

<u>Error Code</u>	<u>Description</u>
19	<p>End of Operation status occurred on buffer data channel after output of 67 or 68 words, but word count is wrong.</p> <p>A = line printer status Q = BDC status before termination A = current word address Q = expected word address</p>
1A	<p>Unexpected End of Operation status occurred after buffered output of 69 words.</p> <p>A = line printer status Q = BDC status before termination A = current word address Q = expected word address</p>
1B	<p>Unexpected external reject after an expected external reject during print.</p> <p>A = status Q = 0</p>
1C	<p>Internal reject on attempt to initiate output on buffer data channel</p> <p>A = first word address minus one of the buffer area Q = contents of Q when reject occurred</p>
1D	<p>External reject on attempt to initiate output on buffer data channel</p> <p>A = first word address minus one of the buffer area Q = contents of Q when reject occurred</p>
1E	<p>Internal reject on attempt to terminate buffer and get current address</p> <p>A = not applicable Q = equipment code</p>
1F	<p>External reject on attempt to terminate buffer and get current address</p> <p>A = not applicable Q = equipment code</p> <p>In sections 3 through 9, 12, and 13 Alarm interrupt is selected unless bit 15 of the Stop/Jump parameter is set. If bit 15 of the Stop/Jump is set, the Alarm interrupt selecting is bypassed.</p>

II. TEST DESCRIPTION
 A. BLOCK DIAGRAM



B. TEST DESCRIPTION

0. Initialization

- a. Determine initial and last addresses of test.
- b. Determine whether equipment address is legal.
- c. Type out title.
- d. Get status. Typeout if bit 7 set.
- e. Parameter stop if bit 0 of Stop/Jump word set.
- f. Return control to monitor.

1. Section 1 - STATIC STATUS CHECK

- a. Section 1, Loop 1
 - 1) Get status, drop bits 7 and 9 if set.
 - 2) Expect 0019, End of Operation, Data, Ready. Error code 07 if not.
 - 3) Loop to 1) if bit 4 of Stop/Jump word set.
- b. Section 1, Loop 2
 - 1) Select Clear Printer.
 - 2) Get status.
 - 3) Expect 0019, error code 07 if not.
 - 4) Loop to 1) if Stop/Jump bit 4 set.
- c. Section 1, Loop 3
 - 1) Select Clear Interrupts.
 - 2) Get status.
 - 3) Expect 0019, error code 07 if not.
 - 4) Loop to 1) if Stop/Jump bit 4 set.
- d. Section 1, Loop 4
 - 1) Select Data Interrupt.
 - 2) Expect interrupt, error code 08 if none.
 - 3) Get status upon interrupt.
 - 4) Expect 001D, error code 07 if not.
 - 5) Clear printer.
 - 6) Loop to 1) if Stop/Jump bit 4 set.

- e. Section 1, Loop 5
 - Same as Loop 4 except step 5) becomes
 - 5) Select Clear Printer.
- f. Section 1, Loop 6
 - Same as Loop 4 except steps 1) and 2) become
 - 1) Select Data Interrupt and Clear Interrupts simultaneously.
 - 2) Expect interrupt, error code 09 if none.
- g. Section 1, Loop 7
 - Same as Loop 4 except steps 1) and 2) become
 - 1) Select Data Interrupt and Clear Printer simultaneously.
 - 2) Expect interrupt, error code 0A if none.
- h. Section 1, Loop 8
 - 1) Select EOP Interrupt.
 - 2) Expect no interrupt, error code 10 if unexpected EOP interrupt.
 - 3) Get status.
 - 4) Expect 0019, error code 07 if not.
 - 5) Clear printer.
 - 6) Loop to 1) if Stop/Jump bit 4 set.
- i. Section 1, Loop 9
 - 1) Select Alarm Interrupt.
 - 2) Expect no interrupt, error code 10 if unexpected Alarm interrupt.
 - 3) Get status.
 - 4) Expect 0019, error code 07 if not.
 - 5) Clear printer.
 - 6) Loop to 1) if Stop/Jump bit 4 set.
- j. End of Section 1
 - 1) Stop if bit 1 of Stop/Jump word set.
 - 2) Repeat section if bit 5 of Stop/Jump word set.
- 2. Section 2 - DATA TRANSFER
 - a. Section 2, Loop 1A
 - 1) Clear printer.
 - 2) Wait Not Busy, get status.

- 3) Expect 0019, error code 07 if not. Also displayed in error message are the current word count (initially zero) and the current word being outputted (initially 2020, the first character on the drum).
- 4) Attempt output of a word.
- 5) Expect no external reject, error code 12 if reject.
- 6) Get status.
- 7) Increment word count, loop to 2) if not 68.
- 8) Wait Not Busy.
- 9) Get status.
- 10) Expect 0011, error code 07 if not.
- 11) Attempt output.
- 12) Expect external reject, error code 11 if none.
- 13) Loop to 1) if Stop/Jump bit 4 set.
- 14) Increment word, loop to 1) if not 6060.

b. Section 2, Loop 1

- 1) Clear printer and select Data interrupt simultaneously.
- 2) Expect interrupt, error code 08 if none.
- 3) In Interrupt routine output a word; do not clear interrupts in Interrupt routine.
- 4) Wait Not Busy.
- 5) Expect 68 data interrupts to have occurred, error code 14 if not.
- 6) Get status.
- 7) Expect 0011, error code 07 if not.
- 8) Loop to 1) if Stop/Jump bit 4 is set. Go to section 2, loop 5 if no 1706 (W of the equipment address is zero).

c. Section 2, Loop 2

- 1) Clear printer.
- 2) Initiate buffered output of 68 words to line printer via the 1706.
- 3) Initialize 2-millisecond counter.
- 4) Expect EOP status on 1706 before counter overflows, error code 18 if none.
- 5) Get 1706 current word address.
- 6) Expect equal to contents of first word address minus one, error code 19 if not.
- 7) Get line printer status.
- 8) Expect 0011, error code 07 if not.
- 9) Loop to 1) if Stop/Jump word 4 is set.

- d. Section 2, Loop 3
 - Same as Loop 2, except steps 2) and 8) which are:
 - 2) Initiate buffered output of 67 words to line printer via the 1706.
 - 8) Expect 0019, error code 07 if not.
- e. Section 2, Loop 4
 - 1) Clear printer.
 - 2) Initiate buffered output of 69 words to line printer via the 1706.
 - 3) Initialize 2-millisecond counter.
 - 4) Expect no EOP status on 1706 before counter overflows, error code 1A if EOP occurs.
 - 5) Terminate and get 1706 current word address.
 - 6) Expect equal to first word address plus 68, error code 19 if not.
 - 7) Get line printer status.
 - 8) Expect 0011, error code 07 if not.
 - 9) Loop to 1) if Stop/Jump bit 4 is set.
- f. Section 2, Loop 5
 - 1) Clear printer.
 - 2) Output illegal character (00).
 - 3) Wait Not Busy, get status.
 - 4) Expect 0039, error code 07 if not.
 - 5) Loop to 1) if Stop/Jump bit 4 is set.
- g. Section 2, Loop 6
 - 1) Clear printer.
 - 2) Output illegal character (00).
 - 3) Select Alarm interrupt.
 - 4) Expect interrupt, error code 0C if none.
 - 5) Get status upon interrupt.
 - 6) Expect 003D, error code 07 if not.
 - 7) Clear printer in interrupt routine.
 - 8) Loop to 1) if Stop/Jump bit 4 is set.
- h. Section 2, Loop 7
 - 1) Select EOP interrupt, Data interrupt, Clear interrupt, and Clear Printer simultaneously.
 - 2) Expect Data interrupt, error code 0B if none.

- 3) In Data interrupt routine.
 - a) Output illegal character (00).
 - b) Select Alarm interrupt, EOP interrupt, and Clear interrupts simultaneously.
 - c) Select Clear Printer.
 - 4) Expect Alarm interrupt, error code 0C if none.
 - 5) Get status upon Alarm interrupt.
 - 6) Expect 003D, error code 07 if none.
 - 7) Expect no EOP interrupt, error code 10 if EOP interrupt occurs.
 - 8) Clear printer.
 - 9) Loop to 1) if Stop/Jump bit 4 is set.
- j. End of Section 2
- 1) Stop if bit 1 of Stop/Jump Word is set.
 - 2) Repeat section if Stop/Jump bit 5 is set.
3. Section 3 - PRINT
- a. Section 3, Loop 1
- 1) Clear printer (and select Alarm interrupt simultaneously if bit 15 of the Stop/Jump word is clear)
 - 2) Output a word (initially 2021).
 - 3) Increment word, loop to 2) if not 6061.
 - 4) Print (all characters).
 - 5) Get status.
 - 6) Wait Not Busy, get status.
 - 7) Expect 0019, error code 07 if not.
 - 8) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
 - 10) Loop to 1) if Stop/Jump bit 4 is set.
- b. Section 3, Loop 2
- 1) Clear printer (and select Alarm interrupt simultaneously if bit 15 of of the Stop/Jump word is clear)
 - 2) Output all characters (32 words).
 - 3) Print.
 - 4) Attempt Clear Printer.
 - 5) Attempt Clear interrupt.
 - 6) Expect no external reject, error code 1B if reject occurs.
 - 7) Wait Not Busy.
 - 8) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
 - 9) Loop to 1) if Stop/Jump bit 4 is set.

c. Section 3, Loop 3

- 1) Clear printer (and select Alarm interrupt simultaneously if bit 15 of the Stop/Jump word is clear).
- 2) Output all characters (32 words).
- 3) Print.
- 4) Wait Not Busy.
- 5) Output a word.
- 6) Expect no external reject, error code 14 if reject occurs.
- 7) Wait Not Busy.
- 8) Increment word count (initially zero), loop to 5 if not 68.
- 9) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
- 10) Loop to 1) if Stop/Jump bit 4 is set.

d. Section 3, Loop 4

- 1) Clear printer and select EOP interrupt simultaneously (and select Alarm interrupt simultaneously if bit 15 of the Stop/Jump word is clear).
- 2) Output all characters (32 words).
- 3) Print.
- 4) Wait Not Busy.
- 5) Expect EOP interrupt, error code 0D if no EOP interrupt.
- 6) Get status upon interrupt.
- 7) Expect 001D, error code 07 if not.
- 8) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
- 9) Loop to 1) if bit 4 of Stop/Jump word is set.

e. Section 3, Loop 5

- 1) Clear printer (and select Alarm interrupt simultaneously if bit 15 of the Stop/Jump word is clear).
- 2) Output all characters (32 words).
- 3) Print.
- 4) Select Data interrupt and Clear interrupts simultaneously (and select Alarm interrupt simultaneously if bit 15 of the Stop/Jump word is clear).
- 5) Wait Not Busy.
- 6) Expect Data interrupt, error code 0E if none.
- 7) Get status upon interrupt.
- 8) Expect 001D, error code 07 if not.
- 9) Clear printer in interrupt routine.
- 10) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
- 11) Loop to 1) if bit 4 of Stop/Jump word is set.

- f. Section 3, Loop 6
 - 1) Clear printer.
 - 2) Output illegal character (00).
 - 3) Wait Not Busy, get status.
 - 4) Expect 0039, error code 07 if not.
 - 5) Print.
 - 6) Wait Not Busy, get status.
 - 7) Expect 0019, error code 07 if not.
 - 8) Clear printer.
 - 9) Loop to 1) if bit 4 of Stop/Jump word is set.
- g. End of Section 3
 - 1) Stop if bit 1 of Stop/Jump word is set.
 - 2) Repeat section if bit 5 of Stop/Jump word is set.
- 4. Section 4 - OPERATIONS
 - a. Section 4, Loop 1
 - 1) Clear printer (and select Alarm interrupt simultaneously if bit 15 of Stop/Jump Word is clear).
 - 2) Space.
 - 3) Wait Not Busy, get status.
 - 4) Expect 0019, error code 07 if not.
 - 5) Expect no Alarm interrupt, error code 10, if Alarm interrupt occurs.
 - 6) Loop to 1) if bit 4 of Stop/Jump word is set.
 - b. Section 4, Loop 2
 - 1) Clear printer and select EOP interrupt simultaneously (and select Alarm interrupt simultaneously if bit 15 of Stop/Jump word is clear).
 - 2) Space
 - 3) Wait Not Busy.
 - 4) Expect EOP interrupt, error code 0D if no EOP interrupt.
 - 5) Get status upon interrupt.
 - 6) Expect 001D, error code 07 if not.
 - 7) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
 - 8) Loop to 1) if bit 4 of Stop/Jump word is set.
 - c. Section 4, Loop 3
 - 1) Clear printer (and select Alarm interrupt simultaneously if bit 15 of Stop/Jump word is clear).

- 2) Space.
 - 3) Attempt Clear Printer.
 - 4) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
 - 5) Loop to 1) if bit 4 of Stop/Jump word is set.
- d. Section 4, Loop 4
- 1) Clear printer (and select Alarm interrupt simultaneously if bit 15 of Stop/Jump word is clear).
 - 2) Space.
 - 3) Select Data interrupt.
 - 4) Expect Data interrupt, error code 0E if none.
 - 5) Get status upon interrupt.
 - 6) Expect 000F, error code 07 if not.
 - 7) Select Clear Printer in interrupt routine.
 - 8) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
 - 9) Loop to 1) if bit 4 of Stop/Jump word is set.
- e. Section 4, Loop 5
- 1) Clear printer.
 - 2) Space.
 - 3) Select EOP interrupt, Data interrupt, and Clear interrupts simultaneously.
 - 4) Expect Data interrupt, error code 0E if none.
 - 5) Get status upon interrupt.
 - 6) Expect 000F, error code 07 if not.
 - 7) In Data Interrupt routine:
 - a) Output illegal character (00).
 - b) Select Alarm interrupt, EOP interrupt, and Clear interrupt simultaneously.
 - 8) Expect Alarm interrupt, error code 0C if none.
 - 9) Get status upon interrupt.
 - 10) Expect 002F, error code 07 if not.
 - 11) In Alarm Interrupt routine, select EOP interrupt and Clear interrupt simultaneously.
 - 12) Wait Not Busy.
 - 13) Expect EOP interrupt, error code 0D if not.
 - 14) Get status upon interrupt.
 - 15) Expect 003D, error code 07 if not.
 - 16) Clear printer.
 - 17) Loop to 1) if bit 4 of Stop/Jump word is set.

- f. End of Section 4
 - 1) Stop if bit 1 of Stop/Jump word is set.
 - 2) Repeat section if bit 5 of Stop/Jump word is set.
- 5. Section 5 - RIPPLE TEST
 - a. Clear printer (and select Alarm interrupt simultaneously if bit 15 of Stop/Jump word is set).
 - b. Select format level 1 (page eject).
 - c. Output all characters in drum order (68 words initially starting with character code 20).
 - d. Wait Not Busy.
 - e. Print.
 - f. Wait Not Busy.
 - g. Space.
 - h. Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
 - i. Loop to c if bit 4 of Stop/Jump word is set.
 - j. Shift characters left end-around one character position in computer memory.
 - k. Increment line count (initially zero), loop to c if not 136.
 - l. Wait Not Busy.
 - m. End of Section 5
 - 1) Stop if bit 1 of Stop/Jump word is set.
 - 2) Repeat section if bit 5 of Stop/Jump word is set.
- 6. Section 6 - ALL POSITIONS
 - a. Clear printer (and select Alarm interrupt if bit 15 of Stop/Jump word is set).
 - b. Output 68 words of one character (initially character code 20).
 - c. Wait Not Busy.
 - d. Print.
 - e. Wait Not Busy.
 - f. Space.
 - g. Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.

- h. Loop to b if bit 4 of Stop/Jump word is set.
 - i. Increment character code, loop to b. if not 60.
 - j. End of Section 6
 - 1) Stop if bit 1 of Stop/Jump word is set.
 - 2) Repeat section if bit 5 of Stop/Jump word is set.
7. Section 7 - VARIABLE BUFFER
- a. Clear printer (and select Alarm interrupt if bit 15 of Stop/Jump word is set).
 - b. Section 7, Loop 1 - Ascending Word Count
 - 1) Output words (initially zero words). Each word (each pair of characters) is equal to the number (in hexadecimal) of words being outputted.
 - 2) Wait Not Busy.
 - 3) Print.
 - 4) Wait Not Busy.
 - 5) Space.
 - 6) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
 - 7) Loop to 1) if Stop/Jump bit 4 is set.
 - 8) Increment word count, loop to 1) if not 68.
 - c. Section 7, Loop 2 - Decrement Word Count
 - 1) Output words (initially 68 words).
 - 2) Wait Not Busy.
 - 3) Print.
 - 4) Wait Not Busy.
 - 5) Space.
 - 6) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
 - 7) Loop to 1) if bit 4 of Stop/Jump word is set.
 - 8) Go to Loop 3 if word count is zero.
 - 9) Decrement word count, loop to 1).
 - d. Section 7, Loop 3 - Alternate Ascending and Descending Word Count
 - 1) Output words (initially zero).
 - 2) Wait Not Busy.
 - 3) Print.
 - 4) Wait Not Busy.
 - 5) Space.
 - 6) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.

- 7) Loop to 1) if bit 4 of Stop/Jump word is set.
 - 8) Increment line count (initially zero), go to end of section if 138.
 - 9) If line count is even, word count equals word count (initially zero) for last even line count plus one.
 - 10) If line count is odd, word count equals word count (initially 68) for last even line count minus one. That is, the word count will be 0, 68, 1, 67, 2, 66, through 68, 0.
 - 11) Loop to 1) for next line.
- e. End of Section 7
- 1) Wait Not Busy.
 - 2) Stop if bit 1 of Stop/Jump word is set.
 - 3) Repeat section if bit 5 of Stop/Jump word is set.
8. Section 8 - CLARITY TEST
- a. Clear printer (and select Alarm interrupt simultaneously if bit 15 of Stop/Jump word is clear).
 - b. Output 68 words of two characters each, the character code being that chosen at the parameter stop, or if no parameter stop, 4D (M).
 - c. Wait Not Busy.
 - d. Print.
 - e. Wait Not Busy.
 - f. Space.
 - g. Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
 - h. Loop to b if bit 4 of Stop/Jump word is set.
 - i. Increment line count (initially zero), loop to b if not 120.
 - j. End of Section 8
 - 1) Wait Not Busy.
 - 2) Stop if bit 1 of Stop/Jump word is set.
 - 3) Repeat section if bit 5 of Stop/Jump word is set.
9. Section 9 - ALIGNMENT TEST
- Identical to section 8 except the characters printed are hyphens (character code 2D).

10. Section 10 - ALARM WITH NOT READY TEST

a. Preliminary

- 1) Clear printer.
- 2) Select format tape level 1 (page eject).
- 3) Output message.
- 4) Wait Not Busy.
- 5) Print message to operator instructing him to cause an Alarm condition on the line printer.
- 6) Wait Not Busy.
- 7) Select format tape level 1 (page eject).
- 8) Output same message on Teletypewriter (if any) preceded and followed by carriage returns.
- 9) Select Alarm interrupt.
- 10) Stop with A = Q = 1742.
- 11) Operator must cause Alarm condition and start computer.

b. Section 10, Loop 1

- 1) Get status.
- 2) Expect 003C, error code 07 if not.
- 3) Loop to 1) if bit 4 of Stop/Jump word is set.

c. Section 10, Loop 2

- 1) Set bit for Alarm interrupt line in interrupt mask.
- 2) Expect interrupt, error code 0C if no Alarm interrupt.
- 3) Get status upon interrupt.
- 4) Expect 003C, error code 07 if not.
- 5) Clear bit for Alarm interrupt line in interrupt mask while in interrupt routine.
- 6) Loop to 1) if bit 4 of Stop/Jump word is set.

d. Section 10, Loop 3

- 1) Attempt Clear Printer.
- 2) Loop to 1) if bit 4 of Stop/Jump word is set.

e. Section 10, Loop 4

- 1) Attempt space.
- 2) Loop to 1) if bit 4 of Stop/Jump word is set.

- f. Section 10, Loop 5
 - 1) Attempt data output.
 - 2) Loop to 1) if bit 4 of Stop/Jump word is set.
 - g. End of Section 10
 - 1) Stop if bit 1 of Stop/Jump word is set.
 - 2) Operator should clear Alarm conditions on the printer.
 - 3) Repeat section if bit 5 of Stop/Jump word is set.
11. Section 11 - PROTECTION TEST
- a. Preliminary
 - 1) Clear printer.
 - 2) Select format tape level 1 (page eject).
 - 3) Output message.
 - 4) Wait Not Busy.
 - 5) Print message to operator instructing him to set the Protect switches on the computer console and on the printer.
 - 6) Wait Not Busy.
 - 7) Select format tape level 1 (page eject).
 - 8) Output same message on Teletypewriter (if any) preceded and followed by carriage returns.
 - 9) Clear Protect bits of all memory locations in computer from address 0000 through last address of test.
 - 10) Stop with A = Q = 1742.
 - 11) Operator must set Protect switches on 1704 and line printer and start computer.
 - b. Section 11, Loop 1
 - 1) Get status.
 - 2) Expect bit 7 to be set, error code 07 if not.
 - 3) Loop to 1) if bit 4 of Stop/Jump word is set.
 - c. Section 11, Loop 2
 - 1) Attempt Clear Printer.
 - 2) Expect external reject, error code 17 if not.
 - 3) Loop to 1) if bit 4 of Stop/Jump word is set.

- d. Section 11, Loop 3
 - 1) Attempt space.
 - 2) Expect external reject, error code 17 if not.
 - 3) Loop to 1) if bit 4 of Stop/Jump word is set.
 - e. Section 11, Loop 4
 - 1) Attempt data output.
 - 2) Expect external reject, error code 17 if not.
 - 3) Loop to 1) if bit 4 of Stop/Jump word is set.
 - f. End of Section 11
 - 1) Stop if bit 1 of Stop/Jump word is set.
 - 2) Operator must clear Protect switch on the computer.
 - 3) Repeat section if bit 5 of Stop/Jump word is set.
12. Section 12 - SPACES AND FORMAT TAPE LEVELS TEST
- a. Section 12, Loop 1
 - 1) Clear printer (and select Alarm interrupt simultaneously if bit 15 of Stop/Jump word is clear).
 - 2) Print out words "SINGLE SPACE".
 - 3) Wait Not Busy.
 - 4) Print.
 - 5) Wait Not Busy.
 - 6) Single space.
 - 7) Increment line count (initially zero), loop to 2) if not 9.
 - 8) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
 - 9) Loop to 2) if bit 4 of Stop/Jump word is set.
 - b. Section 12, Loop 2
 - Same as Loop 1 except for steps 2) and 6) which become:
 - 2) Print out words "DOUBLE SPACE".
 - 6) Double space.
 - c. Section 12, Loop 3
 - 1) Clear printer (and select Alarm interrupt if bit 15 of the Stop/Jump word is clear).
 - 2) Select format tape level (initially level 1).

- 3) Print out words: FORMAT TAPE LEVEL 0X, where X is the format tape level in hexadecimal. Also print out words "TOP OF FORM", if level 1, or "LAST LINE OF FORM" if level 12, or "OY SPACE (s) SHOULD PRECEDE THIS LINE" if level 2 through 11, where Y = 1 for levels 2 and 7, Y = 2 for levels 3 and 8, Y = 3 for levels 4 and 9, Y = 4 for levels 5 and 10, and Y = 5 for levels 6 and 11. (For the 1740/501, Y = 1 for levels 2 and 7, Y = 2 for level 3, Y = 3 for level 4, Y = 4 for level 5, and Y = 5 for level 6.)
- 4) Wait Not Busy.
- 5) Print.
- 6) Wait Not Busy.
- 7) Loop to 2) if bit 4 of Stop/Jump word is set.
- 8) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
- 9) If level equals 12, go to end of section.
- 10) Increment level, loop to 2.

d. End of Section 12

- 1) Clear printer.
- 2) Stop if bit 1 of Stop/Jump word is set.
- 3) Repeat section if bit 5 of Stop/Jump word is set.

13. Section 13 - 6 AND 8 LINE SELECT TEST

a. Section 13, Loop 1

- 1) Clear printer (and select Alarm interrupt simultaneously if bit 15 of Stop/Jump word is clear).
- 2) Select format tape level 1 (page eject).
- 3) Wait Not Busy, get status.
- 4) Expect status bit 9, error code 07 if not set.
- 5) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
- 6) Loop to 1) if bit 4 of Stop/Jump word is set.

b. Section 13, Loop 2

- 1) Clear printer (and select Alarm interrupt simultaneously if bit 15 of Stop/Jump word is clear).
- 2) Space.
- 3) Wait Not Busy.
- 4) Increment line count (initially zero), skip to step 8) if 4.
- 5) Get status.
- 6) Expect no bit 9, error code 07 if set.

- 7) Loop to 2).
- 8) Get status.
- 9) Expect bit 9, error code 07 if not set.
- 10) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
- 11) Loop to 1) if bit 4 of Stop/Jump word is set.

c. Section 13, Loop 3

- 1) Clear printer (and select Alarm interrupt simultaneously if bit 15 of Stop/Jump word is clear).
- 2) Select eight lines per inch.
- 3) Print out words "08 LINES PER INCH".
- 4) Wait Not Busy.
- 5) Print.
- 6) Wait Not Busy.
- 7) Space.
- 8) Increment line count (initially zero), loop to 3) if not 16.
- 9) Expect no Alarm interrupt, error code 10 if Alarm interrupt occurs.
- 10) Loop to 1) if bit 4 of Stop/Jump word is set.

d. Section 13, Loop 4

Same as Loop 3 except step 2) is omitted and step 3) becomes:

- 3) Print out words "06 LINES PER INCH".

e. End of Section 13

- 1) Stop if bit 1 of Stop/Jump word is set.
- 2) Repeat section if bit 5 of Stop/Jump word is set.

14. End of 1740/501, 1742 Test

- a. Type out end of test message if bit 8 of Stop/Jump word is clear.
- b. Stop if bit 2 of Stop/Jump word is set.
- c. Repeat test if bit 6 of Stop/Jump word is set, exit test if not. Monitor may run test again.
- d. If test is repeated, parameter stop occurs if bits 10 and 0 (and 6) are set in Stop/Jump word.

III. PHYSICAL REQUIREMENTS

- A. SPACE REQUIRED - about 2500₁₀ locations
- B. TIMING - 1 min. 30 sec.
- C. EQUIPMENT CONFIGURATION
 - 1. 1704 Computer with 4K memory
 - 2. 1705 Interrupt Data Channel
 - 3. 1706 Buffer Data Channel (optional)
 - 4. 1742 Line Printer or a 1740 Line Printer Controller and a 501 Line Printer
 - 5. A device for loading program

The following special characters will differ on the pattern print out depending on whether the BCD or ASCII drum is used.

DRUM SYMBOL

	BCD	ASCII
Not equal (Apostrophe)	≠	'
Less than or equal (Ampersand)	<=	&
Arrow Right (At Sign)	→	@
Identity (Underline)	=	—
Logical and (Question Mark)	∧	?
Logical or (Exclamation point)	∨	!
Logical not (Quotation Mark)	—	"
Arrow Up (Number Sign)	↑	#
Arrow Down (Reversed slash)	↓	/
Greater than or equal (Circumflex)	>=	^

FF524-A/1742-120/512 PRINTER TEST
(LP5A23 Test No. 23)

I. OPERATING PROCEDURE

A. RESTRICTIONS

1. This test must be run alone.
2. This test assumes a format tape has already been installed and punched as shown in Table 1.

B. LOADING PROCEDURE

1. This test is called through normal procedure from SMM17.

2. Parameter Stops

First Stop: A = 2341 Q = Stop/Jump Word
 Second Stop: A = Section Select Bit - Bit 0 corresponds to Section 0.
 Bit 1 corresponds to Section 1, etc.

Q = Train Select Set Q = 0000 for 63 character train. Set Q = 8000 for 48AN Train
 Set Q = Any positive number for 48HN Train.

Third Stop: A = Data Interrupt Line Q = End of Operation Interrupt Line
 Fourth Stop: A = Alarm Interrupt Q = N. A.

Section Select Assignments

<u>Bit</u>	<u>Section</u>	<u>Title</u>
0	0	Status Check
1	1	Spacing Test
2	2	Interrupt Status
3	3	Ripple Left Test
4	4	Ripple Right Test
5	5	Hammer Clarity
6	6	On Character in Alternating Position
7	7	Variable Buffer
8	8	Buffer Memory Test
9	9	Format Level Test
10	A	Random Pattern Test
11 through 14		Not used
15		Printer Has No Image Memory (Setting of this bit allows you to run on 1742-20 or 1742-30.)

II. MESSAGES

A. NORMAL MESSAGES

1. Title of Test: LP5A23 1742-120 LINE PRINTER W/595-4 TRAIN.
 VER. 3.1 CP2C
 IA = XXXX

TABLE 1. FORMAT TAPE CONFIGURATION TO BE USED WITH 1742-120 TESTS

Frame	Levels to be Punched												Frame	Levels to be Punched											
	1	2	3	4	5	6	7	8	9	10	11	12		1	2	3	4	5	6	7	8	9	10	11	12
1 - 67	x		x	x	x	x	x	x	x	x	x		34 - 100			x		x							
2 - 68			x										35 - 101			x	x								
3 - 69			x	x									36 - 102			x				x		x			
4 - 70			x		x								37 - 103			x	x	x	x		x		x		
5 - 71			x	x		x							38 - 104			x									
6 - 72			x					x					39 - 105			x	x								
7 - 73			x	x	x				x				40 - 106			x		x							
8 - 74			x							x			41 - 107			x	x					x			
9 - 75			x	x		x					x		42 - 108			x									
10- 76			x		x							x	43 - 109			x	x	x		x	x				
11- 77			x	x				x					44 - 110			x									
12- 78			x										45 - 111			x	x		x						
13- 79			x	x	x	x			x				46 - 112			x		x					x		
14- 80			x										47 - 113			x	x								
15- 81			x	x						x			48 - 114			x									
16- 82			x		x			x					49 - 115			x	x	x	x		x		x		
17- 83			x	x		x					x		50 - 116			x					x				
18- 84			x										51 - 117			x	x			x					
19- 85			x	x	x				x				52 - 118			x		x							
20- 86			x										53 - 119			x	x		x						
21- 87			x	x		x	x						54 - 120			x									
22- 88			x		x					x			55 - 121			x	x	x		x			x		
23- 89			x	x									56 - 122			x				x					
24- 90			x										57 - 123			x	x		x		x		x		
25- 21			x	x	x	x			x			x	58 - 124			x		x							
26- 92			x							x			59 - 125			x	x								
27- 93			x	x									60 - 126			x									
28- 94			x		x							x	61 - 127			x	x	x	x	x	x				
29- 95			x	x		x				x			62 - 128			x									
30- 96			x										63 - 129			x	x	x					x		
31- 97			x	x	x				x	x			64 - 130												
32- 98			x										65 - 131												
33- 99			x	x		x						x	66 - 132												

Cut the tape on the line at frame *132 and glue together. After the tape is glued into a loop be sure to repunch the holes in the last two frames.

B. ERROR STOPS

1. All error stops are set up like normal SMM17 errors.

First Stop: A = Ident Word
 Q = Stop/Jump Word

Second Stop: A = XYWW
 X = Section
 Y = Subsection
 WW = Error Code
 Q = Return Address (Biased)

If there are more than two stops, an explanation will be given in the description of the error.

