SPERRY UNIVAC

Maintain III Test Programs

User Manual

W

Mini-Computer Operations

2722 Michelson Drive
P.O. Box C-19504
Irvine, California 92713

UP-8672 Rev. 1

SPERRY UNIVAC®

MAINTAIN III TEST PROGRAMS
USER MANUAL

UP-8672 Rev. 1
98A 9952 075
Update A
OCTOBER 1980

The statements in this publication are not intended to create any warranty, express or implied. Equipment specifications and performance characteristics stated herein may be changed at any time without notice. Address comments regarding this document to Sperry Univac, Mini-Computer Operations, Publications Department, 2722 Michelson Drive, P.O. Box C-19504, Irvine, California, 92713.

Sperry Univac is a division of Sperry Corporation

Printed in U.S.A.

PAGE STATUS SUMMARY

ISSUE: Update A - UP-8672 Rev. 1

Part/Section	Page Number	Update Level
Cover		
Title Page		A
PSS	1	A
CR	1	A
Preface	1	Orig.
Contents	1 thru 10	Orig.
1	1, 2	Orig.
2	1 thru 17	Orig.
3	1 thru 12	Orig.
4	1 thru 6	Orig.
5	1 thru 6	Orig.
6	1 thru 4	Orig.
7	1 thru 4	Orig.
8	1 thru 7	Orig.
9	1 thru 8	Orig.
10	1 thru 5	Orig.
11	1 thru 13	Orig.
12	1 thru 6	Orig.
13	1 thru 15	Orig.
14	1, 2	Orig.
	3	A
	4 thru 7	Orig.
	8	A
	9 thru 16	Orig.
15	1 thru 6	Orig.
16	1 thru 5	Orig.
17	1, 2	Orig.
	3	A
	4 thru 9	Orig.
	10, 11	A
	12	Orig.
18	1 thru 6	Orig.
19	1 thru 4	Orig.
Appendix A	1 thru 7	Orig.

**New pages*

All the technical changes are denoted by an arrow (→) in the margin. A downward pointing arrow (↓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (↑) is found. A horizontal arrow (→) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.

CHANGE RECORD

Change Designation	Issue Date	Change Description
	June 1977	Inserted new sections. Renumbered sections. Deleted reference to Varian.
	Jan 1978	Deleted, corrected, and added information to Sections 2, 7, 12, and 14.
Update A	June 1979	Adds magnetic tape, disk, and V77-800 test information.
Rev. 1	Sept 1979	Typeset Sections 15 thru 18 and added Section 19.
Update A	Oct 1980	Added subtests to Section 17.

Change Procedure:

When changes are made to this manual, updated pages are issued. These updated pages are either added to this manual or used to replace obsolete pages. The specific pages affected by each change are identified on the PAGE STATUS SUMMARY page.

Printed in U.S.A.

PREFACE

This manual describes the MAINTAIN III test-program system for verifying the correct operation and detecting and isolating malfunctions in Sperry Univac computer systems.

The reader should be familar with the instruction set of the system for which he uses these programs and some assembly-language programming. The person who runs these tests should also know the operating procedures for the control panel and peripheral devices on his system.

The organization of this manual is based on the organization of the test system. The first chapter presents an overview of the entire system. The following chapters present the components of the system. In a chapter for a specific component the reader finds an overview in more detail and a definition of the minimal hardware necessary for using the component, a description of its design and structure, followed by the information needed to use the test in the order needed: first the preliminary procedures such as loading and setting sense switches, then the execution procedures, followed by an explanation of any error indication that may occur during execution or cause termination, and finally examples of the program input and output.

As new systems are developed, the existing test system is expanded to include the new systems. In those cases where tests are applicable to more than one system, reference is made to those systems. Unless specifically called out, the terms 620, V70, and V77 should be applied as indicated in the following listing.

SYSTEM NUMBER	MODEL NUMBERS
620	620/f, 620/i, 620/I
V70	V72, V73, V75 and V76 plus V77 series
V77	V77-200, V77-400, V77-600, V77-800

RELATED DOCUMENTATION

The V70 Series Architecture Manual provides a complete description of the instruction set, word formats, and addresing modes. Each computer is documented by a system reference manual and manuals for each of the major components: processor, memory, memory map (or megamap), writable control store, and cache. Peripheral controllers are also documented separately. These manuals include information concerning installation, operation, theory of operation, and maintenance.

GENERAL OPERATING DESCRIPTIONS

In this manual references to the instruction register designate the I register of the 620/f and the U register on other 620-series computers. Similarly, references to START on the 620/f are the same as RUN on other systems. RESET is SYSTEM RESET on 620 computers except the 620/f which is RESET. The applicable system handbook gives detailed descriptions of control-panel switches and indicators and general operating procedures.

CONTENTS

SECTION 6
POWER-FAILURE/RESTART TEST

SECTION 7
PRIORITY-INTERRUPT-MODULE TEST

SECTION 8
REAL-TIME CLOCK TEST PROGRAM

SECTION 9
620/f AND V70 MEMORY-PROTECTION TEST PROGRAM

SECTION 10
BUFFERED-I/O-CONTROLLER TEST PROGRAM

SECTION 11
WRITABLE CONTROL STORE TEST PROGRAM

SECTION 12
V77-400 WRITABLE CONTROL STORE TEST PROGRAM

SECTION 13
CACHE MEMORY TEST PROGRAM

SECTION 14
MEGAMAP PROGRAM

SECTION 15
V77-800 MICRODIAGNOSTIC TEST PROGRAM

SECTION 16
V77-800 CACHE MEMORY TEST PROGRAM

SECTION 17
V77-800 MAP/MEMORY TEST PROGRAM

SECTION 18
V77-800 WRITABLE CONTROL STORE TEST PROGRAM

SECTION 19
V77-800 FLOATING-POINT PROCESSOR TEST PROGRAM

APPENDIX A
CONTROL PANEL OPERATION

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF TABLES *(continued)*

SECTION 1
SYSTEM OVERVIEW

The V70/620 MAINTAIN III Test Program System is a system approach to testing and maintaining Sperry Univac 70 and 620-series computers, internal options, and peripherals. MAINTAIN III provides an effective and uniform interface between the computer and the user.

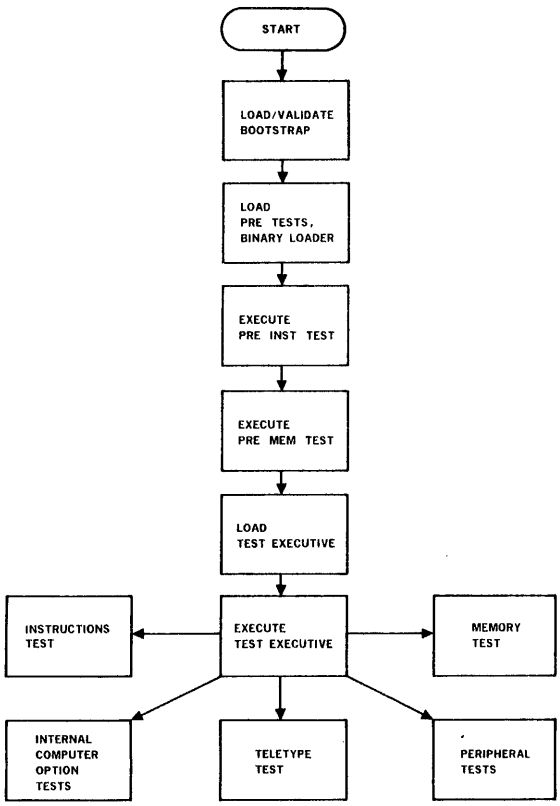
The test programs in this manual cover only the computers and internal options. Refer to the Bibliography for a list of Software Performance Specifications (SPS) covering peripheral test programs that operate in the MAINTAIN III Test Program System.

The test programs are to be used in conjunction with the maintenance manuals for the system, which include theory of operation, installation, and maintenance information.

The MAINTAIN III system programs are designed to verify correct system operation, including internal instructions, memory, internal computer options, and peripherals and their controllers. Malfunctions can be isolated to a specific area of the system and corrected.

1.1 STRUCTURE

The MAINTAIN III system consists of the following elements (figure 1-1):



NOTE: The Test Executive operates with only one test program in memory at a time.

VTII-3185

Figure 1-1. MAINTAIN III System Block Diagram

a. Test executive program which includes preliminary instruction and memory tests, binary loader, and test executive

b. Instructions test program

c. Memory test program

d. Teletype (TTY) test program

e. Internal computer option test programs

f. Peripheral test programs

The test executive program:

a. Loads test program

b. Accepts control directives and parameters from the user

c. Executes test programs

d. Contains a utility package, consisting of aids for debugging, program maintenance, and hardware troubleshooting

e. Includes standard test program subroutines, i.e., TTY input/ output, time delay, memory size determination, SENSE switch option, etc.

The preliminary instructions test portion of the executive test program validates basic CPU operation, the preliminary memory test checks basic functions of the first 8K memory module, and the binary loader reads binary data and stores it in memory.

The memory test program verifies correct operation of memory. It is applicable to 16-bit systems with from 8,192 to 32,768-word memories.

The instructions test program tests and verifies execution of internal, I/O, and optional instructions.

The TTY test program verifies correct operation of the Sperry Univac-modified 33/35 ASR TTY unit.

The internal computer option test programs individually test each option to ensure correct operation.

The peripheral test programs verify correct operation of associated system peripherals (i.e., line printer, disc, paper tape system, etc.) and their controllers.

1.2 MAINTENANCE CONCEPTS

MAINTAIN III minimizes maintenance time for the V70/620-series computers. The programs can be executed when the computer is off-line and not transferring data or performing control functions.

MAINTAIN III test programs are normally on punched paper tape; other media, such as object card decks or magnetic tape are available. The programs exercise the computer, internal options, and peripherals and their controllers with sequences of instructions. If an instruction is improperly executed, the sequence is halted and an error message is output to indicate the failing instruction or operation. The user can then repeat, continue, or halt the program until the fault is isolated and corrected.

To correct hardware malfunctions:

a. Isolate the fault to a functional area, such as memory, control, arithmetic/logic, operations register, input/output, or peripheral device or its controller. Eliminate the functional areas that are operating properly.

b. Execute, repeat, or modify the applicable test program for the area of the suspected fault.

c. Correct the fault by replacing the faulty component or circuit card and restore the system to normal operation.

d. Verify system operation by rerunning the test program.

The maintenance manuals appropriate to the user's system describe the theory of operation of all major functional areas of the computer, internal options, and peripheral controllers. Also given are system checkout procedures using the control panel and specified electronic test equipment.

Specific operating procedures for MAINTAIN III basic computer test programs are given in the following chapters, which also include descriptions of error conditions and error messages.

SECTION 2
TEST EXECUTIVE PROGRAM

The **Test Executive Program** is the controlling factor in the MAINTAIN III test program system. In addition to loading, executing, and monitoring the other MAINTAIN III test programs, the test executive program:

a. Provides utility aids for debugging, program maintenance, and hardware troubleshooting

b. Includes standard subroutines for use by associated test programs, i.e., TTY I/O, time delay/time out, memory size determination, power failure/restart protection, SENSE switch options, etc.

The test executive program is designed for a minimum hardware configuration of a V70/620-series computer with 8K of memory (maximum, 32K) and a 33/35 ASR TTY. All system sizes can be tested, but the test programs operate in the first 8K memory module only.

The test executive object program is normally supplied as punched paper-tape loaded from either the TTY or high-speed paper-tape reader. Magnetic-tape or a punched card object deck is also available.

For current MAINTAIN III test programs object file directory see Usage Description bulletin 92W0106-013.

2.1 COMPONENTS OF THE SYSTEM

The test executive program consists of:

a. Preliminary instructions test

b. Preliminary memory test

c. Binary loader

d. Test executive

The preliminary instructions test validates central processing unit (CPU) operation by testing the machine instructions listed in table 2-1. Successful execution of this test indicates that MAINTAIN III test programs can be correctly loaded.

The preliminary memory test verifies correct operation of the first 8K of memory. Memory addresses 000044 through 017777 are tested in two passes. The first pass checks each address with a pattern of 052525; the second pass, 0125252. The original contents of memory are saved and restored by the program.

Table 2-1. Preliminary Instructions Test Summary

Mnemonic	Description
ADD	Add memory to A register
ADDI	Add immediate
ANAI	AND immediate
DAR	Decrement A register
DBR	Decrement B register
DECR 02	Set B register to − 1
DXR	Decrement X register
ERA	Exclusive-OR memory and A register
ERAI	Exclusive-OR immediate
IAR	Increment A register
IBR	Increment B register
INCR 03	Set A and B registers to + 1
IXR	Increment X register
JAN	Jump if A register negative
JAP	Jump if A register positive
JAZ	Jump if A register zero
JBZ	Jump if B register zero
JIF 011	Jump if A register = 0 and OVFL is set
JMP	Jump (unconditional)
JMPM	Jump and mark (unconditional)
JMP*	Jump indirect
JOF	Jump if overflow indicator set
JXZ	Jump if X register zero
LDA	Load A register
LDAI	Load A register immediate
LDB	Load B register
LDBI	Load B register immediate
LDX	Load X register
LDXI	Load X register immediate
LLRL	Load logical rotation left
LLSR	Long logical rotation right
LRLA	Logical rotation left A register
LSRA	Logical shift right A register
MERG 032	Transfer ORed A and B registers to B register
NOP	No operation
ORAI	Inclusive-OR immediate
ROF	Reset overflow indicator
STA	Store A register
STAI	Store A register immediate
STB	Store B register
STX	Store X register
SUB	Subtract memory from A register
TBA	Transfer B register to A register
TBX	Transfer B register to X register
TXA	Transfer X register to A register
TZA	Transfer zero to A register
TZB	Transfer zero to B register
TZX	Transfer zero to X register
XAZ	Execute if A register zero
XBZ	Execute if B register zero
XIF 022	Execute if B register = 0 and A register = ≥0

The binary loader loads formatted object data into computer memory, computes the check-sum, and transfers program control as directed.

The test executive is integral to the MAINTAIN III test program system. In addition to providing test control and user interface, it contains standard subroutines commonly required by the associated test programs, i.e., TTY I/O routines, SENSE switch routines, etc.

The test executive program utility package consists of aids for debugging, program maintenance, and hardware troubleshooting:

a. CPU registers and memory can be displayed or altered.

b. The user can specify memory data pattern searches.

c. Areas of memory can be set to specified data patterns.

d. Object code can be punched or written.

e. During execution, test programs can be trapped.

The utility routines are summarized in table 2-2; standard executive data items, in table 2-3; and standard I/O routines, in table 2-4. Refer to the listing supplied with the program for the entry addresses of these routines.

Table 2-2. Test Executive Utility Routines

Mnemonic	Description
EARG	Print/change the contents of the pseudo-A register
EBPN	Punch a tape on the Teletype (binary)
EBRG	Print/change the contents of the pseudo-B register
ECNG	Print/change the contents of memory
EDUM	Dump (print) the contents of memory on the Teletype printer
EGOT	Transfer to the specified address
EPUN	Punch a tape on the Teletype (object)
ESRC	Search memory
ETRP	Trap to the specified address
EXRG	Print/change the contents of the pseudo-X register
INIT	Initialize memory

Table 2-3. Standard Test Data Items

Mnemonic	Description
$CON	Control panel/Teletype mode flag
$DCT	Digit counter for I/O routine INPG
$FLG	Loop on error flag
$LWE	Lowest address used by test executive
$MEM	Highest available memory address
MSG3	Memory size message
$TTY	Teletype device address

Table 2-4. Test Executive Standard Data Routines

Mnemonic	Description
ESZC	Determine memory size
INPA	Input one character
INPB	Input and print one character
INPC	Input one edited character
INPD	Input one alphabetic character
INPE	Input two alphabetic characters
INPF	Input terminating control character
INPG	Input octal number
INPH	SENSE Teletype buffer ready
INPI	Initialize Teletype (clear input buffer)
OUTA	Output one character
OUTB	Output two characters
OUTC	Output carriage return and line feed
OUTD	Output message
OUTE	Output octal word
OUTF	Output octal address
OUTG	Output typing error message
OUTH	Output control character
SSWT	Standard SENSE switch routine
TDLY	Time delay
TOUT	Time out

2.2 OPERATING PROCEDURES

2.2.1 PRELIMINARY PROCEDURES

A variety of program loading devices are available for use with Sperry Univac computers. When using the following procedures, the operator must determine what equipment is installed and then use the procedures described in the appropriate manual.

After bringing up computer system power:

a. Enter step mode.

b. Reset SENSE switches 1, 2, and 3.

c. Initialize the computer control circuits by activating RESET. If using a virtual console, enter an A to reset the system.

d. Load the object program. Use the applicable following procedure:

   If using the high-speed paper tape (HSPT) reader, set the LOAD/RUN switch to LOAD. Position the tape in the reader with the first nonblank binary frame at the read station. Set the LOAD/RUN switch to RUN.

   When using the high-speed paper tape reader-punch, open the cover of the read head and insert tape. Position the tape in the reader with the first nonblank binary frame at the read station. Close the cover of the read head.

NOTE

The Sperry Univac part number is punched in the leader portion of the object tape, e.g., 92U0106-01-3x in the test executive tape, where x indicates the revision level. Position tape in the reader past this area.

If the computer is equipped with TTY or HSPT automatic bootstrap loader (ABL), initiate RESET. Place the computer in run mode and press BOOT.

To use a card reader, place the test executive object card deck in the card reader hopper. Place the card reader in the ready status and refer to step e.

If using magnetic tape, place the MAINTAIN III test object tape on the appropriate tape drive unit at proper density. Then:

(1) Manually load the magnetic tape bootstrap routine listed in Table 2-5 or 2-6. Refer to the appropriate equipment operations manual for loading procedures.

(2) Place the MAINTAIN III magnetic tape in the tape unit and position the tape to the load point.

(3) Enter 000 212 into the P register, 07000 into the X register (register R2 on the virtual console), and zero in the A and B registers (R0 and R1 on the virtual console).

(4) Set STEP/RUN to RUN on control panel or press key R on the virtual console to enter the test executive into memory.

If the TTY is used, initialize the TTY by setting control to LOCAL (off-line). Type CONTROL, D, T, and Q. Return control to LINE (on-line). Position test program tape in the reader with the first nonblank binary frame at the reading station. Set the reader control switch to STOP.

e. If a magnetic tape unit is not used, enter the test executive program into memory by using one of the following procedures.

To enter the program via the computer control panel, momentarily place the load switch to the LOAD position. The high-speed tape reader (reader-punch) is selected when the load switch is activated.

If the virtual console is used:

(1) Select the desired program loader routine from the following listing. Reference tables are listed in the event that the routine must be loaded manually.

PROGRAM LOADER ROUTINE	OCTAL CODE	REFERENCE TABLE
Teletypewriter paper tape reader	000 000	2-7
High-speed paper tape reader	000 001	2-8
F3094-0x/F3096-0x disk memory	000 002	2-9, 2-10
2842-0x and 2826-0x disk memory	000 003	

(2) Enter the program loader code into register R0 by pressing key 0 on the virtual console and typing the desired code. Press key 0 again to verify the contents of register R0.

(3) Press key B on the virtual console to load the program loader and object program. When the tape stops, press R on the virtual console or set STEP/RUN switch to RUN on the control console.

When using a card reader, the program loader routine must be manually loaded. Use the listing in Table 2-11 and the procedures outlined for the magnetic tape unit.

If the TTY is used, set the reader control switch to START/RUN.

After the preliminary tests and binary loader are read into memory, the bootstrap routine jumps to address 007000. The paper-tape reader is turned off, and the preliminary instructions test is automatically executed, starting at address 007002.

Following successful execution of the instructions test, the program automatically executes the preliminary memory

test. The program then jumps to the binary loader, which loads the test executive. Setting SENSE switch 3 during their execution causes the program to loop on the combined preliminary instructions and memory tests.

Preliminary test error conditions are described in section 2.2.3.

Table 2-5. Magnetic-Tape Bootstrap Routine

Address	Instructions Code		Symbolic Coding		
		1 *	MAGNETIC TAPE BOOTSTRAP		
		2 *			
*	0000ZZ	3 MT	SET	ZZ	
*	00000Y	4 TU	SET	Y	
000200		5	ORG	0200	
*000200	1012ZZ	6 MTS	SEN	0200+MT,07002	SENSE IF DONE
000201	007002				
*000202	1011ZZ	7	SEN	0100+MT,MTST	SENSE IF DATA IN
000203	000206				
000204	001000	8	JMP	MTS	SENSE DATA IN
000205	000200				
*000206	1025ZZ	9 MTST	CIA	MT	GET WORD
000207	055000	10	STA	0,1	STORE WORD
000210	001000	11	JMP	MTSA	
000211	000214				
		12 *			
		13 *	START HERE WITH X=07000		
		14 *			
*000212	104YZZ	15 ENTR	EXC2	(TU*64)+MT	SELECT UNIT
000213	1000ZZ	16	EXC	MT	READ ONE RECORD BINARY
000214	005144	17 MTSA	IXR		STEP INDEX
000215	001000	18	JMP	MTS	LOOP
000216	000200				

* = where

Y = Drive number 1, 2, 3, or 4
Z = Device address, (normally 010)

Table 2-6. Magnetic Tape Bootstrap Routine
(F3093-0x, F3062-00, 0870-99, and 0870-35 Tape Units)

Address	Instructions Code		Symbolic Coding		
000200	1031WW	BOOT	OAR	BIC	SET INITIAL ADDRESS
000201	005301		DECR	1	A=0177777 FOR LAST ADDRESS
000202	1031XX		OAR	BIC+1	SET FINAL ADDRESS
000203	1000WW		EXC	BIC	ACTIVATE BIC
000204	1000ZZ		EXC	MT	READ ONE RECORD FROM MAG TAPE
000205	1012ZZ	WAIT	SEN	0200+MT,07002	TO START IF MAG TAPE READY
000206	007002				
000207	005000		NDP		
000210	001000		JMP	WAIT	LOOP UNTIL READY
000211	000205				
		*			
		*		START AT 0212	
		*		WITH X=07000	
000212	1000XX	START	EXC	BIC+1	INITIALIZE BIC
000213	104YZZ		EXC2	(TU*0100)+MT	SELECT MAG TAPE UNIT
000214	005141		INCR	041	SET A+07001 (START ADDRESS)
000215	001000		JMP	BOOT	START BIC
000216	000200				

where:

WW Even BIC device address
XX Odd BIC device adress
Y Drive number 1, 2, 3, or 4
ZZ MT device address

Table 2-7. Teletype Paper-Tape Bootstrap Routine

Address	Instruction Code		Symbolic Coding		
		1 *	TELETYPE PAPER TAPE BOOTSTRAP		
		2 *			
*	0000ZZ	3 TY	SET	ZZ	
000200		4	ORG	0200	
*000200	1026ZZ	5 READ	CIB	TY	8 BITS TO B
000201	004011	6	ASLB	9	SAVE LS6
000202	004041	7	LRLB	1	
000203	004446	8	LLRL	6	MERGE INTO A
000204	001020	9	JBZ	SEL	MORE IF B ZERO
000205	000214				
000206	055000	10	STA	0,1	
000207	001010	11	JAZ	07000	EXIT IF ZERO
000210	007000				
000211	005144	12	IXR		STEP INDEX
		13 *			
		14 *	START HERE WITH X=07000		
		15 *			
000212	005101	16 ENTR	INCR	1	SET A BIT 0
*000213	1026ZZ	17 SEL	CIB	TY	CLEAR TTY BUFFER
*000214	1012ZZ	18	SEN	0200+TY,READ	SENSE READ READY
000215	000200				
000216	001000	19	JMP	*-2	LOOP
000217	000214				

* = where Z = Device address, (normally 01)

Table 2-8. High-Speed Paper-Tape Bootstrap Routine

Address	Instruction Code		Symbolic Coding		
		1 *	HIGH-SPEED PAPER TAPE BOOTSTRAP		
		2 *			
*	0000ZZ	3 PT	SET	ZZ	
000200		4	ORG	0200	
*000200	1026ZZ	5 READ	CIB	PT	8 BITS TO B
000201	004011	6	ASLB	9	SAVE LS6
000202	004041	7	LRLB	1	
000203	004446	8	LLRL	6	MERGE INTO A
000204	001020	9	JBZ	SEL	MORE IF ZERO
000205	000214				
000206	055000	10	STA	0,1	STORE WORD
000207	001010	11	JAZ	07000	EXIT IF ZERO
000210	007000				
000211	005144	12	IXR		STEP INDEX
		13 *			
		14 *	START HERE WITH X=07000		
		15 *			
000212	005101	16 ENTR	INCR	1	SET A BIT 0
*000213	1005ZZ	17 SEL	EXC	0500+PT	READ A FRAME
*000214	1015ZZ	18	SEN	0500+PT,READ	SENSE READ READY
000215	000200				
000216	001000	19	JMP	*-2	LOOP
000217	000214				

* = where Z = Device address, (normally 037)

Table 2-9. Disk Bootstrap Routine (F3094-0x and F3096-0x RMD)

Address	Instructions Code		Symbolic Coding		
001130	1004ZZ	START	EXC	0400+DISK	INITIALIZE DISK
001131	011167		LDA	DBUNT	POSITION TO CYLINDER
001132	1002ZZ		EXC	0200+DISK	
001133	1031ZZ		OAR	DISK	
001134	101XZZ	SLP	SEN	(UNIT*01000)+DISK	GO SEEK COMPLETE?
001135	001140				
001136	001000		JMP	SLP	NO, LOOP
001137	001134				
001140	1025ZZ	GO	CIA	DISK	CHECK STATUS
001141	151166		ANA	DBSTS	
001142	001010		JAZ	GOOD	I.O ERROR
001143	001145				
001144	000000		HLT		
001145	011167	GOOD	LDA	DBUNT	POSITION TO SECTOR
001146	1003ZZ		EXC	0300+DISK	
001147	1031ZZ		OAR	DISK	
001150	1000WW		EXC	BIC+1	INITIALIZE BIC
001151	011170		LDA	DBSRT	SET START
001152	1031VV		OAR	BIC	
001153	011171		LDA	DBEND	SET END
001154	1031WW		OAR	BIC+1	
001155	1000VV		EXC	BIC	ACTIVATE BIC
001156	1000ZZ		EXC	DISK	READ FROM DISK
001157	1014ZZ	BLP	SEN	0400+DISK,BLP	BUSY?
001160	001157				
001161	1025ZZ		CIA	DISK	CHECK STATUS
001162	151166		ANA	DBSTS	
001163	001010		JAZ	0600	START PROGRAM
001164	000600				
001165	000000		MLT		
001166	005760	DBSTS	DATA	05760	STATUS BITS
001167	0Y0000	DBUN	DATA	(UNIT*020000)	
001170	000000	DBSRT	DATA	0	START LOAD
001171	001130	DBEND	DATA	01130	END LOAD

Where:

- VV Even BIC device address
- WW Odd BIC device address
- X Unit number 0, 1, 2, or 3
- Y Unit number times 2
- ZZ Disk device address

Table 2-10. Disk Bootstrap Routine
(70-7606, F3094-0x, F3310-xx, and F3096-0x RMD's)

Address	Instructions Code		Symbolic Coding		
001130	1004ZZ	START	EXC	0400+DISC	INITIALIZE DISK
001131	011167		LDA	DBUNT	POSITION TO CYLINDER
001132	1031ZZ		OAR	DISK	
001133	1002ZZ		EXC	0200+DISK	
001134	101XZZ	SLP	SEN	((UNIT/2)*100)+DISK,	GO SEEK COMPLETE?
001135	001140				
001136	001000		JMP	SLP	NO, LOOP
001137	001134				
001140	1025ZZ	GO	CIA	DISK	CHECK STATUS
001141	151166		ANA	DBSTS	
001142	001010		JAZ	GOOD	NO ERROR
001143	001145				
001144	000000		HLT		
001145	011167	GOOD	LDA	DBUNT	POSITION TO SECTOR
001146	1031ZZ		OAR	DISK	
001147	1003ZZ		EXC	0300+DISK	
001150	1000WW	EXC	BIC+1	INITIALIZE BIC	
001151	011170		LDA	DBSRT	SET START
001152	1031VV		OAR	BIC	
001153	011171		LDA	DBEND	SET END
001154	1031WW		OAR	BIC+1	
001155	1000VV		EXC	BIC	ACTIVATE BIC
001156	1000ZZ		EXC	DISK	READ FROM DISK
001157	1014ZZ	BLP	SEN	0400+DISK,BLP	BUSY?
001160	001157				
001161	1025ZZ		CIA	DISK	CHECK STATUS
001162	151166		ANA	DBSTS	
001163	001010		JAZ	0600	START PROGRAM
001164	000600				
001165	000000		HLT		
001166	173760	DBSTS	DATA	0173760	STATUS BITS
001167	0Y0000	DBUNT	DATA	UNIT*020000	SETUP WORD
001170	000000	DBSRT	DATA	0	START LOAD
001171	001130	DBEND	DATA	01130	END LOAD

Where:

- VV Even BIC device address
- WW Odd BIC device address
- X Integer of Unit/2
- Y Unit number times 2
- ZZ Disk device number

Table 2-11. Card Bootstrap Routine

Address	Instruction Code		Symbolic Coding		
		1 *	CARD	BOOTSTRAP	
		2 *		ZZ	
*	0000ZZ	3 CR	SET		
000200		4	ORG	0200	
*000200	1025ZZ	5 BOOR	CIA	CR	INPUT ODD COLUMN
000201	004250	6	LRLA	8	MOVE TO HIGH ORDER
*000202	1011ZZ	7	SEN	0100+CR,BOOS	SENSE CHARACTER READY
000203	000221				
000204	001000	8	JMP	*-2	LOOP
000205	000202				
000206	001010	9 BOOT	JAZ	07000	END OF PRELIM
000207	007000				
000210	001000	10	JMP	ENTR	
000211	000212				
		11 *			
		12 *		START HERE WITH X=07000	
		13 *			
*000212	1002ZZ	14 ENTR	EXC	0200+CR	READ A CARD
*000213	1011ZZ	15 BOOU	SEN	0100+CR,BOOR	SENSE CHARACTER READY
000214	000200				
*000215	1016ZZ	16	SEN	0600+CR,BOOT	SENSE END OF CARD
000216	000206				
000217	001000	17	JMP	*-4	LOOP
000220	000213				
*000221	1021ZZ	18 BOOS	INA	CR	MERGE EVEN COLUMN INTO A
000222	055000	19	STA	0,1	STORE WORD
000223	005144	20	IXR		STEP INDEX
000224	001000	21	JMP	BOOU	MORE ON CARD
000225	000213				

* = where Z = Device address, (normally 030)

2.2.2 Operating the Test Executive

This program can be executed using the systems Teletype

For Teletype operation when the test executive program is loaded and halts with 000000 in the instruction register, press START or RUN to begin execution. This procedure assumes that the TTY device address is 01; if it is not, load the device address in the A register and press START or RUN.

To start the test executive program manually:

a. Clear the instruction register to zero.

b. Load 014000 in the P register.

c. Press RESET, and, in run mode, press START or RUN.

d. Load the desired device address (if the TTY device address is other than 01) in the A register, and press START or RUN.

The program begins execution by outputting the message:

```
THIS IS THE V70/620 TEST EXECUTIVE

MEMORY SIZE IS nK
```

For a V75/V77 system the message is:

```
THIS IS THE V75 TEST EXECUTIVE

MEMORY SIZE IS nk
```

where n indicates memory size (for example, 8 or 12, or multiples of 4). At this time, cache memory (if included in the system) is disabled. The test executive program then waits for a control statement input (table 2-12).