<u>Error Code</u>	<u>Description</u>
01	Printer not ready
02	False busy status
03	No busy status when there should be
07	No data interrupt
08	No End of Operation interrupt. The third stop will be: A = status Q = 0000
0B	Unexpected interrupt. A third stop will show: A = status Q = 0000
12	No alarm interrupt. A third stop will show: A = status Q = 0000
1D	Busy too long. If busy status did not drop within 750 milliseconds after a print operation, paper motion, or error memory read, this error is given

<u>Error Code</u>	<u>Description</u>
1E	External reject on output. A third stop will show: A = status Q = 0000
1F	External reject on input
20	Internal reject on output. A third stop will show: A = status Q = 0000
21	Internal reject on input
25	Interrupt bit not set on interrupt. A third stop will show: A = status Q = 0000
26	Data bit not set in status. A third stop will show: A = status Q = 0000
27	End of operation bit not set on EOP. A third stop will show: A = status Q = 0000
28	Alarm bit not set on alarm interrupt. A third stop will show: A = status Q = 0000
29	Abnormal End of Operation bit not set on alarm interrupt. A third stop will show: A = status Q = 0000

III. SECTION DESCRIPTIONS

A. SECTION 0

1. Ready Status (Subsection 0)
 - a. Perform a clear printer function.
 - b. Check for ready status.
 - c. Repeat steps a through b 50 times. If not ready error 1 (printer not ready).
2. Busy Status - During No Operation (Subsection 1)
 - a. Perform a clear printer.
 - b. Check for busy status. If busy, error 2 (false busy status).
 - c. Repeat steps a through b 50 times.

3. Busy Status - During and After Print Operation (Subsection 2)
 - a. Perform a clear printer.
 - b. Check for busy status. If busy, error 2 (false busy status).
 - c. Print "During and after the printing of this data status responses are being checked".
 - d. Check busy status. If not busy, error 3 (no busy status when there should be).
 - e. Wait for busy to drop. If not dropped by 750 milliseconds, error 1D (busy too long).
 - f. Repeat steps c through e 10 times.

B. SECTION 1

1. Single Space Six Lines Per Inch (Subsection 0)
 - a. Advance to top of form and print: "Single space six lines per inch".
 - b. Single space, suppress space, print: "Function Code 01 is for single spacing".
 - c. Repeat step b 25 times.
2. Double Space Six Lines Per Inch (Subsection 1)
 - a. Advance to top of form and print: "Double space six lines per inch".
 - b. Double space.
 - c. Suppress space, print: "Function code 02 is for double spacing", double space.
 - d. Repeat step c 25 times.
3. Last Line of Form (Subsection 2)
 - a. Advance to last line of form. Print "Last line of form".
 - b. Repeat step a three times.
4. Top of Form (Subsection 3)
 - a. Function last line of form.
 - b. Print "Last line of form".
 - c. Advance to top of form. Suppress space and print "top of form".
 - d. Repeat c two times, each time addressing another TOF message. The first TOF message will be printed over itself three times, the second one two times, and the third one once.

5. Page Eject (Subsection 4)
 - a. Function page eject.
 - b. Print "Top of form".
 - c. Repeat a through b three times.

C. SECTION 2

1. Interrupt on Data Function (Subsection 0)
 - a. Perform a clear printer.
 - b. Start printing "Checking Data Interrupt Status".
 - c. Function Data interrupt.
 - d. Wait for busy status to drop.
 - e. Verify Data interrupt. If no interrupt, error 7 (no Data interrupt).
 - f. Check interrupt bit. If not set, error 25 (interrupt bit not set on interrupt).
 - g. Check data bit. If not set, error 26 (data bit not set on data interrupt).
 - h. Repeat steps b through g 25 times.
2. Check Clear Interrupt on Data Function
 - a. Perform a clear printer.
 - b. Start printing "Checking Clear Data Interrupt Function".
 - c. Function Data interrupt.
 - d. Function Clear Data interrupt.
 - e. Wait for busy status to drop. If interrupt, error B (unexpected, interrupt).
 - f. Repeat steps b through e 10 times.
3. Interrupt on End of Operation - Print Operation (Subsection 2)
 - a. Perform a clear printer.
 - b. Start printing "Checking End of Operation Interrupt-Print Operation".
 - c. Function EOP interrupt.
 - d. Wait for busy status to drop.
 - e. Verify EOP interrupt. If no interrupt, error 08 (no EOP interrupt).

- f. Check Interrupt bit. If not set, error 25 (interrupt bit not set on interrupt).
 - g. Check data bit. If not set, error 26 (data bit not set on data ready).
 - h. Check EOP bit. If not set, error 27 (end of operation bit not set on end of operation).
 - i. Repeat steps b through h 10 times.
4. Interrupt on End of Operation - Paper Motion (Subsection 3)
- a. Same as subsection 2 with the exception of step b. Step b is replaced with: Start printing "Checking End of Operation Interrupt - Paper Motion".

Wait for busy status to drop.

Function a double space.

This test is repeated 10 times.
5. Check Clear End of Operation Interrupt (Subsection 4)
- a. Function EOP interrupt.
 - b. Function clear EOP interrupt.
 - c. Start printing "Checking Clear End of Operation Interrupt Function".
 - d. Wait for busy status to drop.
 - e. If interrupt is received, error 0B (unexpected interrupt).
 - f. Repeat steps a through e 10 times.
6. Check Alarm Interrupt
- a. Function Alarm interrupt.
 - b. Output 10 codes.
 - c. Function print.
 - d. Wait for busy to drop.
 - e. Verify interrupt. No interrupt, error 12 (no alarm interrupt).
 - f. Check Interrupt bit. If not set, error 25 (interrupt bit not set on interrupt).
 - g. Check alarm bit. If not set, error 28 (alarm bit not set on alarm interrupt).

- h. Check abnormal EOP bit. If not set, error 29 (abnormal EOP bit not set on alarm interrupt).
- i. Repeat steps a through h 10 times.

7. Check Clear Alarm Interrupt

- a. Function clear printer.
- b. Function alarm interrupt.
- c. Function clear alarm interrupt.
- d. Output 10 codes.
- e. Function print.
- f. Verify no interrupt. If interrupt occurs, error 0B (unexpected interrupt).

D. SECTION 3

- 1. Ripple Left - six lines per inch
 - a. Print one line of all characters on train.
 - b. Shift line one character left for each line of print.
 - c. Print 136 lines.

E. SECTION 4

- 1. Ripple Right - six lines per inch
 - a. Print one line of all characters on train.
 - b. Shift line one character left for each line of print.
 - c. Print 136 lines.

F. SECTION 5

- 1. Hammer Adjustment and Clarity - six lines per inch
 - a. Print 14 lines of the letter H in even columns.
 - b. Print 14 lines of the letter H in odd columns.
 - c. Alternate rows of M and W are printed, 28 lines.

G. SECTION 6

1. Print One Character Alternating with a Space - six lines per inch
 - a. Print five lines of a character alternating with a space.
 - b. Single space between each five line group.
 - c. Repeat a and b for each character on train.

H. SECTION 7

1. Variable Buffer - six lines per inch
 - a. Print one line of one word (two characters).
 - b. Increase each successive line by one word until one full line is printed.
 - c. Decrease each successive line by one word until one word is left.

I. SECTION 8

NOTE

This section is not executed if bit 15 of parameter word A2 is set.

1. Buffer Memory Test - six lines per inch
 - a. Change image code for 4 to \$CC.
 - b. Print one full line of 3 and 4.
 - c. Repeat a and b 32 times.
 - d. Print one full line of 4 and 3.
 - e. Repeat d 32 times.

J. SECTION 9

1. Format Tape Level Test (pre-print levels)

- a. All pre-print levels are selected in such an order as to show a double space between each printed line. Each line that is printed gives the pre-print level that was selected. All lines are double spaced including the last line, level 12. The following is the order in which the preprint levels were selected.

1	4	6
4	5	4
6	4	5
8	9	7
10	8	6
7	6	8
6	4	9
9	11	4
10	4	5
11	7	12
6	8	

K. SECTION A

1. Random Pattern Test

- a. Image memory is reloaded with codes 0 through 3E.
- b. One full line of random characters is generated.
- c. Step b is repeated 132 times. If a print error occurs, an alarm interrupt will be received.
An error 0B (unexpected interrupt) will be received.

Director Functions

A00=1	Clear Printer
A01=1	Clear Interrupt
A02=1	Data Interrupt Request
A03=1	End of Operation (EOP) Interrupt Request
A04=1	Alarm Interrupt Request
A05=1	Print Request
A06-A15	Not Used

Status Codes

XXX1	Ready
XXX2	Busy
XXX4	Interrupt
XXX8	Data
XX1X	End of Operation
XX2X	Alarm
XX4X	Abnormal End of Operation
XX8X	Protected
X1XX	Load Image

ASCII CODES FOR EACH CHARACTER ON THE TRAIN

<u>Codes</u>	<u>63</u>	<u>AN</u>	<u>HN</u>
20	Space	Space	Space
21	!	None	None
22	"	+	&
23	#	None	None
24	\$	\$	\$
25	&	None	┘
26	,	None	None
27	#	@	None
28	(%	(
29)	*)
2A	*	*	*
2B	+	&	+
2C	, (Comma)	,	,
2D	-	-	-
2E	.	.	.
2F	/	/	/
30	0	0	0
31	1	1	1
32	2	2	2
33	3	3	3
34	4	4	4
35	5	5	5
36	6	6	6
37	7	7	7
38	8	8	8
39	9	9	9
3A	:	None	None
3B	;	None	None
3C	< (Less)	None	None
3D	=	*	=
3E	> (Greater)	None	None
3F	>	None	None
40	┘	None	None
41	A	A	A
42	B	B	B
43	C	C	C

<u>Codes</u>	<u>63</u>	<u>AN</u>	<u>HN</u>
44	D	D	D
45	E	E	E
46	F	F	F
47	G	G	G
48	H	H	H
49	I	I	I
4A	J	J	J
4B	K	K	K
4C	L	L	L
4D	M	M	M
4E	N	N	N
4F	O	O	O
50	P	P	P
51	Q	Q	Q
52	R	R	R
53	S	S	S
54	T	T	T
55	U	U	U
56	V	V	V
57	W	W	W
58	X	X	X
59	Y	Y	Y
5A	Z	Z	Z
5B	[None	None
5C	\	None	None
5D]	None	None
5E	^	None	None
5F	-	None	None

Order	Train Set						AN	HN
	63-1	63-2	63-3	63-4	63-5	63-6		
1	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1
3	2	2	2	2	2	2	2	2
4	3	3	3	3	3	3	3	3
5	4	4	4	4	4	4	4	4
6	5	5	5	5	5	5	5	5
7	6	6	6	6	6	6	6	6
8	7	7	7	7	7	7	7	7
9	B	\$	J	V	\$	(8	8
10	Z	→	K	W	→	G	9	9
11	,	≠	Q	X	≠	H	G	G
12	/	=	*	Y	=)	H	H
13	8	8	8	8	8	8	+	+
14	9	9	9	9	9	9	-	-
15	A	A	A	A	A	A	.	.
16	C	C	C	C	C	C	E	E
17	(B	≤	J	V	≤	A	A
18	G	Z	≥	K	W	≥	B	B
19	H	,	<	Q	X	<	C	C
20)	/	>	*	Y	>	D	D
21	R	R	R	R	R	R	R	R
22	I	I	I	I	I	I	I	I
23	F	F	F	F	F	F	F	F
24	L	L	L	L	L	L	L	L
25	M	M	M	M	M	M	M	M
26	N	N	N	N	N	N	N	N
27	O	O	O	O	O	O	O	O
28	P	P	P	P	P	P	P	P
29	{	(B	{	J	V	J	J
30	%	G	Z	%	K	W	K	K
31	"	H	,	"	Q	X	Q	Q
32	})	/	}	*	Y	x	x
33	D	D	D	D	D	D	/	/
34	U	U	U	U	U	U	S	S
35	S	S	S	S	S	S	T	T
36	T	T	T	T	T	T	U	U

Train Set

<u>Order</u>	<u>63-1</u>	<u>63-2</u>	<u>63-3</u>	<u>63-4</u>	<u>63-5</u>	<u>63-6</u>	<u>AN</u>	<u>HN</u>
37	V	#	(B	#	J	V	V
38	W	\	G	Z	\	K	W	W
39	X	^	H	,	^	Q	X	X
40	Y	!)	/	↓	*	Y	Y
41	+	+	+	+	+	+	\$	\$
42	-	-	-	-	-	-	&	&
43	,	,
44	E	E	E	E	E	E	Z	Z
45	J	V	:	(B	:		(
46	K	W	;	G	Z	;	#	=
47	Q	X	-	H	.	-	@	=
48	*	Y	=)	/	=	%)



1728 CARD READER/PUNCH TEST

(CRPAOD Test No. D)
(CP=OF)

I. OPERATING INSTRUCTIONS

A. RESTRICTIONS

1. Section 12, an optional section, requires manual intervention. A stop occurs near the start of the section for the operator to set the Protect switches on the 1704 console and on the 1728. A nonstandard stop will occur near the end of the section, with A register contents equal to the system M register setting. At this stop, clear the PROTECT switch on the console and RUN. Check the message to determine if it is End of Section or an error message.
2. Section 13, an optional section, requires manual intervention. A stop occurs near the start of the section for the operator to cause an alarm condition on the 1728, for example, by making the input hopper empty. He must clear the alarm condition at the end of the section for further testing.
3. Sections 7 and 11 will not be executed unless the 1728 is on a 1706 (or 1716) Buffered Data Channel. However, if section 7 is selected, section 6 will be executed using a random pattern and a sync check pattern.
4. The cards punched in sections 4 through 7 are to be read in sections 8 through 11. In each of sections 4 through 7, 100 cards are punched followed by an end-of-file card which is offset. More cards will be punched if Stop/Jump bit 4 is set. Cards having punch errors will also be offset.
5. The test can not be loaded into a 4K computer.
6. This test must be run alone.
7. Bits 2 and 3 of SMM parameter word must specify correct machine type.

B. LOADING PROCEDURE

Call as external test number D under SMM17. The equipment address must have bit 0 set and bits 1 through 6 all clear.

Restart test after loading by MC, set P = IA and RUN.

C. PARAMETERS

If bit 0 of the SMM stop/jump word is set at the start of the test or if bits 10 and 0 are set at the start of succeeding passes through the test, a parameter stop occurs. (Selected parameters are typed out.)

1. First stop, A = 0D31, Q Stop/Jump word.
2. Second stop, A = 00FF, Q = 0XYZ. The bits in the A register specify the sections to be tested; i. e., sections 0 through 7. The sections available are:

Section 0 - Punch Static Check

Section 1 - Reader Static Check

Section 2 - Punch Feed, Interrupt Check

Section 3 - Reader Feed, Interrupt Check

Section 4 - Punch when No Reject (52 Pattern)

Section 5 - Punch when Data Status (Shifting Ones Pattern)

Section 6 - Punch when Data Interrupt (Shifting Zeros Pattern)

Section 7 - Buffered Output (Random and Single Column Patterns)

Section 8 - Read when No Reject

Section 9 - Read When Data Status

Section 10 - Read when Data Interrupt

Section 11 - Buffered Read

Section 12 - Protection Test (optional)

Section 13 - Alarm with Not Ready Test (optional)

Q specifies the interrupt lines.

X - Data Interrupt Line

Y - End-of-Operation Interrupt Line

Z - Alarm Interrupt Line

Note the the hexadecimal digits X, Y, and Z must correspond to the physical connections of the interrupt lines.

C. 3. Third stop, A = 0028, Q = 0000

The number in A specifies the column (3-4E in hex) to be punched in the single column pattern. The number in Q specifies the delay (in milliseconds) between cards.

4. Most loops of the program are executed 100 times. To exit a loop, set bit 15 of the stop/jump word.

D. MESSAGES

No timeouts occur if bit 8 of the stop/jump word is set.

1. Test title and initial address and frequency count timeout.

CRPAOD, 1728 CARD READER/PUNCH TEST
CP2F, VER. 3.1
IA = XXXX, FC = XX

XXXX is the initial address of the test, XX is frequency count of test.

2. Start of Section 12

SET PROTECT SWITCHES ON 1704 CONSOLE AND 1728

3. Start of Section 13

CAUSE ALARM ON 1728

4. End of 1728 test.

A	Q	A	Q
OD24	Stop/Jump word	Pass number	Return address

E. ERROR MESSAGES

1. All error messages are in the SMM17 format, e.g.,

A	Q	A	Q
ODX8	Stop/Jump word	0YZZ	Return address

where

X = number of stops (if any) or number of pairs of words typed (if any),

Y = section number,

ZZ = error code.

The section number and return address tell where in the test the error occurred. The error code indicates the type of error. Additional information will be displayed, depending on the type of error, if X is greater than 2.

I. E. 2. 0D

No alarm interrupt when expected.

Additional information:

A	Q
Level 1 status	Level 2 status

0E

Unexpected alarm interrupt.

Additional information:

A	Q
Level 1 status upon interrupt	Level 2 status upon interrupt

0F

Unexpected 1706 Buffered Data Channel status

Additional information:

A	Q
Actual status	Expected status

10

Data error in card just read.

Additional information:

A	Q
Actual data	Expected data
Column number	Pattern number

Where the patterns are

Pattern 0 - Two-five (555, AAA, 555, AAA, etc.)

Pattern 1 - Shifted one (001, 002, 004, ---, 800, 001, 002, etc.)

Pattern 2 - Shifted zero (FFF, FFD, FFB, ---, 7FF, FFE, FFD, etc.)

Pattern 3 - "Random". Column 1 = Column 2 = ADDEND.
Word (N+1) = Word (N) + ADDEND.

Pattern 4 - Single-column. All columns blank except selected column, 79, 80. Selected column, Column 79 = 801. Column 80 = 004.

Pattern 5 - End-of-File card. Column 1 = 003. Rest of card blank (no parity or hole count bits).

Pattern 6 - Unidentified pattern.

Pattern 0 through 4 have even parity in column 79 and hole count in column 80.

Note: A blank card (all 0's) qualifies as a random card.

Note: If a data error occurs in one of columns 1 thru 78, no more of columns 1 through 78 will be checked unless Stop/Jump bit 4 is set.

- | | | | | | | |
|----------------|-----------------------------|---|---|---|----------------|-----------------------------|
| I. E. 2. | 11 | Wrong column count | | | | |
| | | <table border="0"> <tr> <td style="padding-right: 20px;">A</td> <td>Q</td> </tr> <tr> <td>Actual Count</td> <td>Expected count (80 decimal)</td> </tr> </table> | A | Q | Actual Count | Expected count (80 decimal) |
| A | Q | | | | | |
| Actual Count | Expected count (80 decimal) | | | | | |
| | 12 | Unidentifiable pattern.
Additional information: | | | | |
| | | <table border="0"> <tr> <td style="padding-right: 20px;">A</td> <td>Q</td> </tr> <tr> <td>First column</td> <td>Second column</td> </tr> </table> <p>Column 79 (parity) and column 80 (hole count) will be checked.</p> | A | Q | First column | Second column |
| A | Q | | | | | |
| First column | Second column | | | | | |
| | 13 | Wrong 1706 current word address.
Additional information: | | | | |
| | | <table border="0"> <tr> <td style="padding-right: 20px;">A</td> <td>Q</td> </tr> <tr> <td>Actual address</td> <td>Expected address</td> </tr> </table> | A | Q | Actual address | Expected address |
| A | Q | | | | | |
| Actual address | Expected address | | | | | |
| | 14 | Unidentifiable interrupt | | | | |
| | | <table border="0"> <tr> <td style="padding-right: 20px;">A</td> <td>Q</td> </tr> <tr> <td>Level 1 status</td> <td>Not applicable</td> </tr> </table> | A | Q | Level 1 status | Not applicable |
| A | Q | | | | | |
| Level 1 status | Not applicable | | | | | |

II. TEST DESCRIPTION

00. Initialization

- a. (INITIAL). Type title and initial address.
- b. (INITD). Determine whether legal equipment address. Error code 1 if not.
- c. (INITA). Determine whether sufficient memory. Error code 2 if not.
- d. Parameter stop. Error code 3 if parameter error.

0. Section 0 - PUNCH STATIC CHECK

- a. Initialization
 - 1) (SEC0). Go to Section 1 if parameter bit is not set.
 - 2) Initialize section.
- b. Section 0, Loop 0.
 - 1) (LOOP00). Input station 0, Level 1 status.
 - 2) Expect reply (hang on reject). Error code 4 (internal reject) or error code 5 (external reject) if not.
 - 3) (S00A). Expect ready status. Error code 7 (unexpected level 1 status) if not.

- II. 0. b. 4) Exit loop if Stop/Jump 15 is set. Loop 100 time if not. Loop indefinitely if Stop/Jump 4 is set.
- c. Section 0, Loop 1
 - 1) (LOOP01). Input station 3, Level 1 status.
 - 2) Expect reply (hang on reject). Error code 4 or 5 if not.
 - 3) (S01A). Expect Ready status. Error code 7 if not.
 - 4) Loop 100 times.
- d. Section 0, Loop 2
 - 1) (LOOP02). Select all functions on station 0, expect internal reject.
 - 2) Error code 6 if reply.
 - 3) Error code 5 if external reject and not on 1706.
 - 4) (S02A). If on 1706 expect 1706 status to be not reply, not reject, and Busy. Error code F if not. Terminate buffer.
 - 5) (S02B). Get station 0, Level 1 status. Expect ready. Error 7 if not.
 - 6) Loop 100 times.
- e. Section 0, Loop 3.
 - 1) (LOOP03). Select all functions on station 2, expect internal reject.
 - 2) Error code 6 if reply.
 - 3) Error code 5 if external reject and not on 1706.
 - 4) (S03A). If on 1706 expect 1706 status to be not reply, not reject, and Busy. Error code F if not. Terminate buffer.
 - 5) (S03B). Get station 0 status. Expect Ready. Error 7 if not.
 - 6) Loop 100 times.
- f. Section 0, Loop 4.
 - 1) (LOOP04). Input level 1, station 2 status. Expect reply (hang on reject). Error code 4 or 5 if not.

- 2) (S04A). Expect Ready and Data status. Error 7 if not.
 - 3) Loop 100 times.
- g. Section 0, Loop 5.
- 1) (LOOP05). Input level 2, Station 2 status. Expect reply (hang on reject). Error 4 or 5 if not.
 - 2) (S05A). Expect Zero status. Error 8 if not.
 - 3) Loop 100 times.
- h. Section 0, Loop 6.
- 1) (LOOP06). Clear controller on station 2. Expect reply (hang on reject). Error 4 or 5 if not.
 - 2) (S06A). Get station 2, Level 1 status. Expect Data and Ready status. Error 7 if not.
 - 3) Loop 100 times.
- i. Section 0, Loop 7.
- 1) (LOOP07). Clear interrupts on station 2. Expect reply (hang on reject). Error 4 or 5 if not.
 - 2) (S07A). Get station 2 level 1 status. Expect Data and Ready status. Error 7 if not.
 - 3) Loop 100 times.
- j. Section 0, Loop 8.
- 1) (LOOP08). Set interrupt mask bits for all three interrupts.
 - 2) (S08A). Request data interrupt on station 2. Expect reply (hang on reject). Error 4 or 5 if not.
 - 3) (S08B). Expect data interrupt. Error 9 if none.
 - 4) Expect no End-of-Operation interrupt. Error C if EOP interrupt.
 - 5) Expect no alarm interrupt. Error E if alarm interrupt.
 - 6) Get station 2, Level 1 status. Expect Data, Interrupt, and Ready status. Error 7 if not.

- II. 0. j. 7) (S08C). Clear controller on station 2. Expect reply (hang on reject). Error 4 or 5 if not.
- 8) (S08D). Loop 100 times.
- k. Section 0, Loop 9.
 - 1) (LOOP09). Set mask bits for all three interrupts.
 - 2) (S09A). Request EOP interrupt on station 2. Expect reply (hang on reject). Error 4 or 5 if not.
 - 3) (S09B). Error A if data interrupt occurs.
 - 4) Error C if EOP interrupt occurs.
 - 5) Error E if alarm interrupt occurs.
 - 6) Get station 2, Level 1 status. Expect Data and Ready status. Error 7 if not.
 - 7) (S09C). Clear controller on station 2. Expect reply (hang on reject). Error 4 or 5 if not.
 - 8) (S09D). Loop 100 times.
- l. Section 0, Loop 10.
 - 1) (LOOP0A). Set mask bits for all three interrupts.
 - 2) (S0BA). Get station 2, Level 1 status. Expect Data and Ready. Error 7 if not.
 - 3) Loop 100 times.
- n. Section 0, Loop 12.
 - 1) (LOOP0C). Select all undefined functions on station 2. Expect reply (hang on reject). Error 4 or 5 if not.
 - 2) (S0CA). Get station 2, Level 1 status. Expect Data and Ready. Error 7 if not.
 - 3) Loop 100 times.
- o. Section 0, Loop 13.
 - 1) (LOOP0D). Set mask bits for all interrupts.

- 2) (S0DA). Select all functions except Feed on station 2. Expect reply (hand on reject). Error 4 or 5 if not.
 - 3) (S0DB). Expect data interrupt. Error 9 if none.
 - 4) Error C if EOP interrupt occurs.
 - 5) Error E if alarm interrupt occurs.
 - 6) Get station 2, Level 1 status. Expect Data, Interrupt, and Ready status. Error 7 if not.
 - 7) (S0DC). Clear controller on station 2. Expect reply (hang on reject). Error 4 or 5 if not.
 - 8) (S0DD). Loop 100 times.
- p. End of Section 0.
- 1) (SEC0A). End-of-Section 0.
 - 2) (SEC0B). Repeat section if Stop/Jump 5 is set.
1. Section 1 - READER STATIC CHECK
- a. Initialization
 - 1) (SEC1). Go to Section 2 if parameter bit is not set.
 - 2) Initialize Section 1.
 - b. Execute Section 0, Loops 4 through 13 using station 1 instead of station 2 for all functions and status inputs. Expect status to be Ready instead of Data and Ready. Expect No Data interrupt and No Interrupt status upon data interrupt requests in Loops 8 and 13.
 - c. Execute Section 1, Loop 14.
 - 1) (LOOP0E). Attempt data input on station 1. Expect external reject. Error 6 or 4 if not.
 - 2) Get Station 1 status. Expect Ready status. Error 7 if not.
 - 3) Loop 100 times.
 - d. End of Section 1.
 - 1) (SEC0A). End-of-Section stop if Stop/Jump 1 is set.
 - 2) (SEC0B). Repeat section if Stop/Jump 5 is set.

II. 2. Section 2 - PUNCH FEED, INTERRUPT CHECK

a. Initialization

- 1) (SEC2). Go to Section 3 if parameter bit is not set.
- 2) Initialize section.

b. Section 2, Loop 0.

- 1) (LOOP20). Initialize column counter.
- 2) Delay
- 3) (MOTOR). Increment loop counter.
- 4) (MOTORA). Select feed and clear on reader. Expect reply (hang on reject). Error 4 or 5 if not.
- 5) (MOTORB). Get station 1 status. Expect Busy and Ready. Error ' if not.
- 6) (MOTORC). Wait for reader Not Busy.
- 7) (MOTORD). Clear controller on station 1. Expect reply (hang on reject). Error 4 or 5 if not.
- 8) (S20A). Select feed, clear, and all interrupts on station 2. Expect reply (hang on reject). Error 4 or 5 if not.
- 9) (S20D). Wait for column 1 data status.
- 10) (S20G). Expect status to be Data, Interrupt, and Ready. Error 7 if not.
- 11) Expect data interrupt. Error 9 if none.
- 12) (S20GA). Expect no alarm interrupt. Error E if alarm interrupt.
- 13) Restart loop if alarm interrupt and Stop/Jump 15 is clear. Exit loop if alarm interrupt and Stop/Jump 15 set.
- 14) (S20H). Output data (zeros) on station 2. Expect reply (restart loop if not). Error 4 or 5 if not.
- 15) (S20C). Get station 2 status. Expect Busy and Ready. Error 7 if not.

- 16) Execute steps 9) through 15) for columns 2 through 80. Wait 9 milliseconds maximum for data status in step 9). Expect status to be Data, Interrupt, Busy, and Ready in step 9).
- 17) (S20D). Wait 9 milliseconds maximum for EOP status.
- 18) (S20E). Expect status to be EOP, Interrupt, and Ready. Error 7 is not.
- 19) (S20I). Error B if not EOP interrupt.
- 20) (S20J). Clear interrupts on station 2. Expect Reply (hang on reject). Error 4 or 5 if not.
- 21) (S20K). Get Station 2 status. Expect EOP and Ready status. Error 7 if not.
- 22) (S20L). Clear controller on station 2. Expect Reply (hang on reject). Error 4 or 5 if not.
- 23) (S20M). Get Station 2 status. Expect EOP and Ready status. Error 7 if not.
- 24) (S20N). Loop 50 times.
- 25) Offset last card.

c. End of Section 2.

- 1) (SEC2A). Clear controller on station 2. Expect Reply (hang on reject). Error 4 or 5 if not.
- 2) (SEC2B). End-of-Section stop.
- 3) Repeat section if Stop/Jump 5 is set.

3. Section 3 - READER FEED, INTERRUPT CHECK

a. Initialization

- 1) (SEC3). Go to Section 4 if parameter bit is not set.
- 2) Initialize section.

b. Section 3, Loop 0.

- 1) (LOOP30). Delay
- 2) Feed a card on the reader to start the motor. Wait Not Busy.

- II. 3. b. 3) (S30A). Select Feed, Clear, and all interrupts on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
- 4) (S30B). Set mask bits for all three interrupts.
- 5) (S30C). Get station 1 status. Expect Busy and Ready.. Error 7 if not.
- 6) (S30D). Wait 32 milliseconds maximum for column 1 data status.
- 7) (S30F). Expect status to be Data, Interrupt, Busy, and Ready. Error 7 if not.
- 8) Error E if alarm interrupt.
- 9) Input data on station 1. Expect reply. Error 4 or 5 if not.
- 10) Execute steps 5) through 9) for 80 columns. In step 6) wait about 700 microseconds maximum for data status.
- 11) (S30D). Wait about 700 microseconds maximum for EOP status.
- 12) (S30DA). Expect EOP interrupt, Ready status. Error 7 if not.
- 13) (S30G). Error B if not EOP interrupt.
- 14) (S30H). Clear interrupt on station 1. Expect Reply (hang on reject). Error 4 or 4 if not.
- 15) (S30I). Get station 1 status. Expect EOP and Ready status. Error 7 if not.
- 16) (S30J). Clear controller on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
- 17) (S30K). Get station 1 status. Expect EOP and Ready status. Error 7 if not.
- 18) Loop 50 times.
- c. Section 3, Loop 1 - Force Lost Data
- 1) (LOOP31). Delay
- 2) Feed a card and wait Not Busy to start motor.
- 3) (S31A). Select Feed, Clear, EOP and Alarm interrupts. Expect Reply (hang on reject). Error 4 or 5 if not.

- 4) (S31B). Get station 1 status. Expect Busy and Ready. Error 7 if not.
- 5) (S31C). Wait 120 milliseconds maximum for EOP interrupt.
- 6) Error B if no EOP interrupt.
- 7) (S31D). Expect status upon EOP interrupt to be lost data alarm, EOP, Data, Interrupt, Not Busy, Ready. Error 7 if not.
- 8) (S31E). Error D if no alarm interrupt.
- 9) (ALA13). When alarm interrupt occurs, get status and save. Clear interrupts and select EOP interrupt.
- 10) (S31F). Expect status upon alarm interrupt to be Lost Data, Alarm, Data, Interrupt, Busy, Ready. Error 7 if not.
- 11) (S31G). Clear interrupts on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
- 12) (S31H). Get station 1 status. Expect lost data, Alarm, EOP, Data, Ready. Error 7 if not.
- 13) (S31I). Clear controller on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
- 14) (S31J). Get station 1 status. Expect EOP and Ready. Error 7 if not.
- 15) Loop 50 times.

d. End of Section 3.