The Test Executive can be restarted at any time by initializing the computer and entering RUN from location 0 or by pressing the console interrupt (INT) switch.

Table 2-12. Test Executive Utility Routine Commands

Control	Description
A	Print/change the contents of the pseudo-A register.*
B	Print/change the contents of the pseudo-B register*
Cx.	Print/change the contents of memory address x.
Dx.	Dump (list) memory on the Teletype printer beginning at memory address x.
Gx.	Load the contents of the pseudo-registers into the respective A, B, and X registers, and transfer to memory address x.
Ix,y,z.	Initialize memory addresses x through y with the value of z.
L.	Load a test program (object) and transfer control to the loaded program.
Px,y,z.	Generate in object format on associated peripherals. x is the address of the first word; y is the address of the last word; and z is the execution address. For noncontiguous areas of memory, set z at minus one except for the final area to be copied.
Rn	Print/Change the contents of the psuedo-n register*(n is any number 0 through 7).
Sx,y,z,m.	Search memory addresses x through y for the z value. m represents a search mask for comparison.
Ty,x	Trap to memory address y, starting at address x.
X	Print/change the contents of the pseudo-X register.*
\	Terminate the control statement and return to the beginning of the test executive supervisor routine. Must be typed prior to inputting the period of the control statement.
←	Delete the last octal digit and substitute the digit following the backarrow.
Carriage Return	Output a carriage return on the Teletype printer.
Line Feed	Output a line feed on the Teletype printer.
. period	Execute the control statement.
, comma	Print/change sequential memory addresses.

*The pseudoregisters are memory cells used for storing and saving the contents of the respective operations registers.

2.2.2.1 Magnetic Tape Commands

If the magnetic-tape version is being used, the following additional commands may be used:

Edc.	Write EOF on drive d, controller c.
Fn,dc.	Position to file n on drive d, controller c.
Ldc(,bic).	Load and execute program on drive d, controller c (using BIC bic).
Px,y,z,dc (,bic).	(see P control)

NOTE

d =	0 for master drive
d =	1 for first slave, etc.
c =	0 for magnetic-tape unit device address 010, etc.
c =	1 for magnetic-tape unit device address 011, etc.

A feature that appears in the magnetic tape version only is a directory of available programs. The directory can be called up after the test executive identification message is displayed. Do not position the tape prior to loading the directory. Type:

Ldc(,bic).

Position the tape.

2.2.2.2 Disk Commands

If the disk revision is being use, the following additional commands may be used:

Fn. Position to file N.

Ln, Load to file N.
or If the terminator is a period, the program is
Ln. loaded and the start address is displayed, control is then returned to MAINTAIN III.

If the terminator is a comma, then the program is loaded and executed.

2.2.2.3 Writing Disk Files

When writing a file from memory to disk:

a. Ensure the disk drive is write-enabled.

b. Select an output file by using F command.

c. Write out the object modules by using the P command.

d. Close the file after all the object modules are written. Use tne E command to close the file.

The current output file is closed when either an L or a second F command is issued. If the selected file is a system file, the following message is displayed:

SYSTEM FILE OK?

Enter a Y if the request was intentional or enter an N if unintentional.

All MAINTAIN III system files are two cylinders long and all non-system (scratch) files are four cylinders long. Capacity is approximately 4K words per cylinder. Programs over 16K words will use at least two scratch files.

2.2.2.4 Examples

In the following examples, operator inputs are represented in bold type. Other entries are program responses output to the TTY printer.

Display the contents of a pseudoregister:

```
A    142340.
B    001000.
X    006003.
```

Display the contents of a pseudo register on a V75 system:

```
R0   143240.
R1   001000.
R2   013421.
R3   000000.
R4   000000.
R5   000000.
R6   000000.
R7   000000.
```

Display and change the contents of a pseudoregister and return to the test executive:

```
A    010454      10406.
B    006016      10406.
X    007413      10406.
```

Display and change the contents of a pseudo register on a V75 system:

```
R0   010454      10406.
R1   006016      10406.
R2   007413      10406.
R3   006234      10406.
R4   013457      10406.
R5   013341      10406.
R6   000000      10406.
R7   000000      10406.
```

Display the contents of memory address 002050 and return to the test executive:

```
C02050. = 102401.
```

Display and change the contents of memory address 002050, then display the next two addresses:

```
C02050. = 102401 103402,
( 002051 ) = 000067,
( 002052 ) = 177777.
```

Dump (display) memory starting at address 006000:

```
D6000.
( 006000 )    010454    002000 . . .
( 006010 )    005145    004543 . . .
( 006020 )    005041    001000 . . .
( 006030 )    006217    001000 . . .
```

Eight columns of data actually follow the reference address in the first column. Space limitaions prohibit an actual representation herein.

Terminate the dump by typing RUBOUT or set SENSE switch 3. The program then completes the current print line before terminating.

Initialize memory addresses 000200 through 000210 to 177777 and return to the test executive:

```
I200,210,177777.
```

Search memory addresses 000200 through 000240 for the contents of 106213; display addresses that compare and return to the text executive:

```
S200,240,106213,177777.
( 000220 )=106213
( 000235 )=106213
```

Trap to memory address 000204 starting at address 000100. Display the trap address, contents of the overflow indicator, and contents of the A, B, and X register.

```
T204,100.
( 000204 )  142340   002000   010405    1
```

Load and execute a test program:

For a V75 system the trap command maintains the same format; however, eight registers (R0 through R7) will be displayed along with the overflow indicator.

```
L.
(TEST IDENTIFIER)
```

Transfer to and execute a test program located at address 000500:

```
G500.
(TEST IDENTIFIER)
```

Punch or write in object format beginning at address 000001 through 000006, after initializing the addresses to the desired values:

```
I0,7,0.
I1,6,1.
I2,5,2.
I3,4,3.
P1,6,7.
```

> In the example immediately above, the initialize memory control statement has been used to establish a specified pattern in memory for validation of the format of the resultant operation.

Terminate an erroneous control statement:

```
P1,6\
```

Cancel an octal digit and replace with the following digit:

```
I0,6← 7.
```

Detailed descriptions of loading and execution procedures of other MAINTAIN III test programs under test executive control are contained in the following chapters.

Briefly, to load a test program:

a. Select the desired test program.

b. Type L. on the TTY keyboard.

c. The program is loaded and a test identifier message output on the TTY printer.

Return to the beginning of a test program is normally controlled by a SENSE switch option, or after the execution of a specified number of cycles.

To return to the test executive from a test program, follow the restarting procedure described in section 2.2.1. Pressing the INT switch on the 620/f or V70 series computer also returns control to the test executive; however, since some programs dynamically alter memory, refer to the applicable chapter of this manual regarding restrictions on interrupting a test in progress.

In general if a test is operating under interrupt control, the program should be terminated via SENSE switch 3, then use the INT switch. This precludes leaving an interrupt hanging that may cause subsequent problems.

To return to a just-executed test program from the test executive, type

```
Gx.
```

where x is the starting address of the test program (refer to the program listing supplied with the software and to the following chapters for starting addresses).

2.2.3 Error Indications

After the preliminary tests and binary loader are loaded, the preliminary instructions test is automatically executed beginning at address 007002 If an error is detected, the program halts with the error code in the instruction register (table 2-13)

Table 2-13. Preliminary Instructions Test Error Codes

Error Code	Instruction Subtest
000001	TZA/DAR/JAZ/JAN
000002	LDA/IAR/STA
000003	LDB/JBZ/TZB
000004	IBR/DBR
000005	LDX/JXZ/TZX
000006	IXR/DXR
000007	LDAI/JAN
000010	LDAI/ERA/JAN
000011	ERAI/JAP
000012	LDBI/TBA
000013	LDXI/TXA
000014	LDB/TBX
000015	LDA/ERA
000016	LDA/STA
000017	LDB/STB
000020	LDX/STX
000021	XAZ
000022	XBZ
000023	ROF/SOF/JOF
000024	ROF/JOF/JMP
000025	JMPM/(JMP)
000027	LRLA
000030	LLSR
000031	LLSR
000032	LLRL
000033	LLRL
000034	ADD
000035	ADDI/ORAI
000036	SUB
000037	NOP
000040	INCR 03 (005103)
000041	DECR 02 (005302)
000042	MERG 032 (005032)
000043	LSRA
000044	LDA
000045	STA
000046	ANAI
000047	STAI
000050	XIF 022 (003022)
000051	JIF 011 (001011)

To continue program execution after an error halt, press START or RUN. To loop on the subtest in error:

a. Set SENSE switch 2.

b. Refer to the program listing for the jump address specified by the preceding JSS2 instruction, and set the P register to that address.

c. Press START or RUN.

Refer to the program listing for the significance of the A, B, and X registers after an error halt, and to the applicable maintenance manual for correction procedures.

If an error is detected in the memory test, the program halts with 000077 in the instruction register, the address of the faulty cell in the X register, and the bits in error in the A register. To continue the test, press START or RUN. To loop on an error:

a. Set SENSE switch 2.

b. Press START or RUN.

The binary loader computes the check-sum of each record of a test program (object) as it is loaded and compares the result with the expected value in the check-sum frames at the end of each record.

If a check-sum error is detected during program loading the program stops and the test executive outputs the message:

```
CHECKSUM ERROR X = xxxxxx
```

where xxxxxx is the error address in the X register.

To restart the program after a check-sum error halt:

a. Position the program tape in the reader at the previous record mark (three all-holes frames).

b. Press START or RUN.

If the record does not cause a halt on restarting, an intermittent fault probably exists in the reader. If a halt again occurs, visually examine the tape and compare it to the illustration of object tape format in the programming section of the applicable system reference manual. If the tape is correct and the reader is operating correctly, refer to the program listing for the address of CKSM and display it on the control panel. Analyzing the ones in the check-sum can indicate the location of the fault.

On a non-V75/V77 system, if the test executive does not print:

```
THIS IS THE V70/620 TEST EXECUTIVE

MEMORY SIZE IS nK
```

or, if on a V75/V77 system, the test executive does not print:

```
THIS IS THE V75 TEST EXECUTIVE
MEMORY SIZE IS nk
```

the TTY or its controller is not operating properly, the program halts with 000077 in the instruction register, and the TTY output routine times out.

Refer to the applicable maintenance manual for troubleshooting and correction procedures.
If an illegal control statement is input, the test executive outputs the message:

```
INVALID
```

→ Enter the correct control statement.

During TTY input activity or while the TTY is waiting for input, setting SENSE switch 3 terminates the input. This internal test executive routine also applies to test programs calling the test executive I/O routines.

If the system includes the power failure/restart (PF/R) option, the test executive PF/R routine permits automatic recovery of operating conditions after a prime power failure and recovery.

2.2.4 Test Examples

Heading Message for V70/620 System

```
THIS IS THE V70/620 TEST EXECUTIVE
MEMORY SIZE IS 16K
```

→ **Heading Message for V75/V77 System**

```
THIS IS THE V75 TEST EXECUTIVE
MEMORY SIZE IS 16K
```

Correct Control Statement A Input

```
A 000000 2.
A 000002 1,
A 000001 .
A 000001 ,
```

Cancelling Control Statement A Input

```
A 000001 \
A 000001 2\
A 000001 .
```

Invalid Control Statement A Input

```
A 000001 X  INVALID
A 000001 2X  INVALID
A 000001 .
```

Correct Control Statement B Input

```
B 000000 2.
B 000002 1,
B 000001 .
B 000001 ,
```

Cancelling Control Statement B Input

```
B 000001 \
B 000001 2\
```

Invalid Control Statement B Input

```
B 000001 X  INVALID
B 000001 2X  INVALID
B 000001 .
```

Correct Control Statement R Input

```
R0   000000 2.
R1   000000 1.
R2   000000.
R3   000000.
R4   000000.
R5   000000 4.
R6   000000.
R7   000000.
```

Cancelling Control Statement R Input

```
R0   000001\
R1   000000
R2   000300 2\
R3   010000 3\
R4   000000 4\
R5   000003 5\
R6   040000 6\
R7   050000 7\
```

Invalid Control Statement R Input

```
R0   000001    X    INVALID
R1   000001    2X   INVALID
R2   000001    2X   INVALID
R3   000001    X    INVALID
R4   000001    X    INVALID
R5   000001    X    INVALID
R6   000001    A    INVALID
R7   000001    B    INVALID
```

Correct Control Statement C Input

```
C10.=000000 1.
C10.=000001 .
C10,=000001 2.
C10,=000002 .

C10.=000002 1,
( 000011 ) =000000 2.
C10.=000001 ,
( 000011 ) =000002 .
C10.=000001 ,
( 000011 ) =000002 ,
( 000012 ) =000000 3.
C10.=000001 ,
( 000011 ) =000002 ,
( 000012 ) =000003 .
```

Cancelling Control Statement C Input

```
C10.=000002 \
C10.=000002 3\
C10,=000002 \
C10,=000002 3\
C10.=000002 .

C10,=000001 ,
( 000011 ) =000002 \
C10,=000001 ,
( 000011 ) =000002 3\
C10,=000001 ,
( 000011 ) =000002 ,
( 000012 ) =000003 4\

C10.=000001 ,
( 000011 ) =000002 \
C10.=000001 ,
( 000011 ) =000002 3\
C10.=000001 ,
( 000011 ) =000002 ,
( 000012 ) =000003 4\
C12.=000003 .
```

Invalid Control Statement C Input

```
C10.=000002 3X  INVALID
C10,=000002 3X  INVALID
C10.=000002 .
C1X  INVALID
C10,=000001 ,
( 000011 ) =000002 X  INVALID
C10,=000001 ,
( 000011 ) =000002 3X  INVALID
C10,=000001 ,
( 000011 ) =000002 ,
( 000012 ) =000003 X  INVALID
C12.=000003 .
```

Cancelling Control Statement D Input

```
D\
D4\
```

Invalid Control Statement D Input

```
DX  INVALID
D4X  INVALID
```

Correct Control Statement E Input

```
I0,7,0.
I1,6,1.
I2,5,2.
I3,4,3.
```

Cancelling Control Statement I Input

```
I\
I0\
I0,\
I0,7\
I0,7,\
I0,7,0\
I,\
I,7\
I,7,\
I,7,0\
I,,\
I,,0\
```

Invalid Control Statement I Input

```
IX  INVALID
I0X  INVALID
I0,X  INVALID
I0,7X  INVALID
I0,7,X  INVALID
I0,7,0X  INVALID
I7,0,0.  INVALID
```

Correct Control Statement L Input

```
L.
```

Cancelling Control Statement L Input

```
L\
```

Invalid Control Statement L Input

```
LX  INVALID
L0  INVALID
```

Cancelling Control Statement P Input

```
P\
P1\
P1,\
P1,6\
P1,6,\
P1,6,0\
P,\
P,6\
P,6,\
P,6,0\
P,,\
P,,0\
```

Invalid Control Statement P Input

```
PX  INVALID
P1X  INVALID
P1,X  INVALID
P1,6X  INVALID
P1,6,X  INVALID
P1,6,0X  INVALID
```

Invalid Control Statement S Input

```
SX  INVALID
S0X  INVALID
S0,X  INVALID
S0,7X  INVALID
S0,7,X  INVALID
S0,7,5X  INVALID
S0,7,5,X  INVALID
S0,7,5,7X  INVALID
```

Cancelling Control Statement S Input

```
S\
S0\
S0,\
S0,7\
S0,7,\
S0,7,5\
S0,7,5,\
S0,7,5,7\
S,\
S,7\
S,7,\
S,7,5\
S,7,5,\
S,7,5,7\
S,,\
S,,5\
S,,5,\
S,,5,7\
S,,,\
S,,,7\
```

Correct Control Statement X Input

```
X 000000 2.
X 000002 1,
X 000001 .
X 000001 ,
```

Cancelling Control Statement X Input

```
X 000001 \
X 000001 2\
```

Invalid Control Statement X Input

```
X 000001 X  INVALID
X 000001 2X  INVALID
X 000001 .
```

Correct Control Statement D Input

```
D0.
( 000000 )  000000 000001 000002 000003 000004 000005 000006 000007
( 000010 )  000010 000011 000012 000013 000014 000015 000016 000017
( 000020 )

D0,
( 000000 )  000000 000001 000002 000003 000004 000005 000006 000007
( 000010 )  000010 000011 000012 000013 000014 000015 000016 000017

D4.
( 000004 )  000004 000005 000006 000007
( 000010 )  000010 000011 000012 000013 000014 000015 000016 000017

D4,
( 000004 )  000004 000005 000006 000007
( 000010 )  000010 000011 000012 000013 000014 000015 000016 000017
```

Correct Control Statement I Input

```
I0,7,0.
I1,6,1.
I2,5,2.
I3,4,3.
D0.
( 000000 )  000000 000001 000002 000003 000003 000002 000001 000000
```

Control Statement L Input With Forced Check-Sum Error

```
I0,7,0.
L.CHECKSUM ERROR  X = 000001
D0.
( 000000 )  000000 000001 000002 000003 000003 000006 000001 000000
```

Correct Control Statement P Input

```
I0,7,0.
I1,6,1.
I2,5,2.
I3,4,3.
P1,6,7.@@F@@A@@A@@B@@C@@C@@B@@A@@G@@@@@G@@G

P1,6,777777.@@F@@A@@A@@B@@C@@C@@B@@A@@B@@A@@G
```

Correct Control Statement S Input

```
D0,
( 000000 )  000000 000001 000002 000003 000004 000005 000006 000007

S0,6,0,7777.
( 000000 ) =000000

S1,7,7,7777.
( 000007 ) =000007

S1,6,0,7777.
S1,6,7,7777.

S0,7,35,7.
( 000005 ) =000005
S1,5,1,1.
( 000001 ) =000001
( 000003 ) =000003
( 000005 ) =000005
S1,2,77,0.
( 000001 ) =000001
( 000002 ) =000002
```

SECTION 3
INSTRUCTION TESTS

The **Instructions Test Program** of MAINTAIN III tests machine instructions, including optional instructions. It operates under the control of the test executive (section 2), which provides the user interface, utility aids, and standard subroutines.

The V70/620-series internal instructions are divided into function groups to facilitate orderly testing. The test program is divided into three independent parts, two of which include, in addition to instruction subtests, an error printout routine, and a subtest-looping routine.

Part 1 of the instructions test program tests the following groups of instructions:

Basic control and switch (subtest 1)

Register change (subtest 2)

Overflow (subtest 3)

Shift/rotate (subtest 4)

Load/store (subtest 5)

Logical (subtest 6)

Jump/execute (subtest 7)

Arithmetic (subtest 8)

Indirect-addressing-limit test (subtest 13)

Register load/store (subtest 14)

Double load/store (subtest 15)

Register/register transfer (subtest 17)

Logical instructions (subtest 18)

Arithmetic instructions (subtest 19)

Byte instructions (subtest 20)

Jump instructions (subtest 21)

Indirect and interrupt vector (subtest 22)

Subtests 14 through 22 are applicable only to the V75 and V77 computers.

Part 2 tests the following groups of instructions:

Extended addressing (subtest 9)

Optional (subtest 10)

Input/output (subtest 11)

Part 3 is the instruction execution verification test (subtest 12). Part 3 is applicable only to the 620/f; it is not applicable to the V70 series.

The supervisor routine (IBGN) calls standard I/O routines for input/output via the Teletype printer and keyboard. The standard I/O routines are called indirectly through the test executive pointer table.

From user inputs, the supervisor controls:

a. Subtest sequencing

b. Test cycles

c. Output of the end-of-cycle message (e.g., END INST #2)

The error printout routine controls the output of error messages on the Teletype printer. This routine:

a. For non-V75 system, saves the contents of the A, B, and X registers at entry; for V75 system, saves registers R0 through R7.

b. Issues a carriage return and line feed.

c. Outputs the address that called the routine (this is the error print address for troubleshooting reference).

d. Outputs the saved contents of all registers.

e. Restores all registers with original contents of entries and exits.

The instructions test program contains special manual entry points for looping and/or troubleshooting a subtest or sequence of subtests.

The basic control portion of this test (subtest 1, part 1) verifies that the computer performs basic operations required for execution of this and subsequent routines properly. The instructions tested are:

Mnemonic	Instruction
ERA	Exclusive-OR memory and A register
JAN	Jump if A register negative
JAZ	Jump if A register zero
JMP	Jump (unconditional)
JMP*	Jump indirect
JOF	Jump if overflow indicator set
LDA	Load A register
LDAI	Load A register immediate
LLRL	Long logical rotation left
NOP	No operation
ROF	Reset overflow indicator
SOF	Set overflow indicator
STA	Store A register

The switch instruction portion of this test (subtest 1, part 2) consists of two routines, each requiring user intervention. The first of these routines tests the following instructions using both direct and indirect addressing:

Mnemonic	Instruction
JSS1	Jump if SENSE switch 1 set
JSS2	Jump if SENSE switch 2 set

Mnemonic	Instruction
JSS3	Jump if SENSE switch 3 set
JS1M	Jump and mark if SENSE switch 1 set
JS2M	Jump and mark if SENSE switch 2 set
JS3M	Jump and mark if SENSE switch 3 set
XS1	Execute if SENSE switch 1 set
XS2	Execute if SENSE switch 3 set

The second routine tests the following V70/620/f instructions:

JS1N	Jump if SENSE switch 1 not set
JS2N	Jump if SENSE switch 2 not set
JS3N	Jump if SENSE switch 3 not set
JS1NM	Jump and mark if SENSE switch 1 not set
JS2NM	Jump and mark if SENSE switch 2 not set
JS3NM	Jump and mark if SENSE switch 3 not set
TSA	Load A register with switches (transfer switches to the A register)
XS1N	Execute if SENSE switch 1 not set
XS2N	Execute if SENSE switch 2 not set
XS3N	Execute if SENSE switch 3 not set

The register change instructions describe four types of register-to-register operation:

a. Transfer

b. Increment

c. Decrement

d. Complement

Both positive and negative numbers are used and the overflow (OVFL) indicator is checked if the sign of a register changes.

The instructions tested (subtest 2) are:

Mnemonic	Instruction
COMP	Complement source-to-destination registers
CPA	Complement A register
CPB	Complement B register
CPX	Complement X register
DAR	Decrement A register
DBR	Decrement B register
DECR	Decrement source-to-destination registers
DXR	Decrement X register
IAR	Increment A register
IBR	Increment B register
INCR	Increment source-to-destination registers
IXR	Increment X register
MERG	Merge source-to-destination registers
TAB	Transfer A register to B register
TAX	Transfer A register to X register
TBA	Transfer B register to A register
TBX	Transfer B register to X register
TXA	Transfer X register to A register
TXB	Transfer X register to B register
TZA	Transfer zero to A register
TZB	Transfer zero to B register
TZX	Transfer zero to X register
ZERO	Clear registers to zero

The overflow instructions are:

Mnemonic	Instruction
AOFA	Add overflow to A register
AOFB	Add overflow to B register
AOFX	Add overflow to X register
SOFA	Subtract overflow from A register
SOFB	Subtract overflow from B register
SOFX	Subtract overflow from X register

This test (subtest 3) is an extension of the register change instructions test. The instructions are executed conditional on the status of overflow. Both true and false operations are tested.

This test (subtest 4) checks the following instructions:

Mnemonic	Instruction
ASLA	Arithmetic shift left A register
ASLB	Arithmetic shift left B register
ASRA	Arithmetic shift right A register
ASRB	Arithmetic shift right B register
LASL	Long arithmetic shift left
LASR	Long arithmetic shift right

Mnemonic	Instruction
LLRL	Long logical rotation left
LLSR	Long logical rotation right
LRLA	Logical rotation left A register
LRLB	Logical rotation left B register
LSRA	Logical shift right A register
LSRB	Logical shift right B register

The following one-word addressing instructions are tested (subtest 5) in all addressing modes (direct, indirect, relative, and indexed):

Mnemonic	Instruction
LDA	Load A register
LDB	Load B register
LDX	Load X register
STA	Store A register
STB	Store B register
STX	Store X register

The following two-word nonaddressing instructions are also tested:

Mnemonic	Instruction
LDAI	Load A register immediate
LDBI	Load B register immediate
LDXI	Load X register immediate

STAI	Store A register immediate
STBI	Store B register immediate
STXI	Store X register immediate

The logical instructions test (subtest 6) checks the following one-word addressing instructions using direct addressing:

Mnemonic	Instruction
ANA	AND memory and A register
ERA	Exclusive-OR memory and A register
ORA	Inclusive-OR memory and A register

The following two-word nonaddressing instructions are also tested:

Mnemonic	Instruction
ANAI	AND immediate
ERAI	Exclusive-OR immediate
ORAI	Inclusive-OR immediate

The jump/execute instructions test (subtest 7) comprises three routines. The first routine tests the following instructions using relative, direct, and indirect addressing. Both true and false conditions are checked.

Mnemonic	Instruction
JAN	Jump if A register negative
JANM	Jump and mark if A register negative
JAP	Jump if A register positive
JAPM	Jump and mark if A register positive
JAZ	Jump if A register zero
JAZM	Jump and mark if A register zero
JBZ	Jump if B register zero
JBZM	Jump and mark if B register zero
JMP	Jump (unconditional)
JMPM	Jump and mark (unconditional)
JOF	Jump if overflow indicator set
JOFM	Jump and mark if overflow indicator set
JXZ	Jump if X register zero
JXZM	Jump and mark if X register zero
XAN	Execute if A register negative
XAZ	Execute if A register zero
XBZ	Execute if B register zero
XEC	Execute (unconditional)
XOF	Execute if overflow indicator set
XXZ	Execute if X register zero

The second routine of the jump/execute instructions test checks the following V70/620/f instructions:

Mnemonic	Instruction
JANZ	Jump if A register not zero
JANZM	Jump and mark if A register not zero
JBNZ	Jump if B register not zero
JBNZM	Jump and mark if B register not zero
JOFN	Jump if overflow indicator not set
JOFNM	Jump and mark if overflow indicator not set
JXNZ	Jump if X register not zero
JXNZM	Jump and mark if X register not zero
XANZ	Execute if A register not zero
XBNZ	Execute if B register not zero
XOFN	Execute if overflow indicator not set
XXNZ	Execute if X register not zero

The third routine tests V70/620/f instructions IJMP (jump indexed) and JSR (jump and set return in X register). IJMP is tested in direct, indirect, relative, indexed relative to X, and indexed relative to B addressing modes. JSR is tested using both the B and X registers for return address storage.

The arithmetic-instructions test (subtest 8) checks standard arithmetic instructions with both positive and negative operands and those causing overflow.The instructions tested are:

Mnemonic	Instruction
ADD	Add memory to A register
ADDI	Add immediate
SUB	Subtract memory from A register
SUBI	Subtract immediate
INR	Increment memory and replace
INRI	Increment memory and replace immediate

The extended-addressing test (part 2, subtest 9) comprises two routines. The first routine tests preindexing in which the selected register contents are added to the second word of the instruction after the effective address has been accessed. The second word of two-word extended addressing instructions contains an effective address. Addressing modes are: direct, indirect, relative, and indexed relative to the X or B register. The instructions tested are:

Mnemonic	Instruction
ADDE	Add extended
ANAE	AND extended
ERAE	Exclusive-OR extended
INRE	Increment memory and replace extended
LDAE	Load A register extended
LDBE	Load B register extended
LDXE	Load X register extended
ORAE	Inclusive-OR extended
STAE	Store A register extended
STBE	Store B register extended
STXE	Store X register extended
SUBE	Subtract extended

The second routine tests V70/620/f postindexing in which the selected register contents are added to the first address not specifying indirect addressing. This effective address specifies the operand address. The instructions tested are: ADDE, LDAE, STAE, and SUBE. Direct, indirect, and postindexed relative to X and B addressing modes are used.

The optional-instructions test (subtest 10) checks the following instructions in all applicable addressing modes:

Mnemonic	Instruction
BT	Bit test (620/f only)
DIV	Divide
DIVE	Divide extended
DIVI	Divide immediate
MUL	Multiply
MULE	Multiply extended
MULI	Multiply immediate
SRE	Skip if register equal to memory (620/f only)

The I/O instructions tested by the input/output-test (subtest 11) are:

Mnemonic	Instruction
CIA	Clear and input to A register
CIAB	Clear and input to A and B registers
CIB	Clear and input to B register
EXC	External control
IME	Input to memory
INA	Input to A register
INAB	Input to A and B registers
INB	Input to B register
OAB	ORed output of A and B registers
OAR	Output from A register
OBR	Output from B register
OME	Output from memory
SEN	Program sense

These instructions are tested (subtest 11) using the TTY; the data transfer out instructions are directed to the TTY printer, and the data transfer in instructions are directed from the TTY keyboard.

This test (subtest 12) is applicable to the 620/f computer only (not to the V70 series). It verifies that all possible instructions can be executed without the computer halting or " hanging up" . Halt and I/O instuction are not tested.

The indirect-addressing-limit test (subtest 13) is applicable to the 620/f and V70 computers only. It verifies that the hardware limits the number of indirect addressing levels to five for one-word instructions, and to four levels with two-word instructions. This is done for real-time operating system considerations.

Subtests 14 through 22 are applicable only to the V75 and V77 computers.

The instructions tested by subtest 14 are:

Mnemonic	Instruction
LD	Load
ST	Store

The following addressing modes are tested utilizing the previous instructions:

direct
indexed
indirect
pre-indexed indirect

The instructions tested by Subtest 15 are:

Mnemonic	Instruction
DLD	Double Load
DST	Double Store
LDI	Load Immediate

The Double load/store instructions are tested utilizing the following addressing modes:

direct
indexed
indirect
pre-indexed indirect

The instructions tested by subtest 16 are:

Mnemonic	Instruction
INC	Increment Register
DEC	Decrement Register

Overflow and underflow operation is also tested using the above instructions.

The instructions tested by subtest 17 are:

Mnemonic	Instruction
T	Transfer

The instructions tested by subtest 18 are:

Mnemonic	Instruction
COM	Complement Register
DOR	Double OR
DER	Double Exclusive OR
DAN	Double AND

The instructions tested by subtest 19 are:

Mnemonic	Instruction
AD	Add
SB	Subtract
ADI	Add Immediate
DADD	Double Add
DSUB	Double Subtract
ADR	Add Register
SBR	Subtract Register

In addition, the overflow and underflow tests are also performed on the instruction of subtest 19.

The instructions listed by subtest 20 are:

Mnemonic	Instruction
LBT	Load Byte
SBT	Store Byte

The load/store byte instructions are tested using the indexed and indexed indirect addressing modes.

The instructions tested by subtest 21 are:

Mnemonic	Instruction
JZ	Jump if Register Zero
JNZ	Jump if Register Not Zero
JN	Jump if Register Negative
JP	Jump if Register Positive
JDZ	Jump if Double-Precision Register Zero
JDNZ	Jump if Double-Precision Register Not Zero

In addition to testing the asserted and non-asserted condition, the indirect jumps are also tested.

Subtest 22 tests the following operations:

- levels of indirect addressing.
- correct execution of instruction at a vectored interrupt address.

The following instruction codes are tested using the RTC interrupt facility:

Mnemonic	Instruction
LD	Load
LDI	Load Immediate
ADI	Add Immediate
SBT	Store Byte
LBT	Load Byte
DADD	Double Add
JZ	Jump if Register Zero
ROF	Reset Overflow
ADR	Add Register
INC	Increment Register

The instructions test program is designed to test the minimum configuration of a V70/620-series computer with 8K of memory (maximum, 32K) and a 33/35 ASR TTY.

The instructions test program object format is normally a punched paper tape for loading from the TTY or high-speed paper tape reader. Other media are available (e.g., card object deck).

3.1 PRELIMINARY PROCEDURES

To load the instructions test program from the Teletype:

a. Load and execute the test executive program (section 2).

b. Position the Instructions Test Program Object on the Object Input Device.

c. Type L, followed by a period, on the TTY keyboard to command the test executive to load the program tape.

When program-loading is complete, the Teletype prints the message:

```
THIS IS V70/620 INSTRUCTION TEST, PART 1
(2 or 3)

CPU TYPE =
```

Respond to the 'CPU TYPE =' message by typing one of the following codes, followed by a period:

Type	Computer
1	620/i or 620/L with standard instruction set
2	620/i or 620/L with optional instructions
3	602/f with standard instruction set
4	620/f with optional instructions/70 system
7	V75 computer
10	V77-200/400 without control panel
11	V77-200/400/600
21	V77-800

620/L and 620/L-100 codes are the same.

The instruction subtests apply to the computer type as follows:

Computer	Subtests
620/i or 620/L with standard instruction set	1-8,11
620/i or 620/L with optional instructions	1-11
620/f with standard instruction set	1-9,11-13
620/f with optional instructions/-70 system	1-13
V75 or V77	14-22

NOTE: For the type 10 computers, no halts are performed in subtest 2.

If, before typing the period to complete computer-type input, the user wishes to change the specification to another computer type:

a. Type a backarrow (←).

b. Type the new computer-type code, followed by a period.

When a valid computer-type code and a period have been entered, the TTY printer outputs `cycles =`.

Type one of the following:

Input	Definition
.(period)	Specifies continuous execution of the test and suppresses the END INST message after each test cycle

Input	Definition
,(comma)	Specifies continuous execution of the test and printing of the END INST message after each test cycle
Octal number followed by a period	Specifies automatic termination of the test after the designated number of cycles suppresses the END INST message after each test cycle
Octal number followed by a comma	Specifies automatic termination of the test after the designated number of cycles and printing of the END INST message after each test cycle

The test can be terminated at the completion of the current test cycle by setting SENSE switch 3, which returns control to the beginning of the test program.

Error conditions are described in control panel mode of operation.

The following SENSE switch options apply to all of the instructions test program except subtests 1 and 12.

Switch	Set	Reset
1	Halt on error	Print error data
2*	Loop on subtest	Halt after subtest
3	terminate test	continue test

* SENSE switch 2 can be used with special program entry points for troubleshooting.

To loop on a subtest for troubleshooting (parts 1 and 2):

a. In the listing supplied with the program, locate the special troubleshooting routine labled ITRS and load the address of ITRS in the P register.

b. Load the appropriate computer number in the A register.

c. Press START or RUN. The program halts with zero in the instruction register.

d. Select the desired subtest for looping. Refer to the address in ITRS.

e. Load the selected address in the P register, and set SENSE switch 2.

f. Press START or RUN.

If the instructions test program is run on the 620/f or V70 computer, pressing the INT switch returns control to the test executive.

3.2 EXECUTING INSTRUCTION TESTS

Upon entry of valid CPU type and number of cycles, the basic control portion of the control and switch subtest is executed; the program then halts with 000600 in the instruction register. To operate the switch portion of the subtest:

a. Set all SENSE switches.

b. Press START or RUN. The program halts with 000500 in the instruction register.

c. Reset all SENSE switches.

d. Press START or RUN.

The program halts with 000700 in the instruction register for 620/i and 620/L testing, and 000400 for V70/620/f testing.

The operation on 620/i and 620/L uses the following procedure. The halt described in item d above indicates completion of the switch test.

a. Select SENSE switch settings.

b. Press RUN to begin execution of subtests 2 through 8.

To execute these tests on the 620/f or V70 series systems, when the program halts, test TSA (transfer switches to A register):

a. Set STEP/RUN to STEP.

b. Display the A register.

c. Set all register entry switches (all ones).

d. Press START twice.

e. Verify that the A register contains all ones.

f. Set register entry switches to 0125252.

g. Press START.

h. Verify that the A register contains 0125252.

i. Reset the register entry switches.

j. Press START.

k. Verify that the A register contains all zeros.

l. Set the P register switch.

m. Set STEP/RUN to RUN and press START.

The program halts with 000700 in the instruction register, completing the switch test. To continue testing:

a. Select sense switch settings.

b. Press START.

Subtests 2 through 8 are executed and the message:

```
END INST #1
```

is output at the end of each cycle of testing unless suppressed.

After execution of these subtests, control is returned to the part 1 supervisor routine and the message:

```
CPU TYPE =
```

is output. To rerun this portion of the instructions test program.

To execute part 2 of the instructions test program, load the program tape.

Part 2 automatically executes the extended addressing instructions test (subtest 9) and the optional instructions test (subtest 10), the specified number of cycles or until SENSE switch 3 is set.

The message:

```
END INST #2
```

is output at the end of each cycle of subtests 9 to 10 unless suppressed.

At the completion of subtests 9 and 10, if applicable, the I/O instructions test (subtest 11) is executed and outputs the message:

```
THIS IS THE I/O INSTRUCTION TEST

PLEASE TYPE IN A LOWER CASE CHARACTER
```

Type any of the standard lower-case characters (section 5, table 5-1) on the TTY keyboard as requested to initiate the testing of the EXC instruction (EXC 1004xx, initialize TTY, where xx is the TTY device address). This instruction resets the controller and sets the sense signal false. If the instruction is successfully executed, the message:

```
THANK YOU
```

is output, followed by:

```
NOW TYPE ASDFASDFASDFAS
```

Type the characters as specified on the TTY keyboard.

When all the I/O instructions have been tested:

a. Control returns to the instructions test program (part 2) supervisor.

b. The message:

```
CPU TYPE =
```

is output, unless SENSE switch 3 has been set, terminating the test and returning control to the beginning of the instructions test programs.

Part 3 of the instructions test program (applicable to the 620/f only) automatically executes the instruction execution verification test (subtest 12) the specified number of cycles or until SENSE switch 3 is set. The message:

```
END INST #3
```

is output at the end of each cycle if it is not suppressed.

When all subtest 12 cycles are complete:

a. Control returns to the supervisor routine.

b. The message:

```
CPU TYPE =
```

is output.

Error conditions are described in section 3.3.

When testing the 620/f and V70 series computers, return control to the test executive by pressing the INT switch.

To return to the instructions test program from the test executive, type G600.

The value 600 represents the starting address of the instructions test program.

3.3 ERROR INDICATIONS

If an illegal entry is typed on the system's Teletype in response to:

```
CPU TYPE =
CYCLES =
```

the message:

```
INVALID
```

is output and the program waits for a correct entry.

During the basic control and switch portion of this subtest, programmed halts allow the setting of the SENSE switches.

All other halts indicate the occurrence of errors. SENSE switch options are not applicable because the switches are being tested.

Refer to the program listing to correlate the instruction(s) under test with the contents of the P register. Error messages are not output.

During the switch portion of this subtest, the program halts upon detection of an error. The P register contains the error address, and the A, B, and X register, their values when the error is detected. Error messages are not output.

Press START or RUN to continue testing after an error halt.

Error reporting during subtests 2 through 8 is a function of SENSE switch 1.

If SENSE switch 1 is set, the program halts when an error is detected. The P register contains the error address. The significance of the A, B, and X register contents can be determined by referring to the program listing. Refer also to the listing to correlate the failing instruction(s) with the P register contents.

If SENSE switch 1 is reset and the error condition does not prohibit normal printout, an error message of the form:

```
(nnnnnn)  aaaaaa    bbbbbb    xxxxxx
```

where

(nnnnnn) is the address of the instruction in error
aaaaaa is the A register contents
bbbbbb is the B register contents
xxxxxx is the X register contents

Refer to the program listing to identify the failing instruction(s) and the significance of the A, B, and X register contents.

For subtest 13 through 22, there are two error messages:

```
NO INDIRECT ADDRESS LIMITING

ERROR-INCORRECT LEVELS OF INDIRECT
```

If SENSE switch 1 is set the program will halt instead of printing the message. Refer to the program listing to correlate the error condition with the testing sequence.

Extended addressing instructions are standard on the V70/620/f, but applicable only to the 620/i or 620/L with optional instructions, and, therefore, are not executed on other 620 computers.

The optional instructions test subtest is executed only when specified by the user, and only those instructions actually present in the system are tested.

The 620/i divide algorithm (DIV, DIVI, and DIVE instructions) does not produce correct results in all cases. When the dividend is negative and the divisor can be evenly divided into the dividend, the quotient in the B register is one less than it should be, and the A register, which normally holds the remainder, contains the absolute value of the divisor with the sign of the dividend. The V70/620/f divide algorithm is corrected.

The multiply algorithm (MUL, MULI, and MULE instructions) is identical in all V70/620 systems and needs no correction.

Error-reporting for this subtest and the I/O instructions subtest uses a common error control routine (K09, refer to the program listing), excluding tests of the BT and SRE instructions. If SENSE switch 1 is set, the program halts upon detection of an error with 000300 in the instruction register. If SENSE switch 1 is reset, the error printout routine (IQ80) is called and the error data stored for printout at the conclusion of the test, and the testing continues until terminated. In this case, the error address that is printed out is the address of the error control routine, and the X register printout is the address of the failing instruction. The original X register contents are saved at the address labeled KSVX.

Tests of BT and SRE contain separate error-reporting calls (refer to the program listing).

The I/O instructions test begins with the message:

```
THIS IS THE I/O INSTRUCTION TEST

PLEASE TYPE IN A LOWER CASE CHARACTER
```

If the first line of this message is not identical to the above, the OBR (output B register) instruction is in error. The first three words of the second line test the OAR (output A register) instruction, and the remainder of the line, the OME (output from memory) instruction.

The EXC (external control) instruction should clear the TTY read buffer. If it does not, the message:

```
EXEC (1004xx) DOES NOT WORK
```

is output. If EXC is correctly executed, the message:

```
THANK YOU
```

is output, followed by:

```
NOW TYPE ASDFASDFASDFAS
```

to test the OAB (output A and B registers) instruction. When the characters are typed exactly as given, the program compares the ASCII code for each character and stores error addresses (if any) in a table for output upon completion of the test.

Part 3 of the program has no programmed error halts or error message printouts. If an error is detected, the test does not run to completion. Refer to the program listing for SINS, which contains the last word executed.

For all subtests, except 10 and 11, the P register contains the error address, and the A, B, and X registers, the values at the time of the halt (refer to the program listing).

For subtest 10, multiply/divide errors halt the program with the instruction register containing 000300; the A and B registers, current values; and the X register, the address of the instruction in error. The contents of the X register when the error is detected are saved at the address labeled KSVX. This address can be displayed if the X register is operated on by the instruction in error.

For subtest 11, errors halt the program with the instruction register at 000200. The A register contains the actual input data, and the B register, the expected data. The X register contains the address of the failing instruction (refer to the program listing).

For subtest 12 (part 3), errors halt (or "hang-up") the program at points that cannot be defined. Refer to the program listing for SINS, which contains the word that is executed last. If this is not the point at which the program halts, there is an error.

Refer to the applicable system maintenance manual for correction procedures.

3.4 TEST VALIDATION EXAMPLES

The results presented in this section were extracted from Teletype printed copy collected during testing.

EXAMPLE 1

Execute part 1 on a type 2 computer (with optional instructions) with an 8K memory:

```
THIS IS THE 620 INSTRUCTION TEST,PART 1
CPU TYPE = 2.
CYCLES = 1,
END INST #1
CPU TYPE = 2.
CYCLES = 350.
CPU TYPE = 2.
CYCLES = 5,
END INST #1
END INST #1
END INST #1
END INST #1
END INST #1
CPU TYPE =
```

EXAMPLE 2

Execute part 1 on a type 2 computer with an 8K memory, but specifying other computer types:

```
THIS IS THE V70/620 INSTRUCTION TEST, PART 1

CPU TYPE = 1.
CYCLES = 1,
END INST # 1
CPU TYPE = 3.
CYCLES = 1.
( 003572 )      000001     000000     000000
( 003675 )      000001     000000     000000
( 004004 )      177777     000000     000000
( 004061 )      177777     000000     000000
( 004105 )      177777     000000     000000 NO INDIRECT  ADDRESS LIMITING
CPU = 4.
CYCLES = 1.
( 003572 )      000001     000000     000000
( 003675 )      000001     000000     000000
( 004004 )      177777     000000     000000
( 004061 )      177777     000000     000000
( 004105 )      177777     000000     000000 NO INDIRECT  ADDRESS LIMITING
     END INST #1
     CPU TYPE =
```

Note the error printouts when computer types 3 and 4 are specified. The program tested the V70/620/f-only jump-if-not and execute-if-not instructions. The error printouts indicate invalid operations.

EXAMPLE 3

Execute part 2 on a type 2 computer:

```
THIS IS THE V70/620 INSTRUCTION TEST,
PART 2

CPU TYPE = 2.
CYCLES = 1,
END INST #2
THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE IN A LOWER CASE CHARACTER
THANK YOU
NOW TYPE ASDFASDFASDFAS
         ASDFASDFASDFAS
CPU TYPE = 2.
CYCLES = 13.
THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE IN A LOWER CASE CHARACTER
```

(I/O input test bypassed with SENSE switch 3)

```
CPU TYPE = 3←2 (Note backarrow to correct input)
CYCLES = 3,
END INST #2
END INST #2
END INST #2
THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE IN A LOWER CASE CHARACTER
THANK YOU

NOW TYPE ASDFASDFASDFAS
         ASDFASDFASDFAS
CPU TYPE = 2.
CYCLES = 100.

THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE IN A LOWER CASE CHARACTER
THANK YOU
NOW TYPE ASDFASDFASDFAS
         ASDFASDFASDFFF  Input  Error

( 003311 ) 000301    000306    003350
( 003311 ) 000323    000306    003351
CPU TYPE =
```

(Runs until terminated with SENSE switch 3)

```
THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE A LOWER CASE CHARACTER
```

(I/O test bypassed with SENSE switch 3)

```
CPU TYPE =
```

EXAMPLE 4

Execute part 2 on a type 2 computer, but specifying other computer types:

```
THIS IS THE V70/620 INSTRUCTION TEST,
PART 2

CPU TYPE = 3.
CYCLES = 2,
END INST #2
END INST #2
THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE IN A LOWER CASE CHARACTER
THANK YOU
NOW TYPE ASDFASDFASDFAS
         ASDFASDFASDFAS
CPU TYPE = 4.
CYCLES = 1,

( 002337 )     177773   000002   001643
( 002337 )     177773   177777   001652
( 002337 )     177777   000001   001715
( 002337 )     177777   000001   001721
( 002337 )     177777   177777   001764
( 002337 )     177777   177777   001770
( 002233 )     125252   000000   000000
( 002320 )     002362   000000   000000

END INST #2
THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE IN A LOWER CASE CHARACTER
```

(I/O input test bypassed with SENSE switch 3)

```
CPU TYPE =
```

Note the error printouts when computer type 4 is specified. The program tested the 620/f-only division instructions and BT and SRE. The error printouts indicate invalid operations.

EXAMPLE 5

Execute part 1 on a type 3 computer (V70/620/f with standard instructions) with an 8K memory:

```
THIS IS THE V70/620 INSTRUCTION TEST,
PART 1

CPU TYPE = 3.
CYCLES = 1,
END INST #1
CPU TYPE = 3.
CYCLES = 5,
END INST #1
END INST #1
END INST #1
END INST #1
END INST #1
CPU TYPE = 3.
CYCLES = 1000.
CPU TYPE = 3.
CYCLES =. (Continuous; terminate with SENSE switch 3)
CPU TYPE =
```

EXAMPLE 6

Execute part 2 on a type 3 computer (8K memory):

```
THIS IS THE V70/620 INSTRUCTION TEST,
PART 2

CPU TYPE = 3.
CYCLES = 1,
END INST #2
THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE IN A LOWER CASE CHARACTER
```

(I/O input test bypassed with SENSE switch 3)

```
CPU TYPE = 3.
CYCLES = 10,
END INST #2
END INST #2
END INST #2
END INST #2
END INST #2
END INST #2
END INST #2
END INST #2
THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE IN A LOWER CASE CHARACTERS
```

EXAMPLE 7

Execute parts 1 and 2 on a type 4 computer (V70/620/f with optional instructions) and 8K of memory:

```
THIS IS THE V70/620 INSTRUCTION TEST,
PART 1

CPU TYPE = 4.
CYCLES = 1,
END INST #1
CPU TYPE = 4.
CYCLES = 300.
CPU TYPE = 4.
CYCLES = 5,
END INST #1
END INST #1
END INST #1
END INST #1
END INST #1
CPU TYPE =

THIS IS THE V70/620 INSTRUCTION TEST,
PART 2

CPU TYPE = 4.
CYCLES = 12,
END INST #2
END INST #2
END INST #2
END INST #2
END INST #2
END INST #2
END INST #2
END INST #2
END INST #2
END INST #2
END INST #2
END INST #2

THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE IN A LOWER CASE CHARACTER
THANK YOU
NOW TYPE ASDFASDFASDFAS
         ASDFASDFASDFAS
CPU TYPE =
```

EXAMPLE 8

Execute part 1 on a type 4 computer (8K memory), but specifying other computer types:

```
THIS IS THE V70/620 INSTRUCTION TEST,
PART 1

CPU TYPE = 1.
CYCLES = 1,
END INST #1
CPU TYPE = 1.
CYCLES = 1.
CPU TYPE = 1.
CYCLES = 10,
END INST #1
END INST #1
END INST #1
END INST #1
END INST #1
END INST #1
END INST #1
END INST #1
CPU TYPE = 1.
CYCLES = 10.
CPU TYPE = 2.
CYCLES = 3,
END INST #1
END INST #1
END INST #1
CPU TYPE = 2.
CYCLES = 100.
CPU TYPE =
```

EXAMPLE 9

Execute part 2 on a type 4 computer (8K memory), but specifying other computer types:

```
THIS IS THE V70/620 INSTRUCTION TEST,
PART 2

CPU TYPE = 1.
CYCLES = 1,
```

continued

```
THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE IN A LOWER-CASE CHARACTER
THANK YOU
NOW TYPE ASDFASDFASDFAS
         ASDFASDFASDFAS
CPU TYPE = 2.
CYCLES = 1,
( 002337 )      000005   000003   001623
( 002337 )      000005   000001   001631
( 002337 )      000001   177777   001700
( 002337 )      000001   177777   001704
( 002337 )      000001   000001   001747
( 002337 )      000001   000001   001753
END INST #2

THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE IN A LOWER CASE CHARACTER
THANK YOU
NOW TYPE ASDFASDFASDFAS
         ASDFASDFASDFAS
CPU TYPE = 3.
CYCLES = 1,
END INST #2


THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE IN A LOWER CASE CHARACTER
THANK YOU
NOW TYPE ASDFASDFASDFAS
         ASDFASDFASDFAS
CPU TYPE = 4.
CYCLES = 1,
END INST #2


THIS IS THE I/O INSTRUCTION TEST
PLEASE TYPE IN A LOWER CASE CHARACTER
THANK YOU
NOW TYPE ASDFASDFASDFAS
         ASDFASDFASDFAS
CPU TYPE =
```

Note the error printouts when the type 2 computer is specified. The program tested the 620/i-only divide instructions on a V70/620/f.

EXAMPLE 10

Execute part 3 (subtest 12). Note that this subtest is only applicable to the 620/f computer (part 3 does not apply to the V70 series). The program does not accept inputs that specify other 620-series computers.

```
THIS IS THE 620 INSTRUCTION TEST,PART 3

CPU TYPE = 1.  INVALID
CPU TYPE = 2.  INVALID
CPU TYPE = 3.
CYCLES = 3.
END INST #3
CPU TYPE = 4.
CYCLES = 4,
END INST #3
END INST #3
END INST #3
END INST #3
CPU TYPE = 5.  INVALID
CPU TYPE = 6.  INVALID
CPU TYPE =
```

EXAMPLE 11

Execute 1, 4, and 5 cycles of the V75 instruction test (subtests 1 through 22).

```
THIS IS THE 620 INSTRUCTION TEST, PART 1:

CPU TYPE = 7.
CYCLES = 1,
END INST #1
CPU TYPE = 7.
CYCLES = 4,
END INST #1
END INST #1
END INST #1
END INST #1
CPU TYPE = 7.
CYCLES = 5,
END INST #1
END INST #1
END INST #1
END INST #1
END INST #1
```

SECTION 4
MEMORY TEST PROGRAM

The **Memory Test Program** of MAINTAIN III tests the operation of memory in the V70/620 series computers. It does not test the read-only memory (ROM). The program ascertains the operational status of the computer memory and assists in locating and correcting faults. Parity errors are also reported if the memory parity option is included in the system, or V77 ERCC error correct memory is used. All available memory sizes can be tested (8K through 32K).

The memory test program is designed to test the minimum configuration of a V70/620 series computer with 8K memory (maximum, 32K), a 33/35 ASR TTY Sperry Univac modified), and, if applicable, the memory parity option.

The format of the memory test program is normally a punched paper tape for loading from the Teletype or high-speed paper-tape reader. Other media are available (e.g., card object deck).

The memory test program consists of two parts: part 1 and part 2.

Unique Address Routine (test 1). The unique address routine tests the memory address register and associated circuits. It determines if a unique address exists for each location in memory. This is accomplished by storing the address of each location in itself, followed by a read and compare process for each address. Error number 1 is used with this test.

Binary Address (test 2, part 2 only). This routine zeros an entire 4K of memory. Then certain locations are set to all ones, and the entire 4K is tested.

Error number 50 is used with this test.

All Zeros Routine (test 3). This routine stores a zero word in each memory location. The contents of each location is then read and checked to determine if any extraneous bits were picked up. The routine is repeated three times before advancing to test 4. Error number 2 is used with this test.

All Ones Routine (test 4). This routine stores an all-ones word in each memory location. The contents of each location is then read and checked to determine if any bits were dropped. The routine is repeated three times before advancing to test 5. Error number 3 is used with this test.

Checkerboard Routine (test 5). This routine tests memory with alternate words of 125252 for 16-bit memories. The checkerboard pattern is reversed on alternate cycles of the test routine. The routine is repeated three times before advancing to test 6. Error number 4 is used with this test.

Circulating Bit (test 6). This routine uses the worst case patterns to determine whether all zeros or all one will be stored in a given address. One bit at a time is toggled to see if any other bits in that word change. The routine is repeated three times per memory test cycle.

Error numbers 40 through 47 are used to denote the worst case pattern used. If the user inputs worst case patterns, error 40 is used for the first pattern, error 41 for the second, etc.

If the default worst case patterns are used, the error number for the worst case pattern masks for the memory stack can be determined from table 4-1.

Table 4-1. Error Numbers for Worst Case Pattern Masks

Error Number	Worst Case Pattern Mask: Bits Set	Worst Case Pattern Mask: Octal
40	0,1,7	(0203)
41	0,11	(04001)
42	2,4	(024)
43	2,5,6	(0144)
44	2,5	(044)
45	9,10	(03000)

Adjacent Cell Disturb (test 7). This routine builds an all zeros background. Each cell is then complemented and the adjacent MOD 64 cells checked for errors. The test is repeated with an all ones background. Error number 60 is used for the all zeros background, and number 61 for the all ones.

N Squared (test 10). This routine builds an all zeros background. Each cell is then complemented and the cells in the test area read and checked for errors. The test is repeated with an all ones background. Error number 70 is used for this test.

4.1 INITIAL CONDITION SELECTION

To load the memory test:

a. Load the test executive (section 2).

b. Position the memory-test-program tape (part 1 or part 2) in the paper-tape reader with leader at the reading station. After executing part 1 of the program, always reload the test executive for further testing.

c. Type L, followed by a period, on the Teletype keyboard.

When loaded, Part 1 halts with 000777 in the instruction register.

Load the TTY device address in the B register. Press START or RUN to continue. At loading time, the B register is preset to 000001 (standard TTY device address). SENSE switch settings can alter test programs as follows:

Switch	Set	Reset
1	Suppress error message printout	Print error messages

Switch	Set	Reset
2	Before error halt:	
	Halt on error	Bypass error halt
	After error halt:	
	Continue testing	Loop on the error
3	Terminate testing	Continue testing

To continue the test after an error halt, leave SENSE switch 2 set and press START or RUN.

To loop on an error, reset SENSE switch 2 and press START or RUN. Looping continues until SENSE switch 2 is again set.

If the memory test program Part 1, is run on the 620/for V70 series computer, pressing the INT (interrupt) switch returns control to the test executive.

4.2 EXECUTING THE MEMORY TEST PROGRAMS

4.2.1 Part 1

After successful loading, the memory test program outputs the message:

```
MEMORY TEST.
```

The memory test then requests the worst case memory patterns with the message:

```
WORST CASE PATTERN(S)
```

The user responds with up to eight octal patterns, separated by commas and terminated by a period. If the user wishes to use the patterns previously input, respond with only a period.

The program outputs the message:

```
TESTS TO EXECUTE =
```

Type one of the following:

Response	Definition
(period)	Specifies that tests 1, 3, 4, 5, and 7 are to be run (used to test semiconductor memories).
(comma)	Specifies that tests 1, 2, 3, 4, 5, and 6 are to be run (used to test core memories).
n	Execute Test n (input as many n's as required, separated by commas, and terminated by a period).

The memory test program outputs the following message:

```
CYCLES =
```

Type one of the following:

Response	Definition
(period)	Specifies continuous execution of the test and suppresses the END MEMO message after each cycle.
(comma)	Specifies continuous execution of the test and printing of the END MEMO message after each test cycle.
Octal number followed by a period	Specifies automatic termination of the test after a designated number of cycles and suppresses the END MEMO message after each test cycle.
Octal number followed by a comma	Specifies automatic termination of the test after a designated number of cycles and causes the END MEMO message to be printed after each test cycle.

The test is executed for the designated number of cycles or until terminated by the setting of SENSE switch 3. If SENSE switch 2 is reset during execution, the OVFL (overflow) indicator on the control panel of a 620 series computer lights momentarily when an error is detected.

The message END MEMO is output at the end of each cycle of the test.

When test execution is complete, the program outputs a message indicating the number of errors detected and the number of cycles the test was run. Control is then returned to the beginning of the program which again outputs the message:

```
MEMORY TEST
```

Error conditions are described in section 4.3.

4.2.2 Part 2

After successful loading, the memory test program outputs the message:

```
MEMORY TEST
```

The following message is output:

```
V70 PARITY ERROR INTERRUPT LOCATION
   (720=0) =
```

This refers to the V70 Parity Error Interrupt. For 620 Series Computers and V77 computers with ERCC Memory Arrays, respond with 0; then the 620 and ERCC parity errors are retained. For other V70 series computers, respond with the even address of the parity error location. (This message is printed only once after loading.) If the incorrect address is input, the program must be reloaded to input the correct address.

The memory test then requests the worst-case memory patterns with the message:

```
WORST CASE PATTERN(S)
```

The user responds with up to eight octal patterns, separated by commas and terminated by a period. If the user wishes to use the patterns previously input, respond with only a period.

The following messages are output:

```
MEMORY SIZE IS nK
4K MODULE(S) TO BE TESTED =
```

where n refers to the size of memory as detected by the program (i.e., 8K, 12K, 16K, 20K, 24K, 28K, or 32K). To test all of memory, type a period. To test specific 4K memory modules, type an octal digit corresponding to each module. For example, to test addresses 030000 through 057777, type

```
3,4,5
```

followed by a period. Nonconsecutive 4K modules can be tested. Separate each digit of the response with a comma. Up to 15 parameters can be specified (note that zero is equivalent to four parameters).

The program outputs the message:

```
TESTS TO EXECUTE =
```

Type one of the following:

Response	Definition
(period)	Specifies that tests 1, 3, 4, 5, and 7 are to run (used to test semiconductor memories).
(comma)	Specifies that tests 1, 2, 3, 4, 5, and 6 are to be run (used to test core memories).
n	Execute test n (input as many n's as required, separated by commas, and terminated by a period).

The memory test program outputs the following message:

```
CYCLES =
```

Type one of the following:

Response	Definition
(period)	Specifies continuous execution of the test and suppresses the END MEMO message after each cycle.
(comma)	Specifies continuous execution of the test and printing of the END MEMO message after each test cycle.
Octal number followed by a period	Specifies automatic termination of the test after a designated number of cycles and suppresses the END MEMO message after each test cycle.
Octal number followed by a comma	Specifies automatic termination of the test after a designated number of cycles and causes the END MEMO message to be printed after each test cycle.

The test is executed for the designated number of cycles or until terminated by the setting of SENSE switch 3. If SENSE switch 2 is reset during execution, the OVFL (overflow) indicator on the 620 control panel lights momentarily when an error is detected.

The message END MEMO is output at the end of each cycle of the test.

When test execution is complete, the program outputs a message indicating the number of errors detected and the number of cycles the test was run. Control is then returned to the beginning of the program which again outputs the message:

```
MEMORY SIZE IS nK
```

Error conditions are described in section 4.3.

4.3 ERROR INDICATIONS

```
TOO MANY PARAMETERS
```

indicates that more than fifteen parameters were supplied in response to nk module(s) to be tested. Specify the

correct number of parameters (15 or less). Also, if a specified module of memory is outside the memory range, the message:

```
MODULE NOT WITHIN MEMORY RANGE
```

is output. Enter the corrected parameters. If an illegal entry is typed in response to:

```
CYCLES =
```

the message:

```
INVALID
```

is output, and the inquiry is repeated. Corectly type the entry.

To cancel an entry before the period is typed, type a backslash or set SENSE switch 3 to return to the beginning of the program. Type a backarrow to delete a single digit in any response before termination, then type the correct digit.

If SENSE switch 1 is reset and the error condition does not prohibit normal printout, an error message of the form:

```
TEST ADDRESS EXPECTED ACTUAL CYCLE
```

will be typed once.

Where:

TEST = the test number
ADDRESS = address of word in error
EXPECTED = expected word
ACTUAL = actual word
CYCLE = current cycle

In Test 2, the location being modified is typed out.

To loop on an error word, perform the following procedure. Set SENSE switch 2. When an error is encountered, the computer will halt. Reset SENSE switch 2 and press START/RUN.

The error word will be written and read repetitively. Each read is preceded by a NOP. Whenever the word is in error, the error message will be printed. (Unless SENSE switch 1 is set.) To continue, set SENSE switch 2.

When Test 6 (Circulating Bit) is run, the test number will be 00xx4Y. Where xx, if 40, indicates that memory was not preloaded with the desired constant (ones or zeros). An xx from 00 to 21 indicates the bit being toggled.

Y indicates the worst case pattern used.

If the memory test program encounters a parity error, one of the following messages is output:

```
INSTRUCTION PARITY ERROR AT xxxxxx
ADDRESS PARITY ERROR AT xxxxxx
OPERAND PARITY ERROR AT xxxxxx
TRAP PARITY ERROR AT xxxxxx
PARITY ERROR AT xxxxxx
```

where xxxxxx is a memory address. For an instruction parity error, this address is 2 greater than the instruction containing the bad parity. For the remaining parity error types, the address is 3 greater than the instruction in error in the case of 1-word instructions and 4 greater for 2-word instructions. A trap parity error indicates a memory parity hardware malfunction.