- 1) (SEC3A). Clear controller on station 2. Expect Reply (hang on reject). Error 4 or 5 if not.
- 2) (SEC3B). End-of-Section stop.
- 3) (SEC3C). Repeat section if Stop/Jump 5 is set.

4. Section 4 - PUNCH WHEN NO REJECT

a. Initialization

- 1) (SEC4). Go to section 5 if parameter bit is not set.
- 2) Initialize section.
- 3) Generate pattern 0 (two-five) in output buffer area.

- II. 4. b. Section 4, Loop 0.
 - 1) (LOOP40). Initialize column count.
 - 2) Delay.
 - 3) (S40A). Select Feed, Clear, EOP and Alarm interrupts. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 4) (S40D). Attempt data output on station 2. Error 4 if internal reject. Try again if external reject.
 - 5) (S40E). Get station 3 status. Expect Busy and Ready status. Error 7 if not.
 - 6) Error E if alarm interrupt. Offset bad card.
 - 7) Error C if EOP interrupt.
 - 8) Execute steps 4) through 7) for 80 columns.
 - 9) (S40F). Wait for EOP interrupt.
 - 10) Get station 3 status. Expect EOP, Interrupt, Ready. Error 7 if not.
 - 11) Loop 100 times.
 - 12) (S40FA). Generate pattern 5 (End-of-File).
 - 13) Punch End-of-File card.
 - 14) (S40G). Offset End-of-File card.
- c. End of Section 4.
 - 1) (SEC4A). Clear controller on station 2. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 2) (SEC4B). End-of-Section stop.
 - 3) Repeat section if Stop/Jump 5 is set.
- 5. Section 5 - PUNCH WHEN DATA STATUS
 - a. Initialization
 - 1) (SEC5). Go to Section 6 if parameter bit is not set.
 - 2) Initialize section.
 - 3) Generate pattern 1 (shifting 1) in output buffer area.

b. Section 5, Loop 0.

- 1) (LOOP50). Initialize column count.
- 2) Delay.
- 3) (S50A). Select Feed, Clear, Alarm and EOP interrupts on station 2.
- 4) (S50C). Wait for data status.
- 5) Expect status to be Data and Ready. Error 7 if not.
- 6) (S50D). Output data on station 2.
- 7) (S50E). Get station 2 status. Expect Busy and Ready status. Error 7 if not.
- 8) (S50EA). Error E if alarm interrupt. Offset bad card.
- 9) Error C if EOP interrupt.
- 10) Execute steps 4) thru 9) for 80 columns. In step 5) expect status to be Data, Busy, and Ready for columns 2 through 80.
- 11) (S50F). Wait for EOP interrupt.
- 12) Get station 2 status. Expect EOP, Interrupt, and Ready. Error 7 if not.
- 13) Loop 100 times.
- 14) (S50FA). Generate pattern 5 (End-of-File).
- 15) Punch End-of-File card.
- 16) (S50G). Offset End-of-File card.

c. End of Section 5.

- 1) SEC5A). Clear controller on station 2.
- 2) (SEC5B). End-of-Section stop.
- 3) Repeat section if Stop/Jump bit 5 is set.

6. Section 6 - PUNCH WHEN DATA INTERRUPT

a. Initialization

- 1) (SEC6). Go to Section 7 if parameter bit is not set.
- 2) Initialize section.

- II.
 - 6.
 - a.
 - 3) Generate pattern 2 (shifting 0) in output buffer area.
 - b. Section 5, Loop 0.
 - 1) (LOOP60). Initialize column counter.
 - 2) Delay.
 - 3) (S60A). Select Feed, Clear, and all interrupts on station 2.
 - 4) (DATA16). When data interrupt occurs, output data. Increment column counter. Do not clear interrupts.
 - 5) (S60C). Wait for EOP or alarm interrupt.
 - 6) Error E if alarm interrupt.
 - 7) Error 11 if column count is not equal to 80.
 - 8) (S60D). Get station 2 status. Expect EOP, Interrupt, Ready. Error 7 if not.
 - 9) (S60DC). Loop 100 times.
 - 10) (S60DB). Generate pattern 5 (End-of-File).
 - 11) Punch End-of-File card.
 - 12) (S60E). Offset End-of-File card.
 - c. End of Section 6.
 - 1) (SEC6B). Clear controller on station 2.
 - 2) (SEC6C). End-of-Section stop.
 - 3) (SEC6D). Repeat section if Stop/Jump 5 is set.
 - 7. Section 7 - BUFFERED OUTPUT
 - a. Initialization
 - 1) (SEC7). Go to Section 8 if parameter bit is not set.
 - 2) Initialize section
 - 3) (SEC7A). If 1728 is not on 1706 (W=0), execute Section 6 with a new random pattern for each pass through Section 6, Loop 0, unless Stop/Jump 12 is set. In that case use the same random pattern for each pass. Also, punch single-column pattern in Section 6.

b. Section 7, Loop 0.

- 1) (LOOP70). Store first word address +80 in first word address -1 of output buffer area.
- 2) (S70A). Select Feed, Clear, EOP and Alarm interrupts on station 2.
- 3) (S70B). Initiate buffered output on 1706.
- 4) (S70C). Wait 9 milliseconds maximum after 1706 becomes Not Busy for EOP interrupt.
- 5) Error B if no EOP interrupt.
- 6) (S70D). Expect 1706 current word address upon EOP interrupt to be first word address +80. Error 13 if not.
- 7) (S70E). Get station 2 status. Expect EOP, Interrupt, and Ready. Error 7 if not.
- 8) Error E if Alarm interrupt.
- 9) (S70EA). Loop 100 times. Generate new random pattern each time unless Stop/Jump 12 is set.
- 10) (S70EB). Generate pattern 5 (end-of-file).
- 11) Punch end-of-file card.
- 12) (S70F). Offset end-of-file card.

c. Initialization of Section 7, Loop 1.

- 1) Store first word address +81 in first word address -1 of output buffer area.
- 2) Generate pattern 4 (single-column). This pattern consists of all zeros except for the selected column and columns 79 and 80.

d. Section 7, Loop 1.

- 1) (LOOP71). Select feed, clear, EOP and alarm interrupts on station 2.
- 2) (S21A). Initiate 81-word buffered output on 1706.

- II. 7. d. 3) (EOP17). When EOP interrupt occurs, get 1706 status and save. Terminate buffer if Busy. Save current word address. Get 1728 status and save.
- 4) Error B if 1706 becomes Not Busy before EOP interrupt occurs.
- 5) (S71B). Expect 1706 status upon EOP interrupt to be Reject and Busy. Error F if not.
- 6) Expect current word address upon EOP interrupt to be first word address +80. Error 13 if not.
- 7) (S71C). Get station 2 status. Expect EOP, Interrupt, Ready. Error 7 if not.
- 8) Error E if alarm interrupt.
- 9) (S71CA). Loop 10 times.
- 10) (S71CB). Generate pattern 5 (End-of-File).
- 11) Punch End-of-File card.
- 12) (S71D). Offset End-of-File card.
- e. End of Section 7.
 - 1) (SEC7B). Clear Controller on station 2.
 - 2) (SEC7C). End-of-Section stop.
 - 3) Repeat Section if Stop/Jump 5 is set.
- 8. Section 8 - READ WHEN NO REJECT
 - a. Initialization
 - 1) (SEC8). Go to Section 8 if parameter bit not set.
 - 2) Initialize section.
 - b. Section 8, Loop 8.
 - 1) (LOOP80). Initialize column counter.
 - 2) Delay.
 - 3) (S80A). Select Feed, Clear, EOP and Alarm interrupts.
 - 4) (S80C). Error E if alarm interrupt.
 - 5) Error C if EOP interrupt.

- 6) (S80D). Attempt data input. Error 4 if internal reject.
Try again if external reject.
 - 7) (S80E). Store data. Get station 1 status. Expect Busy and Ready. Error 7 if not.
 - 8) Execute Steps 4) through 7) for 80 columns.
 - 9) (S80G). Check data. Determine type of pattern from first two columns to generate rest of pattern. Compare actual with expected data. Offset card if data error.
 - 10) (S80H). Wait for EOP interrupt.
 - 11) (S80I). Expect status upon EOP interrupt to be EOP, Interrupt, Ready. Error 7 if not.
 - 12) Go to step 1) until End-of-File card is read.
- c. End of Section 8.
- 1) (SEC8A). Clear controller.
 - 2) (SEC8B). End-of-Section stop.
 - 3) Repeat section if Stop/Jump5 is set.
9. Section 9 - READ WHEN DATA STATUS
- a. Initialization
- 1) (SEC9). Go to Section 10 if parameter bit is not set.
 - 2) Initialize section.
- b. Section 9, Loop 0.
- 1) (LOOP90). Initialize column counter.
 - 2) Delay.
 - 3) (S90A). Select Feed, Clear, EOP and Alarm interrupts.
 - 4) (S90C). Error E if alarm interrupt.
 - 5) Error C if EOP interrupt.
 - 6) (S90D). Wait for data status.
 - 7) Expect status to be Data, Busy, Ready. Error 7 if not.
 - 8) (S90E). Input data.

- II.
 - 9.
 - b.
 - 9) (S90F). Save data. Get station 1 status. Expect Busy and Ready. Error 7 if not.
 - 10) Execute steps 4) through 9) for 80 columns.
 - 11) (S90H). Check data. Offset card if data error.
 - 12) (S90I). Wait for EOP interrupt.
 - 13) (S90J). Expect status upon EOP interrupt to the EOP, Interrupt, and Ready, Error 7 if not.
 - 14) Go to step 1) until End-of-File card is read.
 - c. End of Section 9.
 - 1) (SEC9A). Clear controller.
 - 2) (SEC9B). End-of-Section stop.
 - 3) Repeat section if Stop/Jump 5 is set.
- 10. Section 10 - READ WHEN DATA INTERRUPT
 - a. Initialization
 - 1) (SECA). Go to Section 11 if parameter bit is not set.
 - 2) Initialize section.
 - b. Section 10, Loop 0.
 - 1) (LOOPA0). Initialize column counter.
 - 2) Delay.
 - 3) (SA0A). Select Feed, Clear and all interrupts.
 - 4) (SA0C). Error E if alarm interrupt.
 - 5) (SA0D). Wait for EOP interrupt.
 - 6) (DATAIA). Input and store data in interrupt routine. Increment column count. Do not clear interrupts.
 - 7) Except column count to be 80. Error 11 if not.
 - 8) (SA0E). Check data. Offset card if data error.
 - 9) (SA0F). Expect status upon EOP interrupt to be EOP, Interrupt, Ready. Error 7 if not.
 - 10) Loop to 1) until End-of-File card is read.

- c. End of Section 10.
 - 1) (SECAA). Clear controller.
 - 2) (SECAB). End-of-Section stop.
 - 3) Repeat section if Stop/Jump 5 is set.

11. Section 11 - BUFFERED READ

- a. Initialization
 - 1) (SECB). Go to Section 12 if parameter bit is not set.
 - 2) Go to Section 12 if no 1706 (W=0).
 - 3) Initialize section.
- b. Section 11, Loop 0.
 - 1) (LOOPB0). Select Feed, Clear, EOP and Alarm interrupts.
 - 2) (SB0A). Initiate 80 -word buffered input.
 - 3) Wait 1 millisecond after 1706 becomes Not Busy for EOP interrupt. Error B if no EOP interrupt.
 - 4) Error E if alarm interrupt.
 - 5) (SB0B). Expect 1706 current word address to be first word address +80. Error 13 if not.
 - 6) (SB0C). Check data. Offset card if data error.
 - 7) (SB0D). Expect status upon EOP interrupt to be EOP, Interrupt, Ready. Error 7 if not.
 - 8) Loop to 1) until End-of-File card is read.
- c. Initialize Section 11, Loop 1.

Store first word address +81 in first word address - 1 of input buffer area.
- d. Section 11, Loop 1.
 - 1) (LOOPB1). Select Feed, Clear, EOP and Alarm interrupts.
 - 2) (SB1A). Initiate 81 -word buffered input.

- II. 11. d. 3) (EOP17). When EOP interrupt occurs, get 1706 status and save. Terminate buffer if Busy. Save current word address. Get 1728 status and save.
- 4) Error B if 1706 becomes Not Busy before EOP interrupt.
- 5) Error E if alarm interrupt.
- 6) (SB1B). Expect 1706 status upon EOP interrupt to be Reject and Busy. Error F if not.
- 7) Expect current word address upon EOP interrupt to be first word address +80. Error 13 if not.
- 8) (SB1C). Check data. Offset card if data error.
- 9) (SB1D). Expect 1728 status upon EOP interrupt to the EOP, Interrupt, Ready. Error 7 if not.
- 10) Loop to 1) until End-of-File card is read.
- e. End of Section 11.
 - 1) (SECBA). Clear controller.
 - 2) (SECBB). End-of-Section stop.
 - 3) Repeat section if Stop/Jump 5 is set.
- 12. Section 12 - PROTECTION TEST
 - a. Initialization
 - 1) (SECC). Go to Section 13 if parameter bit is not set.
 - 2) (SECCA). Clear protect bits in all memory locations through End-of-Test.
 - 3) (SECCB). Timeout message to operator to set Protect switches on 1728 and 1704.
 - 4) Stop with A = Q = 1728.
 - 5) Operator must set Protect switches and hit RUN.
 - b. Section 12, Loop 0.
 - 1) (LOOPC0). Get station 0 status. Expect Protected and Ready. Error 7 if not.

- 2) Loop 100 times.
 - c. Section 12, Loop 1.
 - 1) (LOOPC1). Attempt all functions on station 1. Expect external reject. Error 4 or 6 if not.
 - 2) Get station 1 status. Expect Protected and Ready. Error 7 if not.
 - 3) Loop 100 times.
 - d. Section 12, Loop 2.
 - 1) (LOOPC2). Attempt all functions on station 2. Expect external reject. Error 4 or 6 if not.
 - 2) Get station 2 status. Expect Protected, Data, Ready. Error 7 if not.
 - 3) Loop 100 times.
 - e. End of Section 12.
 - 1) Set bit 1 of stop/jump word.
 - 2) End-of-Section stop.
 - 3) Repeat section if Stop/Jump 5 is set.
13. Section 13 - ALARM WITH NOT READY TEST
- a. Initialization
 - 1) (SECD). Go to ENDTEST if parameter bit is not set.
 - 2) (SECDA). Select Clear Controller and Alarm interrupt.
 - 3) (SECDB). Type out message to operator to cause alarm on 1728.
 - 4) Stop with A = Q = 1728.
 - 5) Operator must cause alarm on 1728 and hit RUN.
 - b. Section 13, Loop 0.
 - 1) (LOOPD0). Get station 0 status. Expect Alarm, Interrupt, Not Ready status. Error 7 if not.
 - 2) Set Alarm interrupt mask bit.

- II. 13. b. 3) Error D if no alarm interrupt.
- 4) Loop 100 times.
- c. Section 13, Loop 1.
 - 1) (LOOPD1). Select all functions on punch. Expect external reject. Error 4 or 6 if not.
 - 2) Get station 2 status. Expect Alarm, Interrupt, Not Ready. Error 7 if not.
 - 3) Loop 100 times.
- d. Section 13, Loop 2.
 - 1) (LOOPD2). Select all functions on reader. Expect external reject. Error 4 or 6 if not.
 - 2) Get station 2 status. Expect Alarm, Interrupt, Not Ready. Error 7 if not.
 - 3) Loop 100 times.
- e. End of Section 13.
 - 1) Set Stop/Jump bit 1.
 - 2) End-of-Section stop.
 - 3) Repeat section if Stop/Jump 5 is set.
- 14. End of 1728 test.
 - (ENDTEST). Timeout End-of-Test message. Stop if Stop/Jump 2 is set. Repeat test if Stop/Jump 6 is set.
 - (RESTART). Re-enter parameters if Stop/Jump 10 is set.

III. PHYSICAL REQUIREMENTS

- A. STORAGE REQUIREMENTS - Approximately 3070_{10} .
- B. TIMING - Variable depending upon the size of test deck.
- C. EQUIPMENT CONFIGURATION
 - 1. 17X4 Computer with 8K memory.
 - 2. 1728 Card Reader
 - 3. A device for loading test.

1729-2/3/411 CARD READER TEST

(CR3A13 Test No. 13)
(CP=OF)

I. OPERATING INSTRUCTIONS

A. RESTRICTIONS

1. Section 12, an optional section, requires manual intervention. A stop occurs near the start of the section for the operator to set the Protect switches on the 17X4 console and on the 1729-2. He must clear the Protect switch on the console at the end of the section for further testing.
2. Section 13, an optional section, requires manual intervention. A stop occurs near the start of the section for the operator to cause an alarm condition on the 1729-2 for example, by making the input hopper empty. He must clear the alarm condition at the end of the section for further testing.
3. Section 11 will not be executed unless the 1729-2/3 is on a 1706 (or 1716) Buffered Data Channel.
4. Special test decks are used for each section 8 through 11. In each of sections 8 through 11, cards are read through the end-of-file card. Cards having data errors are offset, except on 1729-3.
5. The test cannot be loaded into a 4K computer.
6. This test must be run alone.
7. SMM parameter bits 2 and 3 must be correctly set before executing this test to ensure correct timing constants.
8. Section 12 and 13 cannot be run on a 1729-3 card reader.

B. LOADING PROCEDURE

Call as external test number 13 under SMM17. The equipment address must have bit 0 set and bits 1 through 6 all clear.

Restart test after loading by MC, set P = IA and run.

C. PARAMETERS

If bit 0 of the SMM Stop/Jump word is set at the start of the test or at the start of succeeding passes through the test, a parameter stop occurs. A typeout of selected parameters will occur.

1. First stop, A = 1341, Q = Stop/Jump word.
2. Second stop, A = 003F, Q = 0XYZ. The bits in the A register specify the sections to be tested, i. e., sections 0, 3, 8, 9, 10, and 11. The sections available are:

<u>Section</u>		<u>A Register</u>
0	- Reader Static Check	Bit 0
3	- Reader Feed, Interrupt Check	Bit 1

<u>Section</u>		<u>A Register</u>
8	- Read when No Reject	Bit 2
9	- Read when Data Status	Bit 3
10	- Read when Data Interrupt	Bit 4
11	- Buffered Read	Bit 5
12 (optional)	- Protection Test	Bit 6
13 (optional)	- Alarm with Not Ready Test	Bit 7

Q specifies the interrupt lines.

X - Data Interrupt Line

Y - End-of-Operation Interrupt Line

Z - Alarm Interrupt Line

Note: The hexadecimal digits X, Y and Z must correspond to the physical connections of the interrupt lines.

3. Third stop, A= 0028, Q - 0000
The number in A specifies the column (3-4E in hex) to be expected in the single column pattern. The number in Q specifies the maximum delay (in milliseconds) between cards. This value will be decremented down to 0 then turn back to original amount in 50₁₀ even increments.
4. Fourth stop, Card Reader Type (A = 0 1729-2, A = 1 1729-3)
(Q = zero)
5. Most loops of the program are executed 100 times. To exit a loop, set bit 15 of the Stop/Jump word.

D. MESSAGES

No timeouts occur if bit 8 of the Stop/Jump word is set

1. Test title, initial address and frequency count typeout.

CR3A13, 1729-2/3 CARD READER TEST

CPOF, Ver. 3.1
IA = XXXX, FC = XX

XXXX is the initial address of the test, XX is the frequency count

2. Start of Section 12

SET PROTECT SWITCHES ON 17X4 CONSOLE AND 1729-2/3, HIT RUN - IF NON-STD STOP WITH A-REG EQUAL M-REG-CLEAR PROTECT SWITCHES AND HIT RUN, CHECK TYPEOUT FOR END SECT OR ERROR

3. Start of Section 13

CAUSE ALARM ON 1729-2/3, HIT RUN

4. End of 1729-2/3 test

A	Q	A	Q
1324	Stop/Jump word	Pass number	Return address

E. ERROR MESSAGES

1. All error messages are in the SMM17 format, i. e.,

A	Q	A	Q
13X8	Stop/Jump word	0YZZ	Return Address

where

X = number of stops (if any) or number of pairs of words typed (if any),

Y = section number,

ZZ = error code

The section number and return address tell where in the test the error occurred.

The error code indicates the type of error. Additional information will be displayed, depending on the type of error, if X is greater than 2.

2. Types of errors.

<u>Error Code</u>	<u>Meaning</u>				
01	Equipment address in error (operator error) Test must be called again				
02	Insufficient memory for test.				
03	Parameter in error (operator error) Parameters must be selected again.				
04	Unexpected internal reject				
05	Unexpected external reject				
06	Unexpected reply				
07	Unexpected level 1 status Additional information: <table border="0" style="margin-left: 40px;"> <tr> <td style="text-align: center;">A</td> <td style="text-align: center;">Q</td> </tr> <tr> <td>Actual status</td> <td>Expected status</td> </tr> </table>	A	Q	Actual status	Expected status
A	Q				
Actual status	Expected status				
08	Unexpected level 2 status Additional information: <table border="0" style="margin-left: 40px;"> <tr> <td style="text-align: center;">A</td> <td style="text-align: center;">Q</td> </tr> <tr> <td>Actual status</td> <td>Expected status</td> </tr> </table>	A	Q	Actual status	Expected status
A	Q				
Actual status	Expected status				
09	No data interrupt when expected Additional information: <table border="0" style="margin-left: 40px;"> <tr> <td style="text-align: center;">A</td> <td style="text-align: center;">Q</td> </tr> <tr> <td>Level 1 status</td> <td>0000</td> </tr> </table>	A	Q	Level 1 status	0000
A	Q				
Level 1 status	0000				

0A Unexpected data interrupt
 Additional information:
 A Q
 Level 1 status upon interrupt 0000

0B No End-of-Operation interrupt when expected
 Additional information:
 A Q
 Level 1 status 0000

0C Unexpected End-of-Operation interrupt
 Additional information:
 A Q
 Level 1 status upon interrupt 0000

0D No Alarm interrupt when expected
 Additional information:
 A Q
 Level 1 status Level 2 status

0E Unexpected alarm interrupt
 Additional information:
 A Q
 Level 1 status upon interrupt Level 2 status upon
 interrupt

0F Unexpected 1706 Buffered Data Channel status
 Additional information:
 A Q
 Actual status Expected status

10 Data error in card just read
 Additional information:
 A Q
 Actual data Expected data
 Column number Pattern number

Where the pattern is
 Pattern 0 - Two-five (555, AAA, 555, AAA, etc.)
 Pattern 1 - Shifted one (001, 002, 004, ---, 800, 001,
 002, etc.)

Pattern 2 - Shifted zero (FFE, FFD, FFB, ---, 7FF, FFE, FFD, etc.)

Pattern 3 - "Random". Column 1 = Column 2 = ADDEND.
Word (N+1) = Word (N) + ADDEND.

Pattern 4 - Single-column. All columns blank except selected column, 79, 80. Selected column, Column 79 = 801. Column 80 = 004.

Pattern 5 - End-of-File card. Column 1 = 003. Rest of card blank (no parity or hole count bits).

Pattern 6 - Unidentified pattern.

Pattern 0 through 4 have Even parity in column 79 and hole count in column 80.

Note: A blank card (all zeros) qualifies as a random card.

Note: If a data error occurs in one of columns 1 through 78, no more of columns 1 through 78 will be checked unless Stop/Jump bit 4 is set.

11 Wrong Column Count

A = Actual Count Q = Expected Count (80 decimal)

12 Unidentifiable pattern

Additional information:

A	Q
First column	Second column

Column 79 (parity) and column 80 (hole count) will be checked.

13 Wrong 1706 current word address

Additional information:

A	Q
Actual address	Expected address

14 Unidentifiable interrupt

A	Q
Level 1 status	0000

II. TEST DESCRIPTION

00. Initialization

- a. (INITIAL). Type title and initial address.
- b. (INITD). Determine whether legal equipment address. Error code 1 if not.
- c. (INITA). Determine whether sufficient memory. Error code 2 if not.
- d. Parameter stop. Error code 3 if parameter error.

0. Section 0 - READER STATIC CHECK

a. Initialization

- 1) (SECO). Go to Section 3 if parameter bit is not set.
- 2) Initialize section.

b. Section 0, Loop 0.

- 1) (L00P00). Input station 0, level 1 status.
- 2) Expect Reply (hang on reject). Error code 4 (internal reject) or error code 5 (external reject) if not.
- 3) (S00A). Expect Ready status. Error code 7 (unexpected level 1 status) if not.
- 4) Exit loop if Stop/Jump 15 is set. Loop 100 times if not. Loop indefinitely if Stop/Jump 4 is set.

c. Section 0, Loop 1

- 1) (L00P01). Input station 3, level 1 status.
- 2) Expect Reply (hang on reject). Error code 4 or 5 if not.
- 3) (S01A). Expect Ready status. Error code 7 if not.
- 4) Loop 100 times.

d. Section 0, Loop 2

- 1) (L00P02). Select all functions on station 0, expect internal reject.
- 2) Error code 6 if reply.
- 3) Error code 5 if external reject and not on 1706.
- 4) (S02A). If on 1706 expect 1706 status to be Not Reply, Not Reject, and Busy. Error code F if not. Terminate buffer.
- 5) (S02B). Get station 0, level 1 status. Expect Ready. Error 7 if not.
- 6) Loop 100 times.

- e. Section 0, Loop 3.
 - 1) (L00P03). Select all functions on station 3, expect internal reject.
 - 2) Error code 6 if reply.
 - 3) Error code 5 if external reject and not on 1706.
 - 4) (S03A). If on 1706 expect 1706 status to be Not Reply, Not Reject, and Busy. Error code F if not. Terminate buffer.
 - 5) (S03B). Get station 0 status. Expect Ready. Error 7 if not.
 - 6) Loop 100 times.
- f. Section 0, Loop 4.
 - 1) (L00P04) Input level 1, station 1 status. Expect Reply (hang on reject). Error code 4 or 5 if not.
 - 2) (S04A). Expect Ready and Data status. Error 7 if not.
 - 3) Loop 100 times.
- g. Section 0, Loop 5.
 - 1) (L00P05). Input level 2, station 1 status. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 2) (S05A). Expect zero status. Error 8 if not.
 - 3) Loop 100 times.
- h. Section 0, Loop 6.
 - 1) (L00P06). Clear controller on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 2) (S06A). Get station 1, level 1 status. Expect Data and Ready status. Error 7 if not.
 - 3) Loop 100 times.
- i. Section 0, Loop 7.
 - 1) (L00P07). Clear interrupts on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 2) (S07A). Get station 1 level 1 status. Expect Data and Ready status. Error 7 if not.
 - 3) Loop 100 times.

- j. Section 0, Loop 8.
 - 1) (L00P08). Set interrupt mask bits for all three interrupts.
 - 2) (S08A). Request Data interrupt on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 3) (S08B). Expect Data interrupt. Error 9 if none.
 - 4) Expect no End-of-Operation interrupt. Error C if EOP interrupt.
 - 5) Expect no Alarm interrupt. Error E if Alarm interrupt.
 - 6) Get station 1, level 1 status. Expect Data, Interrupt, and Ready status. Error 7 if not.
 - 7) (S08C). Clear controller on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 8) (S08D). Loop 100 times.
- k. Section 0, Loop 9
 - 1) (L00P09). Set mask bits for all three interrupts.
 - 2) (S09A). Request EOP interrupt on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 3) (S09B). Error A if Data interrupt occurs.
 - 4) Error C if EOP interrupt occurs.
 - 5) Error E if Alarm interrupt occurs.
 - 6) Get station 1, level 1 status. Expect Data and Ready status, error 7 if not.
 - 7) (S09C). Clear controller on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 8) (S09D). Loop 100 times.
- l. Section 0, Loop 10.
 - 1) (L00P0A). Set mask bits for all three interrupts.
 - 2) (S0AA). Request Alarm interrupt on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 3) (S0AB). Error A if Data interrupt occurs.
 - 4) Error C if EOP interrupt occurs.
 - 5) Error E if Alarm interrupt occurs.

- 6) Get station 1, level 1 status. Expect Data and Ready. Error 7 if not.
 - 7) (S0AC). Clear controller on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 8) (S0AD). Loop 100 times.
- m. Section 0, Loop 11.
- 1) (L00P0B). Select offset on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 2) (S0BA). Get station 1, level 1 status. Expect Data and Ready. Error 7 if not.
 - 3) Loop 100 times.
- n. Section 0, Loop 12.
- 1) (L00P0C). Select all undefined functions on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 2) (S0CA). Get station 1, level 1 status. Expect Data and Ready. Error 7 if not.
 - 3) Loop 100 times.
- o. Section 0, Loop 13.
- 1) (L00P0D). Set mask bits for all interrupts.
 - 2) (S0DA). Select all functions except Feed on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 3) (S0DB). Expect data interrupt. Error 9 if none.
 - 4) Error C if EOP interrupt occurs.
 - 5) Error E if alarm interrupt occurs.
 - 6) Get station 1, level 1 status. Expect Data, Interrupt, and Ready status. Error 7 if not.
 - 7) (S0DC). Clear controller on station 2. Expect Reply (hang on reject). Error 4 or 5 if not.
 - 8) (S0DD). Loop 100 times.
- p. End of Section 0.
- 1) (SEC0A). End-of-Section 0.
 - 2) (SEC0B). Repeat section if Stop/Jump 5 is set.

3. Section 3 - READER FEED, INTERRUPT CHECK

a. Initialization

- 1) (SEC3). Go to Section 8 if parameter bit is not set.
- 2) Initialize section.

b. Section 3, Loop 0.

- 1) Delay
- 2) Feed a card on the reader to start the motor. Wait Not Busy.
- 3) (S30A). Select Feed, Clear, and all interrupts on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
- 4) (S30B). Set mask bits for all three interrupts.
- 5) (S30C). Get station 1 status. Expect Busy and Ready. Error 7 if not.
- 6) (S30D). Wait 40 milliseconds for column 1 data status.
- 7) (S30F). Expect status to be Data, Interrupt, Busy, and Ready. Error 7 if not.
- 8) Error E if alarm interrupt.
- 9) Input data on station 1. Expect reply. Error 4 or 5 if not.
- 10) Execute steps 4) thru 8) for 80 columns. In step 5) wait about 1200 microseconds for data status.
- 11) (S30D). Wait about 1200 microseconds for EOP status.
- 12) (S30DA). Expect EOP interrupt, Ready status. Error 7 if not.
- 13) (S30G). Error B if no EOP interrupt.
- 14) (S30H). Clear interrupt on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
- 15) (S30I). Get station 1 status. Expect EOP and Ready status. Error 7 if not.
- 16) (S30J). Clear controller on station 1. Expect reply (hang on reject). Error 4 or 5 if not.
- 17) (S30K). Get station 1 status. Expect EOP and Ready status. Error 7 if not.
- 18) Loop 50 times.

c. Section 3, Loop 1 - Force Lost Data

- 1) Delay
- 2) Feed a card and wait Not Busy to start motor.
- 3) (S31A). Select Feed, Clear, EOP and Alarm interrupts. Expect Reply (hang on reject). Error 4 or 5 if not.
- 4) (S31B). Get station 1 status. Expect Busy and Ready. Error 7 if not.
- 5) (S31C). Wait XXX milliseconds for EOP interrupt.
- 6) Error B if no EOP interrupt.
- 7) (S31D). Expect status upon EOP interrupt to be Lost Data, Alarm, EOP, Data, Interrupt, Not Busy, Ready. Error 7 if not.
- 8) (S31E). Error D if no alarm interrupt.
- 9) (ALA13). When alarm interrupt occurs, get status and save. Clear interrupts and select EOP interrupt.
- 10) (S31F). Expect status upon alarm interrupt to be Lost Data, Alarm, Data, Interrupt, Busy, Ready. Error 7 if not.
- 11) (S31G). Clear interrupts on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
- 12) (S31H). Get station 1 status. Expect Lost Data, Alarm, EOP, Data, Ready. Error 7 if not.
- 13) (S31I). Clear controller on station 1. Expect Reply (hang on reject). Error 4 or 5 if not.
- 14) (S31J). Get station 1 status. Expect EOP and Ready. Error 7 if not.
- 15) Loop 50 times.

d. End of Section 3

- 1) (SEC3A). Clear controller on station 2. Expect Reply (hang on reject). Error 4 or 5 if not.
- 2) (SEC3B). End-of-Section stop.
- 3) (SEC3C). Repeat section if Stop/Jump 5 is set.