Error messages for the V77 ERCC are:

V77 ERCC PARITY INTERRUPT AT XXXXXX
PARITY ERROR IN MODULE M PHYSICAL PAGE PPPPPP
PARITY BIT BB COORDINATES RR CC
DATA BIT BB COORDINATES RR CC
DOUBLE BIT ERROR

where:
xxxxxx is the address of the instruction being executed
m is the array module
PPPPPP is the page number
BB is the faulty bit
RR is the row coordinate
CC is the column coordinate

Following detection of a parity error, the instruction register contains a code corresponding to the type of error:

Error Code	Description
000020	Instruction parity error
000021	Address parity error
000022	Operand parity error
000023	Trap parity error (hardware malfunction)
000024	Parity error (V70 series)

and the B register contains the corresponding trap address:

Trap Address	Description
000100	Instruction parity error
000104	Address parity error
000110	Operand parity error
000114	Trap parity error

NOTE

Parity error detection is disabled at the beginning of the termination routine (term) of part 2. To enable parity interrupts again, press RESET.

After a parity error halt, press START or RUN to return to the beginning of the program (via the termination reporting routine).

An accumulated total of errors is output at the completion of the specified number of test cycles:

```
ERROR TOTAL = xxxxxx
NUMBER OF CYCLES RUN = xxxxxx
```

SENSE switch 1 settings do not affect this output.

Refer to the applicable system maintenance manual for correction procedures.

4.4 TEST VALIDATION EXAMPLES

The results presented in this section were extracted from TTY hardcopy collected during validation of the program.

EXAMPLE 1 -- V70/620/f Computer

Part 1:

```
MEMORY TEST
CYCLES = 3.
ERROR TOTAL = 000000
NUMBER OF CYCLES RUN = 000003

MEMORY TEST
CYCLES = 2,
END MEMO
END MEMO
ERROR TOTAL = 000000
NUMBER OF CYCLES RUN = 000002
```

(SENSE switch 3 set)

```
MEMORY TEST
CYCLES =
MEMORY TEST
CYCLES = W INVALID
CYCLES =
```

Part 2:

```
MEMORY TEST
MEMORY SIZE IS 32K

4K MODULES(S) TO BE TESTED = 0, 1, 2.
CYCLES = 3.
ERROR TOTAL = 000000
NUMBER OF CYCLES RUN = 000003
MEMORY SIZE IS 32K

4K MODULE(S) TO BE TESTED = 1.
CYCLES = .
ERROR TOTAL = 000000
```

(SENSE switch 3 set)

```
NUMBER OF CYCLES RUN = 000004
MEMORY SIZE IS 32K

4K MODULES(S) TO BE TESTED = 0, 1.
CYCLES = 4,
END MEMO
END MEMO
END MEMO
END MEMO
ERROR TOTAL = 000000
NUMBER OF CYCLES RUN = 000004
MEMORY SIZE IS 32K
```

EXAMPLE 2 -- 620/i Computer

Part 1:

```
MEMORY TEST
CYCLES = 3.

ERROR TOTAL = 000000
NUMBER OF CYCLES RUN = 000003

MEMORY TEST
CYCLES =

TEST EXEC
(Started at 000027 in the P register)
```

Part 2:

```
MEMORY TEST
MEMORY SIZE IS 16K

4K MODULE(S) TO BE TESTED = 0.
CYCLES = 2.
ERROR TOTAL = 000000
NUMBER OF CYCLES RUN = 000002
MEMORY SIZE IS 16K

4K MODULE(S) TO BE TESTED = 7,
        MODULE NOT WITHIN MEMORY RANGE
MEMORY SIZE IS 16K

4K MODULE(S) TO BE TESTED = 0,0,0,0,0,0,
TOO MANY PARAMETERS
MEMORY SIZE IS 16K

4K MODULE(S) TO BE TESTED = 0.
CYCLES = 1,
END MEMO
ERROR TOTAL = 000000
NUMBER OF CYCLES RUN = 000001
MEMORY SIZE IS 16K
```

EXAMPLE 3 -- 620/L Computer

Part 1:

```
MEMORY TEST
CYCLES =,
END MEMO
END MEMO
      .
      .
      .
END MEMO
END MEMO
ERROR TOTAL = 000000
NUMBER OF CYCLES RUN = 000052

MEMORY TEST
CYCLES =
```

continued

Part 2:

```
MEMORY TEST
MEMORY SIZE IS 8K

4K MODULE(S) TO BE TESTED = 0.
CYCLES = 1.
ERROR TOTAL = 000000
NUMBER OF CYCLES RUN = 000001
MEMORY SIZE IS 8K

4K MODULE(S) TO BE TESTED = 1.
CYCLES = 2,
END MEMO
END MEMO
ERROR TOTAL = 000000
NUMBER OF CYCLES RUN = 000002
MEMORY SIZE IS 8K

4K MODULE(S) TO BE TESTED =.
CYCLES = 1.
ERROR TOTAL = 000000
NUMBER OF CYCLES RUN = 000001
MEMORY SIZE IS 8K


4K MODULE(S) TO BE TESTED =.
CYCLES =.
TEST      ADDRESS    EXPECTED  ACTUAL
000005    (015465)   177777    000000
```

(SENSE switch 3 set)

```
ERROR TOTAL = 000001
NUMBER OF CYCLES RUN = 000020
MEMORY SIZE IS 8K
```

SECTION 5
TELETYPE TEST PROGRAM

The **Teletype (TTY) test program** of MAINTAIN III tests the operation of the TTY and isolates malfunctions. The Teletype units that can be tested are models 33 ASR, 35 ASR, and 35 KSR (Sperry Univac modified) and compatible CRT units.

Acceptable ASCII characters and their representations are listed in table 5-1.

The Teletype test program operates under the control of the test executive (section 2), which provides the user interface, utility aids, and standard subroutines. The following are the elements of the Teletype test program.

The printer test (PT) tests the printed output of the TTY. All 64 TTY characters are output in a specified pattern (section 2.2). Each line output starts with the second character of the previous line, thereby testing all characters in each of the 72 possible print positions.

The keyboard echo (KE) test accepts the input of characters from the TTY keyboard and outputs them to the printer so that input can be compared with output (section 2.3).

The keyboard character (KC) test verifies correct operation of the TTY keyboard. The user enters the characters, both upper and lower case, by pressing the applicable keys. The specified characters are immediately output on the TTY printer for visual comparison (section 2.4).

For ASR models only, the reader test (RT) verifies that the TTY paper tape reader reads known data patterns correctly and that it starts and stops in response to on and off commands (section 2.5). Sperry Univac supplies the patterned paper tape (part number 92V0107-005).

For ASR models only, the punch/reader (PR) test verifies punch and reader accuracy and correct response to on and off commands (section 2.6).

The print suppression (PS) test verifies proper print suppression for the model 35 ASR only (section 2.7).

Table 5-1. (ASCII) Standard Characters

BITS b_4 b_3 b_2 b_1	ROW / COLUMN	b_7 b_6 b_5 = 0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
		0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	¦
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

VTII-3157

The TTY test program is designed to test the minimum configuration of a V70/620-series computer with 8K of memory (maximum, 32K) and a 33/35 ASR TTY (Sperry Univac modified).

The program object format is normally a punched paper tape for loading from the TTY or high-speed paper tape reader. Other media are available (e.g., card object deck).

5.1 INITIAL CONDITION SELECTION

To load the TTY test program:

a. Load the test executive program.

b. Position the TTY test program tape in the reader with leader or any frame before the first data frame at the reading station.

c. Type L, followed by a period, on the keyboard to command the test executive to load the program tape.

When loading is complete, the printer outputs:

```
THE TELETYPE TEST IS LOADED

TELETYPE DA =
```

Type the one- or two-digit octal device address of the selected TTY unit. If the test halts, the device whose address was input is not on-line. To restart the test:

a. Ensure that the TTY is on-line.

b. Clear the instruction register to zero.

c. Load 000500 in the P register.

d. Press START or RUN. The printer again outputs the above-described message.

e. Type the correct device address.

When the program is successfully loaded and a valid device address entered, the bell on the selected TTY rings and the message:

```
TTY TEST IDENTIFIER =
```

is output. Respond by typing one of the two-letter test identifiers listed below, followed by a period.

Test	Identifier
Printer	PT
Keyboard echo	KE
Keyboard character	KC
Reader	RT
Punch/reader	PR
Print suppression	PS

Only SENSE switches 1 and 2 are applicable to the reader (RT) and punch/reader (PR) tests.

SENSE switch settings can alter test programs as follows:

Switch	Set	Reset
1	Suppress error halt	Halt with expected value in a register and actual value in B register
2	Suppress error table printout	Print error table
3	Return control to Teletype test selector (for RT and PR only, first print error totals)	Continue testing

If the TTY test program is run on the 620/f or V70 series computer, pressing the INT switch returns control to the test executive.

5.2 EXECUTING THE TESTS

To operate the printer test:

a. Select PT. The program responds by outputting successive lines of all 64 characters. Each line starts with the second character of the previous line.

b. Visually inspect output.

c. Set, then reset, SENSE switch 3 to terminate PT and return control to the test selection routine.

To operate the keyboard echo test:

a. Select KE. The program responds by outputting:

```
KEYBOARD ECHO TEST
```

b. Type selected characters on the TTY keyboard. The program responds by outputting the typed characters.

c. Visually inspect output.

d. Set, then reset, SENSE switch 3 to terminate KE and return control to the test selection routine.

To operate the keyboard character test:

a. Select KC. The program responds by outputting a line of lower case characters for reference.

b. Type an identical line. If input is correct, the program responds by outputting a line of upper case characters for reference.

c. With the SHIFT key depressed, type an identical line. If input is correct, the program responds by outputting the reference line of lower case characters.

d. Set, then reset, SENSE switch 3 to terminate the test and return control to the test selection routine.

To operate the reader test:

a. Position the test tape in the reader on any frame past the first RUBOUT (all-holes) character.

b. Select SENSE switch options, if desired (section 2.1.2).

c. Select RT.

The program:

a. Reads the test tape, and, if errors are detected, stores the expected results and actual values in an error table.

b. Executes the reader-off test.

c. Outputs the error table in accordance with SENSE switch settings.

d. Executes the reader-on test.

e. Repeats the cycle.

All 256 data bit combinations are tested, except:

Code	Subcode	Function
0214	014	Form
0221	021	Reader on
0223	023	Reader off
0222	022	Punch on
0224	024	Punch off

Set, then reset, SENSE switch 3 to terminate the test and return control to the test selection routine. When the test is terminated, the program outputs:

a. The number of times the tape was repeated (looped)

b. The number of errors (data and reader on/off)

c. The number of reader-on errors

d. The number of reader-off errors

PUNCH/READER TEST

To operate the punch/reader test:

a. Select PR. The program responds by outputting the message:

```
TYPE OF TTY (33 OR 35) =
```

b. Type the digits corresponding to the TTY type. The program responds by punching approximately 10 inches of leader, then pauses.

c. Position the punched leader in the reader with approximately two inches of slack.

d. Select SENSE switches, if desired.

e. Turn on the reader.

The program:

a. Punches an ascending binary pattern (excluding certain control codes.

b. Executes the punch on/off test.

c. Outputs 10 inches of blank tape.

d. Reads the tape in the reader and enters errors in the error table.

e. Prints the error table in accordance with SENSE switch settings.

f. Repeats the test.

Set, then reset, SENSE switch 3 to terminate the test and return control to the test selection routine. When the test is terminated, the program outputs:

a. The number of test cycles

b. The number of errors (data and punch on/off)

c. The number of punch on/off errors

Error conditions are described in section 3.5.

PRINT SUPPRESSION TEST

To operate the print suppression test:

a. Select PS. The program responds by continuously outputting:

```
ASR TTY PRINT SUPPRESSION TEST
```

b. Set, then reset, SENSE switch 3 to terminate the test and return control to the test selection routine.

Error conditions are described in section 3.6.

Note: The Print Suppression Test is only applicable to Teletype model 35. The Teletype model 33 does not have print suppression capability.

RETURNING TO THE TEST EXECUTIVE

To terminate the TTY test and return to the test executive when testing the 620/i or 620/L, computer:

a. Clear the instruction register.

b. Load 014000 in the P register.

c. Press SYSTEM RESET.

d. Press RUN.

When testing the 620/f or or V70 series computer, return control to the test executive by pressing the INT switch.

To return to the TTY test program from the test executive, type:

```
G500.
```

The value 500 represents the starting address of the TTY test program.

5.3 ERROR INDICATIONS

If an illegal entry is typed in response to TELETYPE DA = the message INVALID is output and the program waits for a correct entry. If an incorrect test identifier is input in response to the message:

```
TTY TEST IDENTIFIER =
```

the program outputs the message:

```
INVALID TEST IDENTIFER

TTY TEST IDENTIFIER =
```

Type the correct identifier, followed by a period.

PRINTER TEST

Errors in the printer test are determined by a visual examination of the test output. Successive lines of all 64 characters in the 72 possible positions, each line starting with the second character of the previous line, produce a diagonal pattern of all characters in all positions. If the pattern is broken by the printing of a character out of sequence, the printer is not operating properly.

EXAMPLE OF PRINTER TEST

Refer to the applicable system maintenance manual for correction procedures.

KEYBOARD ECHO TEST

As each character is input from the keyboard (section 2.3), visually examine the printer output. If the output is not identical to the character input, refer to the applicable system maintenance manual for correction procedures.

Example:

Keyboard Input

```
ASDFGHJKL;'ZXCVBNM,./
```

Printer Output

```
ASDAGHHKL;'XZCVBNM,.
```

Note that the absence of the last input character (/) indicates a TTY error.

KEYBOARD CHARACTER TEST

If, in inputting the characters in the reference lines of this test (section 2.4), an incorrect character is transmitted from the keyboard, the TTY bell rings and printer output is inhibited. The input portion of the test recycles so that the character can be retyped. To determine what portion of the input is invalid (if the character is printable), press the space bar to advance the internal pointer to the next character.

EXAMPLE OF KEYBOARD CHARACTER TEST

If errors persist, refer to the applicable system maintenance manual for correction procedures.

READER TEST

The reader test error output consists of a printout of expected and actual values.

Data error indications are of the form:

000xxx 000aaa

where xxx represents the expected result and aaa is the actual value.

Example:

```
EXPECTED        ACTUAL
000050          000040
```

The expected bit configuration was:

0 000 000 000 101 000

The reader, however, read:

0 000 000 000 100 000

Reader off error indications are of the form:

000000 00f000

where f represents the number of frames read after the reader off command was issued.

Example:

```
EXPECTED        ACTUAL
000000          004000
```

Four frames were read after the issuance of the reader-off command.

Reader-on error indications are of the form:

```
EXPECTED        ACTUAL
077777          00f000
```

where f = the first frame read.

After test termination, the program outputs error verification information of the form:

```
000477
000000
000000
000000
```

The first line represents the number of times the test was repeated. There were no errors.

PUNCH/READER TEST

The punch/reader test error output consists of a printout of expected and actual values.

Data error indications are of the form:

000xxx 000aaa

where xxx represents the expected result and aaa is the actual value.

Example:

```
EXPECTED        ACTUAL
000104          000100
```

The expected bit configuration was:

000 000 000 001 000 100

The punch, however, produced:

000 000 000 001 000 000

Punch on/off errors are indicated by the following output:

```
EXPECTED        ACTUAL
000000          000000
```

After test termination, the program outputs error verification information of the form:

```
000477
000000
000000
```

The first line represents the number of times the test was executed. Lines 2 and 3 indicate there were no data or punch on/off errors (section 2.6).

PRINT SUPPRESSION TEST

If errors occur during the print suppression test, asterisks appear in the continuous output.

5.4 TEST VALIDATION EXAMPLES

The results presented in this section were extracted from TTY hardcopy collected during validation of the program.

Begin testing:

```
THE TELETYPE TEST IS LOADED

TELETYPE DA =
```

Input valid device address:

```
TELETYPE DA = 01.
```

Test selection request:

```
TTY TEST IDENTIFIER =
```

Input valid test identifier:

```
TTY TEST IDENTIFIER = PT.
```

Examples of invalid test identifier inputs:

```
TTY TEST IDENTIFIER = 7 INVALID FG.

INVALID TEST IDENTIFIER

TTY TEST IDENTIFIER = HIJ INVALID
K INVALID


INVALID TEST IDENTIFIER
```

Keyboard echo test heading:

KEYBOARD ECHO TEST

Keyboard character test reference lines:

```
1234567890-QWERTYUIOPASDFGHJKL;ZXCVBNM,./
 '#$ &'()*=-@[ +↑<>
```

Example of reader test error printout:

```
TTY TEST IDENTIFIER = RT.

EXPECTED        ACTUAL
000363          000364
000364          000365

000001
000002
000000
000000
```

Example of punch/reader test error printout:

```
TTY TEST IDENTIFIER = PR.

TYPE OF TTY (33 or 35) = 33.

EXPECTED        ACTUAL
000373          000374
000374          000375
000375          000376
000376          000377
000000          000000

000001
000006
000001
```

SECTION 6
POWER-FAILURE/RESTART TEST

The Power-Failure/Restart Test Program of Maintain III tests the operation of the 620-series and 70 systems power failure/restart (PF/R) option.

The PF/R provides an orderly shutdown in case of power failure or turn-off and, when power is restored, restarts the program in progress when power was lost.

Power input to the computer is indirectly monitored by the PF/R. A power-failure-monitor voltage in the computer power supply is constantly being sensed to determine power status. If the monitor voltage drops (due to power failure or power switch turn-off), the PF/R causes an interrupt. This interrupt has the highest priority in the system (unless the memory protect option is used, then this interrupt will follow it in the priority order). In a V75 system, contents of the additional volatile registers R3 through R7 are also placed in memory. The CPU then executes a user-programmed service routine (table 6-1) that places the contents of volatile registers (A, B, X, P, and overflow) into memory. The program halts, the memory is disabled, and the system is reset. The power-down service routine (SAVE) cannot be interrupted by lower-priority options or controllers.

When power is restored, the PF/R enables the memory. The CPU executes a user-programmed power-up service routine (RESTORE) that restores the contents of the volatile registers, and the system resumes service of the program in progress at the time of the interrupt.

For a detailed description of the PF/R, refer to the applicable 620-series option manual.

The PF/R test program is designed to test the minimum configuration of a V70/620 series computer with 8K of memory, PF/R, and 33/35 ASR Teletype (TTY). The test can be performed in both TTY and control panel modes of operation.

The PF/R test program object format is normally a punched paper tape for loading from the TTY reader or a high-speed paper tape reader. Other media are available (e.g., card object deck).

Background programs and other test programs cannot be executed simultaneously with the PF/R test program, and the operations of other internal computer options (i.e., priority interrupt module, buffer interlace controller, real-time clock, etc.) are not monitored.

Table 6-1. Typical PF/R Service Routine

```
        ORG     040
        JMPM    PWRD
        JMP     PWRU

        POWER-DOWN PROCESSOR (SAVE)

        ORG     1000
PWRD    ENTR
        STA     SAVA        SAVE A, B, X REG
        STB     SAVB
        STX     SAVX
        TZA                 CHECK/SAVE OVFL
        DATA    005511      INCR A IF OVFL SET
        STA     SAVO
        INR     HLTF        SET PF/R FLAG
PHLT    HLT

        POWER-UP PROCESSOR (RESTORE)

PWRU    LDA     HLTF        CHK PWRUP FROM RUN
        JAZ     PHLT
        TZA                 CLEAR PF/R FLAG
        STA     HLTF

    (Coding to reinstate optional hardware after a
    power failure, if desired, must be defined here;
    refer to the PF/R manual for timing restric-
    tions. The PF/R test program makes no
    provision for monitoring or restoring option
    conditions.)

        LDA     SAVO        SETUP OVFL FLAG
        ROF
        JAZ     *+3

        SOF
        LDA     SAVA        RETU A, B, X REG
        LDB     SAVB
        LDX     SAVX
        JMP*    PWRD        RETU TO INT ADDR
                .
                .
                .
SAVA    DATA    0
SAVB    DATA    0
SAVX    DATA    0
SAVO    DATA    0
HLTF    DATA    0
        END
```

The PF/R test program consists of the following subtests:

a. Halt test

b. Volatile registers test

c. Memory test

The HALT test checks PF/R operation in computer halt mode. If power loss occurs in this mode:

a. The PF/R interrupt is not acknowledged.

b. The CPU and memory are immediately disabled.

c. The contents of the volatile registers are lost.

d. The program halts when power is restored to indicate that the PF/R power-down SAVE routine was not initiated.

The test is repeated four times, each pass setting up one of the following background bit patterns in the unused portion of memory:

a. All zeros

b. All ones

c. Ones in alternate bits

d. Alternate bits complemented

In each of the four power-down/power-up sequences, the PF/R test program compares the expected bit configuration with the actual value; if different, error messages are output.

The volatile-registers two-pass test verifies that the A, B, X, P, and overflow registers are not modified (prior to storage in memory) by a power-down SAVE routine.

The registers are loaded with predetermined bit configurations and these initial contents are compared with the actual values after the power-down/ SAVE routine is executed. Discrepancies produce error messages.

The memory test verifies that memory is not modified by a power-down/power-up sequence. It is repeated four times using the bit patterns: All zeros, all ones, ones in alternate bits, and alternate bits complemented.

After each pass of the test, the actual contents of memory are compared with the expected values; if different, error messages are output.

Memory locations above the test program are not saved.

6.1 INITIAL CONDITION SELECTION

To load the PF/R test program:

a. Load the test executive program.

b. Position the PF/R test program tape in the reader with leader at the reading station.

c. Type L, followed by a period, on the Teletype keyboard.

SENSE switch settings can alter test programs as follows:

Switch	Set	Reset
1	Suppress error message printout	Print error messages
2	Halt on error	Continue testing without halting
3	Terminate testing and return to the test program beginning	Continue testing

To continue the test after an error halt, set SENSE switch 2 and press START or RUN.

To loop on an error, reset SENSE switch 2 and press START or RUN. Looping continues until SENSE switch 2 is again set.

If the PF/R test program is run on the 620/f or V70 series computer, pressing the INT (interrupt) switch returns control to the test executive.

6.2 EXECUTING THE PF/R TEST PROGRAM

To operate the program after successful loading:

a. The PF/R test outputs the message:

```
POWER FAILURE/RESTART TEST
TIME DELAY =
```

b. Type the desired time delay constant:

For 620/i or 620/L = 010
For 620/f or 620/f-100 = 0134
For 620/L-100 = 032
For V70 with first 8K of Core Memory = 0123
For V70 with first 8K of SC Memory = 0230
For V77-800 = 0372 ←

c. The program outputs the message HALT TEST and waits for input from the operator.

To continue the halt test execution:

a. Initiate a power-down/power-up sequence by turning off, then restoring, CPU power.

For systems with MOS memories, place the computer in HOLD mode then to PWR ON to restore power.

To turn off power to the 620/f or V70 series computer, turn the key-operated power switch to PWR OFF and to PWR ON to restore power. On the 620/i, and 620/L computers, the POWER indicator/switch lights when pressed and power is on; pressing the switch then turns off the indicators and power to the CPU.

b. The program executes the first pass of the halt test, re-outputs the test title, and rings the TTY bell.

c. Repeat steps a and b three more times to complete the remaining passes of this four-pass test. ←

At the completion of pass 4, the program outputs an error message, if errors were detected (section 6); terminates the halt test; and outputs the message:

```
VOLATILE REGISTER TEST
```

on the TTY printer. The program waits in a loop, and the TTY bell rings.

To continue volatile-register test execution:

a. Turn off, then turn on, CPU power.

b. The program executes the first pass of this two-pass test; outputs an error message, if errors were detected (section 3); and rings the TTY bell.

c. Turn off, then turn on, CPU power to execute pass 2.

At the completion of pass 2, the program outputs an error message, if errors were detected; terminates the volatile registers test; and outputs the message:

```
CORE VALIDITY CHECK
```

on the TTY printer. The program waits in a loop, and the TTY bell rings.

To continue memory test execution:

a. Turn off, then turn on, CPU power.

b. The program executes the first pass of the memory test, outputs an error message (section 3) if errors were detected, and rings the TTY bell.

c. Repeat steps a and b for the remaining passes of this four-pass test.

At the completion of pass 4, the program terminates the memory test and outputs the message:

```
HALT TEST
```

To terminate the PF/R test program and return to the test executive when testing the 620/i or 620/L computer.

a. Clear the instruction register.

b. Clear the P register.

c. Press SYSTEM RESET.

d. Press RUN.

When testing the 620/f or V70 series computer, return control to the test executive by pressing the INT switch.

To return to the PF/R test program from the test executive, type G500. The value 500 represents the starting address of the PF/R test program.

6.3 ERROR INDICATIONS

HALT TEST

If, during the halt test, the program detects a discrepancy between the specified background bit configurations and the actual value, an error message of the form:

```
ERROR-CORE MODIFIED xx TIMES
LOC          INITIAL    FINAL
(xxxxxx)     xxxxxx     xxxxxx
```

is output at the completion of each pass. Up to 20 such errors can be listed.

Sense switch options are described in section 6.1.

Refer to the applicable system and PF/R maintenance manuals for correction procedures.

Volatile-Registers Test

During this two-pass volatile-registers test, if the program detects a discrepancy between the specified bit configurations and the actual value, an error message of the form:

```
             REGISTER   ERROR
             INITIAL    FINAL
A            xxxxxx     xxxxxx
B            xxxxxx     xxxxxx
X            xxxxxx     xxxxxx
P            xxxxxx     xxxxxx
OF           ON or      ON or
             OFF        OFF
```

is output at the completion of both passes.

For a V75 system, the form of the message for both passes is:

```
     REGISTER
     INITIAL

A XXXXXX XXXXXX
B XXXXXX XXXXXX
X XXXXXX XXXXXX
R3  XXXXXX XXXXXX
R4  XXXXXX XXXXXX
R5  XXXXXX XXXXXX
R6  XXXXXX XXXXXX
R7  XXXXXX XXXXXX
P XXXXXX XXXXXXXX
OFXXXXXX XXXXXXXX
```

Sense switch options are described in section 6.1.

Refer to the applicable system and PF/R maintenance manuals for correction procedures.

Memory Test

During this memory test, the program detects a discrepancy between the specified bit configurations (section 6.1) and the actual value, and error message of the form:

```
ERROR-CODE MODIFIED xx TIMES
LOC        INITIAL   FINAL
(xxxxxx)   xxxxxx    xxxxxx
```

is output at the completion of each pass. Up to 20 such error are listed.

Sense switch options are described in section 6.1.

Refer to the applicable system and PF/R maintenance manuals for correction procedures.

Using the control panel the program halts between power-down/power-up sequences of the test program, the instruction register contains one of the following error codes describing the type of error, the volatile register contents define error conditions.

Error Code	Description
000000	The power-down sequence had insufficient time for completion of execution.
000001	Programmed halt in the halt test to alert the operator to initiate a power-down/power-up sequence.
000002	Error in the halt test using the background value of zero. A register = number of modified words B register = error table address X register = address of the first modified word
000003	Error in the halt test using all ones.
000004	Error in the halt test using 0125252.
000005	Error in the halt test using 052525.
000006	Error in the volatile registers test, first pass A register = type of error 001 overflow 002 A register 004 B register 010 X register 020 P register A composite of the above B register = initial value X register = actual value
000007	Error in the volatile registers test, second pass.
000010	Error in the memory test using the background value of zero. A register = number of modified words B register = error table address X register = address of the first modified word
000011	Error in the memory test using all ones.
000012	Error in the memory test using 0125252.
000013	Error in the memory test using 052525.
000402 to 000776	Interrupt address error. An interrupt executed the instruction at the address defined in bits 0-7 of the instruction register.
000777	Halt for operator input.

Sense switch options (sense switch 1 does not apply) are described in section 6.1.

Refer to the applicable system and PF/R maintenance manuals for correction procedures.

6.4 TEST VALIDATION EXAMPLES

The results presented in this section were extracted from TTY hardcopy collected during validation of the program.

EXAMPLE 1 -- No Errors (For V75 and non-V75 systems)

```
POWER FAILURE/RESTART TEST
TIME DELAY = 10.

HALT TEST
VOLATILE REGISTERS TEST
CORE VALIDITY CHECK
HALT TEST
```

EXAMPLE 2 -- With Errors (V75 system only)

```
POWER FAILURE/RESTART TEST
TIME DELAY = 110.

HALT TEST
CORE VALIDITY CHECK
ERROR-CORE MODIFIED 4 TIMES
LOC        INITIAL   FINAL
(003243)   000000    000011
(003346)   111111    010000
(003455)   125252    000000
(003532)   052525    077777

VOLATILE REGISTER TEST
REGISTER ERROR
           INITIAL   FINAL
A          001504    001500
B          003060    000306
X          000002    000000
P          000532    000533
OF         ON        OFF
CORE VALIDITY CHECK
ERROR-CORE MODIFIED 2 TIMES
(112157)   125252    000000
(003243)   052525    000000
HALT TEST
```

SECTION 7
PRIORITY-INTERRUPT-MODULE TEST

The **Priority Interrupt Module Test** of MAINTAIN III tests the operation of the model 620-16 (7X - 3101) priority interrupt module (PIM).

The PIM establishes eight levels of interrupt priority for selected peripheral device controllers and stores and processes, in order of their priority, interrupt requests from these controllers.

The PIM automatically scans the interrupt lines every 900 nanoseconds or 468 nanoseconds for the 620/L-100. If signals occur on more than one interrupt line, the highest-priority signal is acknowledged. The remaining interrupt requests are stored until each has been acknowledged. The PIM permits any or all of the eight interrupt lines to be enabled or disabled.

Acknowledgement of an interrupt by the CPU executes the instruction at the memory address specified by the PIM. This instruction can be any of the instruction set, excluding I/O instructions. Thus, an interrupt can be serviced in one instruction execution period.

The PIM responds to five external control and three data transfer instructions (table 7-1). A typical PIM service routine is given in table 7-2.

For a detailed description of the PIM, refer to the applicable option manual.

Table 7-1. PIM Input/Output Instructions

Mnemonic	Code	Description
	External Control	
EXC 014x*	010014x*	Clear interrupt registers
EXC 024x	010024x	Enable the PIM
EXC 0244	0100244	Enable all PIMs
EXC 034x	010034x	Clear interrupt registers and enable the PIM
EXC 044x	010044x	Disable the PIM
EXC 0444	0100444	Disable all PIMs
EXC 054x	010054x	Clear interrupt registers and disable the PIM
	Data Transfer	
OME 044x	010304x	Transfer memory to the mask register
OAR 014x	010314x	Transfer A register contents to the mask register
OBR 024x	010324x	Transfer B register contents to the mask register

* x = PIM device address.

Table 7-2. Typical PIM Service Routine

```
STRT ORG     01000
     LDA     MASK       FETCH INT MASK
     OAR     040        STORE IN REG
     LDAI    0377       INIT OUTPUT DATA
     OAR     037        PRIME INT MODULE
     EXC     0240       ENABLE PIM
     NOP
     JMP     *-1        INTERRUPT DELAY
MASK DATA    0376

INTERRUPT PROCESSING SUBROUTINE

INTR ENTR
     DAR                DECR OUTPUT DATA
     OAR     037        DATA TO PUNCH
     EXC     0240       REENABLE PIM
     JAZ     *+4
     JMP     INTR       EXIT
     EXC     0440       CLEAR PIM
     HLT                END OF PROGRAM

INTERRUPT ADDRESS

     ORG     0100
     JMPM    INTR
     END
```

The PIM test program tests four logical phases of PIM operation. The PIM device address and an associated block of 16 interrupt addresses can be selected at run time. Thus, the test is applicable to all PIM device/interrupt address combinations, and, in a system with more than one PIM each can be tested in turn.

The PIM test program consists of six subtests:

Subtest 1 verfies that disabling the mask register inhibits interrupts when the PIM is enabled.

Subtest 2 verifies that interrupts occur at the specified addresses and that the PIM can be enabled.

Subtest 3 verifies that the PIM can be disabled when the mask register is enabled.

Subtest 4 verifies that outstanding interrupts are cleared (i.e., do not occur) by an external control instruction to clear the (interrupt) line register.

Subtest 5 verifies the group disable.

Subtest 6 verifies the group enable.

The PIM test program is designed to test the minimum configuration of a V70/620-series computer with 8K of memory, PIM, and 33/35 ASR Teletype (TTY).

More than one PIM can be included in a system, but only one such device can be exercised at a time.

The PIM test program object format is normally a punched paper tape for loading from the TTY reader or a high-speed paper tape reader. Other media are available (e.g., card object deck).

7.1 INITIAL CONDITION SELECTION

To load the PIM test program:

a. Load the test executive program (section 2).

b. Position the PIM test program tape in the tape reader with leader at the read station.

c. Type L, followed by a period, on the Teletype keyboard.

SENSE switch settings can alter test programs as follows:

Switch	Set	Reset Reset
1	Suppress error message printout	Print error messages
2	Halt on error	Continue testing without halting
3	Terminate testing and return to the test program beginning	Continue testing

If the PIM test program is run on the 620/f or V70 series computer, pressing the INT (interrupt) switch returns control to the test executive.

7.2 EXECUTING THE PIM TEST PROGRAM

The PIM test program operation is performed as follows:

a. The PIM test program outputs the message:

```
PIM TEST
TTY INTERRUPTS
```

b. Type the PIM Device Address, the Read Ready Interrupt Trap, and the Write Ready Interrupt Trap, separated by commas and terminated with a period, or 0, after the message:

```
ENTER PIM DEVICE ADDRESS
```

c. Type the device address of the PIM to be tested, followed by a period. The program then outputs the message:

```
ENTER ORIGIN OF TRAP ADDRESSES
```

d. Type the starting address of the address block (origin) followed by a period.

Subtest 1. After the interrupt addresses are selected the user can select the subtest to run.

a. The program outputs the message:

```
ENTER SUBTEST NUMBER
```

b. For subtest 1 type a one, followed by a period. The program then outputs the message:

```
SET INTERRUPTS
```

NOTE: Do *not* ground the interrupt line associated with teletype receive on V77-200 processors.

c. Momentarily ground selected PIM interrupt lines (IL00-IL07) (refer to the PIM manual) or use an interrupt simulator to set selected interrupts; press the TTY space bar.

d. Step c can be repeated any time during the delay period after the SET INTERRUPTS message. This delay is 5 seconds for the 620/i, or 620/L computers and 2 seconds for the 620/f or V70.

The program executes subtest 1, in which all interrupts are inhibited, and outputs the message:

```
NO INTERRUPTS
```

Subtest 2. For subtest 2 type a 2, followed by a period. The program then outputs the message:

```
ENTER NUMBER OF 5-SECOND INTERVALS
```

This message requires operator input of the delay time he requires to set interrupt requests. The operator has the option to test all lines or selected groups of lines in one or more passes, or repeatedly test a single line, selected groups of lines, or all lines.

The 5-second interval is applicable to the 620/i or 620/L computers. On the 620/f or V70, this interval is approximately 2 seconds.

Set selected interrupts.

Type the desired number of delay intervals, followed by a period.

If a zero is typed, the program will wait for further interrupt simulation until terminated by setting, then resetting, SENSE switch 3.

The program executes subtest 2, and outputs the number of the interrupt line on which an interrupt occurred. If multiple interrupts were set, the line numbers are in order of priority (1 through 8).

Subtest 3. For subtest 3 type a 3 followed by a period. The program then outputs the message:

```
SET INTERRUPTS
```

Set selected interrupts, press the Teletype space bar.

The program executes subtest 3, in which the PIM is disabled and should recognize no interrupts, and outputs the message:

```
NO INTERRUPTS
```

Error conditions are described in section 3.

Subtest 4. To executed subtest 4 type a four, followed by a period. The program then outputs the message:

```
SET INTERRUPTS
```

Set selected interrupts; press the Teletype space bar.

The program executes subtest 4, in which the PIM is disabled and the interrupt line register is cleared, and outputs the message:

```
NO INTERRUPTS
```

Subtest 5. For subtest 5 type a five, followed by a period. The program then outputs the message:

```
ENTER NUMBER OF 5-SECOND INTERVALS
```

This message requires operator input of the delay time he requires to set interrupt requests. The operator has the option to test all lines or selected groups of lines in one or more passes, or repeatedly test a single line, elected groups of lines, or all lines.

The 5-second interval is applicable to the 620/i or 620/L computers. On the 620/f or V70, this interval is approximately 2 seconds.

Set selected interrupts.

Type the desired number of delay intervals, followed by a period.

If a zero is typed, the program will wait for further interrupt simulation until terminated by setting, then resetting, SENSE switch 3.

The program executes subtest 5 and outputs the number of the interrupt line on which an interrupt occurred. If multiple interrupts were set, the line numbers are in order of priority (1 through 8).

Subtest 6. For subtest 6 type a six followed by a period. The program then outputs the message:

```
SET INTERRUPTS
```

Set selected interrupts, press the Teletype space bar.

The program executes subtest 6 in which the PIM is disabled and should recognize no interrupts and outputs the message:

```
NO INTERRUPTS
```

Error conditions are desribed in section 3.

When the subtests have been executed, set SENSE switch 3. This returns operation to the beginning of the PIM test program and restores the contents of memory in the selected interrupt addresses to pre-testing status.

The program can be executed again to test another PIM, or control can be returned to the test executive.

For 620/i and 620/L computers, perform the following steps to terminate the PIM test programs and to return to the test executive:

a. Clear the instruction register.

b. Load 014000 in the P register.

c. Press SYSTEM RESET.

d. Press RUN two times.

For 620/f or V70 computers, return control to the test executive by pressing the INT switch.

To return to the PIM test program from the test executive, type:

```
G500.
```

The value 500 represents the starting address of the PIM test program.

7.3 ERROR INDICATIONS

The Teletype provides the following error responses. If an incorrect trap address block origin is typed in response to the ENTER ORIGIN OF TRAP ADDRESS message, the PIM test program outputs the message:

```
INVALID INTERRUPT
```

and halts with 000004 in the instruction register. Press START or RUN to continue testing.

During subtest 1, 3, 4, and 6 the message:

```
NO INTERRUPTS
```

indicates successful execution of these subtests. If interrupts occur, however, the PIM test program prints out on the Teletype the number of the interrupt line(s) on which an interrupt was detected, e.g., 12345678.

Refer to the applicable system and PIM maintenance manuals for correction procedures.

On subtest 2 if the printout of interrupt line numbers does not correspond to the interrupts actually simulated during this subtest, refer to the applicable system and PIM maintenance manuals for correction procedures.

When SENSE switch 3 is not reset after returning to the beginning of the PIM test program, the message:

```
RESET SENSE SWITCH 3
```

is output. Reset the switch to continue testing.

7.4 TEST VALIDATION EXAMPLES

The results presented in this section are from Teletype printed output collected during validation of the program.

```
PIM TEST

ENTER PIM DEVICE ADDRESS 40.

ENTER ORIGIN OF TRAP ADDRESSES 120.

ENTER SUBTEST NUMBER 1.
SET INTERRUPTS
NO INTERRUPTS
```

The program correctly reported that no interrupts occurred.

```
ENTER SUBTEST NUMBER 2.
ENTER NUMBER OF 5 SECOND INTERVALS 2.
12345678
```

Interrupts were raised on all eight lines and line numbers correctly reported (in order of priority).

```
ENTER SUBTEST NUMBER 2.
ENTER NUMBER OF 5 SECOND INTERVALS 2.
12345678666666666
```

Subtest 2 was again run; additional interrupts were raised on line 6 during the delay interval.

```
ENTER SUBTEST NUMBER 3.
SET INTERRUPTS
NO INTERRUPTS

ENTER SUBTEST NUMBER 4.
SET INTERRUPTS
NO INTERRUPTS

ENTER SUBTEST NUMBER
PIM TEST
```

SENSE switch 3 was set, then reset, to return to the beginning of the program and restore the contents of memory.

```
ENTER PIM DEVICE ADDRESS 40.

ENTER ORIGIN OF TRAP ADDRESSES 100.

ENTER SUBTEST NUMBER 2.
ENTER NUMBER OF 5 SECOND INTERVALS 1.
INVALID INTERRUPT
```

An incorrect interrupt address origin was specified and an interrupt raised on line 1, producing the **INVALID INTERRUPT** message.

```
ENTER SUBTEST NUMBER
PIM TEST
RESET SENSE SWITCH 3
RESET SENSE SWITCH 3
```

Sense switch 3 was set to return to the beginning of the test, but was not then immediately reset so that testing could continue.

```
ENTER PIM DEVICE ADDRESS 40.

ENTER ORIGIN OF TRAP ADDRESSES 120.

ENTER SUBTEST NUMBER 2.
ENTER NUMBER OF 5 SECOND INTERVALS 2.
12342345678

ENTER SUBTEST NUMBER 4.
SET INTERRUPTS
12345678
```

Interrupts were raised during the delay interval in subtest 4, resulting in the printout of interrupt line numbers instead of the correct message: `NO INTERRUPTS`.

SECTION 8
REAL-TIME CLOCK TEST PROGRAM

The Real-Time Clock (RTC) on the V70/620 series computers generates interrupts at a specified rate. On the V70 series and 620/f, this rate is variable under program control. In addition, the 70 and 620/f models RTC drives a readable 16-bit free-running Counter. The purpose of the RTC test program will be to provide the user with an interface to evaluate the performance of these features of the Real-Time Clock.

The RTC test program has two main goals. The first is to provide output with which the user can validate the correct operation of the features of the RTC. The second is to operate in as many environments as the RTC is found while interfacing with the user as simply as possible.

A software timer which could validate correct RTC operation would have been the ideal solution to the first goal. Due to variations in cycle time, however, such a software timer would be very CPU model sensitive. Thus, in interest of the second goal, an alternative method was adopted. The RTC output will be translated as directly as possible into output which the user can evaluate by checking against an external time source, such as a stop watch.

8.1 FUNCTIONAL CAPABILITIES

The RTC test program will provide two main services. First, an I/O instruction and interrupt test will be run. This will check the correct functioning of the RTC-oriented I/O instructions. The test will also verify that RTC interrupts are occurring, though it makes no attempt to time them or interpret them. The second test will allow timing of the interrupts. This is done by using the interrupts to drive an elapsed-time counter and an interval timer. By comparing their outputs with an external time source, RTC performance can be evaluated.

I/O INSTRUCTIONS

V70 and 620/f Instructions

Mnemonic	Octal	Function	Description
EXC 0147	100147	Enable RTC	Enables both variable interval interrupts and overflow interrupts.
EXC 047	100047	Clear Free Running Counter (FRC)	The only way to clear the FRC.
EXC 0447	100447	Inhibit RTC (Initialize)	Inhibits all interrupts; resets interrupts register and divide-by-eight counter.
EXC 0247	100247	Inhibit Overflow	Inhibits only overflow interrupts
EXC 0347	100347	Enable Increment/Inhibit Overflow	Enables variable interval interrupts; inhibits overflow interrupts
EXC 0647	100647	Initialize Variable Interval Interrupt (VII) counter	Loads (VII) counter from Interval Select Register
EXC 0747	100747	Inhibit Variable Interval Interrupt (VII)	Disallows VII's
OAR 047	103147	Output to Interval Select Register	
OBR 047	103247		
*OME 047	103047		
INA 047	102147	Input FRC	
INB 047	102247		
*IME 047	102047		

Mnemonic	Octal	Function	Description
CIA 047	102547	Clear and input	
CIB 047	102647	FRC	

* These commands should not be used if the RTC is used in a system containing the PMA option.

620-i and L Instructions

EXC 0147	100147	Enable RTC	Enables both incrementation and overflow interrupts.
EXC 0447	100447	Inhibit RTC (Initialize)	Inhibits all interrupts: resets interrupt register and divide-by-eight counter
EXC 0247	100247	Inhibit Overflow	Inhibits only overflow interrupts
EXC 0347	100347	Enable Increment/ Inhibit Overflow	Enables incrementation interrupts; inhibits overflow interrupts.

8.2 HARDWARE SUMMARY

8.2.1 Major Modules and Performance

The Real-Time Clock (RTC) Test Program tests the real-time clock mainframe option for the V70 and the 620/i, 620/L, 620/f series computers. The following RTC functions are exercised:

a. On the V70 and 620/f:

1. The Variable-Interval Interrupt (VII)
2. The Memory-Overflow Interrupt (MOI)
3. The Free-Running Counter (FRC)

b. On the 620/i, 620/L:

1. The Interval-Interrupt (II)
2. The Memory-Overflow Interrupt (MOI)

c. On the V77

1. The Internal Interrupt (II)

8.2.1.1 Free-Running Counter (V70, 620/f)

The free-running Counter (FRC) is a 16-bit counter that is continually updated and can be read under programmed I/O control. The clock for the FRC is hardwired selectable and can either be the Line Frequency Source (60 Hz, at 16.7 milliseconds, 50 Hz at 20.0 millisecond, or 10 KHz at 100 microsecond), the external source supplied by the customer, or the variable-interval rate. The counter can only be reset by the clean free-running Counter (EXC 047) command and will continue to count when the 620/f is in the step mode. Source will be the line frequency unless otherwise specified by the customer.

8.2.1.2 Variable Interval Interrupt (V70, 620/f)

The variable-interval interrupt (VII) memory-address interrupt is 044. The interrupt rate is selectable under programmed I/O control. The formula for calculating the rate is:

$$\text{variable-interval rate} = \frac{\text{Source Frequency}}{\text{Selected Count}}$$

The source is hardwired selectable and can either be a 10 KHz source derived from a crystal controlled oscillator, a line frequency source derived from the power supply (50 or 60 Hz) or an external source supplied by the customer. The selected count can be any count from 1 to 4095 and is selectable by software. The count is hardware preset to 0012 upon initialization. Source will be 10 KHz unless otherwise specified by the customer.

8.2.1.3 Interval Interrupt (620/i, 620/L)

The Interval Interrupt (II) memory-address interrupt is 044. The interrupt rate is normally 1 interrupt in a millisecond. The external source may be supplied by the customer.

8.2.1.4 Memory-Overflow Interrupt (All CPU's)

The Memory-Overflow interrupt (MOI) memory address interrupt is 046. This interrupt is used in conjunction with the (Variable) Interval Interrupt. An Increment Memory and Replace instruction is put in the (Variable) Interval Interrupt address and the Memory-Overflow logic monitors the selected memory location. When the memory location is incremented to 040,000 by the (V)II, the overflow interrupt request will occur after the next (Variable) Interval Interrupt request. The memory location will contain a count of 040,001 when the Memory-Overflow Interrupt request occurs. If RTC interrupts are disabled on the V70 or 620/f, any interrupt requests that would normally occur will be saved and the CPU will receive an interrupt request for each interrupt type that has had a request when the interrupts are re-enabled. On the 620/i and 620/L, only the first II and first MOI will be saved if the RTC interrupts are disabled.

8.2.2 Configurations

The minimum configuration for the RTC test is 8K memory and one of the following:

a. Model 700X CPU

b. 620/f-10X CPU or 620/f-00X CPU

c. 620/L-10X CPU or 620/L-00X CPU

d. 620/i CPU with RTC option (620-13)

e. 620-06, -08 Teletype

8.3 SOFTWARE DESIGN SUMMARY

The Real-Time Clock Test consists of two parts, one testing the basic I/O instructions and interrupts and another for interrupt timing.

8.3.1 I/O Instruction and Interrupt Test

This test is executed once upon entrance to the RTC test. All RTC I/O instructions and the computers ability to detect (variable) interval interrupts and memory overflow interrupts are verified. Upon detection of an error, the computer will either halt with an error code in the instruction register or print an error message. This test must be passed before executing the interrupt timing test.

8.3.2 Interrupt Timing Test

The test program will request that the operator specify the selectable hardwire connections (for the free-running counter on the V70 and 620/f) and for the (Variable) Interval Interrupt.

The performance of the RTC may be checked in two ways. First, either the FRC or the (V)II may be used to drive an interval timer. Second, both may be used to run an elapsed time counter. These produce outputs which may be compared to an external time source for checking RTC performance. No software timing checks are included in this test program.

8.4 USER FACILITIES

8.4.1 Interval Timer

The interval timer will signal the user every 'n' seconds, where 'n' is the current display interval. The time for groups of signals can be measured with a stop watch and an estimate made on RTC performance. On 620 series computers, the interval timer signals the user by complementing the overflow light. On the V70 series computers, the 16 data lights on the control panel are complemented.

In addition to ringing the bell, the overflow light is complemented. Thus, a signal is visible when operating without a Teletype. Finally, since the V70 series has no overflow light, its console lights will be complemented.

The display-interval may be varied, but may be no greater than the number of seconds equivalent to the capacity of the interval timer. The capacity is 040,000 interrupts (= 16,384 interrupts). Thus,

Interrupts Per Second	Maximum Display Interval
10,000	1 second
1,000	16 seconds
60	273 seconds

The test program checks the range of the display period when input and will signal if it is too large.

8.4.1.1 Interval Timer Accuracy

The interval timer has an accuracy of ± 1 interrupt per interval at best. Thus, for a VII at 10,000 interrupts per second and a select count of 1, this inaccuracy is only one-ten thousandths of a second. However, with a select count of 4095, this changes to an accuracy of about ± one-half second. Thus, using the interval timer to time 1 second intervals would produce gross error in the latter case. In general, when the number of interrupts per second is small, (e.g., large VII select count), a long display period is best.

8.4.2 Elapsed Time Counters

The elapsed time counters maintain a total of elapsed minutes and seconds since the beginning of the interrupt-timing test. They run at the same time as the interval timer

but produce no external display unless requested. When a request is made, the current elapsed time is computed and output. During this computation, the interval timer takes second priority and thus may miss intervals. Shortly after the elapsed time has been output, the interval timer will return to normal operation.

The elapsed time counter may also be requested to reset its counters. This will also restart the interval timer.

On the 620-i, 620/L, the interval interrupt drives the elapsed time counter and its current value is output on request. On the V70 and 620/f, both the variable interval interrupt and the free running counter drive elapsed time counters and thus two values are output when a request is made.

8.5 LOADING PROCEDURE

The test executive must be loaded before the real-time clock test program will operate correctly. Teletype input/output subroutines resident in the test executive are called by the RTC program.

a. Load the test executive, which includes the binary object tape loader, per the procedure outlined in section 2.

b. The real-time-clock-test program tape contains the test part number punched in leader. Position the tape past this area at the read station.

c. Type L on the keyboard, followed by a period, to command the test executive to load the tape.

8.6 OPERATING INSTRUCTIONS

The execution of this test is performed by the use of the Teletype interface. The real-time clock test program requires the operator to supply all optional parameters.

For systems that do not contain a paper-tape unit, test programs will be loaded via the available object input device (card reader, magnetic tape, etc.).

8.6.1 Initial Condition Selection

Switch	Set	Reset
SS1	Suppress error printouts	Print error messages
SS2*	Halt on error (Continue after error halt).	Do not halt on error/ loop after error halt.
SS3	Terminate test and return to beginning of test program.	Continue test

* SS2 can be used to loop on an error following an error halt, or to continue the test following the halt:

a. To continue to the next error halt, keep SS2 set and press START on the computer.

b. To loop on the error condition, reset sense switch 2 and press START on the computer. Looping will continue until sense switch 2 is set, then the program continues to the next error halt.

8.6.2 Mode of Operation

8.6.2.1 I/O Instruction and Interrupt Test

The Real Time Clock test starts by printing the following message:

```
REAL TIME CLOCK TEST
RTC TYPE =
```

The Teletype printer then pauses after the message and waits for the user to input the number indicating the CPU type =; i.e., 0 for 620/i, 620/L or 1 for V70 or 620/f, 2 for V77-200/400, or 3 for V77-800. ←

```
I/O INSTRUCTION AND INTERRUPT TEST
```

The following messages are printed after testing each option of the real time clock for the V70 and 620/f.

```
VARIABLE INTERVAL INTERRUPT CHECK
MEMORY OVERFLOW INTERRUPT CHECK
FREE RUNNING COUNTER CHECK
```

For the other CPU's, the following messages are printed after each option is tested:

```
INTERVAL INTERRUPT CHECK
MEMORY OVERFLOW INTERRUPT CHECK
```

If any errors are noted, the following message is printed:

```
ERROR NO. = x where x is a number from 1 to 12.
```

If the test is being run in the console mode of operation, a halt is executed with the error code in the instruction register (section 8.3).

The I/O instruction and interrupt test must be passed before the test can be continued.

8.6.2.2 Input of Hardware Parameters

Upon completion of the I/O instruction and interrupt test, the hardware parameters must be defined by the operator.

For the V70 and 620/f, the program requests:

```
INPUT FRC INCREMENTS PER SECOND
```

The operator inputs the decimal number followed by a period. The correct value depends on the hardwired selectable FRC source. The following are acceptable inputs and their corresponding sources. The first is the standard value.

SOURCE	FRC INCREMENTS PER SECOND
Crystal Controlled Oscillator	10000
Line Frequency	60 (or 50 for 50 Hz)
Customer's External Source	Appropriate Value
Variable Interval Rate	Basic interrupts per second divided by select count.

For all CPU's, the test program will request:

```
INPUT BASIC INTERRUPTS PER SECOND
```

The operator inputs the decimal number followed by a period. The correct value depends on the hardwired selectable clock source. The following are acceptable inputs and their corresponding sources. The first is the standard value.

For V70 and 620/f:

SOURCE	BASIC INTERRUPTS PER SECOND
Crystal Controlled Oscillator	10000
Line Frequency	60 (or 50 for 50 Hz)
Customer's External Source	Appropriate Value

For 620/i, 620/L:

SOURCE	BASIC INTERRUPTS PER SECOND
Standard Source	1000
Customer's External Source	Frequency (in Hz) divided by 8.

For V77-800:

SOURCE	BASIC INTERRUPTS PER SECOND
Crystal Controlled Oscillator	25,50,100,or 200 (50 is standard)

8.6.2.3 Interrupt Timing Test Inputs

After the RTC hardware setup has been defined, the interrupt-timing test is begun. The test types 'INTERRUPT TIMING TEST' to identify itself and then requests the test parameters.

For the V70 and 620/f the following requests are made:

```
INTERVAL TIMER =
```

Typing '0.' will result in the FRC driving the interval timer. Typing '1.' will result in the VII being used instead.

```
VII SELECT COUNT =
```

This count is used to vary the VII rate. The user should type in a decimal number from 1 to 4095, followed by a period. 10 is the standard value. The hardware default value of 10 may be tested by entering a zero followed by a period.

```
INTERVAL DISPLAY PERIOD IN SECONDS =
```

This sets the number of seconds to be measured by the interval timer. Type in a decimal number followed by a period. If the number typed exceeds the interval timer capacity, 'unacceptable' will be typed out and the request repeated.

For the 620/i and 620/L, only the 'interval display period' request will be made out of the above three questions. This is because the interval interrupt alone is available on those CPU's for timing intervals. Also, those CPU's do not permit varying the interval interrupt with the select count.

8.6.2.4 Interrupt Timing Test Execution

Once the above initialization has been completed, the test program outputs

```
BEGIN TEST
```

and starts the interval timer and elapsed time counters. During execution, communication through the Teletype is in the following manner:

CHARACTER TYPED	RESULT
Space	Values of elapsed time counters typed
R	All counters and timers reset
K	Return to initialization
Any other character	No effect

The test continues until interrupted by sense switch settings or console interrupt.

The format for the elapsed-time printout is as follows:

For V70 and 620/f:

```
(V)II:     x MIN.     y SEC.
FRC:       x MIN.     y SEC.
```

For 620/i, 620/L:

```
(V)II:     x MIN.     y SEC.
```

8.7 SUMMARY OF TELETYPE/PRINTER OUTPUT STATEMENTS

V70 and 620/f Messages

```
REAL TIME CLOCK TEST
RTC TYPE =
I/O INSTRUCTION AND INTERRUPT TEST
VARIABLE INTERVAL INTERRUPT CHECK
MEMORY OVERFLOW INTERRUPT CHECK
FREE RUNNING COUNTER CHECK

INPUT FRC INCREMENTS PER SECOND
INPUT BASIC INTERRUPTS PER SECOND

INTERRUPT TIMING TEST
INTERVAL TIMER =
VII SELECT COUNT =
INTERVAL DISPLAY PERIOD IN SEC =
BEGIN TEST

FRC: x-xx MIN, y-yy SEC
(V)II: x-xx MIN, y-yy SEC
UNACCEPTABLE
```

620-i, 620/L Messages

```
REAL TIME CLOCK TEST
RTC TYPE =
I/O INSTRUCTION AND INTERRUPT TEST
INTERVAL INTERRUPT CHECK
MEMORY OVERFLOW INTERRUPT CHECK

INPUT BASIC INTERRUPTS PER SECOND

INTERVAL TIMING TEST
INTERVAL DISPLAY PERIOD IN SEC =
BEGIN TEST

(V)II: x-xx MIN, y-yy SEC
UNACCEPTABLE
```

8.8 SUMMARY OF TELETYPE INPUT STATEMENTS

```
RTC TYPE = x
```

Where x = 0 for 620-i, 620/L
= 1 for 620/f and V70

```
INPUT FRC INCREMENTS PER SECOND
x-xxx.

INPUT BASIC INTERRUPTS PER SECOND
x-xxx.
```

where x-xxx is from 1 to 10 decimal digits, followed by a period.

```
INTERVAL TIMER = x.
```

Where x = 0 for FRC
= 1 for VII

```
VII SELECT COUNT = xxxx.
```

Where xxxx, is a decimal number followed by a period, from 0 to 4095.

```
INTERVAL DISPLAY PERIOD IN SECONDS = xxx.
```

Where xxx. is a decimal number, followed by a period.

8.9 ERROR INDICATIONS

ERROR CODE DESCRIPTION (Error code is in the instruction register)

ERROR CODE	DESCRIPTION
1	Initialized RTC and enabled RTC interrupt did not cause a (Variable) Interval Interrupt.
2	Inhibit (Variable) Interval Interrupt did not inhibit interrupt.
3	Initialize Variable Interval Interrupt and enable (Variable) Interval Interrupt did not cause interrupt.
4	Initialize RTC and enable RTC with increment and replace count of 037775 did not cause an overflow interrupt.
5	Upon receiving MOI the memory count value was not 040001.
6	Inhibit MOI did not inhibit MOI interrupt.
7	Inhibit MOI inhibiting (V)II also.
10	Enable (V)II and inhibiting MOI not inhibiting MOI.
11	Free running counter not incrementing.
12	Clear FRC not clearing counter.

8.10 ERROR HALT DESCRIPTIONS

Instruction Register	Description
000400 to 00777	Illegal interrupt to a non-real-time clock interrupt address. 000 to 0377; A, B, and X have no meaning.
000	Console-mode halt A = 0 Operator set A register to 1 for 73 and 620/f or leaves as 0 for the other CPU's.
020	Console-mode halt. X = B = A = 0 The operator stores the following in each register and pushes run. A, B = Double-precision free running counter increments per second.
021	Console-mode halt. X = B = A = 0 The operator stores the following in each register and pushes run. A, B = Double-precision (Variable) Interval Interrupts per second.
022	Console-mode halt X = B = A = 0 Initialization for interrupt timing test operator stores the following in each register and pushes run: A = Interval timer B = VII Select count X = Display period.
023	Console-mode halt for communication with elapsed-time counters. A, B registers contain the elapsed time.
001 to 012	I/O Instruction and Interrupt Test error. A = Error Code No. (01 to 012) (see 2.6 for error description) B = Location calling error routine X = 0

SECTION 9
620/f AND V70 MEMORY-PROTECTION TEST PROGRAM

The Memory Protection Test Program of MAINTAIN III tests the operation of the V70 system and 620/f memory protection (MP) option, which is not applicable to other 620-series computers.

The MP partitions core memory so that the contents of certain memory areas (designated protected areas) cannot be altered by programs operating in unprotected areas. Memory is partitioned into equal blocks of 512 words. A 4,096-word memory increment is divided into eight such blocks. Each area can then be selectively designated protected or unprotected.

When a program is operating from an unprotected area, the following operations are prohibited:

a. Writing in a protected area

b. Jumping to a protected area

c. All I/O instructions from an unprotected area

d. Program overflow into a protected area

e. Executing a halt instruction

If these operations are attempted, the program aborts and jumps to one of eight preassigned memory addresses. From these addresses, the program can be directed to a user-written subroutine for analysis and correction.

Programs operating from a protected area of memory do not have the above-described limitations.

For a detailed description of the MP, refer to the appropriate maintenance manual.

> The MP test program is designed to test ONLY the **620/f and V70 MP** options.

9.1 PROGRAM DESIGN SUMMARY

The MP test program consists of two subtests:

a. Mask-register test

b. Instruction interrupt address test

9.1.1 Mask-Register Test

This test verifies that the MP establishes protected and unprotected areas in memory. The MP contains one 16-bit mask register for each 8,192 words of memory. Each mask register bit controls 512 words. If the mask register bit is zero, the corresponding 512-word area is protected; if one, unprotected. Mask register 0 controls the lowest-order 8,192 words of memory, and mask register 3, the highest-order 8,192 words. This test is identical for the 620/f and V70.

The mask register test executes the following seven subtests on each 512-word memory block.

a. Enable MP

b. Disable MP

c. Set mask registers

d. Reset mask registers

e. High block boundary

f. Low block boundary

g. Instruction address register

At the beginning of each block test, memory addresses to be modified in the test are saved; they are restored at the conclusion of a block test. Interrupt addresses contain Jump and Mark (JMPM) instructions to an error-reporting subroutine, except programmed interrupts.

The mask Register Test assumes that the memory-protect Jump-error detection is working properly.

9.1.2 Instruction Interrupt Address Test

This test verifies that the MP detects invalid operations and initiates appropriate interrupt action. Invalid operations and their solutions are:

a. **Write Error.** Data cannot be stored in a protected area. If this is attempted, the write instruction is modified to a read instruction to protect memory, the A, B, and instruction registers are unchanged, and the program executes a JMPM to the error-processing subroutine at address 000024 (or 000034 if overflow also exists).

b. **Jump Error.** When a program is operating from an unprotected area, it cannot execute a Jump (JMP) instruction to a protected area. If this is attempted the P register remains unchanged, and, if the instruction is a JMPM, the write instruction is modified to a read instruction. The program then executes a JMPM to address 000026 (or 000036 if overflow also exists).

c. **I/O Error.** If execution of an I/O instruction is attempted from an unprotected area, the I/O instruction is inhibited, and the program executes a JMPM to address 000022 (or 000032 if overflow also exists).

d. **Overflow Error.** The P register cannot be incremented across an unprotected-to-protected boundary:

   (1) To address the next instruction

(2) To address the second word of a two-word instruction

In the first case, the instruction is not executed, and the program executes a JMPM to address 000030.
In the second case, if the instruction is not a write or JMP, it is executed, and the program executes a JMPM to address 000030. A write instruction is not executed, and the program executes the JMPM. If the instruction is a JMP, the JMP address is not transferred to the P register, and the program executes the JMPM.

e. **Halt Error.** If a Halt instruction is executed from a location in an unprotected area or if execution of a Halt instruction located in any area is attempted via an execute instruction which is located in unprotected core, a Halt Error condition exists. When a Halt Error is detected, the Halt instruction is allowed to complete after which the CPU is interrupted to location 020.