8. Section 8 - READ WHEN NO REJECT

a. Initialization

- 1) (SEC8). Go to Section 9 if parameter bit not set.
- 2) Initialize section

b. Section 8, Loop 0

- 1) (L00P80). Initialize column counter.
- 2) Delay
- 3) (S80A). Select Feed, Clear, EOP and Alarm interrupts.
- 4) (S80C). Error E if Alarm interrupt.
- 5) Error C if EOP interrupt.
- 6) (S80D). Attempt data input. Error 4 if internal reject. Try again if external reject.
- 7) (S80E). Store data. Get station 1 status. Expect Busy and Ready. Error 7 if not.
- 8) Execute Steps 4) through 7) for 80 columns.
- 9) (S80C). Check data. Determine type of pattern from first two columns to generate reset of pattern. Compare actual with expected data. Offset card if data error.
- 10) (S8011). Wait for EOP interrupt.
- 11) (S801). Expect status upon EOP interrupt to be EOP, Interrupt, Ready. Error 7 if not.
- 12) Go to step 1) until End-of-File card is read.

c. End of Section 8

- 1) (SEC8A). Clear controller.
- 2) (SEC8B). End-of-Section stop.
- 3) Repeat section if Stop/Jump 5 is set.

9. Section 9 - READ WHEN DATA STATUS

a. Initialization

- 1) (SEC9). Go to Section 10 if parameter bit is not set.
- 2) Initialize section.

b. Section 9, Loop 0.

- 1) (L00P90). Initialize column counter.
- 2) Delay
- 3) (S90A). Select Feed, Clear, EOP and Alarm interrupts.

- 4) (S90C). Error E if Alarm interrupt.
 - 5) Error C if EOP interrupt.
 - 6) (S90D). Wait for data status.
 - 7) Expect status to be Data, Busy, Ready. Error 7 if not.
 - 8) (S90E). Input data.
 - 9) (S90F). Save data. Get station 1 status. Expect Busy and Ready. Error 7 if not.
 - 10) Execute steps 4) through 9) for 80 columns.
 - 11) (S90H). Check data. Offset card if data error.
 - 12) (S90I). Wait for EOP interrupt.
 - 13) (S90J). Expect status upon EOP interrupt to be EOP, Interrupt, and Ready. Error 7 if not.
 - 14) Go to step 1) until End-of-File card is read.
- c. End of Section 9
- 1) (SEC9A). Clear controller.
 - 2) (SEC9B). End-of-Section stop.
 - 3) Repeat section if Stop/Jump 5 is set.
10. Section 10 - READ WHEN DATA INTERRUPT
- a. Initialization
- 1) (SECA). Go to Section 11 if parameter bit is not set.
 - 2) Initialize section.
- b. Section 10, Loop 0
- 1) (L00PA0). Initialize column counter.
 - 2) Delay
 - 3) (SA0A). Select Feed, Clear and all interrupts.
 - 4) (SA0C). Error E if alarm interrupt.
 - 5) (SA0D). Wait for EOP interrupt.
 - 6) (DATAIA). Input and store data in interrupt routine. Increment column count. Do not clear interrupts.

- 7) Expect column count to be 80. Error 11 if not.
 - 8) (SA0E). Check data. Offset card if data error.
 - 9) (SA0F). Expect status upon EOP interrupt to be EOP, Interrupt, Ready. Error 7 if not.
 - 10) Loop to 1) until End-of-File card is read.
- c. End of Section 10
- 1) (SECAA). Clear controller.
 - 2) (SECAB). End-of-Section stop.
 - 3) Repeat section if Stop/Jump 5 is set.
11. Section 11 - BUFFERED READ
- a. Initialization
- 1) (SECB). Go to Section 12 if parameter bit is not set.
 - 2) Go to Section 12 if no 1706 (W = 0).
 - 3) Initialize section.
- b. Section 11, Loop 0
- 1) (L00PB0). Select Feed, Clear, EOP and Alarm interrupts.
 - 2) (SB0A). Initiate 80-word buffered input.
 - 3) Wait 1 millisecond after 1706 becomes Not Busy for EOP interrupt. Error B if no EOP interrupt.
 - 4) Error E if alarm interrupt.
 - 5) (SBOB). Expect 1706 current word address to be first word address + 80. Error 13 if not.
 - 6) (SB0C). Check data. Offset card if data error.
 - 7) (SB0D). Expect status upon EOP interrupt to be EOP, Interrupt, Ready. Error 7 if not.
 - 8) Loop to 1) until End-of-File card is read.
- c. Initialize Section 11, Loop 1
- Store first word address + 81 in first word address - 1 of input buffer area.

d. Section 11, Loop 1

- 1) (L00PB1). Select Feed, Clear, EOP and Alarm interrupts.
- 2) (SB1A). Initiate 81-word buffered input.
- 3) (E0P17). When EOP interrupt occurs, get 1706 status and save. Terminate buffer if Busy. Save current word address. Get 1729-2/3 status and save.
- 4) Error B if 1706 becomes Not Busy before EOP interrupt.
- 5) Error E if alarm interrupt.
- 6) (SB1B). Expect 1706 status upon EOP interrupt to be Reject and Busy. Error F if not.
- 7) Expect current word address upon EOP interrupt to be first word address + 80. Error 13 if not.
- 8) (SB1C). Check data. Offset card if data error.
- 9) (SB1D). Expect 1729-2/3 status upon EOP interrupt to be EOP, interrupt, ready. Error 7 if not.
- 10) Loop to 1) until End-of-File card is read.

e. End of Section 11

- 1) (SECBA). Clear controller.
- 2) (SECBB). End-of-Section stop.
- 3) Repeat section if Stop/Jump 5 is set.

12. Section 12 - PROTECTION TEST

a. Initialization

- 1) (SECC). Go to section 13 if parameter bit is not set.
- 2) (SECCA). Clear protect bits in all memory locations through End -of- Test.
- 3) (SECCB). Timeout message to operator to set Protect switches on 1729-2/3 and 17X4.
- 4) Stop with A = 1729
Q = .2/3
- 5) Operator must set Protect switches and hit RUN.

- b. Section 12, Loop 0
 - 1) (L00PC0). Get station 0 status. Expect Protected and Ready. Error 7 if not.
 - 2) Loop 100 times.
 - c. Section 12, Loop 1
 - 1) (L00PC1). Attempt all functions on station 1. Expect external reject. Error 4 or 6 if not.
 - 2) Get station 1 status. Expect Protected and Ready. Error 7 if not.
 - 3) Loop 100 times.
 - d. End of Section 12
 - 1) Set bit 1 of Stop/Jump word.
 - 2) End-of-Section stop.
 - 3) Repeat section if Stop/Jump 5 is set.
13. Section 13 - ALARM WITH NOT READY TEST
- a. Initialization
 - 1) (SECD). Go to ENDTEST if parameter bit is not set.
 - 2) (SECDA). Select clear controller and alarm interrupt.
 - 3) (SECDB). Type out message to operator to cause alarm on 1729-2/3.
 - 4) Stop with A = Q = 1729-2/3.
 - 5) Operator must cause alarm on 1729-2/3 and hit RUN.
 - b. Section 13, Loop 0
 - 1) (L00PD0). Get station 0 status. Expect Alarm, Interrupt, Not Ready status. Error 7 if not.
 - 2) Set alarm interrupt mask bit.
 - 3) Error D if no alarm interrupt.
 - 4) Loop 100 times.
 - c. Section 13, Loop 2
 - 1) (L00PD2). Select all functions on reader. Expect external reject. Error 4 or 6 if not.
 - 2) Get station 1 status. Expect Alarm, Interrupt, Not Ready. Error 7 if not.

- 3) Loop 100 times.
- d. End of Section 13
 - 1) Set Stop/Jump bit 1.
 - 2) End-of-Section stop.
 - 3) Repeat section if Stop/Jump 5 is set.
14. End of 1729-2/3 test
 - (ENDTEST). Timeout End-of-Test message. Stop if Stop/Jump 2 is set. Repeat test if Stop/Jump 6 is set.
 - (RESTART). Re-enter parameters if Stop/Jump 10 is set.

III. PHYSICAL REQUIREMENTS

- A. STORAGE REQUIREMENTS - Approximately 6K.
- B. TIMING - Variable depending upon the size of test deck.
- C. EQUIPMENT CONFIGURATION
 1. 17X4 Computer with 8K memory.
 2. 1729-2/3 Card Reader.
 3. A device for loading the test.



1725-1 FH302 CARD PUNCH TEST
(CPC088 Test No. 88)

I. OPERATIONAL PROCEDURE

A. REQUIREMENTS

1. 1784-1 or -2 computer with 8K memory
2. FH302 Controller
3. CH101A/CH101B Card Punch
4. Device for entering test into memory
5. CPVAF0 may be used to verify the punched cards

B. RESTRICTIONS

1. The test must be run in an 8K or larger computer.
2. Bits 2 and 3 of SMM parameter word must specify correct machine type.
3. Section 9, an optional section, requires manual entries. A stop occurs near the start of the section for the operator to enter a desired pattern for punching. From one- to four-card columns will be accepted. The pattern will then be repeated across the card.
4. Section 10, an optional section, requires manual intervention. A stop occurs near the start of the section for the operator to set the PROTECT switch on the CPU console. He must clear the PROTECT switch at the end of the section for further testing.

NOTE

The punch board must be Protect-jumpered before this section will execute.

5. Section 11, an optional section, requires manual intervention. A stop occurs near the start of the section for the operator to cause an alarm condition on the 1725-1 (for example, by making the input hopper empty). He must clear the alarm condition at the end of the section for further testing.
6. CPC088 has no capability of reading back its own output.

7. Control is given to the monitor at intervals during the test. The hardware may not respond favorably if delays generated by interactive testing are of lengthy duration, particularly if the test is in status-driven mode. Time-outs may be adjusted through the use of the entry parameters.

C. LOADING PROCEDURE

Call as external test number 88 under SMM17. The equipment address must have bit 0 set, and bits 1 through 5 and 11 through 15 must be clear. Actual equipment number must be in bits 7 through 10, and bit 6 must be set to 1.

Restart test after loading by MC, set P = IA, and RUN.

D. PARAMETERS

There are two parameter entries in this test. One is at the start of the test, and one is at the start of Section 9.

If bit 0 of the SMM Stop/Jump word is set at the start of the test or if bits 10 and 0 are set at the start of succeeding passes through the test, the parameter stop at the start of the test occurs. (Selected parameters are printed.)

1. Start-of-Test Parameter Entry

a. Stop 1

A1 = 8851 test ID (88), five stops (5), parameter stop (1).

Q1 = STJP test Stop/Jump parameter.

b. Stop 2

A2 = XXXX section select parameter, bit 2 = Section 2, bit 3 = Section 3, etc. There are 13 sections (0-12). \$11FF is prestored.

The sections available are:

Section 0	Controller Test
Section 1	Static Test
Section 2	Interrupt Check
Section 3	Status Test (punch blanks)
Section 4	Pattern 1 (Alternating columns have all ones, other columns have all zeros.)
Section 5	Pattern 2 (Alternating columns have half zeros and half ones, other columns have all ones.)
Section 6	Pattern 3 (Alternating columns have half zeros and half ones, other columns have all zeros.)
Section 7	Pattern 4 (Alternating columns have A's, other columns have 5's.)

Section 8 Pattern 5 (Punch alternating sliding 0's and 1's.)
 Section 9 Operator submits a pattern of four columns to be
 repeated across the card.
 Section 10 Protect Test
 Section 11 Alarm When Not Ready
 Section 12 Reject on Data Reset

Q2 = OXYZ Interrupt lines (Cardinal hex number) where:
 X = Data interrupt line
 Y = EOP interrupt line
 Z = Alarm interrupt line

Example: F = Interrupt line 15

c. Stop 3

A3 = 000X Flag indicating if Sections 4 through 9 are to interrupt
 or status drive the hardware (0 = interrupt and 1 =
 status). Prestored as zero.

Q3 = XXXX Maximum delay increment between cards. 2000_{10} is
 prestored. A variable between 0 and the maximum
 is generated by program.

d. Stop 4

A4 = XXXX Delay increment allowance for punching a card.
 Used if punch is interrupt driven. 6144_{10} is pre-
 stored.

Q4 = XXXX Delay increment between card feed and first column
 punch. Used if punch is status driven. 2560_{10} is
 prestored.

e. Stop 5

A5 = XXXX Delay increment between punches. Used if punch is
 status driven. 768_{10} is prestored.

Q5 = 0XXX Number of cards required for each of Sections 4
 through 9. Maximum card count is 255_{10} . 100_{10} is
 prestored.

2. Section 9 Parameter Entry

This stop is presented when Section 9 is executed.

a. Stop 1

A1 = 8831 Test ID (88), three stops (3), parameter stop (1)
Q1 = STJP Test Stop/Jump parameter

b. Stop 2

A2 = XYYY Pattern 1
Q2 = XYYY Pattern 2

c. Stop 3

A3 = XYYY Pattern 3
Q3 = XYYY Pattern 4
X = 8 if YYY is a valid pattern.
= 0 if null entry.

E. MESSAGES

1. Test title, initial address, and frequency count typeout.

CPC088-1725-1 CARD PUNCH TEST
FH302 CONTROLLER FOR THE CH101A/CH101B CARD PUNCH.

CP2C, VER. 3.1-1
IA = XXXX, FC = XX

XXXX is the initial address of the test and XX is the frequency count of the test.

2. End of 1725-1 Test

a. Stop 1

A1 = 8824 Test ID (88), two stops (2), end of test (4)
Q1 = STJP Test Stop/Jump parameter

b. Stop 2

A2 = XXXX Pass count
Q2 = XXXX Return address

3. End of Section

a. Stop 1

A1 = 8822 Test ID (88), two stops (2), end of section stop (2)
Q1 = STJP Test Stop/Jump parameter

b. Stop 2

A2 = 0X00 Section number
Q2 = XXXX Return address

4. Other Messages

a. Start of Section 9

CPC088 SECTION 9 - SUBMIT PATTERN FOR PUNCHING (1 TO 4 COLUMNS). INDICATE THAT THE PARAMETER SUBMITTED IS TO BE USED BY SETTING BIT 15.

This message alerts the operator that the test is not actively testing but awaiting patterns to punch. The old patterns are displayed. The stop occurs after the message is output. (See Section 9 Parameter Entry.)

b. Start of Section 10

CPC088 SECTION 10 - SET CPU PROTECT SWITCH. PROTECT JUMPER MUST BE IN CONTROLLER. PRESS RETURN WHEN READY.

c. End of Section 10

CPC088 SECTION 10 - CLEAR CPU PROTECT SWITCH. PRESS RETURN WHEN READY.

d. Start of Section 11

CPC088 SECTION 11 - MAKE 1725-1 NOT READY WHILE CARDS ARE FEEDING. PRESS RETURN WHEN READY.

F. ERROR STOPS

There is a common error routine, and therefore, the error parameters always have the same meaning. A description of the error parameters and codes follow.

1. Stop 1

A1 = 8888 Test ID (88), eight stops (8), error stop (8)
Q1 = STJP Test Stop/Jump parameter

2. Stop 2

A2 = 0YZZ (Y = section number and ZZ = error code)
Q2 = Return address

3. Stop 3

A3 = Last I/O call address
Q3 = Second-level call address

4. Stop 4
 - A4 = Third-level call address
 - Q4 = Q register output
5. Stop 5
 - A5 = Director function
 - Q5 = Status 1
6. Stop 6
 - A6 = Status 2
 - Q6 = Actual column number last output (-1 if no card)
7. Stop 7
 - A7 = Pattern of column last output (if appropriate)
 - Q7 = Expected number of columns to output
8. Stop 8
 - A8 = Status 1 when last interrupt occurred
 - Q8 = Dummy

<u>Error Code</u>	<u>Error Description</u>
1	Equipment address in error (operator error); test must be called again
2	Unexpected (ghost) interrupt
3	Parameter error (operator error); parameter must be selected again
4	Unexpected internal reject
5	Unexpected external reject
6	External reject when input status 1
7	External reject when input status 2
8	External reject when output director function
9	External reject when output a column of data
\$A	Internal reject when input status 1 in interrupt processor
\$B	External reject when input status 1 in interrupt processor
\$C	Internal reject when input status 2 in interrupt processor
\$D	External reject when input status 2 in interrupt processor
\$E	Internal reject when output director function in interrupt processor
\$F	External reject when output director function in interrupt processor
\$10	Internal reject when output data in interrupt processor
\$11	External reject when output data in interrupt processor

<u>Error Code</u>	<u>Error Description</u>
\$12	Status 2 punch error bit not set when status 1 punch error bit is set
\$13	Punch error, punch error check does not agree
\$14	Card hopper empty
\$15	Status 1 punch error bit not set when status 2 punch error bit is set
\$16	Stand-by, CP is off-line
\$17	CP is in punch inhibit state
\$18	Ready status not up
\$19	No other status set when alarm up
\$1A	EOP occurred at column 1 punch
\$1B	Interrupt occurred without corresponding status
\$1C	Interrupt reject when output function with protect violation
\$1D	Timed out waiting for data status to set
\$1E	Timed out waiting for data interrupt
\$1F	No EOP when punch inhibit or not ready occurred
\$20	Less than 80 card columns punched
\$21	Expect EOP, interrupt, and ready status only (disregard protect)
\$22	Data interrupt occurred, not requested
\$23	Column 80 punched with no EOP interrupt
\$24	Timed out waiting for data status
\$25	Busy status bit not set
\$26	Expect busy and ready status only (disregard protect)
\$27	Expect ready and data status only
\$28	Normal return, should be internal reject
\$29	Normal return, should be external reject
\$2A	External reject, should be internal reject
\$2B	Internal reject, should be external reject
\$2C	Internal reject, should be normal return
\$2D	External reject, should be normal return
\$2E	Expect zero status 2
\$2F	Unexpected EOP interrupt
\$30	Unexpected alarm interrupt
\$31	Expect data, interrupt, and ready only
\$32	No data interrupt occurred after delay
\$33	Unexpected data interrupt
\$34	Motor start up failed, ready status not up
\$35	Illegal bits in status 1 (exclude protect, busy, and ready)

<u>Error Code</u>	<u>Error Description</u>
\$36	Expect protect, data, ready status only
\$37	Expect EOP and ready status only (exclude protect)
\$38	Expect ready only (exclude protect)
\$39	Normal reply when out function with protect violation
\$3A	Internal reject when output function with alarm condition
\$3B	Normal reply when out function with alarm condition
\$3C	Expect alarm and interrupt status only (not ready)
\$3D	Data status not expected
\$3E	Data status not set when clear controller
\$3F	Ready status not set when clear controller
\$40	Expect alarm, EOP, and interrupt status only

II. DESCRIPTION

A. PATTERNS

Following are patterns generated by the test and the sections they are used in. Pattern in card is represented by 1's and 0's. 1 = punched hole, 0 = no punch.

1. Section 4 Column

	1	2	3
Row 12	1	0	
11	1	0	
0	1	0	
1	1	0	
2	1	0	
3	1	0	
4	1	0	
5	1	0	
6	1	0	
7	1	0	
8	1	0	
9	1	0	

Repeated through column 80

2. Section 5

		Column				
		1	2	3	4	5
Row	12	0	1	1	1	
	11	0	1	1	1	
	0	0	1	1	1	
	1	0	1	1	1	
	2	0	1	1	1	
	3	0	1	1	1	
	4	1	1	0	1	
	5	1	1	0	1	
	6	1	1	0	1	
	7	1	1	0	1	
	8	1	1	0	1	
	9	1	1	0	1	

Repeated through column 80

3. Section 6

		Column				
		1	2	3	4	5
Row	12	0	0	1	0	
	11	0	0	1	0	
	0	0	0	1	0	
	1	0	0	1	0	
	2	0	0	1	0	
	3	0	0	1	0	
	4	1	0	0	0	
	5	1	0	0	0	
	6	1	0	0	0	
	7	1	0	0	0	
	8	1	0	0	0	
	9	1	0	0	0	

Repeated through column 80

4. Section 7

		Column				
		1	2	3	4	5
Row	12	1	1	0		
	11	1	0	1		
	0	1	1	0		
	1	1	0	1		
	2	1	1	0		
	3	1	0	1		
	4	1	1	0		
	5	1	0	1		
	6	1	1	0		
	7	1	0	1		
	8	1	1	0		
	9	1	0	1		

Repeated through column 80

5. Section 8

		Column																								
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
Row	12	1	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0
	11	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
	0	1	1	0	0	0	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	0	1	0	0	0
	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
	2	1	1	0	1	0	0	0	1	1	1	0	1	0	1	0	1	1	1	0	0	0	0	1	0	0
	3	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
	4	1	1	0	1	0	1	1	0	0	1	0	1	0	1	0	1	0	0	1	1	0	1	0	1	0
	5	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
	6	1	1	0	1	1	1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	0
	7	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
	8	1	1	1	1	0	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	1	0	1	1	1
	9	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Repeated through column 80

All patterns start at card column 2. Column 1 contains the card sequence number and pattern code. The sequence number is from 0 through \$FF (maximum 256 cards per pattern per pass). The pattern number occupies the four low-order bits. The pattern number is valid only when the sequence number is zero. The sequence number is valid only when the pattern number is zero. (Sequence number zero is a special case.)

Column 1	Pattern (card columns 2 through 80)																								
	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Row	12	11	0	1	2	3	4	5	6	7	8	9													
	Sequence Number (0-255)								Pattern number (1-15) (Number 15 is reserved for operator-submitted pattern in Section 9.)																

The last card of the test deck contains a 5, 6, 7, 8, 9 punch in card column 1 and no pattern punches. The card is offset.

Column 1	Card columns 2 through 80 contain zeros.																								
	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Row	12	11	0	1	2	3	4	5	6	7	8	9													

B. SECTION DESCRIPTION

1. Section 0 - Controller Test

- a. Output clear controller, expect reply.
- b. Input level 1 status when Q06=0, expect internal reject.
- c. Input level 1 status when Q05=1, expect internal reject.
- d. Input level 1 status when Q06=1, Q05=0, Q04 through Q02=1's, Q01=0, and Q00=1, expect reply.
- e. Input level 1 status with Q00=0, expect external reject.
- f. Output clear controller with Q01=1, expect external reject.
- g. Input level 1 status with E field of Q register = 0, expect internal reject.
- h. Input level 1 status with W field of Q register = 1, expect internal reject.

2. Section 1 - Static Check

- a. Output clear controller, expect reply.
- b. Input level 1 status, expect reply. Expect ready and data status.
- c. Input level 2 status, expect reply. Expect zero status.
- d. Output clear interrupt, expect reply. Input level 1 status, expect ready and data status. Input level 2 status, expect reply. Expect zero status.
- e. Clear controller, expect reply. Request offset, expect reply. Input level 2 status, expect reply. Expect zero status.
- f. Select all undefined functions (\$FE60), expect reply. Expect ready and data status. Expect level 2 status of zero.
- g. Clear controller, expect reply. Select all functions except feed (\$FF7F), expect reply. Expect data interrupt. EOP and alarm interrupts unexpected. Expect ready, data, and interrupt status.

3. Section 2 - Interrupt Check
 - a. Clear controller, expect reply. Request data interrupt, expect reply. Expect interrupt. Alarm and EOP interrupts are not expected. Expect ready, data, and interrupt status.
 - b. Clear controller, expect reply. Request EOP interrupt, expect reply. Expect no interrupt. Alarm and data interrupts are not expected. Expect ready and data status.
 - c. Clear controller, expect reply. Request alarm interrupt, expect reply. Expect no interrupt. Data and EOP interrupts are not expected. Expect ready and data status.
4. Section 3 - Punch Status Check (Punch blanks in one card.)
 - a. Clear controller, expect reply. Output feed, set interrupts, and clear. Expect reply.
 - b. Input status 1. Expect ready status if column 1, or busy and ready status if not column 1.
 - c. Input status 1. Expect data status within the range of counts for card feed or column-to-column.
 - d. Expect data, interrupt, and ready status. Expect busy status unless column 1. Expect data interrupt. Expect no alarm or EOP interrupt. Output data (zeros), expect reply. Repeat b through d until 80 columns are punched.
 - e. After column 80, wait for EOP status to be set. Expect EOP, interrupt, and ready status. Expect EOP interrupt. Alarm and data interrupts are not expected.
 - f. Clear interrupts. Expect EOP and ready status. Clear controller, expect data and ready status.
 - g. Offset and feed last card, expect reply.
5. Section 4 - Punch Number 1 Pattern
 - a. Clear controller, expect reply. Start the motor and set variable to pattern number 1.
 - b. If the card punch is to be interrupt driven (see Entry Parameter, Stop 3, A3), the routine PUNCHI is used. If the hardware is to be status driven, the routine PUNCHS is used. (See the description of these routines following Section 12.)

6. Section 5 - Punch Number 2 Pattern
 - a. Clear controller, expect reply. Start the motor and set variable to pattern number 2.
 - b. If the card punch is to be interrupt driven (see Entry Parameter, Stop 3, A3), the routine PUNCHI is used. If the hardware is to be status driven, the routine PUNCHS is used. (See description of these routines following Section 12.)
7. Section 6 - Punch Number 3 Pattern
 - a. Clear controller, expect reply. Start the motor and set variable to pattern number 3.
 - b. If the card punch is to be interrupt driven (see Entry Parameter, Stop 3, A3), the routine PUNCHI is used. If the hardware is to be status driven, the routine PUNCHS is used. (See the description of these routines following Section 12.)
8. Section 7 - Punch Number 4 Pattern
 - a. Clear controller, expect reply. Start the motor and set variable to pattern number 4.
 - b. If the card punch is to be interrupt driven (see Entry Parameter, Stop 3, A3), the routine PUNCHI is used. If the hardware is to be status driven, the routine PUNCHS is used. (See the description of these routines following Section 12.)
9. Section 8 - Punch Number 5 Pattern
 - a. Clear controller, expect reply. Start the motor and set variable to pattern number 5.
 - b. If the card punch is to be interrupt driven (see Entry Parameter, Stop 3, A3), the routine PUNCHI is used. If the hardware is to be status driven, the routine PUNCHS is used. (See the description of these routines following Section 12.)
10. Section 9 - Punch Pattern Submitted by Operator (Pattern 15)
 - a. Output message requesting operator to submit pattern (see I.E. 4. a).
 - b. Accept pattern from operator (see Section 9 Parameter Entries).
 - c. Clear controller, expect reply. Start the motor and set variable to pattern number 15.
 - d. If the card punch is to be interrupt driven (see Entry Parameter, Stop 3, A3), the routine PUNCHI is used. If the hardware is to be status driven, the routine PUNCHS is used. (See the description of these routines following Section 12.)

11. Section 10 - Protect Test
 - a. Clear controller, expect reply.
 - b. Set protect bits in all of core.
 - c. Output message to operator to set PROTECT switch on computer console.
 - d. Save the interrupt trap area that involves protect violations (101 - 103). Set area for skip on protect fault to protected area of director function output routine (IRFUNC).
 - e. Input status 1, expect reply. Expect protect, data, and ready status bits to be set.
 - f. Clear all the protect bits of the output routine IRFUNC except for the last two instructions. Clear the protect bits of three variables in error routine.
 - g. Attempt to output all functions, expect external reject.
 - h. Set the protect bits in IRFUNC.
 - i. Output clear controller; expect reply, protect, data, ready status, output data (zero), and reply.
 - j. Restore the interrupt trap area.
 - k. Output message to operator to clear PROTECT switch.
12. Section 11 - Alarm When Not Ready
 - a. Clear controller and select alarm interrupt, expect reply.
 - b. Output message telling operator to cause an alarm condition on the card punch (make not ready).
 - c. Feed cards until operator causes error condition. Input status 1, expect alarm and interrupt status bits to be set. Expect ready status bit to be reset.
13. Section 12 - Reject on Data Reset
 - a. Clear controller, expect reply.
 - b. Output feed and set interrupts, expect reply.
 - c. Wait for data status to set.
 - d. Expect data, interrupt, and ready status.
 - e. Expect data interrupt. Expect no alarm or EOP interrupt.
 - f. Output data, expect reply.

- g. Input status 1, expect data status to be reset.
- h. Immediately output data again, expect external reject.

C. INTERRUPT AND PUNCH ROUTINES

1. Interrupt Response Routine

- a. The Interrupt Expected flag is interrogated. If the flag is not set, a ghost interrupt is assumed.
- b. If a ghost interrupt occurred, status 1 and 2 are input and saved. The column number and column image are saved (the column number is -1 if no card is being processed). The Ghost Interrupt flag is set. Exit is made to the monitor.
- c. If the Interrupt Expected flag is set, a legitimate interrupt is assumed. The Interrupt Expected flag is reset and status 1 and 2 are input and saved. All interrupt requests are cleared, expect reply.
- d. If the alarm status bit is set, an alarm interrupt is assumed and the Alarm Interrupt flag is set. If the EOP status bit is set, an EOP is also assumed and the EOP Interrupt flag is set. Exit is made to the monitor.
- e. If the EOP status bit is set, the EOP Interrupt flag is set. Exit is made to the monitor.
- f. If the data status bit is set, control goes to g. If not a data interrupt, the No-Status-Upon-Interrupt flag is set (an error condition) and control passes to the monitor.
- g. If the Status Drive flag is set (the hardware is being status driven), an error condition exists and a flag is set. Control goes to the monitor.
- h. If any section other than 4 through 9 is being processed, the flag DINTFL will be set, indicating that a data interrupt has occurred. Control then passes to the monitor.
- i. If a section from 4 through 9 is being executed and column 1 of the end card has just been punched, control will go directly to the monitor.
- j. The arrival of control at this point indicates a data interrupt has occurred and a card column may be punched. The next word from the output buffer is picked up. If the column number is 1 and the punch pattern is for rows 5, 6, 7, 8, and 9, the End-Card flag is set. Whatever the punch pattern is, the pattern and column number are stored for ERROR and the data is output. A reply is expected. The alarm, EOP, and data interrupt requests are output. Reply is expected. The Interrupt Request flag is set and the column counter is bumped.

- k. Return is made to the monitor.
2. Interrupt Driven Punch Subroutine
- a. Move the requested card pattern to the output buffer. Set the card count to zero. Set the expected column count in ERROR to 80 and clear the Offset and End Card flags.
 - b. Clear Interrupt flags. Set the column counter to 1. Delay for a random length of time for card feed hardware purposes.
 - c. Output feed (and offset if required), expect reply. Output requests for alarm, EOP, and data interrupts, expect reply. The Interrupt Expect flag is set (for the interrupt response routine) and a counter is initialized for the card timeout. (See Entry Parameter, Stop 4, A4.) The Offset flag is cleared.
 - d. The Status-Read flag is reset. A temporary control release is made to the monitor. When control returns, a check is made to ascertain if an alarm interrupt has occurred. If one has, a status check is made and the error reported. The Offset and Status-Read flags are set. If the ready status bit is not set or if punch inhibit status is on and an EOP interrupt did not occur, an error is declared.
 - e. A check is made for the occurrence of an EOP interrupt if one occurred. If the End-Card flag is set, an error condition is reported and control passes to g. If not end card, control goes to i.
 - f. If the No-Status-Upon-Interrupt flag was set by the interrupt response routine, an error is reported. The Status-Read and Offset flags are set.
 - g. The last card has been punched if the End-Card flag is set. An error is reported if a ghost (unexpected) interrupt occurred while the last card was being punched. Output feed and offset, expect reply. The column index is set to -1 for unexpected (ghost) interrupt occurrences while cards are not being punched. Control is then returned to the using section.
 - h. A delay of one count is made. If the time allocated for the punch wait has not expired, a return is made to subsection d. If the time has expired, the interrupt requests are cleared, the Interrupt Expect flag is reset, the error is reported, and control goes to i. If status was not read by interrupt processor, status 1 is read before going to i.

- i. An error is reported if an unexpected (ghost) interrupt has occurred during the punching of the previous card and the Offset flag is set. An error is also reported if all 80 columns were not punched in the previous card.
 - j. If EOP, interrupt, and ready status bits are not set, an error condition is reported.
 - k. The Offset flag is set if the following was in error:
 - 1. EOP, interrupt, and ready status bits are not set.
 - 2. No-Status-Upon-Interrupt flag is set.
 - 3. 80 columns were not punched.
 - 4. An alarm interrupt occurred.
 - l. If all the cards have been punched (see Entry Parameter, Stop 5, Q5), column 1 in the output buffer is set for the 5, 6, 7, 8, 9 punch. The expected column count in Error is set to 1 and control passes back to b.
 - m. If all the cards have not been punched, the card count is bumped. Control then goes to b.
3. Status Driven Punch Subroutine
- a. Move the requested card pattern to the output buffer. Set the card count to zero. Set the expected column count in Error to 80 and End Card flags. Set the Status-Drive flag for the interrupt processor.
 - b. Clear the interrupt flags. Set the column counter to 1. Delay for a random length of time for card feed hardware purposes.
 - c. Output feed (and offset if required), expect reply. Output requests for alarm and EOP interrupts, expect reply. The Interrupt Expect flag is set (for the interrupt response routine), and a counter is initialized for the timeout between card feed and first column punch. (See Entry Parameter, Stop 4, Q4.) The Offset flag is cleared.
 - d. The Status-Read flag is reset. A temporary control release is made to the monitor. When control returns, a check is made to ascertain if an alarm interrupt has occurred. If one has, a status check is made and the error reported. The Offset flag is set. If the ready status bit is not set, or if punch inhibit status is on and an EOP interrupt did not occur, an error is declared.

- e. A check is made for the occurrence of an EOP interrupt. If one occurred and the current card is not an end card, control goes to o. If the End-Card flag is set, an error condition is reported and control passes to t.
- f. If the No-Status-Upon-Interrupt flag was set by the interrupt response routine, an error is reported. The Status-Read and Offset flags are set.
- g. If the Data Interrupt flag is set, an error is reported. The Status-Read and Offset flags are set.
- h. A check is made for the availability of status 1. If the Status-Read flag is set, status 1 was read by the interrupt response routine. If the flag is not set, status 1 is read at this time.
- i. If the data status bit is set, output clear interrupts, expect reply. Clear interrupt, expect flag. Control then passes to k. If the data status is not set, a delay is made and the time counter is bumped. If a timeout has not occurred, control goes to d. If time is out, interrupt requests are cleared, the Interrupt Expect flag is cleared, the timeout error is reported, and control goes to j.
- j. If column 80 was just punched, an error occurred because no EOP interrupt came in. The Offset flag is set, the error is reported, and control goes to p.
- k. If the ready status is not set, the error is reported and the Offset flag is set.
- l. If the column counter has passed 1, the busy status bit is checked. If busy is not set, the error is reported and the Offset flag is set.
- m. If the end card has just been punched, control goes to t. If not, the next word from the output buffer is picked up. The pattern is stored for Error and the data is output. A reply is expected. The interrupt flags are cleared. If the end card was just punched, the final End-Card flag is set. The alarm and EOP interrupt requests are output. Reply is expected. The Interrupt Expect flag is set. The timeout counter is reinitialized (see Entry Parameter, Stop 5, A5).
- n. Input status 1, expect reply. If busy and ready status bits are not set, an error condition is reported and the Offset flag is set. If any other bits are set, the error is reported. The column counter is bumped and the interrupt status 1 storage location is cleared; control then passes to d.

- o. If 80 columns have not been punched, the error is reported (an EOP interrupt occurred) and the Offset flag is set.
- p. If EOP, interrupt, and ready status bits are not set, the error is reported and the Offset flag is set. If any other bits in the status 1 word are set, the error is reported and the Offset flag is set.
- q. An error is reported and the Offset flag is set if a ghost interrupt occurred while the previous card was being punched.
- r. If all the cards have been punched (see Entry Parameter, Stop 5, Q5), column 1 in the output buffer is set for the 5, 6, 7, 8, 9 punch. The End Card flag is set, the expected column count in Error is set to 1, and control passes back to b.
- s. If all the cards have not been punched, the card count is bumped. Control then goes to b.
- t. Output clear interrupts, expect reply. Clear the Interrupt Expect flag. Output feed and offset for end card, expect reply.
- u. Set the column counter to -1 for ghost interrupts while not punching a card. Clear the End-Card, Column, and Status-Drive flags. Return to the using section.

III. PHYSICAL REQUIREMENTS

A. STORAGE REQUIREMENTS

To be defined at a later date.

B. TIMING

Variable, depending upon the size of the test deck.

C. EQUIPMENT CONFIGURATION

- 1. 1784 computer with 8K memory
- 2. 1725-1 Card Reader
- 3. Device for loading test



1731/601, 602, 612 MAGNETIC TAPE TEST

(MT1007 Test No. 7)

(CPOF)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

Specific sections will only run on certain types of tape units whether the specific section is selected or not.

1. No sections will run on 7-track tape at 800 BPI.
2. No sections will run on 9-track tape if density is 200 or 556 BPI or if mode is BCD.
3. Bit 15 of Unit Select parameter must be set for BDC direct I/O.

B. LOADING PROCEDURE

This test operates under the control of 1700 SMM. The calling sequence is that specified by SMM. To restart test after loading MC, set P = IA and RUN.

C. PARAMETERS

If Bit 0 of the Stop/Jump word is set, the program will have one monitor stop displaying \$741 in the A register and the Stop/Jump word in the Q register, and three program stops to enter parameters in A and Q as follows (after last stop parameters selected will be typed out):

- Stop 2 A = Section Selects
Q = Unit Selects, Bit 15 = Direct BDC I/O
- Stop 3 A = Data Interrupt line
Q = Alarm Interrupt line
- Stop 4 A = End of Operation Interrupt line
Q = Zero

1. Section Select Bits

<u>Bit</u>	<u>Section</u>	<u>Function</u>
0	0	Ready Not Busy status
1	1	Write Enable Load Point, Protect status
2	2	Binary parity and File Mark status
3	3	BCD Parity and File Mark status
4	4	Read wrong mode parity
5	5	Write, read, and check file mark and binary data
6	6	Binary patterns
7	7	BCD data
8	8	Variable length records
9	9	Creep Test
10	10	Ladder Test
11	11	Interrupts

2. Unit Select Bits

<u>Bit</u>	<u>Unit</u>	<u>Function</u>
0	0	Units selected must be Ready and Write
1	1	Enabled and may be any combination of
2	2	601's, 602's, and 612's. No new units may
3	3	be selected after initial parameter.
4	4	
5	5	
6	6	
7	7	
15	N/A	Set for direct I/O on BDC channel

D. STOP AND JUMP SETTINGS

Bit Settings in STJP Word

<u>Bit</u>	<u>Stops</u>
0	Stop Enter Parameters
1	Stop End of Section (bits 12, 13, and 14 indicate unit number)
2	Stop End of Test
3	Stop on Error

<u>Bit</u>	<u>Jumps</u>
4	Repeat Conditions
5	Repeat Section
6	Repeat Test
7	Omit Simulation
8	Omit Typeouts
9	Bias Return Address Display
10	Re-enter Parameters (changeable parameters are section select bits and direct or buffered BDC I/O)

E. MESSAGES

1. Alarms and Typeouts

a. Normal Program Typeouts

MT1007, 1731 MAGNETIC TAPE TEST IA = XXXX, FC = XX, CPOF, VER. 3.1

b. Error Alarm Typeouts

- 0001 - External Connect reject
- 0002 - Unit Not Ready
- 0003 - Status reject
- 0004 - Unit Busy
- 0005 - Unit protected
- 0006 - No write enable
- 0007 - W = 02, 07, or 0C
- 0008 - No units available
- 0009 - External Function reject
- 000A - No Load Point status
- 000B - Internal reject on output
- 000C - Internal reject on input
- 000D - No Busy status
- 000E - No File Mark/Tape Mark status
- 000F - No Parity status
- 0010 - Parity status
- 0011 - No 7-track status
- 0012 - Deselect reject
- 0013 - Data Compare error
- 0014 - Internal Connect reject
- 0015 - No Parity status on binary file mark

- 0016 - BDC External reject on I/O
- 0017 - No Interrupt status
- 0018 - Non-requested interrupt
- 0019 - Unidentified interrupt
- 0020 - Alarm interrupt - Not Ready
- 0021 - Alarm interrupt - Lost Data
- 0022 - Alarm interrupt - Parity Status
- 0023 - Alarm interrupt - End of Tape
- 0024 - Alarm interrupt - File Mark/Tape Mark
- 0025 - No alarm interrupt condition
- 0026 - End of Operation interrupt not cleared
- 0027 - Alarm interrupt not cleared
- 0028 - Data interrupt not cleared
- 0029 - Parity status on binary tape mark
- 0030 - Internal function reject
- 0031 - Internal Mode/Density Select reject
- 0032 - Internal Deselect reject
- 0033 - No alarm status of file mark/tape mark
- 0034 - Unidentified interrupt not cleared
- 0035 - No End of Operation interrupt
- 0036 - No Data interrupt
- 0037 - I/O reject following data interrupt
- 00FF - Not sufficient core length

2. Error Displays

There are three types of error displays: data error (ECC\$13), core length (ECC\$FF), and non-data errors. Displays are as follows:

	A	Q	A	Q		
Non-data	ID	STJP	SECECC*	RETAD**	A	Q
Data	ID	STJP	SECECC*	RETAD**	Actual data	Expected data
<u>Core Length</u>	ID	STJP	SECECC*	RETAD**	Core Length	Last AD

*SECECC: Section Number - bits 8-15; Error Code - bits 0-7
 **RETAD: Return Address biased

F. ERROR STOPS

For all errors there are two monitor stops with displays in A and Q. Bits 4 to 7 of the ID word indicate the total number of stops.

II. DESCRIPTION

This test consists of 12 sections which can be individually selected. All sections will be executed unless the Section Select bits are altered by the operator. Each section selected is run on each unit in consecutive unit number order until all sections have been run on all units. This procedure is repeated for each density.

A. SECTION MT0

Ready Not Busy Status

1. Check Ready status.
2. Check Repeat conditions.
3. Check Busy status.
4. Check Repeat conditions.
5. Repeat 500_{10} times beginning with step 1.

B. SECTION MT1

Write Enable, Load Point and Protected Status

1. Rewind unit.
2. Check Write Enable status.
3. Check Repeat conditions.
4. Check Load Point status.
5. Check Repeat conditions.
6. Check Protected status.
7. Check Repeat conditions.
8. Repeat 500_{10} times beginning with step 2.

C. SECTION MT2

Busy on BDC Write, Parity and File Mark/Tape Mark Status

1. Check Channel.
 - a. BDC channel - continue with step 2
 - b. AQ channel - jump to step 5
2. Write 300_{10} word record to all "1's".
3. Check Busy status of the BCD 10 times.

4. Wait for Not Busy.
5. Write File Mark/Tape Mark .
6. Wait for End of Operation status .
7. Check track number of unit.
 - a. If 7 track - check for presence of parity status
 - b. If 9 track - check for absence of parity status
8. Check Repeat conditions.
9. Check File Mark/Tape Mark status.
10. Check Repeat conditions.
11. Check Alarm status.
12. Repeat 10 times from step 5.

D. SECTION MT3

Write BCD File Mark, Check Busy and File Mark Status

1. Write File Mark .
2. Checks for Busy .
3. Check Repeat conditions .
4. Wait for Not Busy .
5. Check File Mark status .
6. Check Repeat conditions .
7. Repeat 10 times from step 1 .

E. SECTION MT4

Parity Status - Write/Read Opposite Mode

1. Write two 125_{10} word binary records .
2. Write two 125_{10} word BCD records .
3. Backspace four times .
4. Read first binary record as BCD .
5. Wait for Not Busy .
6. Check for expected parity status .

7. Read second binary record as binary.
8. Wait for Not Busy.
9. Check for absence of parity status.
10. Read first BCD record as binary.
11. Wait for Not Busy.
12. Check for expected parity status.
13. Read second BCD record as BCD.
14. Wait for Not Busy.
15. Check for Absence of parity status.
16. Check 7-track status.
17. Check repeat conditions.
18. Repeat 100_{10} times from step 16.

F. SECTION MT5

Write, Read, and Check Binary Record and File Mark/Tape Mark data

1. Rewind unit.
2. Write six file marks/tape marks.
3. Write a 40-word binary record of \$FF.
4. Write six file marks/tape marks.
5. Backspace seven times.
6. Read and check 40-word binary record.
7. Backspace once.
8. Backspace once.
9. Wait for Not Busy.
10. Check File Mark/Tape Mark status.
11. Repeat six times from step 8.
12. Backspace once.
13. Wait for Not Busy.
14. Check Load Point status.
15. Read and check one file mark/tape mark.
16. Repeat step 15 six times.

G. SECTION MT6

Write, Read, and Check Five Binary Patterns

1. Patterns used.
 - a. all "0's"
 - b. all "1's"
 - c. \$AA55
 - d. left-shifting 0
 - e. left-shifting 1
2. Write five 100_{10} word record of pattern.
3. Backspace five times.
4. Read and check data of the five 100_{10} word record.
5. Repeat from step 2 until all patterns have been written, read and checked.

H. SECTION MT7

Write, Read, and Check BCD Records

1. Write file mark.
2. Wait for Not Busy.
3. Generate 100_{10} BCD codes.
4. Write five 100_{10} word BCD record.
5. Backspace five times.
6. Read and check data five 100_{10} word BCD records.
7. Repeat from step 1 five times.

I. SECTION MT8

Variable Length Records of \$20

1. Write file mark/tape mark.
2. Wait for Not Busy.
3. Write six binary records of length 28_{10} , 97_{10} , 275_{10} , 20_{10} , 75_{10} , and 5_{10} .
4. Backspace six times.
5. Wait for Not Busy.
6. Read and check each record.
7. Repeat from step 3 five times

J. SECTION MT9

Creep

1. Write 20_{10} file marks/tape marks.
2. Write a one word binary record of \$F0.
3. Backspace.
4. Read and check one word record.
5. Repeat from step 2 fifty times.

K. SECTION 10

Ladder Test

1. Write file mark/tape mark.
2. Write \$FF binary record of variable length and comparable data.

Length (words)	Data
1	\$01
2	\$02
3	\$03
.	.
.	.
\$20	\$20
.	.
.	.
\$FF	\$FF

3. Backspace \$FF times.
4. Read and check each binary record.

L. SECTION MT11

Interrupts - Alarm interrupt is selected throughout the entire test.

1. Select End of Operation interrupt.
2. Set End of Operation interrupt request flag.
3. Write a 500_{10} word record of one's.
4. Wait for Not Busy .
5. Check End of Operation occur flag.

6. Clear End of Operation occur flag.
7. Reset End of Operation interrupt.
8. Backspace once.
9. Wait for Not Busy.
10. Check End of Operation occur flag.
11. Clear End of Operation occur flag.
12. Read 500_{10} word record that was written.
13. Wait for Not Busy.
14. Check End of Operation occur flag.
15. Clear End of Operation occur flag.
16. Clear all interrupts.
17. Clear End of Operation Request flag.
18. Check data.
19. Repeat from step 1 ten times.
20. Check channel.
 - a. BDC - set Alarm Interrupt Request flag and exit from section.
 - b. AQ - check data interrupts as follows:
 - 1) select Data interrupt
 - 2) set Data Interrupt request flag
 - 3) select Write function
 - 4) wait for Not Busy
 - 5) check data occur flag
 - 6) clear Data occur flag
 - 7) check repeat conditions
 - 8) repeat from step 1 selecting a Read function
 - c. Reset Read function back to Write.
 - d. Store 25_{10} words in output area.
 - e. Clear interrupts.
 - f. Clear data line from Mask register.

- g. Select Data interrupt.
- h. Select Write function.
- i. Check Interrupt status bit.
- j. Write one word.
- k. Repeat from step i 25 times.
- l. Backspace.
- m. Wait for Not Busy.
- n. Repeat from step e to step k changing Write's to Read's.
- o. Set data line back in Mask register.
- p. Reset Alarm Request flag.

III. INITIALIZATION AND SUBROUTINE DESCRIPTION

A. INITIALIZATION

1. Convert bias value and frequency count to ASCII and store in typeout routine.
2. Check core length.
3. Set all section select bits.
4. Set up title for typeout.
5. Set up interrupt line display for parameter stop.
6. Check W for zero or correct value for BDC channel.
7. Store unit selected by operator or units Ready and Write Enabled in a table.
8. Clear controller.
9. Store pattern for Section MT6.
10. Store pattern lengths for Section MT8.
11. Store density for pass 1 of test.

B. SUBROUTINES

1. TESTEX - Intermediate exit to monitor
 - Stores return address of test at IA + 5
2. WNB - Wait for Not Busy
 - a. Check BDC for Not Busy if on BDC channel
 - b. Obtain status 1
 - c. Check Busy status
 - d. If Not Busy, exit
 - e. If Busy, store direct locations used both by monitor and test such as I and \$10 to \$19
 - f. Exit through TESTEX
 - g. Restore values stored at step e.
 - h. Repeat from step until Not Busy
3. CON - Connect
 - a. Connect unit
 - b. Check Ready Not Busy, Protected, and Write Enable status
 - c. Determine whether particular unit connected will run at specified density and mode
 - d. If no, next unit in table is connected
 - e. If yes, mode and density (MDEN) are selected
4. SECT - Check Section Select Bit
 - a. Section bit is stored
 - b. Mode and density are stored
 - c. Determine whether section is selected
 - d. If yes, add one to return address and do an RTJ to CON and exit SECT
 - e. If no, exit SECT
5. DATA - Generate and Compare Data
 - a. Store number of words and pattern
 - b. Generate data output one word at a time and compare it with that input

6. STOP - Routine Used for Error and End of Section Stops
 - a. Set up stop display
 - b. Unit number is bits 12-14 of ID word
7. OUT - Write Data on AQ or BDC Direct or Buffered
 - a. Generate and store all data to be written except for section 7 which generates its own data
 - b. Write and generate data in the manner requested by W and bit 15 of unit parameter.
 - c. BDC present - do an RTJ to DS1
 - d. RTJ - WNB
8. INP - Read Data on AQ or BDC Direct or Buffered
 - a. Type of input is determined by W and bit 15 of units parameter
 - b. RTJ - WNB
9. SLFUN - Select All Controller Functions
 - a. Check status for End of Operation
 - b. If not present, do an RTJ to WNB
 - c. If present, select requested function
10. MDEN - Select Mode and Density
 - a. Loop on controller active
 - b. Controller inactive, select mode and density requested
11. DS1 - Obtain Status 1
 - a. Check BDC for Not Busy if on BDC channel
 - b. Store status of 1731 at STAT1.
12. DS2 - Obtain Status 2
 - a. Check BDC for Not Busy if on BDC channel
 - b. Store status of 1731 at STAT2
13. ENDSC - Check Repeat Section
 - a. Set A and Q for End of Section stop
 - b. Check repeat section bit
 - c. Deselect units

- d. Loop on controller active
- e. Update address of units table
- 14. PARM - Re-enter Parameters
 - a. Reinitialize address of units table
 - b. Check Re-enter Parameter bit
 - c. If set, restore Section Select bits
- 15. FINAL - End of Test
 - a. Make sure all densities have been run
 - b. Display End of Test information
 - c. Check Repeat Test bit

IV. PHYSICAL REQUIREMENTS

- A. SPACE REQUIREMENTS - 2438_{10} or 986_{16} locations
- B. TEMPORARY STORAGE REQUIREMENTS - 500_{10} or $1F4_{16}$ locations
- C. TIMING - 3 min 15 sec with one unit
- D. EQUIPMENT CONFIGURATION
 - 1. 17X4 Computer with 8K memory
 - 2. 1705 Interrupt Data Channel
 - 3. 1706 Buffer Data Channel (optional)
 - 4. 1731 Magnetic Tape Controller
 - 5. Minimum of one 601, 602, or 612 tape unit and a maximum of eight such units in any combination.

1731/601, 602, 612 MAGNETIC TAPE TEST
(MT200E Test No. 0E)
(CP2F)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

The test attempts to determine units available and ready for testing and stores the unit information in parameter A2. Therefore, parameter A2 cannot be prestored.

B. LOADING PROCEDURE

This test operates under the control of the 1700 SMM. The calling sequence is that specified by SMM. Test can be restarted at initial address.

C. PARAMETERS

The initialization portion of the test sets parameters to test all 601's, 602's, and/or 612's connected to the 1731 that are Ready and have file protect rings inserted. It also sets the interrupt line(s) parameters. If a parameter stop is selected, these parameters may be altered. A timeout of parameters will occur after last stop.

1. Selective Stops

- | | | |
|-------------------|---------|---|
| a. Parameter | Stop 1: | A = 0E41 OVERFLOW light on
Q = Stop Jump parameter |
| | Stop 2: | A = Selected units
Q = Selected test sections |
| | Stop 3: | A = Data interrupt line
Q = Alarm interrupt line |
| | Stop 4: | A = End of Operation interrupt line
Q = Units that are 602's |
| b. End of Section | Stop 1: | A = 0E22 OVERFLOW light on
Q = Stop Jump parameter |
| | Stop 2: | A = 0X00 (x = section number)
Q = Return address |
| c. End of Test | Stop 1: | A = 0E24 OVERFLOW lite on
Q = Stop Jump parameter |
| | Stop 2: | A = Pass count
Q = Return address |

d. Error (See Section E)

2. Selecting Stops and Jumps (Stop Jump Word)

<u>Bit</u>	<u>Stops</u>
0	Stop to enter parameters
1	Stop at end of section
2	Stop at end of test
3	Stop on error

<u>Bit</u>	<u>Jumps</u>
4	Repeat identical conditions of error
5	Repeat section
6	Repeat test
7	Omit simulation
8	Omit typeouts
9	Bias return address display

D. MESSAGES

1. Typeouts

a. Normal Program Typeouts

1) Initialization of test

```
MT200E, 1731 MAGNETIC TAPE TEST
IA = XXXX, FC = XX. CP2F, VER. 3.1
```

2) End of one pass through test

A	Q	A	Q
0E24	Stop/Jump	Pass Count	Return Address

2. Error Types

a. Sense Test Errors (Section 0)

- 00 - Unit should be Ready at start
- 01 - Unit does not have write enable
- 02 - Unit should be Not Busy
- 03 - Unit should be in 800 BPI
- 04 - Unit should not be in 556 BPI
- 05 - Unit should be in 556 BPI
- 06 - Unit should not be in 800 BPI
- 07 - Unit should be in 200 BPI

- 08 - Should be at load point
- 09 - False End of Tape
- 0A - False End of Operation
- 0B - Should sense Busy
- 0C - Should not sense Parity Error, Alarm, Load Point, End of File, Lost Data
- 0D - Should sense controller not active
- 0E - Should sense controller active
- 0F - Should sense Parity Error alarm
- 10 - Should sense Parity Error
- 11 - Should sense Lost Data
- 12 - Should sense Lost Data alarm
- 13 - Should sense End of File mark
- 14 - Should sense End of File Mark alarm
- 15 - Illegal reply after Lost Data condition
- 16 - Should not sense Lost Data

b. All others

- 01 Write parity error
- 02 Read parity error
- 03 Data error
- 04 Block too short
- 05 Lost data
- 06 End of Operation interrupt error
- 07 Data interrupt error
- 08 Alarm interrupt error
- 7E Internal reject
- 7F Illegal external reject

E. ERROR STOPS AND TYPEOUTS

A	Q	A	Q	A	Q
NEX8	Stop/Jump	0YZZ	Return Address	VVVV	WWWW

Where N = tape unit number
 X = number of stops
 Y = section (always zero for reject error)
 ZZ = error type

VVVV = 1) failing data word for data error
 2) (A) when illegal reject occurred

WWWW = 1) expected data word for data error
 2) (Q) when illegal reject occurred

II. DESCRIPTION

A. The program consists of seven sections. It tests all units being used simultaneously. Each section is run in all recording densities.

1. Section 0 - Sense

Checks all sensing - both for the condition and for absence of condition where applicable. Repeat five times.

2. Section 1 - Interrupt

- a. Write file mark. Check End of Operation interrupt
- b. Read file mark. Check Alarm interrupt
- c. Check Write Data interrupt
- d. Check Read Data interrupt
- e. Repeat section in all densities

3. Section 2 - Backspace

- a. Write 18_{10} -word binary record of all zeros
- b. Write eight file/tape marks
- c. Backspace nine times
- d. Read all "0's" record
- e. Write 18_{10} -word binary record of all "1's"
- f. Write 50_{10} 18_{10} -word binary records of all "0's"
- g. Backspace 51_{10} times
- h. Read all "1's" record
- i. Repeat section in all densities

4. Section 3 - Long Record

- a. Write 4000_{16} -word binary record
- b. Backspace
- c. Read 4000_{16} -word record (no data checking)
- d. Repeat section in all densities

5. Section 4 - Variable Record Length - BCD

- a. Write BCD pattern in records ranging from 18_{10} words to 258_{10} words in increments of four

- b. Backspace to first record
 - c. Read all records
 - d. Repeat section in all densities
6. Section 5 - Random Block Length
- a. Generate random data in write data area
 - b. Generate random record length
 - c. Write 64_{10} random length records
 - d. Backspace 64_{10} records
 - e. Read 64_{10} records
 - f. Repeat section in all densities
7. Section 6 - Nonstop Read and Write
- a. Write 80_{10} records in Start/Stop mode
 - b. Backspace 80_{10} records nonstop
 - c. Read 80_{10} records nonstop
 - d. Write 80_{10} records nonstop
 - e. Backspace 80_{10} records start/stop
 - f. Read 80_{10} records start/stop
 - g. Repeat section in all densities

B. INITIALIZATION AND SUBROUTINE DESCRIPTION

1. Initialization

- a. Convert bias value and frequency count to ASCII and store in typeout routine.
- b. Store return address.
- c. Find converter number for BCD operation to determine which Busy switch to use, if applicable.
- d. Find all units that are ready to write.
- e. Find interrupt lines to 1731.
- f. Type test title, number, initial address, and frequency count.

2. Subroutines

- a. BSYCON - Passes control to and accepts control from SMM17. Checks and sets or clears the converter Busy switch if applicable (1706, 1716)
- b. SUNO - Select Unit
 - 1) Selects next available unit
 - 2) Upon entry A = parity mode parameter. If A = 0, use previous parameter
 - 3) Upon exit A = 0 if another unit is selected. A ≠ 0 if no more units.
 - 4) Checks SELECTIVE SKIP switch.
- c. ILREJ - Reject Error
 - 1) Stores A and Q in display
 - 2) Go to SMM Stop subroutine
- d. STATO - Level 1 Status
 - 1) Upon entry if A = 0, read status and exit
 - 2) Upon entry if A ≠ 0, wait for condition in A and give control to SMM17 until condition is present
 - 3) Upon entry A can be either positive or negative when not zero
- e. LEV2 - Level 2 Status
 - 1) Read status and exit
- f. FSELO - Perform function in A to selected unit
- g. FALLO - Perform function in A to all units
- h. WSELO - Write on selected unit
 - 1) A contains record length
 - 2) Control given to SMM17 after buffer initiate
- i. WALLO - Write one record on all units
 - 1) A contains record length
 - 2) Check for errors
 - 3) Uses WSELO

- j. RSELO - Read on selected unit
 - 1) A contains record length, if known
 - 2) If A = 0, record length will be determined and recorded
 - 3) Uses End of Operation interrupt
 - 4) Gives control to SMM17 after buffer initiate
- k. RALLO - Read one record all units
 - 1) Uses RSELO
 - 2) Checks all errors
- l. DENO - Update density select parameter
- m. PATTO - Stores A in 80_{10} consecutive buffer locations
- n. TOERR - Section 0 error
 - Processes all errors detected in section 0

III. PHYSICAL REQUIREMENTS

A. TEMPORARY STORAGE REQUIREMENTS

After all tests called by SMM have been loaded and initialized, the program will attempt to reserve 1024_{10} locations. If not available, it will use any amount less than 1024_{10} .

B. TIMING - 2 min 30 sec with one unit.

C. EQUIPMENT CONFIGURATION

- 1. 1704 Computer with 4K memory
- 2. 1705 Interrupt Data Channel
- 3. 1706 Buffer Data Channel (optional)
- 4. 1731 Magnetic Tape Controller
- 5. Minimum of one 601, 602, or 612 Tape Unit and a maximum of eight such units in any combination.

1732/608, 609 MAGNETIC TAPE TRANSPORT TEST
(MT3015 Test No. 15)
(CP2F)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

The test attempts to determine unit available and ready for testing and stores the unit information in parameter A2. Therefore, parameter A2 cannot be prestored.

B. LOADING PROCEDURE

This test operates under control of SMM17. The calling sequence is that specified by SMM17. This test may be restarted at initial address.

C. TESTING PROCEDURE

If the 1732 is on a buffered data channel, the whole test will run either in A/Q or DSA mode but not both at the same time.

D. PARAMETERS

The initialization portion of the test sets parameters to test all 608's and 609's connected to the 1732 that are ready and have file protect rings inserted. The test can run buffered if a 1706/16 channel is available. A typeout of parameters will occur after last stop.