9.2 620/f-V70 MEMORY-PROTECTION DIFFERENCES

The following differences in the Instruction Interrupt Address Test exists between the 620/f and V70.

Test 1:	For the 620/f, the address saved at the halt interrupt location (020) is the address of the executed HLT instruction plus 1. For the V70, the address is that of the HLT instruction.
TEST 10:	For the 620/f, the expected type of interrupt is an overflow interrupt. For the V70, the expected type of interrupt is a Halt interrupt.
Test 35:	This test is optional for the 620/f and standard for the V70
Test 37:	This test is optional for the 620/f and standard for the V70.

9.3 SYSTEM CONFIGURATION

The MP test program is designed to test the minimum configuration of a 620/f or V70 series computer with 8K or memory (maximum, 32K), MP, and 33/35 ASR Teletype (TTY).

The MP test program object format is normally a punched paper tape for loading from the TTY reader or a high-speed paper tape reader. Other media are available (e.g., card object deck).

9.4 PRELIMINARY PROCEDURES

To load the MP test program:

a. Load the test executive program (section 2).

b. Position the MP test program tape in the tape reader with leader at the reader station.

c. Type L, followed by a period, on the TTY keyboard,

or

Load zero in the A register and 07600 in the I register, set RESET, and, in run mode, press START.

SENSE Switch Options

SENSE switch settings can alter test program execution as follows:

Switch	Set	Reset
1	Suppress error message printout	Print error messages
2	Halt on error (continue testing after error halt)	Do not halt on error (loop after error halt)
3	Terminate testing and return to the test program beginning	Continue testing

To continue the test after an error halt, set SENSE switch 2 and press START.

To loop on an error, reset SENSE switch 2 after an error halt and press START. Looping continues until SENSE switch 2 is again set.

V70/620/f INT Switch

Pressing the INT (interrupt) switch returns control to the test executive.

9.5 OPERATING THE MEMORY-PROTECTION TEST PROGRAM

9.5.1 Mode of Operation

After successful loading of the MP test program:

a. The test program outputs the message:

```
MEMORY PROTECT TEST
ENTER CPU TYPE 0 = 620/f 1 = V70
       2 = V77
```

b. If 620/f is specified, the following message is output:

```
OPTIONAL INST. PRESENT 0 = YES 1 = NO
```

If testing the 620/f containing the optional instruction set (document number 98 A 9908 430), type a 0, followed by a period. If the 620/f does not contain the optional instruction set, type a 1, followed by a period.

c. The program then outputs the message:

```
START TEST 0. =MASK REG. OR 1. =INST TEST
```

d. Type a 0, followed by a period, to execute the mask register test or type a 1, followed by a period, to execute the instruction-interrupt-address test first.

e. The program then outputs the message:

```
CYCLES =
```

f. Type one of the following:

Response	Description
A period	Specifies continuous testing and suppression of nonerror messages
A comma	Specifies continuous testing and output of nonerror messages
An octal number, followed by a period	Specifies test termination after the designated number of cycles and suppression of nonerror messages
An octal number, followed by a comma	Specifies test termination after the designated number of cycles and output of nonerror messages

With each response, the test can be terminated at the completion of the current cycle by setting SENSE switch 3. The maximum number of cycles that can be specified is 077777 (32,768 decimal). Zero specifies continuous test execution.

g. The program then outputs the message:*

```
MASK REG. TEST
```

h. The mask register test is automatically executed and, if nonerror message output is not suppressed, the program outputs the message:

```
INST. INT. ADDR. TEST
```

i. The instruction interrupt address test is automatically executed, and, when the requested number of cycles of the complete test are complete, the program outputs the message:

```
MP TEST COMPLETE
CYCLES =
```

* If the test is started with the INST. TEST, the message outputs of steps g and h will be reversed.

9.6 ERROR INDICATIONS

9.6.1 Mask-Register Test

If, during the execution of this test, an error is detected by the MP test program, a message of the following form is output:

ERROR BLOCK = xx TYPE = n

where

xx = one of the 512-word memory blocks
- 01 = addresses 000513-001024
- 02 = addresses 001025-001536
- 03 = addresses 001537-002048
- etc.

n = one of the following error conditions:
- 1 = bottom of block boundary test failed
- 2 = top of block boundary test failed
- 3 = interrupt occurred when disabled in unprotected area
- 4 = interrupt occurred when disabled in protected area
- 5 = interrupt occurred when enabled in protected area
- 6 = interrupt did not occur when enabled in unprotected area
- 7 = incorrect address in the instruction address register

The MP test restarts at the beginning if an error is detected. To test the failing block in a loop, refer to section 2.1.2 for the appropriate SENSE switch settings. In an error loop, all memory addresses from 000002 through 000200 are set to the error address with bit 8 set (except MP interrupt addresses). Thus, interrupts attempting to execute instructions at these addresses result in a "fatal" error halt.

9.6.2 Instruction Interrupt Address Test

If, during the execution of this test, the MP test program detects an error, a message of the following form is output:

ERROR TYPE = xnn

where

x = one of the following conditions:
- 0 = expected interrupt or condition not present
- 1 = test executed correctly, but the interrupt address not correct

nn = one of the following error conditions:

Error Code	Test Description	Expected Result
01	Execute a HLT instruction in an unprotected address	Interrupt to halt address
02	Execute a HLT instruction in the last address of an unprotected memory block	Interrupt to halt address
03	In unprotected memory, execute a HLT instruction using an XEC instruction in an unprotected address	Interrupt to halt address
025	In unprotected memory, execute a write instruction that modifies a protected address using an XEC instruction in unprotected memory	Interrupt to write address
026	In protected memory, execute a write instruction that modifies an unprotected address using an XEC instruction in unprotected memory	No interrupt
027	Execute a one-word write in protected memory instruction with the instruction in the last address of unprotected memory	Interrupt to write/ overflow address
030	In unprotected memory, execute a JMP instruction to protected memory	Interrupt to jump address
031	In unprotected memory, execute a JMPM instruction to protected memory	Interrupt to jump address
032	In unprotected memory, execute a JMP instruction with the first word in the last address of unprotected memory	Interrupt to jump/ overflow address
033	In unprotected memory, execute a IJMP instruction to protected memory	Interrupt to jump address
034	In unprotected memory, execute a jump and set return in B instruction (JSR) that causes a jump to protected memory.	Interrupt to jump address
035	In unprotected memory, execute a BT instruction that causes a jump to protected memory (optional test for 620/f)	Interrupt to jump address

Error Code	Test Description	Expected Result
036	Check the interrupt address return location and the instruction address register after a jump error	Interrupt to jump address
037	Execute an SRE instruction with the fourth word in the last address of unprotected memory; the skip exit is taken (optional test for 620/f)	Interrupt to jump address
04	Execute a non-I/O, nonstore, one-word instruction in the last address of an unprotected memory block	Interrupt to overflow address
05	Execute a non-I/O, nonstore, two-word instruction with the second word in the last address of an unprotected memory block	Interrupt to overflow address
06	Execute a non-I/O, nonstore, two-word instruction with the first word in the last address of an unprotected memory block	Interrupt to overflow address
07	Execute a JMP instruction with the jump condition not met and the second word in the last address of an unprotected memory block	Interrupt to overflow address
010	Execute an XEC instruction that executes a halt in unprotected memory and with the second word in the last address of an unprotected memory block	1. Interrupt to overflow address for 620/f 2. Interrupt to halt address for 73 system
011	Execute an INRE (indirect) through protected-to-unprotected memory with the second word on INR in the last address of an unprotected memory block	Interrupt to overflow address
012	Execute a two-word extended read instruction (non-INR) with the second word in the last address of unprotected memory	Interrupt to overflow address
013	Execute an extended write to unprotected memory instruction with the second word of the instruction in the last address of unprotected memory	Interrupt to overflow address

Error Code	Test Description	Expected Result
014	Execute a one-word I/O instruction in unprotected memory	Interrupt to I/O address
015	Execute a one-word I/O instruction in unprotected memory using an XEC instruction in protected memory	No error
016	Execute a one-word I/O instruction in unprotected memory using an XEC instruction in unprotected memory	Interrupt to I/O address
017	Execute a one-word I/O instruction in protected memory using an XEC instruction in unprotected memory	Interrupt to I/O address
020	Execute a two-word I/O instruction with the second word in the last address of unprotected memory	Interrupt to I/O overflow address
021	In unprotected memory, execute a one-word write instruction that writes in protected memory	Interrupt to write address
022	In unprotected memory, execute a two-word write instruction that writes in protected memory	Interrupt to write address
023	In unprotected memory, execute a one-word write instruction that writes in unprotected memory	No interrupt
024	In unprotected memory, execute a two-word write instruction that writes in unprotected memory	No interrupt

9.7 TEST VALIDATION EXAMPLES

The results represented in this section were extracted from TTY hardcopy collected during validation of the program.

EXAMPLE 1 on 620/f

```
MEMORY PROTECT TEST
ENTER CPU TYPE 0 = 620/f     1 = V70
0.
OPTIONAL INST. PRESENT   0 = YES,      1. = NO
0.
START TEST      0. = MASK REG. OR 1. = INST. TEST
0.
CYCLES = 1,
MASK REG. TEST
INSTR. INT. ADDR. TEST
MP TEST COMPLETE
CYCLES = 2,
MASK REG. TEST
INSTR. INT. ADDR. TEST
MASK REG. TEST
INSTR. INT. ADDR. TEST
MP TEST COMPLETE
CYCLES =.
```

EXAMPLE 2

```
MEMORY PROTECT TEST
ENTER CPU TYPE 0 = 620/f     1 = V70
0.
OPTIONAL INST. PRESENT   0. = YES,     1. = NO
1.
START TEST      0. = MASK REG. OR 1. = INST. TEST
1.
CYCLES = 2,
INSTR. INT. ADDR. TEST
MASK REG. TEST
INSTR. INT. ADDR. TEST
MP TEST COMPLETE
CYCLES =
```

EXAMPLE 3

Running the test on a 620/f but specifying V70.

```
MEMORY PROTECT TEST
ENTER CPU TYPE 0 = 620/f     1 = V70
1.
START TEST      0. = MASK REG. OR 1. = INST. TEST
0.
CYCLES = 1,
MASK REG. TEST
INSTR. INT. ADDR. TEST
ERROR TYPE = 000101
ERROR TYPE = 000010
MP TEST COMPLETE
CYCLES =.
ERROR TYPE = 000101
ERROR TYPE = 000010
ERROR TYPE = 000101
ERROR TYPE = 000010
ERROR TYPE = 000101
ERROR TYPE = 000010
ERROR TYPE = 000101
```

EXAMPLE 4

```
MEMORY PROTECT TEST
ENTER CPU TYPE 0 = 620/f     1 = V70
1.
START TEST      0. = MASK REG. OR 1. = INST. TEST
0.
CYCLES = 4,
MASK REG. TEST
INSTR. INT. ADDR. TEST
MASK REG. TEST
INSTR. INT. ADDR. TEST
MASK REG. TEST
INSTR. INT. ADDR. TEST
MASK REG. TEST
INSTR. INT. ADDR. TEST
MP TEST COMPLETE
CYCLES =
```

EXAMPLE 5

Running the test on a V70 but specifying 620/f.

```
MEMORY PROTECT TEST
ENTER CPU TYPE 0 = 620/f     1 = V70
0.
OPTIONAL INST. PRESENT    0. = YES,     1. = NO
0.
START TEST      0. = MASK REG. OR 1. = INST. TEST
0.
CYCLES = 2,
MASK REG. TEST
INSTR. INT. ADDR. TEST
ERROR TYPE = 000101
ERROR TYPE = 000010
MASK REG. TEST
INSTR. INT. ADDR. TEST
ERROR TYPE = 000101
ERROR TYPE = 000010
MP TEST COMPLETE
CYCLES =
```

SECTION 10
BUFFERED-I/O-CONTROLLER TEST PROGRAM

The buffered I/O controller test program of MAINTAIN III tests the operation of the V70 series system and 620-series buffered I/O controller (BIOC) options.

The BIOC monitors 16-bit word transfers between the CPU I/O bus and an external device. The data are transferred under program control or, optionally, under the control of the buffer interlace controller (BIC). The BIOC can also send a control signal (on up to four channels) to the external device and receive a SENSE signal (on up to eight lines) from it. In addition, the BIOC processes four interrupt lines in route to the priority interrupt module (PIM) if included in the computer system.

Computer control is extended to external devices through the BIOC. All BIOC functions are programmable.

The BIOC responds directly to three external control, one sense, and three data transfer instructions (table 10-1). A typical service routine is given in table 10-2.

Program Design Summary

The BIOC test program consists of five subtests:

a. I/O register test (subtest 1)

b. Pulse output test (subtest 2)

c. Sense line test (subtest 3)

d. Load input buffer via BIC (subtest 4)

e. Load output buffer via BIC (subtest 5)

The subtests can be individually selected for execution; the number of test cycles can also be specified.

If the tested system contains more than one BIOC, each can be tested by specifying the appropriate device address at the beginning of the test.

Table 10-1. BIOC Input/Output Instructions

Mnemonic	Code	Description
	External Control	
EXC 0x62	0100x62	Output a control pulse on line x (x = 00 through 03)
EXC 0662	0100662	Connect the BIOC for output
EXC 0762	0100762	Connect the BIOC for input
	Sense	
SEN 0x62	0101x62	Test the state of line x (x = 00 through 07)
	Data Transfer	
OME 062	0103062	Load the output buffer register from memory
OAR 0162	0103162	Load the output buffer register from the A register
OBR 0262	0103262	Load the output buffer register from the B register

Data input transfers are under the control of the BIC.

Table 10-2. Typical BIOC Service Routine

```
MUX     DATA    1               MUX CHANNEL
ANS     DATA    0               STORE DATA
        ORG     0100
        SEN     0360,SEL        ADC NOT BUSY
        NOP
        JMP     *-2
SEL     OME     0160,MUX
        EXC     0560
        SEN     0260,DATA
        NOP
        JMP     *-2
DATA    IME     060,ANS
        HLT
        END
```

I/O Register Test

This subtest sequentially outputs data from the computer to the BIOC output register and returns it to the BIOC input register for comparison. A comparison discrepancy results in the output of error messages (section 10.3).

Pulse Output Test

This subtest sequentially activates the output control pulse lines (60 times per test cycle) and verifies that a corresponding pulse level is returned to the BIOC input register. If a discrepancy is detected, data comparison error messages are output.

Sense Line Test

This subtest tests the eight BIOC sense lines by applying data to the output register, routing it to the sense lines, and verifying the response. Incorrect sense responses result in the output of error messages.

Load Input Buffer Via BIC Test

This subtest stores a one-word data pattern in the BIOC input register and connects the BIC to the input register for transfer of a 16-word block of data to memory. Each test cycle transfers one data block. The data pattern can be changed using the memory-altering feature of the test executive program. Incorrect data in the memory block following the transfer result in error messages.

Load Output Buffer Via BIC Test

This subtest stores a one-word data pattern in each word of a 16-word block of memory and connects the BIC to the BIOC output register for transfer of the data from memory. Each test cycle transfers one data block. The data pattern can be changed using the memory-altering feature of the test executive program. If the contents of the input register at the completion of the transfer are not identical to the transmitted data pattern, error messages are output.

System Configuration

The BIOC test program is designed to test the minimum configuration of a V70 or 620-series computer with 8K of memory (32K maximum), the BIOC, and a 33/35 ASR Teletype, using special test cables.

If more than one BIOC is included in the system configuration, each can be tested by specifying the appropriate device address when initiating the test.

If a BIOC operating with a BIC is to be tested, the BIC option is a prerequisite.

The BIOC test program object format is normally a punched paper tape for loading from the TTY or high-speed paper tape reader. Other media are available (e.g., card object deck).

10.1 PRELIMINARY PROCEDURES

To load the BIOC test program:

a. Load the test executive program (section 2).

b. Position the BIOC test program tape in the tape reader with leader at the reading station.

c. Type L, followed by a period, on the Teletype keyboard.

SENSE switch settings can alter test program execution as follows:

Switch	Set	Reset
1	Suppress error message printout	Print error message
2	Halt on error (continue testing after error halt)	Do not halt on error (loop after error halt)
3	Terminate testing and return to the test program beginning	Continue testing

To continue the test after an error halt, set SENSE switch 2 and press START or RUN.

To loop on an error, reset SENSE switch 2 after an error halt and press START or RUN. Looping continues until SENSE switch 2 is again set.

If the BIOC test program is run on the 620/f or V70 series computer, pressing the INT (interrupt) switch returns control to the test executive.

10.2 EXECUTING THE BIOC TEST PROGRAM

The BIOC is operated in the Teletype mode. After successful loading of the BIOC test program, the test program outputs the messages:

```
BUFFERED I/O TEST

ENTER BUFFER I/O DEVICE ADDRESS
```

On the TTY, type the appropriate device address, followed by a period. The program then outputs the message:

```
BIC TO BE USED (Y/N)
```

If the BIC capability is not to be tested, type an N. If it is, type a Y. If the response is Y, the program outputs the message:

```
ENTER BIC DEVICE ADDRESS
```

Type the BIC device address, followed by a period. The program then outputs the message:

```
ENTER SUBTEST NUMBER
```

CAUTION

Before attempting subtest execution, connect the appropriate test cable to circulate data between the BIOC input and output registers.

Subtests 1, 4, 5	Test cable A
Subtests 2, 3	Test cable B

Refer to the Buffered I/O Controller Manual (98 A 9902-626) for connection procedures.

A clock signal for testing under BIC control must also be provided.

Type the desired subtest number, followed by a period. This last message is output immediately after the **BIC TO BE USED** message if the response to that message is an N.

Following entry of the subtest number, the program outputs the message:

```
CYCLES =
```

Type the desired number of test cycles, followed by a period. If a zero is input, the test cycles continuously until terminated by the setting of SENSE switch 3. The program indicates termination of each subtest by requesting a new subtest number.

If more than one BIOC is included in the system, set, then reset, SENSE switch 3 to return to the beginning of the BIOC test program for a new device assignment.

When testing the 620/f or V70 series computer, return control to the test executive by pressing the INT switch.

To return to the BIOC test program from the test executive, type:

```
G500.
```

on the TTY. The value 500 represents the entry address for the BIOC test program. The actual starting address of the program is 000600, and it can be entered directly at that point.

10.3 ERROR INDICATIONS

If, during the execution of subtests 1, 4, and 5, the BIOC test program detects a discrepancy between the data patterns in the BIOC input and output registers, it outputs the message:

```
OUTPUT    xxxxxx    INPUT    yyyyyy
```

where

xxxxxx = the pattern transmitted to the output register
yyyyyy = the data read from the input register

If, during the execution of subtests 2 and 3, the program detects noncorresponding signal levels (subtest 2) or an incorrect sense response (subtest 3), it outputs an error message of the form:

```
000xxx
```

where xxx is an octal pattern representing the lines in error. This pattern is the exclusive-OR of all errors detected during one pass of the subtest. A one in position 0 (reading from right to left) of the binary conversion of the octal pattern indicates that line 0 is in error; in position 1, line 1, etc. For example, an octal value of 000377 indicates that all eight lines are in error.

If, during the execution of all the subtests, the BIOC input register is not cleared when read, the program outputs the following message:

```
IR RESET ERROR
```

When continuous execution of the program is terminated by the setting of SENSE switch 3, the message:

```
RESET SENSE SWITCH 3
```

is output if the switch is left set. Reset the switch to continue testing.

If an incorrect cable is used in testing the BIOC, error messages appropriate to the subtest being executed are output (see above).

10.4 TEST VALIDATION EXAMPLES

The results presented in this section were extracted from Teletype printed copy collected during validation of the Buffered I/O controller.

EXAMPLE 1:

Execute subtests 1, 4, and 5 of the BIOC test program for one cycle each using test cable A (no errors detected).

```
BUFFERED I/O TEST

ENTER BUFFERED I/O DEVICE ADDRESS 60.

BIC TO BE USED (Y/N) Y
ENTER BIC DEVICE ADDRESS 20.

ENTER SUBTEST NUMBER 1.
CYCLES = 1.

ENTER SUBTEST NUMBER 4.
CYCLES = 1.

ENTER SUBTEST NUMBER 5.
CYCLES = 1.

ENTER SUBTEST NUMBER
```

EXAMPLE 2:

Execute subtests 2 and 3 for one cycle each using test cable B (no errors detected).

```
BUFFERED I/O TEST

ENTER BUFFERED I/O DEVICE ADDRESS 60.

BIC TO BE USED (Y/N) Y
ENTER BIC DEVICE ADDRESS 20.

ENTER SUBTEST NUMBER 2.
CYCLES = 1.

ENTER SUBTEST NUMBER 3.
CYCLES = 1.

ENTER SUBTEST NUMBER
```

EXAMPLE 3:

Execute subtests 1 and 4 using the wrong test cable (B).

```
BUFFERED I/O TEST

ENTER BUFFERED I/O DEVICE ADDRESS 60.

BIC TO BE USED (Y/N) N

ENTER SUBTEST NUMBER 1.
CYCLES = 1.
OUTPUT    000001    INPUT    000000
OUTPUT    000002    INPUT    000000
OUTPUT    000003    INPUT    000000
OUTPUT    000004    INPUT    000000
 .         .         .        .
 .         .         .        .
 .         .         .        .

ENTER SUBTEST NUMBER 4.
CYCLES = 1.
OUTPUT    177777    INPUT    000000
OUTPUT    177777    INPUT    000000
OUTPUT    177777    INPUT    000000
OUTPUT    177777    INPUT    000000
OUTPUT    177777    INPUT    000000
 .         .         .        .
 .         .         .        .
 .         .         .        .

ENTER SUBTEST NUMBER
```

EXAMPLE 4:

Execute subtests 2 and 3 using the wrong test cable (A).

```
BUFFERED I/O TEST

ENTER BUFFERED I/O DEVICE ADDRESS 60.

BIC TO BE USED (Y/N) Y
ENTER BIC DEVICE ADDRESS 20.

ENTER SUBTEST NUMBER 2.
CYCLES = 1.
000377
000377
000377
000377
000377
000377
000377
000377
 .
 .
 .
ENTER SUBTEST NUMBER 3.
CYCLES = 1.
IR RESET ERROR
IR RESET ERROR
IR RESET ERROR
IR RESET ERROR
IR RESET ERROR
 .
 .
 .
```

EXAMPLE 5:

Execute subtest 4, specifying continuous execution. Terminate the test by setting SENSE switch 3.

```
ENTER SUBTEST NUMBER 4.
CYCLES = 0.

BUFFERED I/O TEST
RESET SENSE SWITCH 3
RESET SENSE SWITCH 3
```

Note that if the test is restarted with SENSE switch 3 remaining set the program requests that the switch be reset.

SECTION 11
WRITABLE CONTROL STORE TEST PROGRAM

The following discussion does not apply to the V77-400 computer system with Writable Control Store (WCS) option. The V77-400 WCS Test Program is explained in section 12. The following discussion applies to all other V70 series computers having WCS.

The WCS test program, which is controlled by the MAINTAIN III test executive program, is used to verify correct operation and to isolate malfunctions. It is modular with independent tests that can be executed individually or in groups.

Minimum hardware requirements for using the test program consists of a V70 series processor with 8K of memory (core or semiconductor), a 256-word WCS, and an ASR-33 TTY. For a more efficient operation, a high-speed paper tape reader is recommended.

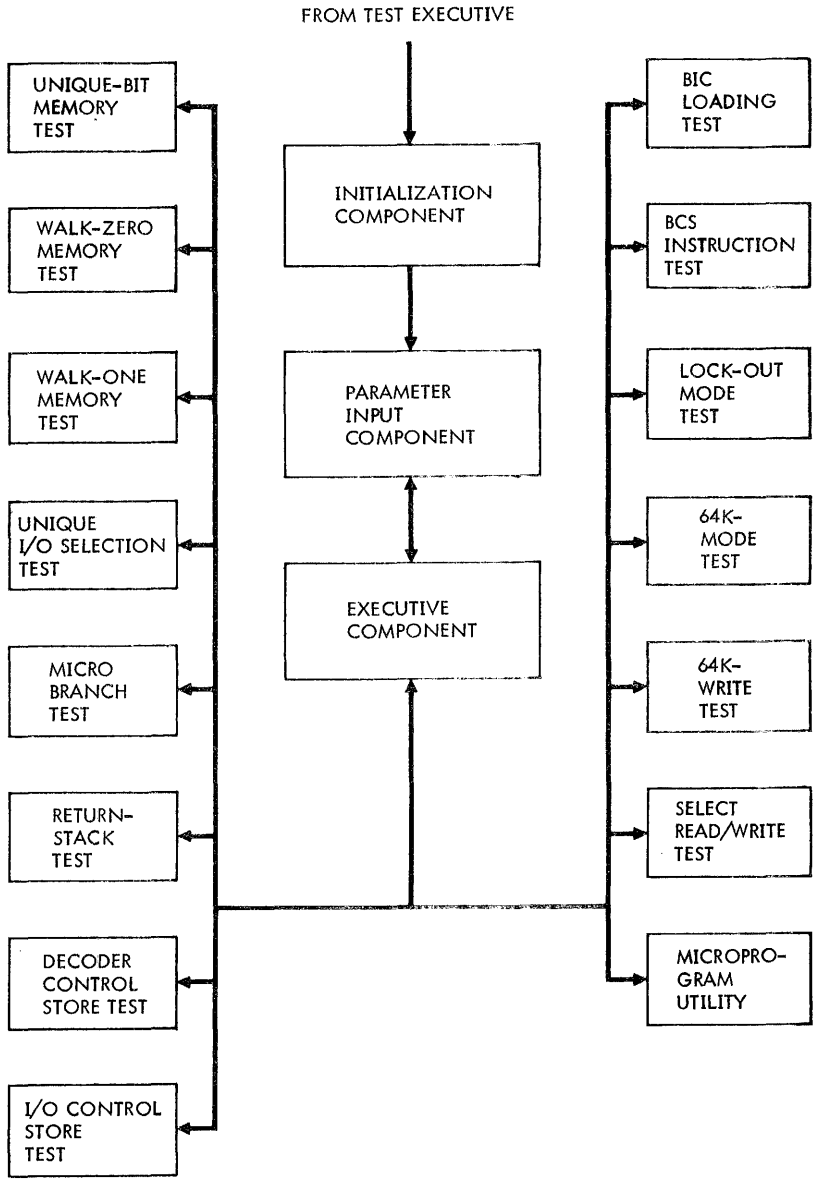


Figure 11-1. Test Program Flow Diagram

11.1 TEST PROGRAM ORGANIZATION

As illustrated in figure 11.1, the test program consists of three supervisory components, fourteen independent tests, and a microprogram utility. The initialization component determines the WCS configuration, the parameter input component selects which control stores in a WCS are to be tested, and the executive component selects the test (or tests) to be performed. The following subsections describe the 14 tests and the microprogram utility.

11.1.1 Unique-Bit Memory Test

This test verifies that any bit of the first CCS word can be a one while all other bits are zeros. The first word of each 256-word array specified by the test program is tested in this manner. Before the test begins, all locations are loaded with all ones and then verified. At the end of the test locations are verified again to ensure that none of the data has changed.

The unique-bit memory test can also be performed on decoder and I/O control stores. In these cases, only the first word of each control store is tested.

11.1.2 Walk-Zero Memory Test

In this test, each word in the specified control store is loaded with all ones except the first word which is loaded with all zeros. This condition is verified. The first word is then loaded with all ones and the second word with all zeros. This condition is verified, and the procedure is continued causing a word of zeros to be "walked" through the control store. This test is a worse case condition for the semiconductor memory and detects unique addressing failures.

11.1.3 Walk-One Memory Test

This test is similar to the walk-zero memory test except a word of ones is "walked" through a control store containing all-zero data.

11.1.4 Unique I/O Selection Test

This test verifies that writing into one WCS module does not change the contents of any other WCS module. A data pattern is loaded into location 8 of the specified CCSs, decoder control stores, and I/O control stores. The complement of the data pattern is then loaded into location 8 of the control stores in the fifteen WCS modules that are potentially addressable. The WCS control stores containing the original data patterns are verified after each loading operation to ensure that no unauthorized data changes occurred.

11.1.5 Micro-Branch Test

This is a unique bit test for the CCS address bits. It verifies that any of the nine adddress bits can be a one while the rest are zeros. This is accomplished by loading the specified CCS with a microprogram that produces a jump address containing a single one bit and eight zero bits.

Successful execution of this microprogram causes a successful return to the WCS test program. If one of the branches goes astray in the CCS, an error return to the test program occurs.

11.1.6 Return-Stack Test

This test exercises the subroutine stack by using a combination of macro- and micro-level testing. The test checks the stack-oriented I/O instructions, and tests the stack's overflow detection. If these tests are successful, a unique bit test is performed for each of the 13-bit stack words to verify that any one of the low-order 9 bits can be a one will all others are zeros (the high-order 4 bits of a stack word specifies the page number).

Unique stack addressing is also tested. By "walking ones" through the stack, the test verifies that loading data onto the stack does not change any of the data in locations below it.

11.1.7 Decoder Control Store Test

The specified decoder control store is loaded with test data and exercised. The test verifies that the specified decoding equations are correct and that a full range of outputs are possible.

11.1.8 I/O Control Store Test

The specified I/O control store is loaded with a micro routine that transfers bit patterns to the display indicators on the control panel. After each transfer, the bit pattern is verified by the test program. If the system contains a BIC option, the BIC loading test is performed on the CCS of page 1. The BIC loading test reports any errors through its standard error messages.

11.1.9 BIC Loading Test

The specified control stores are loaded with a block of test data using the BIC. The BIC loading operation is monitored using the appropriate SEN instructions. If the loading operation is not completed after a specified time delay, an error message is produced. Upon a successful completion of the loading operation, the contents of the control store are verified.

11.1.10 BCS Instruction Test

This test first verifies that the BCS instruction is branching to a WCS module and not the processor control store. A unique-bit test is then performed for the BCS instruction by performing test branching to locations 1, 2, 4, 8, 16, 32, 64, and 128. An error message occurs if any of the first five branching operations fail or if any of the last three pass. If no error messages occur, the BCS instruction is branching within the first 32 words as required by the Varian firmware.

11.1.11 Lockout-Mode Test

This test loads and executes a microprogram to set and reset the memory lockout flag. A time delay is provided between setting and resetting of the flag to allow verification with an oscilloscope. The activity of the memory lockout flag can be observed with the oscilloscope by monitoring signal MHGP– on the WCS board (pin 6 of the IC at location D15).

11.1.12 64K-Mode Test

This test loads and executes a microprogram to set and reset the 64K-mode flag. A time delay is provided between setting and resetting of the flag to allow verification with an oscilloscope. The activity of the 64K-mode flag can be observed with the oscilloscope by monitoring signal W65KE– on the WCS board (pin 8 of the IC at location D12).