1. Selective Stops

- a. If a parameter stop is selected, these parameters may be altered;

Stop 1: A = 1541 OVERFLOW light on

Q = Stop Jump Parameter

Stop 2: A = Selected units (e. g. bit 5, unit 5)

Q = Selected sections (e. g. bit 6, section 6)

Stop 3: A = Data Interrupt Line

Q = Alarm Interrupt line

Stop 4: A = End of Operation Interrupt Line

Q = 0000 = 1732 or 1732/17X6 A/Q I/O
= 0002 = 1732/1706 DSA I/O only

- b. End of Section

Stop 1: A = 1522 OVERFLOW light on

Q = Stop Jump Parameter

Stop 2: A = 0X00 (X = section number)

Q = Return Address

c. End of Test

Stop 1: A = 1524 OVERFLOW light on
Q = Stop/Jump Parameter

Stop 2: A = Pass count
Q = Return Address

d. Error (see section E)

2. Selecting Stops and Jumps (Stop Jump Word)

<u>Bit</u>	<u>Stops</u>
0	Stop to enter parameters
1	Stop at end of section
2	Stop at end of test
3	Stop on error

<u>Bit</u>	<u>Jumps</u>
4	Repeat identical conditions of error
5	Repeat section
6	Repeat test
7	
8	Omit typeouts
9	Bias return address display
10	Not used (Use bit 0 instead)

D. MESSAGES

1. Typeouts

a. Normal Program Typeouts

1) Initialization of Test

MT3015, 1732 MAGNETIC TAPE TEST

IA = XXXX, FC / XX CP = 2F VER. 3.1-1

2) End of pass X through test

A	Q	A	Q
1524	Stop/Jump	X	Return Address

2. Error Types

a. Sense Test Error (Section 0)

10 - Ready status should be set

11 - Write enable status not set

- 12 - Busy status set (should not be set)
- 13 - 800 BPI status not set (should be set)
- 14 - 556 BPI status is set (should not be set)
- 15 - 556 BPI status not set (should be set)
- 16 - 800 BPI status is set (should not be set)
- 17 - 200 BPI status is not set (should be set)
- 18 - Load point status is not set (should be set)
- 19 - End of tape status set (should not be set)
- 1A - End of Operation status set (should not be set)
- 1B - Busy status is not set (should be set)
- 1C - One or more of the following status bits are set; Parity error, Alarm, Load point, End of file, Lost data (None of these should be set)
- 1D - Controller Active status is set (should not be set)
- 1E - Controller Active status is not set (should be set)
- 1F - Alarm status is not set (should be set from parity error)
- 20 - Parity Error status is not set (should be set)
- 21 - Lost Data status is not set (should be set)
- 22 - Alarm status is not set (should be set from forcing lost data)
- 23 - File Mark status not set (should be set)
- 24 - Alarm status is not set (should be set from reading file mark)
- 25 - Attempted to input data when Data status was not present. Input to A instruction should have rejected, but it did not.
- 26 - Lost Data status is set (should not be set)
- 27 - Fill status is not set (should be set)

b. All other Errors

- 01 - Parity Error on Write operation
- 02 - Parity Error on Read operation
- 03 - Data compare error (reports only the first data error in a record)
- 04 - Block too short
- 05 - Lost Data
- 06 - End of Operation Interrupt error

- 07 - Data Interrupt error
- 08 - Alarm Interrupt error
- 09 - Fill status set (should not be set)
- 69 - Controller stayed busy too long; possibly caused by a long rewind
- 6A - Did not receive expected status bit
 - A3 = Actual status bit
 - Q3 = Expected status bit
- 7E - Internal Reject
- 7F - Illegal External Reject
- 50 - Misread file marks during backspace
 - A3 - Not used
 - Q3 - Number of files backspaced correctly
- 51 - Misread records during backspace
 - A3 - Not used
 - Q3 - Number of records backspaced correctly

E. ERROR STOPS AND TYPEOUTS

A	Q	A	Q	A	Q	A	Q
15X8	Stop/Jump	NYZZ	Return Address	VVVV	WWWW	WDCNT	RL

Where X = number of stops

N = unit number

Y = section number

ZZ = error type

VVVV=1) actual data on data compare error

2) (A) when illegal reject occurred

WWWW=1) expected data on data compare error

2) (Q) when illegal reject occurred

WC=(A) word number in error (data compare error only)

RL= (Q) expected record length of data error.

II. DESCRIPTION

A. NINE SECTION PROGRAM (NUMBERED 0-8)

All units being used are tested simultaneously. Each section is run in all recording densities.

1. Section 0 - Status Sense Test

This section checks all status bits - both for the condition and for absence of the condition where applicable. Repeat 5 times.

2. Section 1 - Interrupt Test

- a. Rewind all units. Write file mark. Check End of Operation interrupt.
- b. Read file mark. Check Alarm interrupt
- c. Check Write Data interrupt

- d. Check Read Data interrupt
 - e. Repeat section in all densities
3. Section 2 - Backspace Test
- a. Rewind all units. Write 18_{10} - word binary record of all zeros
 - b. Write eight file/tape marks
 - c. Backspace nine times
 - d. Read all zeros record
 - e. Write 18_{10} - word binary record of all ones
 - f. Write 50_{10} - 18_{10} - word binary records of all zeros
 - g. Backspace 51_{10} times
 - h. Read all ones record
 - i. Repeat section in all densities
4. Section 3 - Search File Mark Forward/Backward Test
- a. Rewind all units
 - b. Write file mark on all units
 - c. Write 18_{10} - word record of 0000_{16} on all units
 - d. Write file mark on all units
 - e. Write 18_{10} - word record of 0001_{16} on all units
 - f. Repeat from step d (incrementing the data of each record by 0001_{16}) until a record of 0030_{16} has been written.
 - g. Rewind all units
 - h. Execute 19_{16} search file mark forward functions on all units.
 - i. Read record of 0018_{16} 's on all units
 - j. Execute 14_{16} search file mark forward function on all units.
 - k. Execute 19_{16} search file mark backward functions on all units
 - l. Execute 1 search file mark forward function on all units
 - m. Read record of 0014_{16} on all units
 - n. Repeat from Step a for all densities

5. Section 4 - Variable Record Length - BCD (Binary if 609)

- a. Rewind all units. Write BCD pattern in records ranging from 18_{10} words to 258_{10} words in increments of four (character mode)
- b. Backspace to first record
- c. Read all records (character mode)
- d. Repeat section in all densities

6. Section 5 - Random Block Length - Binary

- a. Rewind all units. Generate random data in write data area
- b. Generate random record length
- c. Write 64_{10} random length records (character mode)
- d. Backspace 64_{10} records
- e. Read 64_{10} records (character mode)
- f. Repeat section in all densities

7. Section 6 - Nonstop Read and Write

- a. Rewind all units. Write 80_{10} records in Start/Stop mode
- b. Backspace 80_{10} records nonstop
- c. Read 80_{10} records nonstop
- d. Write 80_{10} records nonstop
- e. Backspace 80_{10} records start/stop
- f. Read 80_{10} records start/stop
- g. Repeat section in all densities

8. Section 7 - Variable Record Length - BCD (Binary if 609)

This section is identical to Section 4 except that all data is written and read in assembly/disassembly mode rather than character mode.

9. Section 8 - Random Block Length - Binary

This section is identical to Section 5 except that all data is written and read in assembly/disassembly mode rather than character mode.

B. INITIALIZATION

1. Convert bias value and frequency count and store in typeout routine.
2. Store return address.
3. Find converter number for BAQ operation to determine which BUSY switch to use, if applicable.
4. Finds all units that are Ready and Write enabled.
5. Types test title, number initial address, and frequency count.

III. PHYSICAL REQUIREMENTS

A. TEMPORARY STORAGE REQUIREMENTS

After all tests called by SMM have been loaded and initialized, the program will attempt to reserve 1024_{10} locations. If not available, it will use any amount less than 1024_{10} .

B. EQUIPMENT CONFIGURATION

1. 17X4 Computer
2. 1732 Magnetic Tape Controller
3. Minimum of one 608 or 609 Magnetic Tape Transport and maximum of eight of any combination of such units
4. 1706 or 1716 Buffered Channel (optional)



1731/1732, 601, 608, 609 SPECIAL MAG TAPE TEST
(MTSA1F Test No. 1F)
(C POF)

I. OPERATIONAL PROCEDURE

A. RESTRICTIONS

1. This test must be run alone.
2. Bit 5 in the SMM parameter word must be set. This tells the monitor to use the Non-interrupt mode typeout.
3. After loading test, set all unit PROTECT switches on tape controller, and if controller is 1732, set TEST MODE switch on tape controller. All units that have Write enabled and are Ready will be tested.
4. Check special instructions under Test Description, II. D, Section 3, before running test.
5. Sections are run in the following order: 1, 5, 2, 3, 4, 6

B. LOADING PROCEDURE

The calling sequence is that specified by SMM17. The test number for this special mag tape test is 1F. Test can be restarted at initial address.

C. PARAMETERS

1. If bit 0 of the Stop/Jump word is set, a parameter stop will occur.
 - a. First stop, A = 1F31, Q = Stop/Jump parameter. The Stop/Jump parameter may be changed if desired.
 - b. Second stop, A = controller designator and Section selection:
 - Bit 15 = 1: Controller is a 1732
 - Bit 15 = 0: Controller is a 1731
 - Bit 05 = 1: Run test Section 6
 - Bit 04 = 1: Run test Section 5
 - Bit 03 = 1: Run test Section 4
 - Bit 02 = 1: Run test Section 3
 - Bit 01 = 1: Run test Section 2
 - Bit 00 = 1: Run test Section 1
- Q = Data interrupt line. The bit position identifies the interrupt line, i.e.,
bit 6 = interrupt line 6.

- c. Third stop, A = Alarm interrupt line, Q = EOP interrupt line. The bit position identifies the interrupt line.
2. Prestored parameters as follows:
- a. Controller is a 1731, sections 1 through 4. If no sections are selected, the End of Test stop will indicate the units that are Ready and Write enabled.
 - b. Data interrupt is on line 0.
 - c. Alarm interrupt is on line 0.
 - d. End of Operation interrupt is on line 0.
3. A typeout of selected parameters will occur after last stop.

D. MESSAGES

1. Typeouts

a. Normal program typeout

- 1) Test identification at start of test
MTSA1F, 1731/32 Special Test
IA = XXXX, FC = XX, CPOF, VER. 4.0
- 2) End of Test typeout

1st Stop A = 1F24 Q = STJP Word
2nd Stop A = Pass Count Q = Units tested

Units tested: Bits 00-07 correspond to units 0-7.

b. Error typeouts

- 1) Error messages are in format specified by SMM17.

A	Q	A	Q
1F38	S/J Parameter	YYZZ	Return Address

YY = Section Number
ZZ = Error Code

- 2) Additional test information for all errors (third stop):

A	Q
Current Unit	Units Tested

E. ERROR CODES

An error code is displayed in the lower two digits of the A register on the second stop of all error stop sequences.

<u>Error Code</u>	<u>Description</u>
01	No units are Ready and Write enabled.
02	Internal reject on Connect function.
03	Reject on status 2 input.
04	Reject on status 1 input.
05	Incorrect number of words read during CRC check.
06	Incorrect CRC word.
07	Alarm interrupt failed to occur.
08	Incorrect interrupt occurred, received alarm interrupt.
09	Incorrect status after Alarm interrupt.
0A	End of Operation interrupt failed to occur.
0B	Incorrect interrupt occurred, received EOP interrupt.
0C	Incorrect status after EOP interrupt.
0D	Reply received when attempting selection of 9 track and BCD mode, expected Reject.
0E	No parity error status after binary Write and BCD Read.
0F	Reply received from unprotected instruction, expected Reject.
10	Computer PROTECT or tape PROTECT switch was never set, function never rejected.
11	No Load Point status after Rewind.
12	Unexpected EOP or Interrupt status received after a no word Write.
13	Rewind function accepted while unit was Busy.
14	Unexpected EOP status after a no word Write.
15	No load point status after rewind.
16	Backspace function rejected when at load point.

<u>Error Code</u>	<u>Description</u>
17	Unit did not go Busy after backspace from load point.
18	Not used.
19	No load point status after Search File Mark Backward.
1A	Rewind unload function rejected.
1B	Data interrupt failed to occur.
1C	Incorrect interrupt occurred, received data interrupt.
1D	Incorrect status after Data interrupt.
1E	No Protect status after selecting unit.
1F	Unidentified interrupt.
21	Selected Section 6 on a 1731 controller.
XX22	Failed on Search File M ark off of load point. XX - ID record first word, expected 02
XX23	Failed on second Search File Mark off of load point. XX - ID record first word, expected 03
XX24	Failed on third Search File Mark off of load point. XX - ID record first word, expected 04

II. TEST DESCRIPTION

A. INITIALIZATION

1. Typeout title.
2. Parameter stop if bit 0 of Stop/Jump word set.
3. Return control to monitor.

B. SECTION 1 - CRC CHECK

1. If 7-track unit selected, test proceeds to next unit.
2. If 9-track unit selected, a binary count is written from $00FF_{16}$ to 0000.
3. Backspaces and reads 0101_{16} words, last word is CRC word.
4. Check is made for correct CRC word.

C. SECTION 2 - CONTROLLER PARITY ERROR CHECK

1. If 7-track unit selected, a word is written in binary mode.
2. Word is read back in BCD mode.
3. Parity error is expected.
4. If 9-track unit selected, an attempt is made to select BCD mode.
5. A Reject is expected.

D. SECTION 3 - CONTROLLER PROTECT CIRCUITRY CHECK

1. The computer will stop with the unit being tested in A, and either "1731" or "1732" in Q. All units will be run which are Ready and have Write enabled.
2. The computer PROTECT switch should be set and run.
3. Momentarily clear unit PROTECT switch on tape controller for unit which was indicated in "A".
4. After resetting unit PROTECT switch, clear PROTECT switch on computer.
5. Computer will stop with next unit to test indicated in "A" and go to 1 or
6. Test will advance to next section when all units have been tested.
7. If computer stops with Protect fault, clear computer PROTECT switch and run. Error message will be displayed.

E. SECTION 4 - EOT

1. An EOT marker must be on tape for this section.
2. Check for Load Point status set after rewind.
3. Check that EOP status does not set after a Write has been initiated, unit going Busy, with no data written.
4. Check that Rewind is not accepted while controller is Busy.
5. Check for proper End of Tape status.
6. Check that backspace from load point is accepted.
7. Check Rewind unload function.
8. In addition, for the 1732 only, a check is made for Search File Backward function from load point.

F. SECTION 5 - CONTROLLER INTERRUPT AND INTERRUPT STATUS

1. Write File Mark function is executed.
2. Check for Alarm interrupt occurring before EOP interrupt.
3. Check for EOP interrupt occurring after the Alarm interrupt.
4. Data interrupt is checked after a write function has been issued.

G. SECTION 6 - SEARCH TAPE MARK TEST

1. Write special series of tape records:
 - ID record N
 - Data record simulating a tape mark
 - ID record N + 1
 - True tape mark
 - ID record N + 2

Repeat this pattern three times.

2. Rewind the tape.
3. Search tape mark forward.
4. Read a record, must be ID Record 3.
5. Search tape mark forward.
6. Read a record, must be ID Record 6.
7. Search tape mark forward.
8. Read a record, must be ID Record 9.

III. PHYSICAL REQUIREMENTS

A. SPACE REQUIRED: About 1105₁₀ locations.

B. EQUIPMENT CONFIGURATION

1. 17X4 computer with 4K memory
2. 17X5 Interrupt Data Channel
3. 1706 Buffer Data Channel (Optional)
4. 1731 or 1732 Mag Tape Controllers and 601, 608, or 609 Mag Tape Units
5. A device for loading program

1731, 1732/60X MAGNETIC TAPE TEST
(MTXP4A Test No. 4A)

I. INTRODUCTION

This diagnostic test is designed to test the operation of the 1731 and 1732 Magnetic Tape Controllers and any 60X Tape Drive(s) connected to the controller.

II. REQUIREMENTS

A. HARDWARE

17XX Computer with 12K Memory
1731 or 1732 Magnetic Tape Controller
At least one 60X Magnetic Tape Drive

B. SOFTWARE

SMM17 with monitor based subroutines (MBS)
Teletype input package required for Section A

C. ACCESSORIES

Grounding strap for running RGG on 609's

III. OPERATIONAL PROCEDURE

A. RESTRICTIONS

The MTXP4A test cannot be multiplexed with other tests that exercise the input device on which the library is being loaded from due to the overlay loading requirements of MTXP4A.

B. LOADING PROCEDURE

MTX operates under the control of SMM17 MBS System. Bit 8 of the SMM parameter word must be set a load time to select MBS. If Section A is to be run, the teletype input package must also be selected by setting bit 6 of the SMM parameter word.

C. PARAMETERS

If bit 00 of the Stop/Jump word is set when this test is initiated, the following parameters may be entered.

Stop 1

A = 4A41 = Test ID

Q = STJP for M TX

Bit

- 0 - Stop for parameter entry
- 1 - Stop at end of section
- 2 - Stop at end of test
- 3 - Stop on error
- 4 - Repeat condition. (Refer to detailed description of each section.)
- 5 - Repeat section
- 6 - Repeat test
- 7 - Not used
- 8 - Omit typeouts
- 9 - Bias return address on error stops
- 10 - Reenter parameters on repeat test; bit 0 must also be set
- 11 - Report only one data word error on each buffer check
- 12 - Not used
- 13 - Enter parameters for Section 9 (Refer to Section 9 for detailed description.)
- 14 - Repeat Read/Write; bit 4 must not be set when this bit is used
- 15 - Refer to Error Code 42

Stop 2

A = Unit and density select.

- Bits 0-7 - Select the corresponding numbered unit to be tested
- Bit 8 - Select 800 BPI
- Bit 9 - Select 556 BPI
- Bit 10 - Select 200 BPI
- Bits 11-15 - Not used

If no units are selected in this parameter, the test attempts to find all units that are Ready and Write Enabled and will automatically select these units to test.

If no densities are selected, all densities are automatically selected.

One pass through each section is made for each density selected.

9-track units will run all passes at 800 BPI. If 800 BPI and a 601 drive are selected, the pass through each section will be at 556 BPI.

Q = Section select - Bits 0-15 select sections
0-F to be executed

- 0 - Status test
- 1 - Status test
- 2 - File Mark and Backspace test

- 3 - Illegal Functions test
- 4 - Visual Creep test
- 5 - Interrupt test
- 6 - Data Pattern and Record Length test, Character mode
- 7 - Data Pattern and Record Length test, Assembly/Disassembly mode
- 8 -
- 9 - RGG test
- A - Protect test
- B -
- C - Compatibility Write test
- D - Compatibility Read test
- E -
- F -

Stop 3

Interrupt line assignment in the form BLBL in A and BL00 in Q. B = Bit position in director status word associated with assigned line. L = Assigned interrupt line.

A = 3L4L

Q = 5L00

Stop 4

A = Not used

Q = Unit/controller type select parameter

If bit 8 is set, bits 0-7 (corresponding to each unit number) must be set to 1 if the unit is a 7-track or 0 if the unit is a 9-track. Section 0 of the test will check Status 2, bit 3 to ensure that the status bit is correct.

If bit 8 is not set (or if Section 9 is not selected), the FNDUNTP subroutine is called. This routine examines Status 2 of each unit, and sets or clears the bit in this parameter according to what the status bit says it is. All status checks throughout the test use this parameter when checking Status 2.

Bit 9 - 1 = 601 Tape Drive(s)

0 = Non-601 Tape Drive(s)

Bit 10 - 1 = 1731 Controller

0 = 1732 Controller

Bit 11 - Not used
 Bit 12 - Not used
 Bit 13 - 0 = Unload all tapes at the end of the Read compatibility section (Section C)
 1 = Do not unload tapes at the end of the Read Compatibility section (Section C)
 Bit 14 - Not used
 Bit 15 - Not used

D. SECTION INDEX

<u>Number</u>	<u>Description</u>
0	Section 0 tests the execution of Select functions, and the setting and clearing of certain status bits.
1	Section 1 tests the setting and clearing of the Data status and Lost Data status bits.
2	Section 2 tests the Write FM/TM, Read FM/TM, Search FM/TM, and Backspace.
3	Section 3 tests for reject to illegal functions.
4	Section 4 is the Visual Creep test.
5	Section 5 tests interrupts.
6	Section 6 tests the data paths with six different data patterns and ten different record lengths in Character mode.
7	Section 7 is identical to Section 6, except that the data is written and read in Assembly/Disassembly mode.
8	Section 8 is not used.
9	Section 9 is the RGG section.
A	Section A tests the Protect features; requires TTY input package.
B	Section B is not used at this time.
C	Section C is the Write Compatibility section.
D	Section D is the Read Compatibility section.
E	Section E is not used at this time.
F	Section F is not used at this time.

IV. OPERATOR COMMUNICATION

A. MESSAGE FORMATS

1. Initial Message

After MTX has been loaded and initialized, the following message is typed.

```
MTXP4A 1731/1732 MAG TAPE TEST
IA = XXXX   FC = XX   CP03
```

IA is the initial address of where the program is loaded. This is also the restart address. FC is the number of times MTX is to be repeated.

2. Other Messages

a. NO UNITS READY AND WRITE ENABLED REENTER PARAMS

This message is typed if no units were selected in parameter A2 and the test is unable to find any units ready and write enabled. Parameters must be entered after this message is typed.

b. SECTION 7 NOT EXECUTED ON 1731

This message is typed if Section 7 was selected to be run and a 1731 Controller was selected in parameter Q4 bit 9. The test continues to the next selected section.

Refer to the detailed description of Section 9 and A for a description of messages used in those sections.

3. Error Messages

Information typed out on errors is in the following format unless described otherwise in the error description.

A1	Q1	A2	Q2	A3	Q3	A4	Q4
4AX8	STJP	USEE	RTN	MBSERR	IO/RP	(A)	(Q)

A1 - 4A - MTX ID number

X - Number of stops on this error

8 - Error message

Q1 - Stop/Jump parameter word

A2 - U - Unit number

S - Test section number

EE - Error code

Q2 - Address pointer to where within a section the error occurred

A3 - MBS error code

0000 - Status timeout error

0001 - I/O timeout error

0002 - I/O response error

0003 - Status error

0004 - Interrupt timeout error

Q3 - IO - 10 - Output

20 - Input

30 - Function

RP - 10 - Reply

20 - External reject

30 - Internal reject

A4/Q4 - Contents of the A and Q registers when the last operation was executed.

A5/Q5 through A7/Q7 contains the following information:

MBS Error

<u>Code</u>	<u>A5</u>	<u>Q5</u>	<u>A6</u>	<u>Q6</u>	<u>A7</u>	<u>Q7</u>
0000	Actual Status 1	Actual Status 2	Clock	0CBS		
0001	Actual Status 1	Actual Status 2	Clock	N/A		
0002	Actual Status 1	Actual Status 2	N/A	N/A		
0003	Actual Status 1	Expected Status 1	Actual Status 2	Expected Status 2		
0004	Actual Status 1	Actual Status 2	Clock	Interrupt Line	Interrupt Mask	

A6 - Clock is the time in milliseconds

Q6 - C - 0 - Wait for status bit to clear

1 - Wait for status bit to set

B - Status bit being monitored

S - 0 - Status 1

1 - Status 2

Data errors are displayed using a standard SMM error typeout as follows:

A1	Q1	A2	Q2	A3	Q3	A4	Q4
				EXP	ACT	BUFF	BUFF
4A48	STJP	USEC	RTN	DATA	DATA	WORD	LENGTH

A1 through Q2 are the same as in the MBS errors described above

A3 - Data word expected

Q3 - Data word read from tape

A4 - Word number in the buffer for this error

Q4 - Bits 0-14 are the length of the buffer being checked

Bit 15 = 1 if in Character mode

= 0 if in A/D mode

Set bit 11 of the Stop/Jump word to eliminate typing out any more data errors on this buffer check. If bit 11 remains set, only one data error will be typed on all succeeding buffer checks.

B. MESSAGE DICTIONARY

The following paragraphs provide cross reference information for the error codes described.

<u>Error Code</u>	<u>MBS Error</u>	<u>Explanation</u>
00	All	This error is a result of the test attempting to find units available to run the test on. It can occur only if no units were selected at parameter entry time, and occurs before the first selected test section is initiated. It can only be caused by an internal or external reject to an Input status.
01	All	This error occurs in the FNDUNTP routine and is the result of a reject to a Select Unit function.
02	01 02	Reject to a Clear Controller function; Section 0 only.
03	01 02	Reject to a Select Unit and Density function; Section 0 only.
04	03	Status incorrect after a Select Unit and Density function was executed.

<u>Error Code</u>	<u>MBS Error</u>	<u>Explanation</u>
05	01	Reject to a Select Unit and BCD Mode function; Section 0. This function is executed only on 7-track units.
	02	
	03	
06	01	Reject to a Select Unit and Binary Mode function; Section 0.
	02	
	03	
07	01	Reject to a Select Unit and Assembly/Disassembly Mode function; Section 0. This function is not executed if a 1731 controller was selected in parameter Q4.
	02	
	03	
08	01	Reject to a Select Unit and Character mode function; Section 0.
	02	
	03	
09	02	Reject to a Select Unit function.
0A	01	Reply response was not received within the time expected on this tape motion function. A6 is the time allowed in milliseconds. A4 is the motion function attempted. (Refer to the following table.)
	02	Reject to this motion function. A4 is the motion function attempted.

<u>A4</u>	<u>Function</u>
0080	Write
0100	Read
0180	Backspace
0280	Write FM/TM
0300	Search FM/TM Forward
0380	Search FM/TM Reverse
0400	Rewind
0600	Rewind Unload

<u>Error Code</u>	<u>MBS Error</u>	<u>Explanation</u>
0B	02	Reject to a Select Mode and Density function.
0D	00	Busy status did not drop within the time limit specified for this operation. A4 is the operation executed. (Refer to error 0A.) A6 is the time limit in milliseconds.
	03	Status error before Busy dropped and the time limit expired for this operation. Error code 0D is the result of monitoring status during an operation.
0E	02	Status error after completion of an operation. The actual status displayed in this error is the last status read in the Monitor Status routine. A4 is the operation executed. (Refer to error 0A.)
0F	01	Reply response was not received to an input or output of data within the time limit specified. A6 is the time limit in milliseconds. A3 = 1020 = reject to output. Q3 = 2020 = reject to input.
	02	Reject to this output or input data.
10	02	Reply to a Select BCD with a 9-track unit selected.
11	02	Reply to a Select 200 or 556 BPI with a 9-track unit selected. A4 = 0020 = Select 200 BPI A4 = 0010 = Select 556 BPI
12	02	Reply to a Motion function while the controller is active executing a Read. A4 is the function attempted. (Refer to error 0A.) The record is 1000 characters long.
13	02	Reply to Select Unit function while the controller is active executing a Read.
14	02	Reply to a Select Assembly/Disassembly mode on a 1731 Controller.

<u>Error Code</u>	<u>MBS Error</u>	<u>Explanation</u>
15	03	<p>Data status did not set within 30 milliseconds (approximately) after a start Read function. The data that is being read is the remainder of a record after the following steps:</p> <ol style="list-style-type: none"> 1. Write a record 8 inches long on tape. 2. Write a file mark. 3. Backspace twice. 4. Write a file mark. 5. Backspace. 6. Read forward over file mark. <p>The Write file mark in step 4 should erase approximately 6 inches of the 8-inch record. This read is an attempt to read the remaining data.</p>
16	00	Busy status did drop set within 2 seconds after executing the Start Read described in error 15.
	03	Status error before busy dropped when executing the read described in error 15. File mark status in this error indicates that no data remained and the file mark written in step 2 was read.
17	00	Busy status did not drop within 2 seconds after this Read was executed. After this Read was started, one word was input and attempts were made to execute other motion functions. These functions should have been rejected and should not affect this Read operation.
	03	Status error before busy dropped on this Read. Refer to previous errors.
19	None	<p>Reply received to an input data when it should have been rejected. After data status had set, the program delays more than two tape character times before attempting this input. A3, Q3, and A4 are insignificant on this error.</p> <p style="margin-left: 40px;">Q4 = 0 = A/D mode Q4 = 1 = Character mode</p>

<u>Error Code</u>	<u>MBS Error</u>	<u>Explanation</u>
1A	None	Data status did not set within the time expected since the last output. Approximately 10 milliseconds are allowed from Start Write motion until the data status for the second output. Time between the remaining outputs varies from 50 to 300 microseconds, depending on the Density and Character or A/D mode. A3 = Last status input Q3 = Q register at input status time A4 = Number of words output or input before looking for this data status Q4 = 0 = A/D mode 1 = Character mode
1B	None	Data status was set approximately 5 microseconds after an output data was executed.
1C	None	Reject to an output data after data status was set. (Refer to error 1A for A3, Q3, A4, and Q4.)
1D	None	Data status did not set within the time expected since last input. Approximately 10 milliseconds is allowed from Start Read motion to the first data status. Time between the remaining inputs varies from 50 to 300 microseconds, depending on the Density and Character or A/D mode. (Refer to error 1A for A3, Q3, A4, and Q4.)
1E	None	Data status was set approximately 5 microseconds after an input data was executed.
1F	None	Reject to an input data after data status was set. (Refer to error 1A for A3, Q3, A4, and Q4.)
20	None	Data error. This record is four characters long with the first two characters being the same data as in a file mark. This record was written and read in Binary mode.
21	None	Data error. This record is the same as in error 20, except it was written and read in BCD mode.

<u>Error Code</u>	<u>MBS Error</u>	<u>Explanation</u>
22	None	Data error. This record is four characters long with the last two characters being the same data as in a file mark. This record was written and read in Binary mode.
23	None	Data error. This record is the same as in error 22, except it was written and read in BCD mode.
24	None	Data error. This record is one character and is the record being used to check Creep.
25	None	Data error. This record was written and read in Binary mode.
26	None	Data error. This record was written and read in BCD mode.
27	None	Data error. This record was written and read in Binary mode.
28	None	Data error. This record was written and read in BCD mode.
2E	None	Data error on reading a file mark. This FM was read in Binary mode.
2F	None	Data error on reading a file mark. This FM was read in BCD mode.
30	03	Status incorrect after a Clear Controller function.
31	01 02	Reject to an Interrupt Request function.
32	03	Status incorrect after an Interrupt Request function.
33	03	Status incorrect while waiting for a Write file mark function to cause an Alarm interrupt.
34	03	Status incorrect while waiting for a Backspace function to cause an End of Operation interrupt.

<u>Error Code</u>	<u>MBS Error</u>	<u>Explanation</u>
35	03	Status incorrect while waiting for a Read function to cause a Data interrupt.
36	01 02	Reject to a Clear Interrupt function.
37	03	Status incorrect after a Clear Interrupt function.
40	03	Ready and/or Busy status did not drop 10 milliseconds after a Rewind Unload was executed.
41	03	Ready status was still set approximately 5 seconds after a Rewind Unload was executed.
42	None	Unable to read the first record on tape at any density. Set bit 15 of the Stop/Jump word and hit run to retry reading the tape on this unit. If bit 15 is not set, this unit number will be cleared from the units selected for the test. A3 is the status after reading the first record on tape. Q3 is insignificant on this error.
43	None	The first record read from this tape is not the ID record for a compatibility tape. A3 is status 1 and Q3 is the first two characters in the first record on this tape (should be 1731 or 1732). (Refer to error 42 for retrying to read this tape.)
52	01 02	Reject of a protected output of a deselect unit function when the 1700 and the tested unit's PROTECT switches are on.
54	01 02	Reject of an unprotected output of a Clear Controller function when there is no unit selected, the 1700 and the tested unit's PROTECT switches are on.
55	01 02	Reject of an unprotected status instruction when the 1700 and the tested unit's PROTECT switches are on.
55	03	Status incorrect after a protected select unit instruction when the 1700 and the tested unit's PROTECT switches are on.
56	01 02	Did not receive an external reject to an unprotected output of a deselect unit function when the 1700 and the selected unit's PROTECT switches are on.

V. DESCRIPTIONS

A. GENERAL

1. MBS

This test is written using MBS as described in Supplement E. All I/O is accomplished using these subroutines, except in cases where more control is required by the test (for example, testing lost data status, the RGG section, etc.). These cases are described in the section descriptions.

2. Densities

Each section is executed once for each density selected before control is passed to the next section. (Refer to parameter entry A2.)

3. Interrupts

Interrupts are used only in the Interrupt Test Section (Section 5).

B. SECTION DESCRIPTIONS

The sections are described using a word flow type of description. Each major step in a test is numbered and events within the major steps are alpha characters. Repeat conditions parameter (bit 4 of the Stop/Jump word) will repeat the major step on the currently selected unit.

Section 0

Initialize the section and all MBS calls.

1. Execute a Clear Controller function and test Status 1.
2. Select a unit and density. Test Status 1 and 2. If bit 8 is set in parameter Q4, test the 7-track status bit after this select. Repeat step 2 for each unit.
3. If the next unit is 9-track, go to step 4. If it is 7-track, execute a select unit and BCD mode. Test Status 1 and 2.
4. Execute a select unit and Binary mode. Test Status 1 and 2.
5. If the controller is a 1731, repeat from step 3 on the next unit. If not, execute a select unit and Assembly/Disassembly mode. Test Status 1 and 2.
6. Execute a select unit and Character mode. Test Status 1 and 2. Repeat steps 3 through 6 for each unit.