11.1.13 64K-Write Test

This test loads and executes a microprogram that writes a bit pattern into the first word of the upper 32K of memory (octal address 100000). It verifies that the bit pattern is read-back correctly and also if wrap-around addressing occurred at location zero. Meaningful results are produced only if the system contains more than 32K of memory.

11.1.14 Read/Write Selection Test

This test enables the user to read or write a page or part of a page by means of the teletypewriter terminal. Read/write functions may be cycled a prescribed number of times as in any of the other tests.

Test operation entails use of three commands: P, R, and W.

The P command offers the user a means of inserting a test pattern to be used in testing the WCS. The command is invoked by typing Pn following the INPUT COMMAND prompt, where n is the 16-bit pattern in octal notation. The command is terminated by a period.

The R command allows the user to read a number of contiguous words from the specified page and to decide whether to test for pattern errors.

The W command enables the user to write. The R and W commands have identical formats as follows:

< command > , < pattern > < ,startword > < ,lastword >

More specifically, R and W commands are expressed as

$$\left\{\begin{matrix}R\\W\end{matrix}\right\} , \left\{\begin{matrix}0\\1\\2\\3\\4\\5\\6\\7\end{matrix}\right\} \left\{\begin{matrix}.\\,m\end{matrix}\right\} \left\{\begin{matrix}.\\,n.\end{matrix}\right\}$$

where

R = Read selected page,
W = Write selected page,
0 = All zeros pattern,
1 = All ones pattern,
2 = Constant 052525 (octal) pattern,
3 = Alternating 052525 (octal) pattern,
4 = Constant 0125252 (octal) pattern,
5 = Alternating 0125252 (octal) pattern,
6 = User-defined pattern,
7 = No check of data on Read command (undefined for Write),
m = Starting word address ($0 \leq m \leq 777_8$)
n = Last word address ($n \geq m$)
, = Delimiter
. = Command terminator

11.1.15 Microprogram Utility

A subset of the Sperry Univac microprogram utility is provided as an aid in troubleshooting. It performs reading and writing operations on single WCS words, and lists blocks of WCS words. In addition, with the auxiliary computer configuration (see section 11.5.8) controlled execution can be performed.

11.2 TEST IDENTIFICATION

The tests performed by the test program are identified with hexadecimal numbers 1 through E. In general, the higher the identification number the more complex the test. Therefore, to isolate an error to its most basic component, the lowest possible numbered test should be used. For example, if the return stack test fails due to a bad WCS memory location (the most basic component), the error can be isolated more accurately by using tests 1, 2, and 3. If the same test fails due to a bad micro- branching operation, the error can be better identified by using test 6.

Table 11-1 identifies the tests and lists the test prerequisites.

Table 11-1. Test Identifications

Test Number (Hex)	Test	Test Prerequisites
1	Unique Bit Memory	None
2	Walk-Zero Memory	None
3	Walk-One Memory	None
4	Unique I/O Selection	1, 2, 3
5	BIC Loading Test	1, 2, 3, 4
6	Micro Branch	1, 2, 3, 4
7	BCS Instruction	1, 2, 3, 4, 6
8	Return Stack	1, 2, 3, 4, 6
9	Decoder Control Store	1, 2, 3, 4, 6
A	I/O Control Store	1, 2, 3, 4
B	Lockout Mode	1, 2, 3, 4, 6
C	64K Mode	1, 2, 3, 4, 6
D	64K Write	1, 2, 3, 4, 6
E	Microprogram Utility	None
F	Select Read/Write	None

11.3 PROGRAM LOADING

Before loading the WCS test program, the MAINTAIN III test executive must be loaded. Included in the test executive is the object loader for the WCS test program. After the test executive is loaded, mount the object medium of the WCS test program in the input device.

Refer to section 2 of this manual for loading procedures.

11.4 SENSE SWITCHES

Operation of the WCS test program can be modified by setting and resetting SENSE switches on the computer control panel. The switch functions are listed in table 11-2.

Table 11-2. SENSE Switch Settings

SENSE Switch	Set	Reset
1	Suppresses error printouts	Prints error messages
2	Halts program on error*	No halts on error
3	Returns program to the previous parameter request	Continues test

*After the error halt, one of the following operations can be performed:

a. The program can be continued by pressing START

b. The program can be made to loop on the error condition, by resetting SENSE switch 2 and pressing START. If the error condition clears up, looping continues until SENSE switch 2 is set.

11.5 TELETYPE MODE

This section describes test program operating procedures using the TTY.

11.5.1 TTY Input Editing

The TTY character ← can be used to delete the previous character typed. The TTY character \ can be used to delete the current line being typed.

If any parameter or character input is unacceptable to the test program, the TTY prints the message

```
INVALID
```

and the request is repeated.

11.5.2 Determining the Configuration

The TTY identifies the test program by printing:

```
WCS TEST PROGRAM
```

The WCS device address is requested by the message:

```
EVEN WCS DEV ADDR =
```

The operator must then type either 70, 72, or 74 followed by a period. The particular device address used depends on the hardware configuration.

The WCS configuration is determined by writing data into all control stores that could exist (CCS, decoder control store, and I/O control store), and then attempting to read back the data. A successful reading operation indicates a control store is present, and an unsuccessful one indicates a control store is not present.

The TTY prints a message describing the WCS configuration as follows:

```
PAGE: n; x; y; z
```

The letter n is a single hexadecimal character specifying a page number for an existing WCS. If the WCS contains a CCS, then x is either CCS(256) or CCS(512) depending on its size. If no CCS is present, then x is omitted. If the WCS contains a decoder control store, then y is DCS. If no decoder control store is present, the y is omitted. If the WCS contains an I/O control store, then, z is I/O CS. If no I/O control store is present, then z is omitted. The following example shows a WCS configuration and the TTY message that describes it.

CONFIGURATION:

Page 1 contains a 512-word CCS, a decoder control store, and an I/O control store.

Page 2 contains a 256-word CCS and an I/O control store.

Page 3 contains a 256-word CCS.

MESSAGE:

```
WCS CONFIGURATION

PAGE: 01; CCS(512); DCS; I/O CS
PAGE: 02; CCS(256); I/O CS
PAGE: 03; CCS(256);
```

If no control stores are found by the test program, the following message occurs:

```
NO WCS FOUND
USE SS3 TO RE-SPECIFY DEV ADDR
```

The above condition is due to one of the following:

a. An incorrect device address is specified.

b. A WCS malfunction is preventing the WCS from being identified by the test program's write/read operations.

In the first case, SENSE switch 3 should be activated to return the program to the device address specification. In the second case, testing should be continued to isolate the malfunction further.

The BIC device address is then requested as follows:

```
BIC DEV ADDR =
```

The operator should type the BIC address followed by a period. If a BIC is not available, the operator should type a zero followed by a period.

11.5.3 Control Stores to be Tested

The operator must specify the control stores to be tested. The TTY first requests this information for CCS by printing the following message:

```
TEST PAGES
CCS:
```

The operator must then type one of the following four characters:

A N • –

The functions of these characters are listed as follows:

Character	Function
A	Tests all existing CCS (section 11.5.2)
N	Tests none of the CCSs.
•	Tests the CCSs specified during the last request. During the first time through, this character functions the same as character N.
–	Accepts a list of page numbers and tests the CCSs on each of these pages. The page numbers are a series of single hexadecimal digits separated by commas and terminated with a period. Typing this character overrides the configuration definition determined in section 11.5.2.

After determining which CCSs are to be tested, similar requests are made for the decoder and I/O control stores. The TTY messages for these requests are:

```
DCS:
```

and

```
I/O CS:
```

The same four characters A N • – are again used to specify the control stores to be tested. The following is an example of a TTY printout requesting the testing of CCSs on pages 1, 2, and 4; none of the decoder control stores; and all of the I/O control stores (the underlined characters are operator inputs):

```
TEST PAGES

CCS: -1, 2, 4.

DCS: N

I/O CS: A
```

11.5.4 Tests to be Performed

The program requests the sequence of tests to be performed by causing the TTY to print the following message:

```
TEST SEQ
```

The operator must then type one of the following four characters to specify the test sequence: – • **A B**

The functions of these characters are listed as follows:

Character	Function
–	Enables the program to accept a series of test numbers (table 11-2) consisting of single hexadecimal digits separated by commas and terminated with a period. Each hexadecimal digit specifies a test. Up to 30 tests can be specified in any order or any number of repetitions.
•	Enables the program to use the test sequence specified during the last request.
A	Causes all tests to be performed in numerical order (e.g., = 1, 2, 3, 4, 5, 6, 7, 8, 9, A). Tests B and higher are not included because they require a special configuration or special attention.
B	Causes program to return to the requesting of which control stores are to be tested (section 11.5.3).

The entire test sequence can be performed a fixed number of times or repeated indefinitely. This request is indicated by the TTY message:

```
CYCLES?
```

The operator must then type the desired number of repetitions; a zero for an indefinite number of repetitions or a single hexadecimal digit to request up to 15 repetitions. If it is desired to have the individual tests identified as they are called, the operator's response must be terminated with a comma. For no test identification, a period is used as the terminator. The following is an example of a TTY printout specifying a test sequence consisting of the unique bit memory test, two micro-branch tests, and the walk-zero memory test; the sequence is to be performed seven times (the underlined characters are operator inputs):

```
TEST SEQ?
= 1, 6, 6, 2.

CYCLES
7,
```

11.5.5 Test-Sequence Execution

After the number of cycles is specified, execution of the first test in the sequence begins. The program repeats the entire test sequence as many times as specified, and then returns to input a new sequence. The execution can be interrupted at any time by setting SENSE switch 3. This causes the program to return to the request for a new test sequence.

11.5.6 Error Message Formats

There are three basic formats (types 1, 2 and 3) for error messages used in the test program. Type 1 reports discrepancies between the expected and actual contents of a word in the WCS. Type 2 is more general and reports a wide range of conditions through an error code number. Type 3 reports data error conditions generated in Test F.

Information provided by the type 1 format is listed below:

a. Number of the test reporting the error.

b. Page number of the WCS causing the error.

c. Type of control store causing the error.

d. Addressing containing error.

e. Expected data and actual data are listed.

Information provided by the type 2 format is listed below:

a. Number of the test reporting the error.

b. Type of control store causing the error.

c. Error code number (hexadecimal) describing the error condition (section 11.8 describes the error codes).

d. Provision is made for specifying a control store word. This may be used for other information depending on the error code.

Information provided by the type 3 format is listed below:

a. Same information as in the type 1 error message.

b. A printout of the current value of the microword read operation.

c. A printout of the actual value of the microword read operation.

The type 1 error message format is given below:

```
**T a :b,P c; W d; E= e; A= f
```

where:

a is single hexadecimal digit specifying the test number

b is a letter specifying the type of control store (C = CCS, A = array A of decoder control store, B = array B of decoder control store, and I = I/O control store).

c is a single hexadecimal digit specifying the paper number number

d is four hexadecimal digits specifying the word address

e is either 4 or 16 hexadecimal digits specifying the expected data in the address.

f is either 4 or 16 hexadecimal digits specifying the actual data in the address.

The type 2 error message format is given below:

```
**T a :b,P c; W d; ERR n
```

where:

a, b, and c are the same as in the type 1 format

d is either the same as in type 1 format or has a special function specified by n

n is the error code number (in hexadecimal, see section 11.8.)

The type 3 error message format is given below:

```
**T a :b,P c; w d; E= e; A=F
  ccccccccccccccc
  aaaaaaaaaaaaaaa
```

where

a. the first line is the same as the type 1 message;

b. the second line is the correct data; and

c. the third line is the actual data read.

11.5.7 Summary of TTY Requests

The following is a summary of TTY requests with the operator responses underlined.

```
EVEN WCS DEV ADDR = n.
```

where n is 70, 72, or 74.

```
BIC DEV ADDR = n.
INPUT TYPE 2 PAGES
x
```

where

n is either the BIC device address or zero if there is no BIC;

x is one of the following:

```
A
N
.
- p1, p2, p3, etc.
```

such that p is a series of hexadecimal digits specifying tests.

```
TEST SEQ?
x
```

where x is one of the following:

```
A
B
.
- t1, t2, t3 etc.
```

where t is a series of hexadecimal digits specifying tests.

```
CYCLES?
xy
```

where x is either zero or a single hexadecimal digit, and y is either either a period or a comma.

The following TTY requests pertain to test F only.

```
INPUT COMMAND
Pn
x,y {;z}  {;w}
```

where n is an octal number and x is either R or W, and $0 \leq y \leq 7$, $0 \leq z \leq 777_8$, $w \geq z$.

11.5.8 Microprogram Utility

The TTY identifies the microprogram utility by printing:

```
VARIAN 73 MICRO UTILITY

DEBUG CONFIG  (Y OR N)
```

The operator must then type a Y or N to specify whether or not the auxiliary computer configuration is being used. When a directive is ready to be processed, the TTY prints:

```
MU**
```

A detailed description and the directives for the microprogram utility are provided in the Microprogramming Guide. The control-store loading (L) and media selection (M) directives are not available with the microprogram utility under MAINTAIN III. In addition to the standard directives of the microprogram utility, a directive designated RT is available for returning the test program to the MAINTAIN III test executive.

11.6 CONTROL-PANEL INDICATORS

During an output-control transfer (OAR 07X, OBR 07X, or OME 07X), the test program displays each function word on the 16 display indicators of the control panel.

The BCS instruction test also transfers the BCS instruction to the display indicators before it is executed.

11.7 DESCRIPTION OF ERROR CODES

The following is a list of WCS error codes.

Code Hex	Code Octal	Description
8	010	Unable to clear a busy WCS with the initializing instruction. This prevents a WCS I/O operation causing an abort of the test sequence.
A	012	A BIC loading operation is active after the end of the time delay. This prevents a WCS I/O operation causing an abort of the test sequence.
D	015	The unique I/O selection test produced the preceding error message of the type 1 format.
11	021	Micro branch test is unsuccessful. The number of microinstructions correctly executed is indicated by the number following W in the type 2 error message format.
18	030	The return stack test reports that the subroutine stack is full after an initializing instruction.
19	031	The return stack test reports that the subroutine stack is not empty after an initializing instruction.
IE	036	The return-stack test reports that an overflow condition was not detected when it should have been.
IF	037	The return-stack test reports that an overflow detection occurred either too early or too late. The number of push operations required to cause an overflow condition is indicated after the W in the type 2 error message format. This number does include the pushing operation of stack location zero (i.e., the overflow exit).
20	040	The return-stack test reports that an overflow condition was successfully detected after 16 push operations but the branching to an error exit was not successful.
21	041	The return stack test reports that the subroutine stack is not full after an overflow condition.
22	042	The return stack test reports that the subroutine stack is empty after an overflow condition.
23	043	The return-stack test reports that the subroutine stack is full before 16 push operations occur. This is probably caused by the pop microinstruction failing to clear the full status. The number of push operations performed is indicated after the W in the type 2 error message format.
24	044	The test program is aborted for the specified page due to a loading failure of its micro subroutine.
25	045	The test program is aborted for the specified page due to an unsuccessful execution of its micro subroutine.
27	047	The return-stack test was unable to jump (pop operation) to this CCS location.
28	050	This error code always follows the previous one (hex 27, octal 047) to give the stack location (in hexadecimal) for which the failure occurs.
2A	052	The decoder control store failed to decode a test word correctly. The test case number is indicated after the W in the type 2 error message format. Table 11-3 lists the decoder output bits tested and the effective CCS address for the various test cases.
2D	055	The delete-top-of-stack microinstruction probably failed. The stack location is indicated by the number following the W in the type 2 error message format.
2E	056	Pushing ones into this stack location changed the contents of one stack location below it.
2F	057	This error code always follows the previous one (hex 2E, octal 056) to give the stack location that was changed.

Code		Description
Hex	Octal	
31	061	The I/O control store test failed. The type 1 error message format indicates the expected and actual data from the control-panel indicators.
33	063	The BIC loading test failed due to the BIC being busy.
34	064	The BIC loading test failed due to the BIC being inactive immediately after it was started.
35	065	The BIC loading test failed due to the BIC sensing an abnormal device stop.
36	066	The BCS instruction did not branch to WCS.
37	067	The BCS instruction did not branch to the specified location.
38	070	BCS instruction did branch to the specified location.
40	100	The 64K-write test reports that data read from octal address 100000 are different than the data written in.
41	101	The 64K-write test reports that the contents of location 0 changes when data are written into octal location 100000. This is most likely due to a wrap-around addressing configuration.
42	102	The operation of pushing a success exit followed by pushing 15 error exits failed.
43	103	30 (decimal) error exists have been popped without encountering a success exit which should occur after 15 (decimal) error exists.
44	104	Failure of stack to wrap around after 15 (decimal) error exists.
45	105	A data comparison error has occurred during the read function of test F.

11.8 TTY MESSAGE EXAMPLES

Figure 11-2 shows a typical TTY printout that occurs when all tests of the test program are successfully run.

Figure 11-3 shows a typical error message for the I/O control store test. The error producing actual data AAAB is caused by pressing the bit-0 register entry switch on the control panel. The error producing actual data 5557 is caused by pressing the bit-1 register entry switch on the control panel.

Figure 11-4 shows typical error messages caused by a nonexisting WCS page. In both tests (lockout mode and return stack), the actual data read from the non-existing page are all zeros. The printouts of both tests are terminated before completion by setting SENSE switch 3.

Figure 11-5 shows a sequence of commands in test F which will:

a. Write zeros in the entire page (W,O).

b. Read the entire page with no data check (R,7).

c. Read the entire page, computer data having zero pattern (R,O).

d. Read the word 200 (octal) and compare it with ones pattern, freeing a data error.

Figure 11-5 illustrates the commands input by the user and the resulting output and error formats.

TEST CASE NUMBERS	DECODER BITS TESTED								AFFECTIVE WCS CCS ADDRESS
	WDB15+ (CIDS32-)	WDBD14+ (CIDS31-)	WDBD13+ (CIDS30-)	WDBD12+ (CIDT32-)	WDBD10+ (CID00+)	WDAD14+ (CIDS21+)	WDAD13+ (CIDS20+)	WDAD09+ (CIDXX5+)	
0									001
1				X	X				001
2		X							001
3			X						001
4									
5									002
6	X				X				002
7	X					X			002
8	X						X		002
9									
A	X				X	X		X	004
B									004
C				X	X				004
D		X							004
E			X						004
F									
10									008
11	X				X				008
12	X					X			008
13	X						X		008
14									
15									010
16				X					010
17	X								010
18		X		X	X			X	010
19									
1A									020
1B				X					020
1C									
1D									040
1E	X								040
1F									
20									080
21				^					080
22									
23									100
24	X								100

NOTES: 1. SIGNAL MNEMONICS IN PARENTHES ARE CORRESPONDING OUTPUT BITS FROM THE PROCESSOR INSTRUCTION DECODER.
2. THE Xs INDICATE THE DECODER BITS THAT ARE TESTED FOR A PARTICULAR TEST CASE.

Table 11-3. Decoder Control Store Test Cases

```
THIS IS THE 620 TEST EXECUTIVE
MEMORY SIZE IS  32K

L.
WCS TEST PROGRAM

EVEN WCS DEV ADDR=  74.

WCS CONFIGURATION

PAGE: 01; CCS(512); DCS ; I/O CS

BIC DEV ADDR= 0.

TEST PAGES
CCS:  A
DCS:  A
I/OCS:  A

TEST SEQ?
A
CYCLES?
1,

UNIQUE BIT MEMORY TEST (1)

WALK ZERO MEMORY TEST (2)

WALK ONES MEMORY TEST (3)

UNIQUE I/O SELECT TEST (4)

NO BIC TEST

MICRO BRANCH TEST (6)

BCS INSTRUCTION TEST (7)

RETURN STACK TEST (8)

DCS TEST (9)

I/O CS TEST (A)

TEST SEQ?
```

VTII-1826A

Figure 11-2. Printout for a Successfully Run Test Program

```
TEST SEQ?
-A.
CYCLES?
0.
**T 0A :I,P 01; W FFFF; ERR 31
**T 0A :I,P 01; W FFFF;E= AAAA;A= AAAB
**T 0A :I,P 01; W FFFF; ERR 31
**T 0A :I,P 01; W FFFF;E= 5555;A= 5557
```

VTII-1827A

Figure 11-3. An Error Message for I/O Control Store Test

```
TEST SEQ?
B
TEST PAGES
CCS:  -5.
DCS:  N
I/OCS:  N

TEST SEQ?
-B.
CYCLES?
1,

LOCKOUT MODE TEST (B)
**T 0B :C,P 05; W 0060;E= 0308 0884 8007 0001;A= 0000 0000 0000 0000
**T 0B :C,P 05; W 0061;E= 0310 2240 0000 0000;A= 0000 0000 0000 0000
**T 0B :C,P 05; W 0062;E= 0318 0006 E000 0001;A= 0000 0000 0000 0000
**T 0B :C,P 05; W 0063;E= 0490 0001 8001 0102;A= 0000 0000 0000 0000
TEST SEQ?
-8.
CYCLES?
1,

RETURN STACK TEST (8)
**T 08 :C,P 05; W 0000;E= 0490 0001 8001 0202;A= 0000 0000 0000 0000
**T 08 :C,P 05; W 0001;E= 0490 0001 8001 0202;A= 0000 0000 0000 0000
**T 08 :C,P 05; W 0002;E= 0490 0001 8001 0202;A= 0000 0000 0000 0000
```

VTII-1828A

Figure 11-4. Error Messages for a Non-Existing WCS Page

```
TEST SEQ?
-F.
CYCLES?
1,

SELECT READ/WRITE TEST (F)
INPUT COMMAND
W,0.
TEST SEQ?
-F.
CYCLES?
1,

SELECT READ/WRITE TEST (F)
INPUT COMMAND
R,7.
TEST SEQ?
-F.
CYCLES?
1,

SELECT READ/WRITE TEST (F)
INPUT COMMAND
R,0.
TEST SEQ?
-F.
CYCLES?
1,

SELECT READ/WRITE TEST (F)
INPUT COMMAND
R,1,200,200.**T OF :C,P 01; W 0080; ERR 45

FFFFFFFFFFFF FFFF
0000000000000000
```

VTII-3642

Figure 11-5. Error Messages for Test F.

SECTION 12
V77-400 WRITABLE CONTROL STORE TEST PROGRAM

The following discussion applies only to the V77-400 computer system with Writable Control Store (WCS) option. Section 11 explains the WCS Test Program for all other V70 series computers.

The 77-4005 WCS option extends the V77-400 processor's read-only control store to permit addition of new instructions, development of micro-diagnostics, and optimum tailoring of the computer system to its application. The purpose of the WCS Test Program is to verify correct operation and to isolate malfunctions of the WCS option.

The Test Program is modular, having independent tests which can be executed individually or in groups. It requires as its minimum normal configuration 32K words of memory and a model V77 processor with the 77-4005 WCS option. Minimum WCS configuration is one to three WCS boards (modules), each with 1024 words of central control store. Each board contains four WCS pages and has the same I/O device address — which may be either 70 or 77 (octal).

12.1 TEST PROGRAM ORGANIZATION

Figure 12-1 depicts system flow for the V77-400 WCS Test Program. Three supervisory components and seven distinct subtest components are featured.

12.1.1 Supervisory Components

The supervisory components of the Test Program are:

a. the initialization component,
b. the parameter input component, and
c. the executive component.

The initialization component determines the WCS configuration. The parameter input component is the means for selecting which WCS boards (modules) are subject to testing. The executive component allows the calling of individual test programs, either individually or in sequence as a group.

12.1.2 Subtest Components

The subtest components of the Test Program are:

a. the Unique Address Test,
b. the Grouped Bit Memory Test,
c. the Decode Test,
d. the Page Branch Test,
e. the Stack Test,
f. the Execution Test, and
g. the Hog Mode Test.

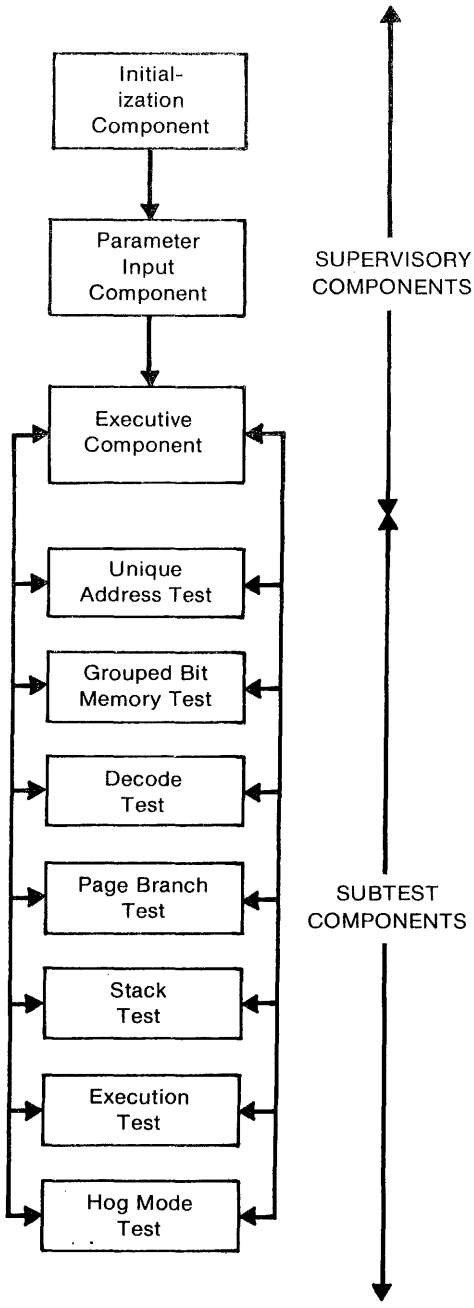


Figure 12-1. V77-400 Writable Control Store Test Program System Flow.

The Unique Address Test writes a unique data pattern into each location of each WCS page, then reads the location to validate the data pattern.

The Grouped Bit Memory Test writes patterns of grouped bits into each location of each WCS page, then reads and verifies the location. Grouped bits are designed to detect shorted bits within each location. (See figure 12-2 for a display of these bit patterns.)

The Decode Test compares results of a function test with results of a software program. (See table 12-1 for a list of these functions.)

The Page Branch Test tests the ability of each page of WCS to branch into every other page.

The Stack Test determines if the hexadecimal return stack returns correctly.

In the Execution Test, a small program is written into the WCS and executed. The results are compared to known values.

The Hog Mode Test checks the WCS memory lockout capability. The test requires two processors with console teletype or the equivalent. These processors share a common memory module. At least one processor, the master, must have WCS. The other processor, the slave, need not. (See figure 12-3.)

The slave processor starts execution of a program. As this program executes, a bell rings at approximately 120 millisecond intervals. During execution of the slave processor's program, the master processor commences execution of its own program. (At this point the system is said to have entered Hog Mode.) If the system is functional, the master processor activity overrides — or "hogs" that of the slave processor. This becomes apparent to the operator in that the interval between rings of the bell becomes noticeably longer.

12.2 PROGRAM LOADING

The WCS test program cannot operate unless the MAINTAIN III test executive is loaded ahead of it. Standard subroutines in the test executive are called by the test program.

To load the V77-400 WCS test program:

1. Load the MAINTAIN III test executive, which includes the binary object loader. (Refer to section 2 of this manual for loading procedures.)

2. Mount the WCS test program object in the object reader. Type L on the operator console.

12.3 SENSE SWITCHES

Operation of the WCS test program can be modified by setting and resetting SENSE switches on the computer control panel. Table 12-2 presents the SENSE switch settings.

Table 12-2. SENSE Switch Settings.

Switch	Set	Reset
SS1	Suppress error printouts	Print error messages
SS2*	Halt on error	No halt on errors
SS3	Return to previous parameter request	Continue test

*SS2 can be used to loop on an error halt, or to continue the test as follows.

After the halt:

1. To continue to the next error halt, keep SS2 set and press START on the computer.

2. To loop on the error condition, reset SS2 and press START on the computer. Looping will continue until SENSE switch 2 is set. If the error condition clears up, looping continues until SS2 is set.

```
Bit      3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
Pattern  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0

0  0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
1  0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
2  0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
3  0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
4  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
5  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
6* 1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 1 0 0 0 0 0 1 1 1 1 0 0 0 1 0 0

*initial value
```

Figure 12-2. Bit Patterns for Grouped Bit Memory Test.

Table 12-1. V77-400 Writable Control Store Decode Test Functions

Reference	Function
1A	(A + B)→Q
1B	(A∧B)→Q
1C	(A xor B)→Q
1D	(A - B)→Q
1E	(A - B - 1)→Q
1F	(B - A)→Q
25	A∧Q
26	A xor Q
27	Q - A
34	A + Q
35	A + Q + 1
36	A - Q
37	A - Q - 1
39	B - A - 1
3A	$\overline{A}$ - B -1
3B	$\overline{A}$∧B B)
41	Lit - 1
45	BC + 1
48	MD + A
49	MD + A
52	Lit
53	Lit
54	$\overline{BC}$
55	BC - 1
56	$\overline{IO}$
57	IO + 1
58	A - MD - 1
59	MD - A - 1
5B	IB xor A
5C	A + Lit
5E	Lit - A
5E	Lit - A
5F	A∧Lit
60	IR + 1
61	IR - 1
66	MD
6B	IB∨A
6C	IR∨A
6D	IR∧A
6E	IR + A
6F	IR - A
73	$\overline{IR}$
76	MD
79	MD - A
7A	MD + A
7B	IB A
7C	IR xor A

NOTES:

1. A & B are inputs to the ALU (2901).
2. Q is a register used during shifting.
3. xor is the EXCLUSIVE OR operator.
4. BC represents the auto boot PROM output control.
5. Lit is a literal (i.e., a character string).
6. IB is the Instruction Buffer.
7. IR is the Instruction Register.
8. MD is Memory Data.

12.4 OPERATING PROCEDURES

12.4.1 Configuration Determination

The test progam identifies itself by outputing

V77-4XX WCS TEST

Next, the hardware configuration is requested via the message

64K MEMORY

If the processor has at least 64K of memory, then the user responds with Y (for yes); otherwise, the user responds with N (for no).

The WCS device address is then requested via the message

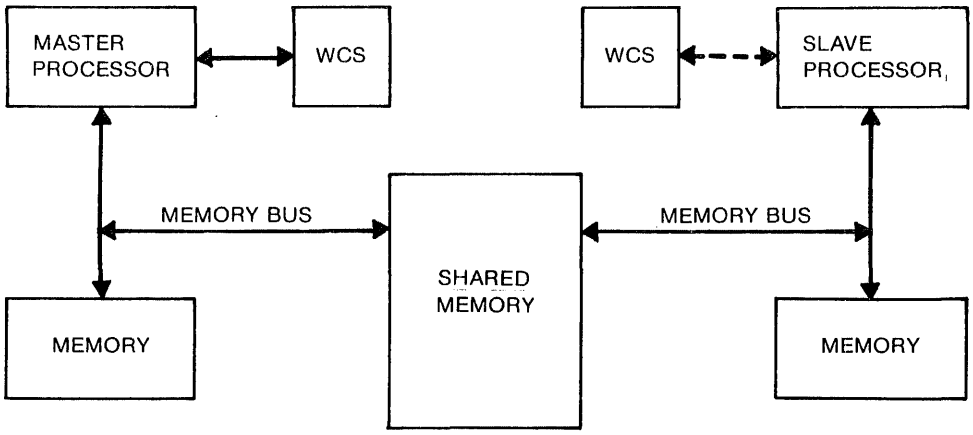
WCS DA

The user responds with the octal value of the device address for the WCS being tested, followed by a period.