7. Select first unit and call REWIND.
Monitor status for 30 seconds until Unit Busy status drops.
Check Status 1 and 2 after busy drops. Repeat step 7 for all units.
8. Select first unit and call WRTFM. Repeat for all units.
9. Do not execute steps 9 and 10 on any 9-tracks. Select the first unit and call BKSPC.
10. Call RDFM. Repeat steps 9 and 10 for all units.
11. Select the first unit and call WRTDAT. Length is one word (Character mode). Repeat for all units.
12. Select the first unit and call BKSPC. Repeat for all units.
13. Select the first unit and call RDDAT. Length is one word (Character mode) and data is not checked. Repeat for all units.
14. Call RWNDALL.
15. Select the first unit and call RDFM. Should read the FM/TM from step 8. Repeat for all units.
16. If the controller is a 1731, go to step 20. If a 1732, set LENGTH for 201 words (Character mode) and call WRTDAT. Repeat for all units.
17. Set A/D mode flag.
18. Select the first unit and call BKSPC.
19. Set LENGTH for 101 words (A/D mode) and call RDDAT. Expect Fill status. Do not check data. Repeat steps 18 and 19 for all units.
20. Check for any more densities. If another density is selected, repeat from step 1. If not, exit Section 0.

Section 1

Initialize the section and all MBS calls and call RWNDALL. Set A/D mode flag.

1. Select the first unit and call WRTFM. Repeat for all units.
2. Select the first unit, Density, Binary, and Character or A/D mode.

Function to Start Write motion.

Ensure that the data status bit sets and clears by monitoring the status and outputting data within 10 microseconds after the data status bit sets and then checking to see that the bit drops approximately 5 microseconds after output. Timeout allowed is approximately 8 milliseconds from when the

Write function is executed until the data status sets for the second output. Timeout between succeeding outputs varies from 50 microseconds at 800 BPI A/D mode to 300 microseconds at 200 BPI Character mode.

After nine outputs, monitor status until unit busy drops. Repeat step 2 for all units.

3. Select first unit and call BKSPC. Repeat for all units.
4. Select the first unit, Density, Binary, and Character or A/D mode.

Function to Start Read motion.

Ensure that data status bit sets and clears by monitoring the status and inputting data within 10 microseconds after the data status bit sets and then checking to see that the bit is not set approximately 5 microseconds after the input. Timeout allowed is approximately 8 milliseconds from when the Read function is executed until the first data status is set. Timeout between succeeding inputs varies from 50 microseconds at 800 BPI A/D mode to 300 microseconds at 200 BPI Character mode.

After nine inputs, monitor status until unit busy drops. Repeat step 4 for all units.

5. Select first unit and call BKSPC. Repeat for all units.
6. Select the first unit, Density, Binary, Character, or A/D mode.

Function a Start Read motion.

Wait for data status to set, input one word, and wait for data status to set again. Delay more than one-word time, attempt another input, and expect a reject.

Monitor status until not busy and check that Lost Data status is set. Repeat step 6 for all units.

7. Set Character Mode flag and repeat steps 1 through 6 in Character mode (except for 800 BPI). Go to step 8.
8. Check for any more densities. If another density is selected, repeat steps 1 through 7 in A/D and Character mode.

Section 2

Initialize the section and all MBS calls and call RWNDALL.

1. Select the first unit and call WRTFM. Repeat for all units.
2. Select the first unit. Set LENGTH for data required to write an 8-inch record on tape and call WRTDAT.
3. Call WRTFM.
4. Call BKSPC to backspace over FM/TM.
5. Call BKSPC to backspace over long record.
6. Call WRTFM. This FM/TM is written over the 8-inch record written in step 2 and should erase approximately 6 inches of the record.
7. If the controller is a 1731, call BKSPC and RDFM. If the controller is a 1732, call SCHFM twice to search reverse and forward.
8. Function a Start Read motion. When data status sets, input one word and then monitor status until unit busy drops. Check for any abnormal status other than Parity Error or Lost Data.
9. To repeat steps 2 through 8 on the current unit, set bit 14 of the Stop/Jump word. Repeat steps 2 through 9 for all units.
10. Select the first unit and if the controller is a 1731, call RDFM. If the controller is a 1732, call SCHFM to search forward. Repeat for all units.
11. Set BCD mode flag. On the first pass, steps 12 through 19 will be executed in BCD on all 7-track units.
12. Select the first unit. Build the following four-character record in the write buffer.

<u>Character</u>	<u>7-Track</u>	<u>9-Track</u>
0	\$F	\$13
1	\$F	\$13
2	\$22	\$22
3	01	01

The first two characters are FM/TM data.

Set LENGTH for four words and call WRTDAT.

13. Call WRTFM.
14. Reverse the data in the write buffer and call WRTDAT again. Repeat steps 12 through 14 for all units.

15. Select the first unit and backspace three times. Repeat for all units.
16. Select the first unit, set LENGTH for four words, and call RDDAT. Check data is the record written in step 12.
17. Call RDFM. Check the FM/TM data.
18. Call RDDAT and check that data is the record written in step 14. Repeat steps 16 through 18 for all units.
19. Select the first unit and if the controller is a 1731, call BKSPC four times (backup to the front of the FM written in step 3). If the controller is a 1732, call SCHFM twice to search reverse. Repeat for all units.
20. Clear BCD mode flag and repeat steps 12 through 19 in Binary mode. Go to step 21.
21. Check for more densities. If another density is selected, repeat steps 1 through 20 at the next density.

Section 3

Initialize the section and all MBS calls. Call RWNDALL.

1. Select the first unit. If this unit is a 7-track, try the next unit. Do not execute steps 2 through 4 on a 7-track unit.
2. Attempt to select BCD mode and expect a reject.
3. Attempt to select 200 BPI and expect a reject.
4. Attempt to select 556 BPI and expect a reject. Repeat steps 1 through 4 for all units.
5. Select the first unit, set LENGTH for 1000 characters, and call WRTDAT. Repeat for all units.
6. Select the first unit and call BKSPC. Repeat for all units.
7. Select the first unit and function a Start Read motion.

Input one word and then set up to expect reject to any function attempts.

Attempt to execute each of the following motion function while the controller is still busy. Expect rejects.

<u>Function Code</u>	<u>Operation</u>
\$ 80	Write
\$100	Read
\$180	Backspace
\$200	Illegal
\$280	Write FM/TM
\$300	Search FM/TM Forward
\$380	Search FM/TM Reverse
\$400	Rewind
\$480	Illegal
\$500	Illegal
\$580	Illegal
\$600	Rewind Unload
\$680	Illegal

If more than one unit is being used, attempt to select another unit and expect a reject.

Monitor status until unit is not busy and check for no abnormal status. Repeat step 7 for all units.

8. If the controller is a 1732, go to step 9. If the controller is a 1731, attempt to select A/D mode and expect a reject.
9. Check for any more densities. If another density is selected, repeat steps 1 through 8 at next density.

Section 4

Initialize section and all MBS calls. Call RWNDALL.

1. Select the first unit and call WRTFM.
2. Call BKSPC.
3. Repeat steps 1 and 2 30 times. This will erase approximately the first 15 feet of tape from load point. Repeat steps 1 through 3 for all units.
4. Select the first unit.
5. Set length for one-word (Character mode) buffer and call WRTDAT.
6. Call BKSPC.
7. Call RDDAT and check data.
8. Call BKSPC.

9. Repeat steps 5 through 8 200 times. Repeat steps 4 through 9 for all units.
10. Check for any more densities. If another density is selected, repeat steps 4 through 9 on all units at the next density.

This section requires that the operator visually watch the tape drives while steps 4 through 8 are executing to check that Creep is forward and not backward. If Creep is backward or excessively forward, adjustment should be made. To hang this section on a single, use the following steps.

- a. Restart the program at IA.
- b. Reselect parameters selecting only the unit desired, Section 4, and repeat section (bit 5 in Stop/Jump word).

Section 5

Initialize section and all MBS calls.

1. Select the first (next) unit.
2. Function clear controller, expecting a reply. Check status after the function.
3. Function Data, End of Operation, and Alarm interrupt requests, expecting a reply and no interrupts. Check status after the function.
4. Check repeat condition. Repeat 2 and 3 if repeat bit is set. If not set, continue to 5.
5. Repeat 1 through 4 for each unit selected.
6. Select the first (next) unit.
7. Function Clear Controller, expecting a reply. Check status after the function.
8. Check for end of tape status. Rewind unit if set.
9. Function Alarm Interrupt request, expecting a reply. Check status after function.
10. Function write FM/TM, expecting a reply.
11. Wait 2 seconds for an Alarm interrupt generated by the FM/TM.
12. When Alarm interrupt occurs, function Clear Interrupt, expecting reply. Check status after function.

13. Monitor status until busy drops. Check last status after busy dropped.
14. Function End of Operation interrupt request, expecting a reply. Check status after function.
15. Function Backspace, expecting a reply.
16. Wait 2 seconds for an End of Operation interrupt generated by the Backspace over FM/TM.
17. When the End of Operation interrupt occurs, function Clear interrupt, expecting reply. Check status after function.
18. Monitor status until busy drops. Check last status after busy dropped.
19. Function Data interrupt request, expecting a reply. Check status after function.
20. Function Read motion, expecting a reply.
21. Wait 2 seconds for a Data interrupt generated by the read FM/TM.
22. When Data interrupt occurs, function Clear interrupt, expecting reply. Check status after function.
23. Monitor status until busy drops. Check last status after busy dropped.
24. Check repeat condition. Repeat 7 through 23 if repeat bit is set. If not set, continue at 25.
25. Repeat 6 through 24 for each unit selected.
26. Repeat 1 through 25 for 20 times.
27. Repeat 1 through 26 for each density selected.

Section 6

Initialize section and all MBS calls. Clear Section 7 flag and set Character mode.

1. Call RWNDALL.
2. Set LENGTH of 1000 words
Set data pattern of all ones.
3. Build data pattern in the Write buffer, select the first unit, and call WRTDAT.
4. Call BKSPC.
5. Call RDDAT and check the data.

6. If bit 14 of the Stop/Jump parameter word is set, backspace and repeat steps 3 through 5 on the current unit. If not, repeat 3 through 6 for all units.
7. Change data pattern and repeat steps 3 through 6. After each of the following data patterns has been used, go to step 8.
 - a. All ones
 - b. All zeros
 - c. \$AA (\$2A for 7-track)
 - d. \$55 (\$15 for 7-track)
 - e. Sliding zeros pattern
 - f. Sliding ones pattern
8. Change buffer length, set data patterns of all ones, and repeat steps 3 through 7. After each of the following record lengths has been used, go to step 9.
 - a. 1000 words
 - b. 440 words
 - c. 225 words
 - d. 98 words
 - e. 59 words
 - f. 29 words
 - g. 15 words
 - h. 6 words
 - i. 2 words
 - j. 1 word
9. Set BCD mode flag and repeat steps 1 through 8 on all 7-track units.
10. Check for any more densities. If another density is selected, clear BCD mode flag and repeat steps 1 through 9.

Section 7

If the controller selected is a 1732, set the Section 7 flag, clear Character mode flag (set A/D mode) and execute steps 1 through 10 of Section 6.

If the controller is a 1731, typeout a message and go to the next section.

Section 9

This section is the 1700 RGG test. It is patterned after the 6000 RGG test described in Supplement A of this writeup. It is intended that this section of MTX provide basically the same maintenance capabilities as the 6000 RGG. However, since the 1700 Magnetic Tape Systems do not have all of the same capabilities as the 6000 tape systems (for example, reverse read, skip bad spot, etc.), some modifications are necessary to run an RGG on the 1700. The major differences between this RGG and the 6000 RGG are: reverse direction is checked by backspacing over the data records, and no attempt is made to check the time between the first two character's input on each record when reading forward.

Parameters for this section may be entered if bit 13 of the Stop/Jump word is set. Enter the following parameters.

- A1 = Test ID word
- Q1 = Stop/Jump word
- A2 = 00FG - Forward gap size
- Q2 = 00RG - Reverse gap size
- A3 = 000X - Bits 0 through 3 select subsections 1 through 4
- Q3 = DDDD - Fixed delay time for subsections 1 and 2

For parameters A2 and Q2, the gap size must be entered as two hexadecimal digits with bits 8 through 15 = 0. If all zeros are entered, that direction is not tested. The test converts the hexadecimal digits* into equivalent hundreds of an inch for record gap size to use. Minimum forward gap is .30 inch and minimum reverse is .50 inch. Increments above the minimum must be by .05. Maximum gap size is .95 inch. For parameter Q3, (enter in hexadecimal) the number of milliseconds** for the fixed delay time to use in subsections 1 and 2. This is the time the program delays between EOP on one record and issuing the Start Read motion for the next record. Minimum time allowed is 3 milliseconds and maximum time allowed is 4 seconds (4000 milliseconds). The prestored parameters are .35 for forward gap size, .50 for reverse gap size, and 3 milliseconds delay time.

* If 0055₁₆ is entered, the record gap size will be .55₁₀ inch.

** If 0250₁₆ is entered, the fixed delay time will be 250₁₀ milliseconds.

Four read options are available as selectable sections. Subsections 1 and 2 use a user-supplied fixed delay time between records. The delay supplied by the user is the number of milliseconds desired, with the minimum being 3 milliseconds. This ensures that tape at least comes to a near stop before the next operation. Sections 3 and 4 normally run an incremental delay from 3 milliseconds to 1 second. This is done in steps of .5 millisecond from 3 to 32 milliseconds; then 64, 128, 256, 512, and 1024 milliseconds are used. Sections 2 and 4 are identical to 1 and 3, respectively, except that prior to reading each group of records, 40 backspaces/forespaces are performed over an EOF in an attempt to reduce vacuum and pressure. If there is a large difference in the number of errors in Sections 2 and 4 compared to 1 and 3, capstan valve leaks or other pneumatic problems could exist. It would be advisable to monitor vacuum/pressure gages while Sections 2 and 4 are running to check for excessive drops.

To create worse-case capstan load, RGG attempts to maintain long tape loops when testing both forward and reverse. To do this, only 50 records are written when generating controlled gaps. These 50 records are read five times to create a sizable number of start/stop operations per read section.

RGG can be made a much more effective test by using it in conjunction with another margin's tool, 50 percent decrease in write and erase current. This technique exaggerates waveform dips etc., and is a definite aid in detection of unit deterioration.

The software controlled gap size is generated in the following manner.

1. Start write motion.
2. Output the correct number of data characters for a 1-inch record at the currently selected density (200, 556, or 800). The data character used is 37_{16} which (in BCD mode) writes ones on the six outside tracks on tape and a zero on the middle track.
3. Output the correct number of all zero characters for the selected gap size. In BCD mode (even parity) this will write all zeros on tape and when read, will appear as a record gap.

The message output to the teletype for this section (following steps 24 and 30) is as follows:

MTX	RGG	UNIT X	DENSITY XXX	SUBSEC X
DIR	GAP	RC SR	LR SE DE	DLY
F	.XX	XXX XXX	XXX XXX	XXX XXXX

The unit number, current density, and subsection number are in the top line. DIR is F or R (forward or reverse) for the direction being tested. GAP is the interrecord gap size in inches. RC is the number of records read (or backspaced over) in five passes over the 50 data records. SR is the number of short records read (less than the correct number of characters in the record). LR is the number of long records read (more than the correct number of characters in the record). SE is the number of records read where status errors occurred on the read. DE is the number of records where data errors occurred. DLY is the fixed delay time between read or backspace functions (subsections 1 and 2 only). SR, LR, SE, and DE will not apply for reverse direction (backspacing).

If RC is not 250 or if SR, LR, SE, and DE are not zero, the gap size is increased by .50 inch and the current unit, density, and direction are retried.

Use the following program flow.

1. Rewind all units and set BCD mode flag.
2. Get a unit to test. If no more units, exit Section 9.
3. Set subsection number to zero.
4. Reset all flags and counts according to the parameters.
5. Increment subsection number and if it is not more than 4, go to step 8.
6. If another density was selected in the MTX parameters and has not been run on this unit, set for the next density and go to step 3.
7. Set for the first selected density and go to step 2.
8. Clear the FWD/REV flag and if the current subsection is selected, proceed to step 9. If not, go to step 5.
9. Increment FWD/REV flag and if forward is selected, go to step 11.
10. Increment the FWD/REV flag and if reverse is selected, go to step 11. If not, go to step 4.

11. Select current Unit, Density, BCD, and Character mode. Write two file marks.
12. Write 50 1-inch records with software controlled gap size as calculated in step 4, 25, or 31.
13. Write two more file marks and backspace over both file marks.
14. Clear all error counts.
15. Rewind and read forward over the first two file marks off load point.
16. Execute reads nonstop until another file mark is read. If 50 records are read with no errors, proceed to step 17. If not, go to step 25 or 31.
17. If forward flag is set, proceed to step 18. If reverse flag is set, go to step 26.
18. Rewind and read forward over the first two file marks off load point.
19. If current subsection is 1 or 3, proceed to step 20. Subsections 2 and 4 backspace and read over the file mark 40 times as fast as possible before going to step 20.
20. Start read motion and inputting data (allow about 15 milliseconds for the first data character in). Check each data character as it is input, and count the number of characters input. Timeout between each character input. When nothing has been input for two character times, wait for EOP status to set and go to step 21.
21. If file mark status is set, go to step 23. If not file mark, increment the record count and check each of the following. If incorrect, increment the proper error count.
 - a. Number of characters input (short record, long record).
 - b. Status 1 should be EOP, Busy, and Ready only.
 - c. Any data errors on this record.
22. Delay a minimum of 3 milliseconds and repeat steps 20 and 21 until a file mark is read.
23. Repeat steps 18 through 22 four more times and go to step 24.
24. Report the results on the console teletype. After the typeout, check to see that 250 records were read and that there were no short records, long records, status errors, or data errors. If all right, rewind the current unit and go to step 10. If not, go to step 25.

25. Increase forward gap size by .05 inch, rewind the current unit, and repeat from step 11. If gap size of .95 inch has been tried, abort this unit.
26. Backspace over the last file mark read. If current subsection is 1 or 3, proceed to step 27. Subsections 2 and 4 read forward and backspace over the file mark 40 times as fast as possible before going to step 27.
27. Execute a Backspace function and wait for EOP to set. If file mark status is set, go to step 29. If not, increment record count and go to step 28.
28. Delay a minimum of 3 milliseconds and repeat step 27 until a file mark is detected.
29. Position the tape forward past the first file mark following the data and repeat steps 26 through 28 four more times. Go to step 30.
30. Report the results on the console teletype. If 250 data records were backspaced over, rewind the current unit and go to step 3. If not, go to step 31.
31. Increase reverse gap size by .05 inch, rewind the current unit, and repeat from step 11. If a gap size of .95 inch has been tried, abort this unit.

NOTE

To run RGG on a 609, a grounding strap must be used; ground inverter I302 on test point D of card location D12 in the 609. Also, the 609 must be described as a 7-track unit in bits 0 through 7 of the unit/controller type select parameter in Q4 of the parameter stop.

Section A

Initialize section and all MBS calls.

1. Type out SET 1700 PROTECT SWITCH AND CLEAR STOP SWITCH. Wait for carriage return on the teletype after A1=4A1F/Q1=00A1 pause.
2. Unprotected select first (next) unit.
3. Repeat 2 for each unit.
4. Repeat 2 and 3 16 times.
5. Type out SET UNIT PROTECT SWITCHES ON CONTROLLER. Wait for carriage return on the teletype after A1=4A1F/Q1=00A2 pause.
6. Protected deselect unit, expecting a reply; error 52 if reject.
7. Unprotected clear controller, expecting a reply; error 54 if reject.
8. Check repeat condition. Repeat 6 and 7 if repeat bit set. Continue at 9 if not set.
9. Protected select first (next) unit.
10. Unprotected status; error 55 if status rejected or status incorrect.
11. Unprotected deselect unit function expecting an external reject; error 56 if no external reject.
12. Check repeat condition. Repeat 10 and 11 if repeat bit set. Continue to 13 if not set.
13. Repeat 9 to 12 for each unit selected.
14. Repeat 9 to 13 16 times.
15. Type out CLEAR PROTECT SWITCHES. Wait for carriage return on the teletype after A1=4A1F/Q1=00A3 pause.

Section C

Initialize section and all MBS calls and call RWNDALL.

1. Write the following on each tape unit selected.

<u>Record</u>	<u>Buffer Word Length</u>	<u>Data</u>	<u>Modes</u>
1	6	ID	Character, Binary
2		FM/TM	Character, Binary
3	512	All ones	Character, Binary
4	512	All ones	Character, Binary
5	512	\$AA (\$2A 7-track)	Character, Binary
6	512	\$55 (\$15 7-track)	Character, Binary
7	512	Sliding zeros	Character, Binary
8	512	Sliding ones	Character, Binary
9	256	Sliding ones	Character, Binary
10	128	Sliding zeros	Character, Binary
11	64	\$55 (\$15 7-track)	Character, Binary
12	32	\$AA (\$2A 7-track)	Character, Binary
13	16	All zeros	Character, Binary
14	8	All ones	Character, Binary
15		FM/TM	Character, Binary
16	512	Sliding Zeros (7-track only)	Character, BCD
17		FM (7-track only)	Character, BCD
18 through 29	Same as records 3 through 14, except the buffer length is half (number of characters written is the same). (These records are not written on the tape if the controller being used is a 1731).		A/D, Binary
30		FM/TM	Character, Binary

The ID record is six characters long and contains information on what type of controller was used to write this tape, if the tape drive is a 601, and the density(s) used.

<u>Character</u>	<u>Data</u>	<u>Description</u>
0	17	1731 or 1732
1	31 or 32	Controller
2	00 or 06	06 only if 601 drive
3	00 or 01	01 only if 601 drive
4	00	
5	0X	X=1-7 for densities selected.

2. If another density is selected, write all of the same records at the next density, except the ID record and the first FM/TM. For 9 tracks, the data will be written again at 800 BPI.
3. Repeat again if a third density is selected.
4. Write a second FM/TM at the end of the tape.
5. If bit 13 of parameter Q4 is set, exit to the next section. If not, proceed with steps 6 and 7.
6. Select the first unit and execute a Rewind Unload function.
 Delay about 10 milliseconds and check that Ready status dropped.
 Delay about 5 seconds and ensure that Ready status has not set again.
 Repeat step 6 for all units.
7. Type TAPES UNLOADED HIT RUN TO CONTINUE message and halt before exiting Section C.

Section D

Initialize section and all MBS calls, and call RWNDALL.

1. Select the first Unit, Character, Binary modes, and (if 7-track) 200 BPI density.
2. Attempt to read the first record on tape. If read completes successfully, go to step 5. If not, proceed to step 3.
3. Repeat step 2 at 556 BPI and if it fails again, try 800 BPI (on a 9-track, all attempts will be at 800 BPI).
4. After reporting the error, check bit 15 of the Stop/Jump word. If it is set, repeat steps 1 through 3 on this unit. If this parameter bit is not set, delete this unit number from the units selected for testing, and repeat steps 1 through 3 on the next unit.
5. Check to see that the first two characters read on the first (ID) record are 1731 or 1732 and proceed to step 6. If not, go to step 4.

6. Read all of the rest of this tape expecting it to contain data as described in Section C. Use only the densities contained in character 6 of the ID record.
7. Repeat steps 1 through 6 for all units.

C. SUBPROGRAM DESCRIPTION

1. WRTDAT

The Unit, Density, Binary, or BCD mode and Character or A/D mode has been selected before this routine is called. This routine sets up MBS calls for a Write Data record. The length of the data output is determined by what has previously been set in location LENGTH. Data is then output from the write buffer (WRBUF). Status is monitored until busy drops and then the status is rechecked. A 2-second maximum is allowed in the Monitor Status routine.

The repeat condition parameter (bit 4 of the Stop/Jump word) is checked in each of the following situations. If it is set, the MVRPT subroutine is called to do a Backspace.

- a) If an error occurs when outputting data.
- b) If an error occurs while monitoring status.
- c) If an error occurs on the recheck status.
- d) Following the recheck status.

This routine does not exit until the write is complete and the repeat condition parameter is cleared.

2. MVRPT

This routine may be called to execute a Backspace, Read forward one record, or a Rewind. The following steps are executed.

- a) Issue a Clear controller.
- b) Reselect the current unit, density, and modes.
- c) Execute the Backspace, Read, or Rewind.
- d) Monitor status until busy drops. Only ready and busy status are monitored. If busy does not drop within 20 seconds, error code 0C is generated.
- e) Issue another Clear controller.
- f) Reselect the current unit, density, and modes.
- g) Exit MVRPT.

3. WRTFM

This routine executes a Write FM/TM. The rules for Select, Monitor status and repeat condition apply to this routine the same as in WRTDAT.

4. RDDAT

This routine executes a Read Data record and the data is input to the Read buffer (RDBUF). If an error code is indicated (A \neq 0 when RDDAT is called), the data in the Read buffer is checked against the data in the Write buffer after the read has completed. The rules for Select, Monitor status, and Repeat condition apply to this routine the same as in WRTDAT, except that Repeat condition is also checked after the data check.

5. RDFM

This routine executes a Read and FM/TM is expected. LENGTH is set for a one-word input. If A/D mode is selected, Fill Status is expected. If the unit is a 7-track, the binary/BCD flag indicates whether or not a parity error is expected. If an error code is indicated (A \neq 0 when RDFM is called), the data input is checked to be file mark data. The rules for Select, Monitor status, and Repeat condition apply to this routine the same as in the WRTDAT routine, except that Repeat condition is also checked after the data check.

6. BKSPC

This routine executes a Backspace. The FM/TM detect flag and the binary/BCD flag dictate which status to check for. The rules for Select, Monitor status, and Repeat condition apply to this routine the same as in WRTDAT, except that on Repeat condition, the MVRPT routine is called to execute a Read forward one record.

7. SCHFM

This routine executes a Search FM/TM forward or reverse. The location SCHFMT must contain the function code to be used when this routine is called. The rules for Select, Monitor status, and Repeat condition apply to this routine the same as in WRTDAT, except that if the operation is a Search reverse, on Repeat condition the MVRPT routine is called to execute a Read forward one record.

8. REWIND

This routine executes a Rewind to the unit that has been previously selected. This routine does not wait for the Rewind to complete, but exits after the function has been executed.

9. RWNDALL

This routine executes a Rewind on all units that have been selected in the parameter entry A2. This routine does not exit until all units are rewound (unit busy status has dropped on all units).

SUPPLEMENT A - RECORD GAP GENERATOR (RGG)

I. PURPOSE

To test Start/Stop characteristics of 604, 607, 657, and 659 Tape Drives on a 3000 Series Controller, 6681 Channel Converter, and 6000 Data Channel on a 6000 Computer.

II. USAGE

A. BACKGROUND

RGG is a tape unit diagnostic written to detect Start/Stop problems and provide the capability to predict potential problems before they create deteriorated performance. The CE should use RGG on each tape transport on a regularly scheduled basis (every 2 weeks) as a replacement of all previous preventive maintenance techniques of testing Start/Stop time. The following benefits are obtained from this procedure.

1. Decrease PM time spent on Start/Stop testing off-line. Testing can be done on-line with a substantial savings of manpower.
2. Prevent over PMing which generally adds to problems due to constant tweeking. The unit is not worked on for Start/Stop problems if RGG runs.
3. Detect deteriorated unit before it is serious so that offline PM can be scheduled to do required tweeking.
4. RGG is a better yardstick for determining that a unit that had a poor Start/Stop time has been repaired. At times, waveforms that look good do not yield an error-free running of RGG. This is generally caused by what different individuals looking at a scope call good.
5. The end result is a better running unit for the customer.

Under normal operation a tape unit performs its Stop/Start operations in a record gap of approximately .75 inch. It has been proven with RGG that the units can do a Forward Read, Stop, Start, and approach full speed in a record gap of .30 inch, only 40 percent of the normal gap available. Testing indicated that reliable operation could be expected with a gap size of .35 inch on forward operations and .50 inch on reverse (.43 forward and .50 reverse on the 604). Normally, you would expect reverse to operate at .45 inch if the unit had a .1 inch creep.

However, this is not the normal situation due to varied techniques of setting up creep. Creep setup is quite important in running RGG in reverse. The more forward creep you have, the bigger gap Start/Stop requires in reverse, since you continue to move tape after detecting end-of-operation, eating up valuable gap before stopping; therefore leaving less gap to get started in before you get to the first character of the next record. The suggested technique of setting up creep on units that are to run RGG at reverse and reliably detect reverse Stop/Start deterioration is as follows:

1. On a 607, set the creep delay to 1.0 microsecond.
2. On a 604, set the creep delay to 2.0 microseconds.

These delay settings should give you a slight forward creep. Any variations between units is now caused by Start/Stop variations and not differences in delay settings. RGG can now effectively measure and report these variations. Reverse should run reliably at .50 inch. In the 604, a 1-microsecond delay occurs on forward after end-of-operation is detected before tape motion is dropped. This delay should be set up properly on all units.

Another important delay setting in the 604/607 is the Start Read delay that enables sprocket pulses to be generated. On a 607, this delay is 2 microseconds and is used to gate out any noise during the time the unit is getting up to speed. If this delay is set too long, RGG will require a longer gap to run error-free. If it is too short, potential problems may get by RGG and not be detected until they are serious. All units should be set to the required delay so that all are consistent.

Another factor that should remain as consistent among units as possible is the quality of tape used. One good tape should be used to run RGG, and it should be used on all units tested. RGG only uses the first 8 feet of tape off load point. However, testing one unit running two gaps forward and two reverse with all four read sections selected will make 166 passes over this 8 feet of tape. The test tape should be properly maintained by moving the load point marker 4 feet up the tape every 30 units tested. This will ensure that RGG is run on two types of tape surfaces within the same run (4 feet of tape with less than 5000 passes and 4 feet with more than 5000 passes).

B. OPTIONS

The user controls the margins of testing by defining the gap sizes to run, as well as, the rate of Start/Stop operations. Normally, the margins selected should include a record gap size that is not expected to run error-free and one that is. If the expected error-free gap does not run, the unit should be worked on off-line. If it does run, monitor the gap expected to create the errors to detect unit deterioration and plan preventive maintenance. The following options exist.

1. The user can define the first gap size in hundredths of inches to start testing forward and/or reverse. Selecting a gap size of zero for forward or reverse will bypass testing that direction.
2. He can also define the maximum number of gap sizes to run. Testing of a particular direction will terminate once this count is reached or when all read options selected run a gap with no failures.
3. The user can define the gap increment. This increment, in hundreds of inches, is added to the gap size each time a new gap is tested. The increment applies to both forward and reverse.
4. Four read options are available as selectable sections. Section 1 and 2 use a user-supplied fixed delay time between records. The delay supplied by the user is the number of .5 microsecond desired, with the minimum being 3 microseconds (delay count of 6). This ensures that tape at least comes to a near stop before the next operation. Sections 3 and 4 normally run an incremental delay from 3 microseconds to 1 second. This is done in steps of .5 microsecond from 3 to 32 microseconds, then 64, 128, 256, 512, and 1024 microseconds are used. An additional parameter (D2) can be used to extend the delays used to include 2, 4, 8, 16, 32, and 64 seconds. Sections 2 and 4 are identical to 1 and 3, respectively, except that prior to reading each group of records, 40 backspaces/forespaces are performed over an EOF in an attempt to reduce vacuum and pressure. If there is a large difference in the number of errors in Sections 2 and 4 compared to 1 and 3, capstan valve leaks or other pneumatic problems could exist. It would be advisable to monitor vacuum/pressure gages while Sections 2 and 4 are running to check for excessive drops.
5. To create worse-case capstan load, RGG attempts to maintain long tape loops when testing both forward and reverse. To do this, only 50 records are written when generating controlled gaps. These 50 records are read

five times to create a sizable number of Start/Stop operations per read section.

6. RGG can be made a much more effective test by using it in conjunction with another margins tool, 50 percent decrease in write and erase current. This technique exaggerates waveform dips, etc., and is a definite aid in detection of unit deterioration.



1732-2/615-73/93 or 1732-3/616-72/93/95 MAGNETIC TAPE SUBSYSTEM TEST

(PET04B Test No. 4B)

I. OPERATIONAL PROCEDURE

A. REQUIREMENTS

1784-1 or -2, 1732-2, 615-73 or 615-93,
1784-1 or -2, 1732-3, 616-72 or 616-92 or 616-95
8K of memory
TTY or equivalent to run protected (optional)

B. RESTRICTIONS

1. Do not run Sections 3, 4, 5, 9, and 10 in Phase-Encode while in A/Q mode.
2. Section O will not run in Phase Encode (1600 BPI) and via the A/Q channel on transports faster than 37 IPS.

C. LOADING PROCEDURE

This test will load and execute correctly only under SMM17. The calling sequence is that specified by the SMM17 system being used. If the equipment address is zero when called, the test will use the prestored equipment address (IA+6). If the director function of the test is 3 when the test is initialized, the A/Q-DSA parameter will be preset to DSA mode.

D. TESTING PROCEDURE

In the example of a 1732-2 system, the A/Q-DSA parameter (Q4) selects the data path. Q4=0000 selects A/Q I/O only and Q4=0002 selects the DSA data path. The test will release control to multiplex with other tests after the controller goes busy, and at such a time that control can be released for an indefinite period without affecting system integrity.

E. PARAMETERS

During initialization the test will status the equipment address provided and attempt to determine the unit select parameter for the user.

Stop 1

A1 = 4B41 Test ID (4B), four stops (4), parameter stop (1)
Q1 = STJP Test unique Stop/Jump parameter (refer to TSTJP).

3. Selection of sections to be executed will be made in accordance with the following criteria (X indicates mode in which each section can be executed):

	<u>Section Select Parameter</u>	<u>NRZI</u>		<u>Phase Encode</u>	
		<u>A/Q</u>	<u>DSA</u>	<u>A/Q</u>	<u>DSA</u>
Bit 0	Section 0 - Status Test	X	X	X*	X
Bit 1	Section 1 - Interrupt Test	X	X	X	X
Bit 2	Section 2 - Backspace Test	X	X		X
Bit 3	Section 3 - File Mark Forward/ Reverse	X	X		X
Bit 4	Section 4 - Variable Block Length Fixed Data	X	X		X
Bit 5	Section 5 - Random Data and Length	X	X		X
Bit 6	Section 6 - Non-Stop Write/Read	X	X	X	X
Bit 7	Section 7 - Variable Block Length Fixed Data in Assembly/ Disassembly Mode	X	X	X	X
Bit 8	Section 8 - Random Data and Length in Assembly/Disassembly Mode	X	X	X	X
Bit 9	Section 9 - Write/Read/Compare Compatibility Test Generator	X	X		X
Bit 10	Section 10 - Read Only-Compatibility Test	X	X		X
Bit 15	----- Set for 7-Track Transport (disables density select)				

*Will not operate in 50 ips.



4. Stops

The following types of stops are capable in this test.

Type 1

Stop to enter parameters. This will occur if Stop/Jump bit 0 is set. (see parameters.)

A1 = 4B41 Q1 = TSTJP

Type 2

Stop at end of section. This stop will only occur if the test STJP bit 1 is set.

A1 = 4B32 (4B = test ID, 3 = number of stops, 2 = end of section stop)

Q1 = TSTJP

A2 = 0S00 (S = section number)

Q2 = RTA - Return address

A3 = Number of errors this section

Q3 = Number of errors so far in test

Type 4

Stop at end of test. This stop will only occur if Stop/Jump bit 2 is set.

A1 = 4B34 (4B = test ID, 3 = number of stops, 4 = end of test)

Q1 = TSTJP

A2 = NNNN - Number of passes completed

Q2 = RTA - Return address

A3 = Number of errors this pass

Q3 = Number of errors since start

Type 8

Stop on error. This stop will only occur if Stop/Jump bit 3 is set.

A1 = 4BN8 (N = number of stops (3 or 4), 8 = error stop)

Q1 = TSTJP

A2 = USEE (U = unit number, S = section number, EE = error code.

Refer to the error codes for their respective meaning and additional stop information.)

Q2 = RTA - Return address

A3, Q3 through A4, Q4 Refer to the respective error code for these meanings.

Type F

Messages

These are not errors, but messages to the operator to perform some function.

A1 = 4B1F (4B = test ID, 1 = number of stops, F = message stop
Q2 = 00MM (MM = message number)

MM = 0000 Messages clear the PROGRAM PROTECT switch, if set. Set the SELECTIVE STOP switch.

MM = 0055 Section 9 only. All selective units have been read and compared. Hit run (CR) to terminate this section. Otherwise, make any array of the selected drives ready again. To repeat the read and compare operation, zero either the A or Q register.

MM = 000F Section 10 only. Clear SELECTIVE STOP switch, and set PROGRAM PROTECT switch.

5. Word Messages

Word messages are only available if there is a TTY or equivalent available.

a. Initialization of the test:

PET04B, 1732-2, -3 MAGNETIC TAPE TEST
IA=XXXX, FC=XX
CP=2F VR4.0

b. Still in initialization:

CLEAR PROTECT SWITCH, IF SET, SET STOP SWITCH.

c. Still in initialization, but only if the TTY input package is enabled:

TO RUN PROTECTED, SET PROGRAM PROT-CLEAR STOP SWITCH.

6. Error Codes

01 Write parity error. A parity error was detected at the end of a write record sequence. The test will not attempt recovery from this error.

A3 = Status 1

Q3 = Equipment address

- 02 Read parity error. A parity error was detected at the end of a read record sequence. No recovery is attempted.
A3 = Status 1
Q3 = Equipment address
- 03 Data compare error. Data read does not equal the expected data.
A3 = Data read
Q3 = Expected data
A4 = Word number in record of failing word of data
Q4 = Record length of failing record.
- 04 Incorrect record length. Record read was not the expected length.
A3 = Actual word length
Q3 = Expected word length
A4 = Actual LWA of the record read
Q4 = Expected LWA of the record
- 05 Device did not go busy after Function 1 command.
A3 = Status 1
Q3 = Function 1 command
- 06 End of operation interrupt error. NO EOP interrupt after requesting and generating an EOP interrupt.
A3 = Status 1
Q3 = Equipment address
- 07 Data interrupt error. No data interrupt after requesting and generating a data interrupt.
A3 = Status 1
Q3 = Equipment address
- 08 Alarm interrupt error. No alarm interrupt after requesting and generating an alarm interrupt.
A3 = Status 1
Q3 = Equipment address
- 09 Fill status set after read. Fill status detected after read when not expected.
A3 = Status 1
Q3 = Equipment address

- 0A Selected units status 2 does not = phase encode + 1600 BPI.
A3 = Status 2
Q3 = Equipment address
- 0B Phase encode error status set, and should not be. Status 2 phase encode warning, ID abort, or phase encode lost data is detected.
A3 = Status 2
Q3 = Equipment address
- 0C Selected 7-track drive status 2 is incorrect immediately after selecting unit.
- 0D Selected 7-track drive status 2 does not equal 7-track, write enable, and correct density.
- 0E-0F Not used.
- 10 Selected unit is not ready after being selected.
A3 = Status 1
Q3 = Equipment address
- 11 Selected unit does not have write enabled, no write ring.
A3 = Status 2
Q3 = Equipment address
- 12 Selected unit busy and should not be.
A3 = Status 1
Q3 = Equipment address
- 13 Selected unit should be in 800 BPI after being commanded to 800 BPI.
A3 = Status 2
Q3 = Equipment address
- 14 Selected unit is in 556 BPI after being commanded to 200 or 800 BPI.
A3 = Status 2
Q3 = Equipment address
- 15 Selected unit should be in 556 BPI after being commanded to 556 BPI.
A3 = Status 2
Q3 = Equipment address

- 16 Selected unit is in 800 BPI after being commanded to 556 or 200 BPI.
A3 = Status 2
Q3 = Equipment address
- 17 Selected unit should be in 200 BPI after being commanded to 200 BPI.
A3 = Status 2
Q3 = Equipment address
- 18 Selected unit should be at load point.
A3 = Status 1
Q3 = Equipment address
- 19 End of tape is detected and should not be.
A3 = Status 2
Q3 = Equipment address
- 1A End of operation is set while controller is still active.
A3 = Status 1
Q3 = Equipment address
- 1B Unit should be busy after a start motion write command.
A3 = Status 1
Q3 = Equipment address
- 1C Unit should not have phase encode, alarm, file mark, load point, or lost data set.
A3 = Status 1
Q3 = Equipment address
- 1D Controller active should not be set after rewind to load, controller is not busy, and at load point.
A3 = Status 1
Q3 = Equipment address
- 1E Controller should be active after accepting a start motion write command.
A3 = Status 1
Q3 = Equipment address

- 1F Controller should have sensed alarm from reading a binary record
in BCD mode.
- A3 = Status 1
 Q3 = Equipment address
- 20 Controller should have sensed a parity error from reading a
binary record in BCD mode.
- A3 = Status 1
 Q3 = Equipment address
- 21 Controller should have sensed lost data after start motion read, wait
for data, wait for no data status, and attempting an input instruction.
- A3 = Status 1
 Q3 = Equipment address
- 22 Alarm status should be set from error 21 conditions.
- A3 = Status 1
 Q3 = Equipment address
- 23 Controller should have sensed end of file mark after initiating a
read to end of file command. Controller became not busy.
- A3 = Status 1
 Q3 = Equipment address
- 24 Controller should have alarm condition from detecting an end of
file mark from error 23 condition.
- A3 = Status 1
 Q3 = Equipment address
- 25 A reply was received to the input instruction from error 21
condition.
- A3 = Status 1
 Q3 = Equipment address
- 26 Lost data is detected after initiating a start motion, record read,
then waiting for EOP. At EOP time, lost data is set.
- A3 = Status 1
 Q3 = Equipment address

- 27 Fill status is not set after writing 19 words of data in Character mode and reading back 10 words in A/D mode.
A3 = Status 1
Q3 = Equipment address
- 28 Selected phase encode unit status 2 does not equal phase encode.
A3 = Status 2
Q3 = Equipment address
- 29 Selected phase encode unit status does not equal 1600 BPI.
A3 = Status 2
Q3 = Equipment address
- 2A Phase encode lost data or warning is set after reading a record in PE mode.
A3 = Status 2
Q3 = Equipment address
- 2B Alarm status is set after reading a record in Phase Encode mode.
A3 = Status 1
Q3 = Equipment address
- 2C-2D Not used.
- 2E Internal reject on input instruction.
A3 = Contents of A after the reject
Q3 = Contents of Q at the reject
- 2F External reject on input instruction.
A3 = Contents of A after the reject
Q3 = Contents of Q at the reject
- 30 Interrupt occurred on one of the selected interrupt lines, but no status was set, ghost interrupt.
A3 = Status 1
Q3 = Equipment address
- 31 Unrequested data interrupt occurred.
A3 = Status 1
Q3 = Equipment address
- 32 Unrequested end of operation interrupt occurred.
A3 = Status 1
Q3 = Equipment address

- 33 Unrequested alarm interrupt occurred.
A3 = Status 1
Q3 = Equipment address
- 34 Unrequested phase encode interrupt occurred.
A3 = Status 2
Q3 = Equipment address
- 35 Unrequested interrupt occurred from this controller with no supporting status.
A3 = Status 1
Q3 = Equipment address
- 36-3D Not used.
- 3E Internal reject on output instruction.
A3 = Contents of A on reject
Q3 = Equipment address
- 3F External reject on output instruction.
A3 = Contents of A on reject
Q3 = Equipment address
- 40-68 Not used.
- 69 Controller is busy too long. In the case of a 17X6, a terminate buffer will be issued following this error. Control will not be accepted by the test until busy drops.
A3 = Status 1
Q3 = Equipment address
- 6A Time-out waiting for Status 1 condition "NNNN". The test will continue after this error.
A3 = Status 1
Q3 = Status bit to set if positive.
= Status bit to clear if negative.

II. SECTION DESCRIPTION

A. SECTION 0 - STATUS CHECK

1. Loop 0

- a. Wait until controller is inactive.
- b. Select unit and binary modes.

- c. Check Ready, error 10.
 - d. Check status 2 for Write Enable, error 11.
 - e. Check status 1 for Not Busy, error 12.
 - f. Repeat conditions.
 - g. Go to next unit.
2. Loop 1
- a. Skip to loop 4 if no density is selected. (Section select parameter bit 15 is set.)
 - b. Select Unit, Binary, and 800 BPI (1600 if PE).
 - c. If unit is phase encoded, check status 2 for PE transport, error 28 and status 2 for 1600 BPI, error 29.
 - d. If unit is not phase encoded, check status 2 for 800 BPI, error 13 and status 2 for non-556 BPI, error 14.
 - e. Repeat conditions.
 - f. Go to next unit.
3. Loop 2
- a. Skip to next unit if this unit is 9-track.
 - b. Select Unit, Binary, and 556 BPI modes.
 - c. Check status 2 for 556 BPI, error 15.
 - d. Check status 2 for non 800 BPI, error 16.
4. Loop 3
- a. Skip to next unit if this unit is 9-track.
 - b. Select Unit, Binary, and 200 BPI modes.
 - c. Check status 2 for non 556 or 800 BPI, error 17.
 - d. Go to next unit.
5. Loop 4
- a. Rewind all units.
 - b. If section select bit 15 is set, do not select 7-track density. Otherwise, select 800 BPI.
 - c. Select Unit, Binary, Density (if any) modes.
 - d. Wait for not busy, error 69.
 - e. Check status 1 for load point, error 18.
 - f. Check status 1 for not controller active, error 1D.
 - g. Check status 1 for end of tape marker, error 29.
 - h. Repeat conditions.
 - i. Go to next unit.

6. Loop 5

- a. Clear DSA mode of operation if set.
- b. Select Unit and Binary modes.
- c. Write 256 words of data, error 01.
- d. Restart DSA I/O mode if it was set.
- e. Check status 1 for controller active, error 1E.
- f. Check status 1 for non-EOP, error 1A.
- g. Check status 1 for busy, error 1B.
- h. Wait for EOP, error 69.
- i. Write 64 words of data, error 01.
- j. Wait for not busy, error 69.
- k. If NRZI unit, check status 1 for not file mark, load point, controller parity error, lost data, or alarm, error 1C.
- l. If phase encode unit, check status 1 for not alarm, error 2B and status 2 for PE transport, 1600 BPI and no I.D. abort, PE warning, PE lost data, error 2A.
- m. Repeat conditions.
- n. Go to next unit.

7. Loop 6

- a. Select Unit and Binary modes.
- b. Generate 80 words of FF pattern.
- c. Write EOF mark on all units, error 69.
- d. Write 64 words of all ones pattern on all units, error 01.
- e. Write EOF mark on all units, error 69.
- f. Backspace three times on all units, error 69.
- g. Wait for not busy, error 69.
- h. Select unit and BCD modes.
- i. Read one record of 64 frames of data.
- j. Wait for not busy, error 69.
- k. Check status 1 for not lost data, error 26.
- l. If 7-track unit, check status 1 for alarm status due to reading binary data in BCD mode, error 1F and check status 1 for parity error status due to reading binary data in BCD mode, error 20.
- m. Go to next unit to read one record.

8. Loop 7

- a. Select Unit and Binary modes.
- b. Start motion read.
- c. If A/Q mode,
 - 1) Wait for data status.
 - 2) Wait for no data status.
 - 3) Try to input data, reject expected if reply, error 25.
 - 4) Wait not busy, error 69.
 - 5) Check status 1 for lost data, error 21.
 - 6) Check status 1 for alarm, error 22.
- d. Go to next unit.
- e. Select Unit and Binary modes.
- f. Search file mark forward.
- g. Wait for not busy, error 69.
- h. Check status 1 for alarm, error 24.
- i. Check status 2 for file mark, error 23.
- j. Go to next unit.

9. Loop 8

- a. Select Unit and Binary modes.
- b. Wait for not busy, error 69.
- c. Write 19 frames of data, error 01.
- d. Wait for not busy, error 69.
- e. Backspace.
- f. Wait for not busy, error 69.
- g. Select A/D mode, error 3E or 3F.
- h. Read record, error 02.
- i. Wait for not busy, error 69.
- j. Check status 1 for fill status, error 27.
- k. Go to next unit.
- l. End of Section 0.

B. SECTION 1 - INTERRUPT TEST

1. Rewind all units, errors 3E/3F and 69.
2. Generate pattern of 00FF.
3. Write EOF on all units.
4. Select unit.
5. Request EOP interrupt line.

6. Enable EOP interrupt request.
7. Write 98-word record on tape, error 01

NOTE

Interrupt is received.

8. Wait for not busy, error 69.
9. Check that interrupt occurred, error 06.
10. Clear EOP interrupt line request.
11. Backspace, error 3E/3F.
12. Wait for not busy, error 69.
13. If not file mark status, backspace again.
14. Clear controller, error 3E/3F.
15. Request alarm interrupt line.
16. Enable alarm interrupt request, error 3E/3F.
17. Start motion and read, error 3E/3F.

NOTE

Interrupt has occurred.

18. Wait for not busy, error 69.
19. Clear alarm interrupt line request.
20. Check that interrupt occurred, error 08.
21. If NRZI unit,
 - a. Request data interrupt line.
 - b. Enable data interrupt request, error 3E/3F.
 - c. Inhibit interrupts, error 3E/3F.
 - d. Start motion write, error 3E/3F.
 - e. Enable interrupts.

NOTE

Interrupt has occurred. If output one character, skip outputting more.

- f. Wait for not busy, error 69.
- g. Clear data interrupt line request.
- h. Check that interrupt has occurred, error 07.
22. Do next unit again.
23. Do next density again (three times).

C. SECTION 2 - BACKSPACE TEST (Do not run on A/Q in Phase Encode.)

1. Rewind all units, errors 3E/3F and 69.
2. Set output buffer to 0000.
3. Select Binary mode.
4. Write 18 words on all units, error 01.
5. Write EOF on all units eight times, error 3E/3F.
6. Wait for not busy on all units, error 69.
7. Backspace all units nine times, error 3E/3F.
8. Wait for not busy all units, error 69.
9. Read and compare one record on all units 18 words long, errors 02, 03, and 04.
10. Write EOF on all units, error 3E/3F.
11. Set output buffer to 00FF.
12. Wait for not busy on all units, error 69.
13. Write 18-word record on all units, error 01.
14. Set output buffer to 0000.
15. Write 18-word record on all units 50 times, error 01.
16. Wait for not busy all units, error 69.
17. Backspace all units 51 times, error 3E/3F.
18. Set output buffer to 00FF.
19. Wait for not busy on all units.
20. Read and compare 18-word record on all units, errors 02, 03, and 04.
21. Do all densities.

D. SECTION 3 - SEARCH FILE MARK FORWARD/REVERSE TEST (Do not run on A/Q in Phase Encode.)

1. Wait for not busy on all units, error 69.
2. Rewind all units, error 3E/3F.
3. Wait for not busy on all units, error 69.
4. Set pattern (write buffer) = 0000.
5. Write EOF mark on all units, error 3E/3F.
6. Wait for not busy on all units, error 69.
7. Write 18-word record on all units, error 01.
8. Wait for not busy.
9. Increase pattern + 1.
10. Do write sequence again until done, 100 times.
11. Rewind all units, error 3E/3F.
12. Wait for not busy on all units, error 69.
13. Set write buffer to \$18 (file mark data).

14. Search file mark forward on all units 26 times.
15. Read and compare 18-word record of \$18 data on all units, errors 02, 03, and 04.
16. Search file mark forward on all units 21 times.
17. Wait for not busy.
18. Search file mark reverse on all units 26 times.
19. Wait for not busy on all units.
20. Search file mark forward on all units one time.
21. Read and compare 18-word record of \$14 data on all units, errors 02, 03, and 04.
22. Do all densities.

E. SECTION 4 - VARIABLE BLOCK LENGTH IN BCD, FIXED DATA (Do not run on A/Q in Phase Encode.)

1. Rewind all units.
2. Set block length to 18 words.
3. Select BCD mode.
4. Set output buffer to \$11EE left-shifted 1 (\$11EE, 23DC, 47B8, 8F70, 1EE1, etc.)
5. Write block length record to all units, error 01.
6. Make block length +4.
7. Repeat 48 times.
8. Wait for not busy on all units, error 60.
9. Backspace 48 times.
10. Set block length to 18 words.
11. Read and compare all units block length, errors 02, 03, and 04.
12. Make block length + 4.
13. Repeat read 48 times.
14. Do all densities.

F. SECTION 5 - RANDOM LENGTH, RANDOM DATA (Do not run on A/Q in Phase Encode.)

1. Rewind all units.
2. Select Binary.
3. Generate 210 words of random data.
4. Generate random block length (18 to 210 words).
5. Write random data, and block length on all units, error 01.
6. Repeat 40 records.
7. Backspace 40 times.
8. Read and compare (block length unknown) on all units, errors 02 and 04. (Read to end of record, remember record length.)

9. Repeat read 40 times.
10. Do all densities.

G. SECTION 6 - NONSTOP WRITE/READ

1. Rewind all units.
2. Wait for not busy on all units, error 69.
3. Select A/D mode and Binary.
4. Generate 16-word buffer of \$55AA, left-shifted one (55AA, AA55).
5. Select unit.
6. Wait for not busy.
7. Write 16-word record, error 01.
8. Repeat write 37 times.
9. Repeat conditions.
10. Do write 16-word record again.
11. Wait for not busy, error 69.
12. Backspace 37 times, error 3E/3F.
13. Wait for not busy, error 69.
14. Read and compare 16-word record, errors 02, 03, and 04.
15. Repeat read 37 times.
16. Repeat conditions.
 - a. Rewind.
 - b. Wait for not busy.
 - c. Do read and compare again.
17. Repeat 1 through 16 next unit.
18. Repeat 1 through 17 next density.

H. SECTION 7 - A/D MODE VARIABLE BLOCK LENGTH, FIXED DATA

1. Select A/D mode.
2. Go to Section 4.

I. SECTION 8 - A/D MODE RANDOM LENGTH, RANDOM DATA

1. Select A/D mode.
2. Go to Section 5.

J. SECTION 9 - WRITE READ COMPARE COMPATIBILITY TEST (Do not run on A/Q
and Phase Encode.)

1. Select A/D and Binary modes.
2. Generate 81-word buffer of \$FEOF, left-shifted 1 data.
3. Rewind all units.
4. Select unit.

5. Wait for not busy, error 69.
6. Write 48-word record, error 01.
7. Repeat write 128 times.
8. Wait for not busy, error 69.
9. Repeat write on next density (3).
10. Repeat write on next unit.
11. Select Character and BCD modes.
12. Select unit.
13. Wait for not busy, error 69.
14. Write 80-word record, error 01.
15. Wait for not busy.
16. Repeat write 128 times.
17. Repeat write next density.
18. Write EOF two times.
19. Repeat write next unit.
20. Go to Section 10 to verify.

K. SECTION 10 - READ AND COMPARE DATA TO THAT WRITTEN BY SECTION 9 OF THIS TEST (Do not run on A/Q in Phase Encode.)

1. Try to select unit 0. (Reject and go to next unit if this unit was not 7.)
2. Status for ready:
 - a. If ready, go to 3.
 - b. If this is not Section 10, go to 3.
 - c. If this unit was not 7, go to 3; otherwise, output message A=4B1F, Q=0055. (Scanned all possible units, serviced as found ready.)
 - d. If the A or Q register is clear, do again; otherwise, go to e.
 - e. End of section.
3. If unit is not selected via parameter, go to next unit.
4. Generate 81 words of \$FEOF data, left-shifted 1.
5. Select A/D and Binary modes.
6. Rewind unit, error 3E/3F.
7. Wait for not busy, error 69.
8. Read and compare 48-word record, errors 02, 03, and 04.
9. Wait not EOP, error 69.
10. Repeat read 127 times.
11. Do read in next density (3 maximum).
12. Select Character and BCD modes, error 3E/3F.
13. Read and compare 80-word record, errors 02, 03, and 04.
14. Repeat read 127 times.

15. Wait for not busy.
16. Do next density (3 maximum).
17. Start motion read, error 3E/3F.
18. Wait for EOP.
19. Check status for EOF error.
20. Start motion read, error 3E/3F.
21. Wait for EOP.
22. Check status for EOF error.
23. If Section 10, rewind unload and go to next unit.
24. Rewind and go to next unit.

III. MAJOR SUBROUTINE DESCRIPTIONS

A. INITIALIZATION

1. Checks bit 2. If set, clears bit 2, sets DSA flag parameter Q4, and modifies the title for typeout to 1732-2.
2. If W \neq 0, generates the correct BDC busy bit for use by BSYCON in the SMM Inform word.
3. Picks up test frequency count and initial address for title display.
4. Determines which units are on-line (ready and write enable), also 7-track, 9-track, and protect status for the Unit Select parameter.
5. Display title heading.
6. Types out a message to CLEAR PROGRAM PROTECT SWITCH, SET STOP SWITCH and stops with A=4B1F, Q=0000 for the operator to clear the PROGRAM PROTECT switch, and the protect bits in the test. If the TTY input driver (SMMPAR bit 6) is turned on, a message is typed out to SET PROGRAM PROTECT, CLEAR SELECTIVE STOP, TO RUN PROTECTED.
7. Return control to SMM.

B. START2 - PARAMETER ENTRY - (Restart Routine)

1. Makes a parameter call to SMM.
2. Breaks up the Unit Select parameters for use by the test.
3. Restores equipment address into IA+6 of the test.
4. Initializes density select to 0 or 556 BPI, depending on the state of section select parameter (bit 15).

C. TNA1 - TEST NUMBER ANALYSIS

1. Determines section to be executed.
2. Initializes program to Binary and Character mode.
3. Initializes interrupts to the Interrupt Error routine.
4. Calls the respective section selection.
5. At end of section, makes end of section call to monitor.
6. If repeat section is set, recalls same section. Otherwise, goes to next section.
7. If all sections have been executed, makes end of test call to monitor.
8. If repeat test is set, the test is restarted from the first selected section.

D. BSYCON - BUSY CONTROL

This routine releases control to SMM and, depending on the state of the A register on entry, will not accept control back until the subsystem is Not Busy.

Each time SMM passes control back to BSYCON, status 1 of the BDC or 1732-X is taken. If the device is busy, a count is incremented by 1 and control returned to SMM.

When the count = 131,068, error 69 is reported. If a BDC is present, a Terminate buffer is issued after the error is reported. In either case, control is released to SMM.

E. SUN0 - SELECT NEXT UNIT

Selects a unit available according to the Unit Select parameters. The condition of the A register on entry determines if the same unit is to be reselected or the next available unit is to be selected.

1. A/D mode is selected or cleared.
2. Density select is made.
3. Unit test cell for error information is updated.

All flags pertaining to the unit are generated. These include protected, 9-track, phase encoded, and unit number. After unit is selected, status 2 is checked for the correct status of 7/9 track and the correct density. Errors can originate from here.

F. WSEL0 - WRITE DRIVER

The purpose of this routine is to write one record of data from the output buffer to the selected tape transport, via one of three data paths.

The two data paths are 1732-2, 1732-3 A/Q or 1732-2, 1732-3 DSA. On entry, the A register contains the number of words to output.

Errors reported are 3E/3F from data outputs and 01 write parity error at the end of the record.

The driver is exited at EOP time.

G. RSEL0 - READ DRIVER

The purpose of this driver is to read one record into the input buffer from the selected tape, via one of the three data paths. On entry, the A register contains either 0 (unknown record length) or the expected record length.

If data is being transferred via DSA, end of operation interrupt is enabled and control released until the interrupt is received.

At end of operation time, status is checked for read parity errors (02), record length errors (04), and phase encode errors (0B).

H. COMPAR - DATA COMPARE

The purpose of this routine is to compare the data in the input and output buffer, and report any discrepancies between them.

The 7/9-track Unit Select parameter and the Assembly/Disassembly flag is examined to determine how much and which bits in the output buffer are significant to the data read. The record is compared to the length determined by the read driver.

Data compare errors (03) are displayed. The actual data (A3) is the word read, and expected data (Q3) is the data written from the write buffer masked to the correct configuration. A4 contains the word number in the record that failed and Q4 is the record length determined by the read driver. All data compare errors within a record are reported, unless the test Stop/Jump bit 11 is set, at which time it reports only the first error. Upon completion of the compare operation, the input buffer is zeroed out.

I. ILIRPT - ILLEGAL INTERRUPT ROUTINE

The normal interrupt request for this test is to this routine. Any time this routine is entered, an illegal interrupt has been encountered on one of the parameter selected lines. Status 1 of the 1732-x is immediately taken and saved. The routine then tries to determine which interrupt was generated and reports the corresponding error.

J. FSELO - FUNCTION 1 SELCT

The purpose of this routine is to output the contents of A function 1 and check that the controller did go busy upon a reply to any motion function. Errors reported are 3E/3F, internal/external reject of the output function, 2E/2F, internal/external reject of the status 1 input, and 05 if the controller did not go busy after replying to a motion function.

IV. HINTS TO THE USER

When running on a 1732-2,3, two data paths are always available. If testing a phase encode transport, there are two modes of operation (NRZI and Phase Encode (unit select bit 2^3)). This should be used to identify transport problems from the phase encode option problems.

A. READING ERRORS

The most important product of this test is the errors. Therefore, when an error occurs, remember the unit number, section number, and look up the error code for the explanation. Often, this will be self-explanatory. However, there are cases in which it is important to know exactly what the test was doing to detect the error condition. Refer to the failing section writeup for the logic flow of the section. The points at which critical errors can be detected and reported are within the section.

B. SECTION 9 AND 10 AIDS

Section 9 generates the test unique data tape to be read by Section 10. The intention of these two sections is to be a go-no-go form of compatibility check by crossing the total worst case conditions of the drive and controller between the Read and Write operations.

C. START-STOP TIMING VERIFICATION

Phase Encode is the most sensitive to start/stop problems. The symptom of such problems is running all sections.

1. Section 0, error 27.
2. Section 2, data error 03. A3, actual data = x and Q3, expected data = 15 or 27 being that the tape is written in groups of data, file mark, etc., and the record data is the group number. The actual data in this case, if solid, will indicate the actual position on the tape.
3. Section 9/10, motion rejects and data errors.

D. VISUAL OPERATIONS

When data is being written to the tape subsystem, the OVERFLOW light is off.
When data is being read from the tape subsystem, the OVERFLOW light is on.



COMMENT SHEET

MANUAL TITLE CDC 1700 System Maintenance Monitor Volume 1
Reference Manual

PUBLICATION NO. 60182000 REVISION R

FROM: NAME: _____
BUSINESS ADDRESS: _____

CUT ALONG LINE

PRINTED IN U.S.A.

AA3419 REV. 7/75

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.
FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

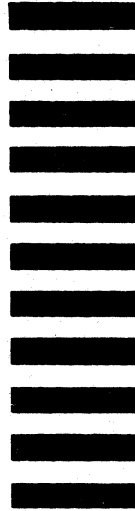
FOLD

FOLD

FIRST CLASS
PERMIT NO. 8241
MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY
CONTROL DATA CORPORATION
Publications and Graphics Division
ARH219
4201 North Lexington Avenue
Saint Paul, Minnesota 55112



CUT ALONG LINE

FOLD

FOLD