The system then determines which pages are accessible. If any pages are, the message

PAGES z y x

is output — where x, y and z are hex page numbers.



Note: broken line (- - -) indicates optional interface.

Figure 12-3. Configuration for the Hog Mode Test

12.4.2 Tests to be Performed

The program next requests identification of the test to be performed by outputting

TEST

The user may respond by making a utility function request (see section 12.6); otherwise the user responds with a command sequence to perform a test. The command sequence is

TEST[,PAGE[,LOCATION[,PATTERN]]]

where

TEST is one of the following single alphabetic characters:

- A Unique Address Test
- D Decode Test
- E Execution Test
- G Unique Address, Grouped Bit Memory, Decode, Page Branch, Stack, and Execution Tests
- H Hog Mode Test
- M Grouped Bit Memory Test
- P Page Branch Test
- S Stack Test

PAGE is an optional parameter specifying the WCS page to be tested (on default, all pages are to be tested)

LOCATION is an optional parameter specifying the data pattern with which to test (on default all precanned patterns are to be used in the test)

PATTERN is an optional parameter specifying the location within the page which is to be tested (on default, all locations are to be tested)

Once the command sequence is input, the program outputs the message

CYCLES?

The user responds with the number of cycles to be performed (0 indicates continuous testing). This is followed by a period to suppress an end cycle message, or a comma to print an end cycle message.

12.4.3 Execution of the Test Sequence

After the number of cycles has been specified, execution of the first test program in the sequence begins. The entire sequence is repeated the required number of times and return is made to the state where a new sequence can be input. The execution may be interrupted at any point by the setting of SENSE switch 3. This causes a return to the request for a new test sequence.

12.5 ERROR MESSAGE FORMATS

12.5.1 Unique Address Test

In the event of an error occurring during the Unique Address Test, the following error message is output:

```
UNIQUE ADDRESS ERROR     CYCLE aaaaaa
T/F PP,ADDR CONTENTS
b c dd eeee ffff
```

where

a is the current cycle
b is T for a true test, or F for complement test
c is the page number
d is the location within the page
e is the most significant half of the data read (should be the same as c if b = T; if b= F, e should be the complement of c)
f is the least significant half of the data read (should be the same as d if b= T; if b= F, f should be the complement of d).

12.5.2 Grouped Bit Memory Test

In the event of an error occuring during the Grouped Bit Memory test, the following error message is output:

```
MEMORY ERROR     CYCLE aaaaaa
T/F PP,ADDR  PATTERN  CONTENTS
b c,dd eeee ffff gggg hhhh
```

where

a is the current cycle
b is T for true test, or F for complement test
c is the page number
d is the location within the page
e is the most significant half of the pattern
f is the least significant half of the pattern
g is the most significant half of the location read (should equal e if b= T; if b= F, g should be the complement of e)
h is the least significant half of the location read (should equal f if b= T; if b= F, h should be the complement of f).

12.5.3 Decode Test

In the event of an error occurring during the Decode Test, the following error message is output:

```
PAGE a DECODE ERROR: f=bb ccc
```

where

a is the WCS page
b is the function code (see table 12-2)
c is the contents of the registers involved and the resulting value.

12.5.4 Page Branch Test

In the event of an error occurring during the Page Branch Test, the following error message is output:

```
PAGE BRANCH ERROR. PAGES a-b
```

where

a and b are the pages involved

12.5.5 Stack Test

In the event of an error occurring during the Stack Test, the following error message is output:

```
STACK TEST ERROR, PAGE a
```

where

a is the page under test at the time of the error

12.5.6 Execution Test

In the event of an error occurring during the Execution Test, the following error message is output:

```
EXECUTION TEST ERROR, PAGE a
```

where

a is the page under test at the time of the error

12.5.7 Hog Mode Test

The message

```
MASTER OR SLAVE (M/S)
```

is output as this test begins. The operator types

```
S
```

on the console keyboard, energizing the slave processor. The bell associated with the slave processor starts to ring periodically at this point (see section 12.1.2). Next the operator types

```
M
```

on the keyboard, energizing the master processor. This causes the interval between rings of the slave processor bell to lengthen.

The operator terminates the Hog Mode Test by use of SENSE Switch 3. By manually halting the slave processor, the operator returns the master processor to the test message mode.

Note: do not attempt to run both master and slave processors simultaneously with any other test.

12.6 UTILITY FUNCTIONS

The user may respond to the

TEST

message (see section 12.4.2) with a utility function request. This is done by typing

- C to change memory pattern 6
- X to have system flow return to MAINTAIN III Executive
- ? to examine or change contents of WCS

12.6.1 Changing Memory Pattern 6

To change memory pattern 6 (see table 12-1), type

Caaaa,bbbb

where

a and b are sixteen bit hex values for the high and low parts of the pattern

12.6.2 Returning to MAINTAIN III Executive

To return to the MAINTAIN III Executive, type

X

12.6.3 Examining/Changing WCS Contents

To examine or change the contents of WCS, type

?a,bbbb

where

- a is the hex page to be displayed
- b is the hex location within the page

This causes output of the following:

a bb cccc dddd

where a and be are as above, c and d are the high and low half contents.

To change the contents, respond with a new c followed by a comma, and a new d followed by a comma for display of the next location or followed by a period to terminate the command.

The previous c or d is retained should an empty new c or d be returned.

SECTION 13
CACHE MEMORY TEST PROGRAM

Cache memory and its supporting logic exist to increase processing speed in V70 series computers. This is done by storing in cache memory the 1024 most frequently used words of main memory and fetching these words at a faster rate than possible with only main memory. The total time for the processor to request a word, transfer the word from cache to processor, and get ready for the next request is 371 nanoseconds. An efficient cache algorithm ensures that data required by the processor is in the cache better than 90 percent of the time.

A test program cannot be practically written to detect all of the faults on a given board. The merit of the cache memory test program lies chiefly in its power to detect a high percentage of "stuck-at" faults, which are a very high proportion of all faults. A "stuck-at" fault is characterized by an input or output of a logic element being "stuck" at either logical one or zero.

From a functional point of view, the cache is almost transparent to software and has virtually no functions to test. But there are thousands of possible stuck-at faults that will cause some program to either fail or run slower than normal.

For example, consider the testing of a 48-bit adder. From a functional point of view it may be possible to check that the adder adds, and even that it adds certain pairs of numbers; but it is not possible to test that the adder correctly adds all 2^{96} possible pairs of numbers. Using the stuck-at approach it can be shown that approximately 100 pairs of numbers adequately test the adder in full.

In the cache command repertoire, the cache enable and disable commands allow a test program to load a word into cache, change the associated memory word without changing cache, and then check for a "hit" by reading the word and checking that the word came from cache, not from memory. A "miss" can be detected in a similar manner.

The page PROM enable and disable commands allow the program to use any physical address for the tag RAM and tag compare logic tests. The purge command sets the cache to an initial state.

The V70 MAINTAIN III Cache Memory Test Program is of a modular design with individual subprograms to test each of the features of cache. For example, one subprogram is a tag comparison test. Depending on user-specified parameters, error conditions will cause looping, restarting or continuation.

13.1 HARDWARE SUMMARY

The V70 Series Cache Manual (where the revision number x is an integer 0-9) provides hardware information of the Cache Memory system.

The minimal normal configuration to run the MAINTAIN III Cache Memory Test Program is

a. 1 V70 series processor
b. 16K main memory
c. an extended instruction set
d. 1 cache memory (76-3500)
e. 1 megamap (76-3301)
f. 1 teletype-compatible operator communication device
g. 1 program input device

13.2 TEST PROGRAM ORGANIZATION

The MAINTAIN III Cache Memory Test Program consists of the MAINTAIN III executive program and several subprograms. The initialization component loads the megamap and prompts for subtest number. The executive component allows the calling of the individual subprograms, each of which is initially in megamap executive mode with logical addresses equal to physical addresses in map zero.

Figure 13-1 depicts the Cache system flow of control. Table 13-1 presents the full array of Cache subtests.

13.3. DESCRIPTIONS OF CACHE DISABLED SUBTESTS

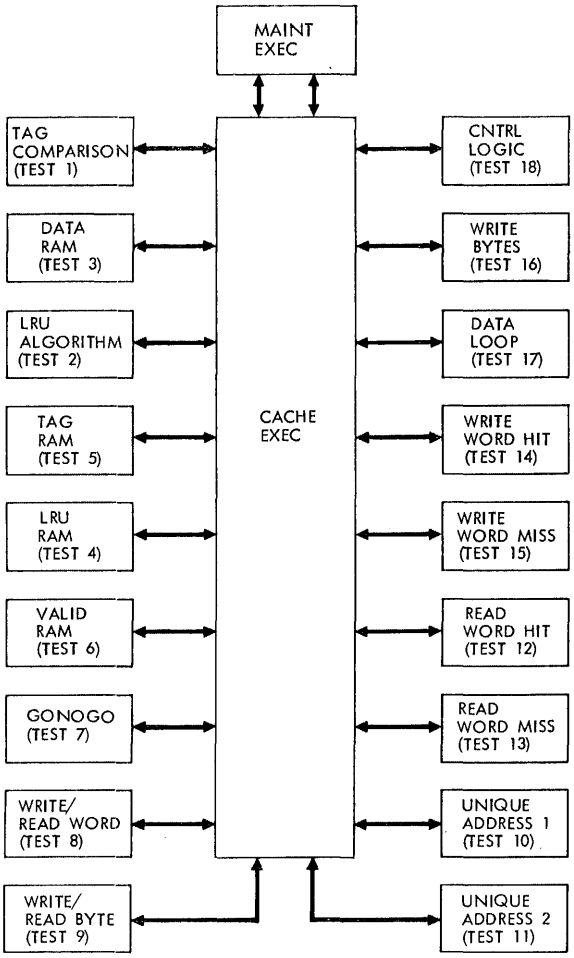
The following tests are run prior to the cache enabled tests.

13.3.1 Go-No-Go Test

Go-No-Go executes all other subtests and hence executes tests with cache enabled as well as tests with cache disabled.

13.3.2 Write/Read Word Tests (with Cache Disabled)

This test validates the uncached memory and word-addressing logic before preceeding with the cache tests. With cache disabled a word of zeros is written, then read and compared. A word of ones is written, then read and compared.



VTII-3653

Figure 13-1. Cache System Flow of Control

Table 13.1. Subtests of the Cache Test Program

Number	Name
1	Tag Comparison Test
2	Least Recently Used Logic Test
3	Data RAM Test
4	LRU RAM Test
5	Tag RAM Test
6	Valid RAM Test
7	Go-No-Go Test
8	Write/Read Word Test (Cache Disabled)
9	Write/Read Byte Test (Cache Disabled)
10	Unique Address Test 1 (Cache Disabled)
11	Unique Address Test 2 (Cache Disabled)
12	Read Word Hit Test
13	Read Word Miss Test
14	Write Word Hit Test
15	Write Word Miss Test
16	Write Bytes Test
17	Data Loop Multiplexor/Driver Test
18	Control Logic Test

13.3.3 Write/Read Byte Test (with Cache Disabled)

This test requires the extended instruction set. The test validates the uncached memory and byte-addressing logic before proceeding with the cache tests. With cache disabled, a left byte is written, then the word is read and compared for left byte changes only. A right byte is written, then the word is read and compared for right byte changes only.

13.3.4 Unique Address Test 1 (with Cache Disabled)

The following test checks the address path through the memory address latch (MA) and drivers (MYAA) on the megamap board, and checks the ability to latch the address.

> Starting at the lowest usable memory address and counting up, write 32K unique data words into 32K sequential addresses (skipping over addresses that contain test program code and skipping over addresses for which there is no memory). Read and check the pattern. For systems with less than 32K words of memory, test the maximum memory size of the system.

13.3.5 Unique Address Test 2 (with Cache Disabled)

This test is identical to Unique Address Test 1 (13.3.4) save that 16 addresses are counted up each time.

13.4 DESCRIPTIONS OF CACHE ENABLED SUBTESTS

The following tests check out the cache system with cache enabled.

Some routines are describable in terms of test patterns and several basic routines. A test pattern consists of a test pattern number, an address (tag and index), a cache data word (which is either to be loaded into cache or is the expected data for a read hit), and a memory data word (which is stored in memory or is the expected data for a read miss). The basic routines are as follows (mnemonic abbreviations in parentheses).

a. **Load Cache using pattern x** (Lx)
This routine loads the cache data word into cache and stores the memory data word at the associated memory address. At the specified address: write cache data word, enable cache, read data word, disable cache, check data word, write memory data word. If there is no memory for the specified address then enable cache, read word (zeros), disable cache, enable cache, write cache data word, disable cache.

b. **Check for a hit using pattern x** (Hx)
At the specified address: enable cache, read data word, disable cache, check that data word equals cache data word.

c. **Check for a miss using pattern x?** (Mx)
At the specified address: enable cache, read data word, disable cache, check that data word equals memory data word. (If memory is not connected, then the data word should be zero.)

d. **Purge cache and delay until purge is completed** (P)
A complete purge lasts 40 microseconds. It is important that the index field of the instruction address be zero during the delay. That is, the decrement register instruction must be at an address with index field equal to zero, followed by a conditional jump to the decrement register instruction.)

e. **Enable cache, write memory data word, disable cache?** (WHx, WMx)
H and M indicate that the cycle should be a hit or a miss, respectively.

Two pairs of routines are required to read or write with cache enabled. One pair must be located so that the index field is less than 100 (this pair is used to read and write into upper index locations). The other pair of routines is located such that the index field is greater than 300_8 (this pair is used to read and write into lower index locations). In the read routines, the index field during the memory cycle prior to the test memory cycle is important. Each of the index field bits should be zero in one of the two read routines.

When the notation "word 0" or "word 1" is used in a test description, the test must be performed twice as follows:

	First Pass	Second Pass
word 0	000000 (data bits = 0)	001001 (parity bits = 0)
word 1	177777 (data bits = 1)	177777 (parity bits = 1)

13.4.1 Read Word Hit

A "hit" means that the requested word is in cache. Memory is initialized with zeros when cache is disabled, then read with cache enabled and the data checked for a hit. This is repeated using a pattern of ones. (A "miss" is an error in this test.) Refer to table 13-2 for a more detailed description.

13.4.2 Read Word Miss

A "miss" means that the requested word is not in cache. Memory is initialized with zeros when cache is disabled. Cache is then purged followed by a cache enable. Memory is then read and the data checked for a read miss. This is repeated for a pattern of ones. (A "miss" is not an error in this test.) See table 13-2.

13.4.3 Write Word Hit

After purging cache, loading it with zeros, then checking for a hit of zeros, cache is enabled, a data word of ones is written into memory and a read-bypass for ones is executed. Then a write bypass of zeros is executed and a check for a hit of ones is made. A second pass, using data which is the complement of the first pass data, completes the test. (A "miss" is an error.) See table 13-2.

13.4.4 Write Word Miss

Cache is purged, loaded with zeros, checked for a hit of zeros, then purged again causing a write miss of ones. This is followed by a read bypass of ones and a write bypass of zeros for a hit of zeros. A second pass consists of the above test, but with the data complemented. (A "miss" is not an error.) See table 13-2.

13.4.5 Write Bytes

This test checks the ability to write bytes with cache enabled. (The extended instruction set is required.) The left byte of a word is written with cache enabled, then the expected data is compared to the actual. The test is repeated for the right byte. See table 13-2.

Table 13-2a. Cache Preliminary Test

TEST PATTERNS

Pattern	Tab	Index	Cache Data	Memory Data
0	A	a	"word 0"	unique
0'	A	a	"word 0"	"word 0"
1	A	a	"word 1"	unique
1'	A	a	"word 1"	"word 1"

TEST SEQUENCES

READ HIT, READ MISS	(Start, loop on error) P,LO,HO,P,MO P,LI,HI,P,MI
WRITE HIT	P,LO,HO,WHI,RBI;WBO;HI P,LI,HI,WHO,RBO;WBI;HO
WRITE MISS	P,LO,HO,P,WMI;RBI;WBO,MO P,LI,HI,P,WMO;RBO;WBI,MI

Table 13-2b. Cache Byte Test

TEST PATTERNS

Pattern	Tag	Index	Cache Data	Memory Data
0	A	a	001011	002022
1	A	a	004044	004044
2	A	a	004011	004022
3	A	a	001044	002044

TEST SEQUENCES

Start, Loop on error
P,LO,WHI(left byte only),RB2,H2
P,LO,WHI(right byte only),RB3,H3

13.4.6 Tag Comparison Test

The "tag" is the most significant 12 bits of the megamap address word. It is stored in one of the four sets of cache corresponding to the address formed by the eight least-significant bits of the megamap word (index).

In this test a tag is loaded into a tag RAM (one of four sets). If a requested word is in cache (using the same tag)

then the tags are equal and the comparator output is correct. Using a new tag with all but one bit the same, the requested word should not be found in cache, verifying the comparator logic. The test is repeated for all twelve tag bits and for all four sets. Refer to table 12-3.

13.4.7 Least-Recently-Used Logic Test

The LRU logic test exhaustively checks every condition - action in the LRU condition - action table.

Five test patterns are required for the LRU test (test pattern numbers 0, 1, 2, 3, and 8). Each test pattern consists of a unique physical address, unique cache data word, and unique memory data word. The five addresses must have the same index but different tags.

Three fundamental routines are required:

Li (i = 0,1,2,3,8 = pattern number)
Loads pattern i into cache as follows: Store cache data word at physical address, enable cache, read data word, disable cache, check the data word, store memory data word.

LPj (j = 0,1,2,3 = set number)
Same as Li except the test pattern number Pj is from table 12-5a, initial pattern orders.

Hi (i = 0,1,2,3,8 = pattern number)
Check for a hit on pattern i as follows: Enable cache, read data word, disable cache, check that data word is cache data word, not memory data word.

Table 13-4 specifies sequences of operations that are required to test LRU logic. Table 13-4a is a list of 34 initial pattern orders (ON = 0,...,33) corresponding to the 34 possible states of the LRU code. NV is the number of valid sets, ON is the order number, and P0, P1, P2, P3 are the pattern numbers.

Each LRU test consists of a purge, followed by one of the test sequences in table 13-4b, followed by one of the check sequences in the table 13-4c. The test is repeated for every combination of initial test pattern order (table 13-4a), test sequence that has the same number of valid sets (NV) as the pattern order (table 13-4b), and check sequence (table 13-4c). Each pattern order in table 13-4a corresponds to one of the lines (initial LRU codes) in the LRU condition-action table. Each test sequence (except TN-0, which is no action in table 13-4b corresponds to one of the actions (columns 3-7) in the LRU condition action table. The final pattern order before the check sequence is 0123. Check sequence CN-0 verifies that all four patterns are in cache. Each of the other check sequences verifies the position of one of the patterns in the pattern order.

Table 13-3. Tag Compare Tests

SET b.	TEST	Purge	Move Test Set to LRU Position	Load Test Tag	Check Hits	Check Tags Equal	Check One Tab Bit Different
0	0	P	- - -	L5	- - -	H5	M6
0	1	P	- - -	L6	- - -	H6	M5
1	0	P	- - L2	L5	- - H2	H5	M6
1	1	P	- - L2	L6	- - H2	H6	M5
2	0	P	- L1 L2	L5	- H1 H2	H5	M6
2	1	P	- L1 L2	L6	- H1 H2	H6	M5
3	0	P	L0 L1 L2	L5	H0 H1 H2	H5	M6
3	1	P	L0 L1 L2	L6	H0 H1 H2	H6	M5

NOTES:
- **P** Means Purge
- **Li** Means Load Pattern i
- **Hi** Means check hit, Pattern i
- **Mi** Means check miss, Pattern i

The 8 sequences shown above are repeated 12 times, once for each tag bit. The tags for patterns 5 and 6 must differ only in the bit position being tested.

Index is the same in all patterns. Five unique data words are required. Tags for patterns 0,1,2 are unique.

LRU Example 1:

If ON = 33, TN = 3, CN = 3, then the initial test pattern order is $P_0 = {}_3$, $P_1 = {}_2$, $P_2 = {}_1$, $P_3 = {}_0$ and the test sequence is:

Purge, LP_0, LP_1, LP_2, LP_3, H0, H3, H1, H2, H3, H0, H1, L8,

Purge LP_0, LP_1, LP_2, LP_3,	Initializes cache, setting valid flags and LRU code to 0. Loads the 4 test patterns into cache, pattern 3 to set 0, pattern 2 to set 1, pattern 1 to set 2, and pattern 0 to set 3. The LRU code order is 0123 (set 0 is LRU, set 3 is MRU) and the corresponding pattern order is 3210.
H0, H3, H1, H2	Four hits on patterns 0312 change the pattern order to 0312 and the corresponding LRU code order to 3021, which is one of the initial states in the LRU condition-action table.
H3	A hit on pattern 3 (set 0) is one of the actions in the LRU condition-action table. The pattern order is 0123 and corresponding LRU code order is 3210.
H0, H1	The pattern order is always 0123 at the start of the check sequence. Check sequence CN = 3, verifies that pattern 2 is in the third position. Two hits on patterns 01 move pattern 2 (set 1) to the LRU position. The pattern order is 2301 and corresponding LRU code order is 1032.
L8	Replaces pattern 2 (set 1) with pattern 8 (set 1). The pattern order is 3018 and corresponding LRU code order is 0321.
H0, H1, H8, H3	Four hits on patterns 0183 verify that pattern 2 (set 1) was replaced by pattern 8 (set 1) and, therefore, pattern 2 (set 1) was in the third position.

Table 13-4. LRU Algorithm Test Procedures.

a. Initial Pattern Orders

NV	ON	SET 0 P_0	SET 1 P_1	SET 2 P_2	SET 3 P_3
0	0	·	·	·	·
1	1	0	·	·	·
2	2	0	1	·	·
2	3	1	0	·	·
3	4	0	1	2	·
3	5	0	2	1	·
3	6	1	0	2	·
3	7	1	2	0	·
3	8	2	0	1	·
3	9	2	1	0	·
4	10	0	1	2	3
4	11	0	1	3	2
4	12	0	2	1	3
4	13	0	2	3	1
4	14	0	3	1	2
4	15	0	3	2	1
4	16	1	0	2	3
4	17	1	0	3	2
4	18	1	2	0	3
4	19	1	2	3	0
4	20	1	3	0	2
4	21	1	3	2	0
4	22	2	0	1	3
4	23	2	0	3	1
4	24	2	1	0	3
4	25	2	1	3	0
4	26	2	3	0	1
4	27	2	3	1	0
4	28	3	0	1	2
4	29	3	0	2	1
4	30	3	1	0	2
4	31	3	1	2	0
4	32	3	2	0	1
4	33	3	2	1	0

b. Test Sequences

PURGE → SET UP INITIAL CONDITION

START LOOP ON ERROR

NV	TN	LOAD PATTERNS INTO CACHE: SET 0	SET 1	SET 2	SET 3	CHANGE ORDER TO SET UP INITIAL LRU CODE: LRU PATTERN MRU PATTERN	ACTION	LOAD REMAINING NON-VALID SETS: SET 0	SET 1	SET 2	SET 3
0	0	·	·	·	·	·	·	L0	L1	L2	L3
1	0	LP_0	·	·	·	H0	·	·	L1	L2	L3
1	1	LP_0	·	·	·	H0	H0	·	L1	L2	L3
2	0	LP_0	LP_1	·	·	H0 H1	·	·	·	L2	L3
2	1	LP_0	LP_1	·	·	H0 H1	H1	·	·	L2	L3
2	2	LP_0	LP_1	·	·	H1 H0	H1	·	·	L2	L3
3	0	LP_0	LP_1	LP_2	·	H0 H1 H2	·	·	·	·	L3
3	1	LP_0	LP_1	LP_2	·	H0 H1 H2	H2	·	·	·	L3
3	2	LP_0	LP_1	LP_2	·	H0 H2 H1	H2	·	·	·	L3
3	3	LP_0	LP_1	LP_2	·	H2 H0 H1	H2	·	·	·	L3
4	0	LP_0	LP_1	LP_2	LP_3	H0 H1 H2 H3	·	·	·	·	·
4	1	LP_0	LP_1	LP_2	LP_3	H0 H1 H2 H3	H3	·	·	·	·
4	2	LP_0	LP_1	LP_2	LP_3	H0 H1 H3 H2	H3	·	·	·	·
4	3	LP_0	LP_1	LP_2	LP_3	H0 H3 H1 H2	H3	·	·	·	·
4	4	LP_0	LP_1	LP_2	LP_3	H3 H0 H1 H2	H3	·	·	·	·
4	5	LP^*_0	LP^*_1	LP^*_2	LP^*_3	H8 H0 H1 H2	L3	·	·	·	·

* If LPj = 3, use L8 instead of LPj

c. Check Sequences

CN	CHECK NEXT CODE								
0	·	·	·	·	H0	H1	H2	H3	CHECK PATTERNS 0,1,2,3.
1	·	·	·	L8	H8	H1	H2	H3	PATTERN 0 IS LRU
2	H0	·	·	L8	H0	H8	H2	H3	PATTERN 1 NEXT
3	H0	H1	·	L8	H0	H1	H8	H3	PATTERN 2 NEXT
4	H0	H1	H2	L8	H0	H1	H2	H8	PATTERN 3 IS MRU

MOVE PATTERN CN-1 TO LRU POSITION.

REPLACE LRU

CHECK HITS

NV means number of valid sets

ON means starting pattern order number

TN means test number

CN means check number

LPj means load pattern Pj into cache set i.

Hi means check for hit on pattern i.

Li means load pattern i into cache.

13.4.8 Data RAM Test

Table 13-5 specifies a write-complement-after read test for the data RAM. The test checks that every cell can hold a one or a zero and checks that every cell is correctly addressed. At each address, the state of the selected cell is read and checked; then the complement state is written into the cell before incrementing (or decrementing) the address. Because part of the test routine is in the RAM, the test is done in two parts. In the first part, the lower addresses are tested, and the test routine is in the upper addresses. In the second part, the upper addresses are tested; and the test routine is in the lower addresses.

13.4.9 LRU RAM Test

The LRU RAM Test uses the same method as the Data RAM Test (see section 13.4.8). Refer to table 13.6 for details.

13.4.10 Tag RAM Test

The Tag RAM Test uses the same method as the Data RAM Test (see section 13.4.8). Refer to tables 13.7a, b, c for details.

13.4.11 Valid RAM Test

The Valid RAM Test uses the same method as the Data RAM Test (see section 13.4.8). Refer to table 13-8 for details.

13.4.12 Data Loop Test

The following tests are intended to detect stuck-at-faults in the data loop multiplexors and drivers (faults that might not be detected by the RAM tests). The tests are repeated twice as follows:

	First Pass	Second Pass
"word 0"	000000 (data bits = 0)	001001 (parity bits = 0)
"word 1"	177777 (data bits = 1)	177777 (parity bits = 1)

Read miss (word 1) followed by a write hit (word 0). All cycles between the read miss and write hit must be read hits. Check that both the word written into cache and the word written into memory are word 0.

13.4.13 Go-No-Go Test

This test executes all other subtests and therefore executes tests with cache enabled as well as tests with cache disabled.

a. Read miss (word 1) followed by a read hit (word 0). All cycles between the read miss and read hit must be read hits. Check that the word in cache is word 1.

b. Read miss (word 1) followed by a read miss (word 0). All cycles between the two read misses must be read hits. Check that the word read and the word in cache are both word 0.

c. Purge, read miss (word 1), purge, write miss (word 0). The write miss must be at the same address as the read miss. Check that the word written into memory is word 0.

d. Execute two back-to-back write hits using word 0 as the first word and word 1 as the second word. Check that the words are written into cache and memory.

e. Execute two back-to-back write hits using word 1 as the first word and word 0 as the second word. Check that the words are written into cache and memory.

13.4.14 Control Logic Test

This tests replacement inhibition during I/O transfer. Four words are loaded into cache at the same index. Using MAP DMA, a word is transferred from memory (same index with cache enabled) to MAP. The 4 cache words are checked for hits. The word which was transferred to the MAP is checked.

Table 13-5. Data RAM Write Complement-After-Read Test

a.

INDEX	OPERATION AT EACH INDEX
000-277	PURGE (Loop on WRITE 0 error)
000-277	CHECK 0, WRITE 1
000-277	CHECK 1, WRITE 0
277-000	CHECK 0, WRITE 1
277-000	CHECK 1, WRITE 0
277-000	CHECK 0, WRITE 1
100-377	PURGE (Loop on WRITE 0 error)
100-377	CHECK 0, WRITE 1
100-377	CHECK 1, WRITE 0
377-100	CHECK 0, WRITE 1
377-100	CHECK 1, WRITE 0
377-100	CHECK 0, WRITE 1

(continued)

Table 13-5. Data RAM Write Complement-After-Read Test (continued)

NOTES: Two passes are required,
data for patterns 0123 is "word 0".
Data for patterns 4567 is "word 1".

First Pass		Second Pass	
"word 0"	000000(Data bits = 0)	"word 0"	001001(Parity bits = 0)
"word 1"	177777(Data bits = 1)	"word 1"	177777(Parity bits = 1)

OPERATION	DEFINITION	REMARKS
WRITE 0	L0, L1, L2, L3	Load 4 patterns.
CHECK 0, WRITE 1	H0, H1, H2, H3. L4, L5, L6, L7.	Hit 4 patterns. Load 4 complement patterns.
CHECK 1, WRITE 0	H4, H5, H6, H7 L0, L1, L2, L3.	Hit 4 complement patterns. Load 4 patterns.

NOTES: Li means load pattern i.
Hi means check hit on pattern i.
Require 8 unique patterns. Data words for patterns 4567 are complement of data words for patterns 0123, respectively.

Table 13-6. LRU RAM Write-Complement-After-Read Test

Repeat following operations for CN = 0,1,2,3,4.

a.

INDEX	OPERATION AT EACH INDEX		
		PURGE	(loop on error)
000-277		WRITE 0	
000-277	CHECK 0	WRITE 1	
000-277	CHECK 1	WRITE 0	
277-000	CHECK 0	WRITE 1	
277-000	CHECK 1	WRITE 0	
277-000	CHECK 0	WRITE 1	
		PURGE	(loop on error)
100-377		WRITE 0	
100-377	CHECK 0	WRITE 1	
100-377	CHECK 1	WRITE 0	
377-100	CHECK 0	WRITE 1	
377-100	CHECK 1	WRITE 0	
377-100	CHECK 0	WRITE 1	

b.

OPERATION	DEFINITION	REMARKS
WRITE 0	L0, L1, L2, L3	Load 4 patterns.
CHECK 0, WRITE 1	LRU check sequence, L3, L2, L1, L0, H0, H1, H2, H3.	See table 13-7c. Load 4 patterns, reverse order. Hit 4 patterns (inverts LRU code).
CHECK 1, WRITE 0	LRU check sequence, L3, L2, L1, L0 H0, H1, H2, H3.	Same as CHECK 0, WRITE 1

NOTES: Li means load pattern i.
Hi means check hit on pattern i.
Requires 5 unique patterns (unique tag and data).

Table 13-7. Tag RAM Write-Complement-After-Read Test

a.

INDEX	OPERATION AT EACH INDEX	
		PURGE (loop on error)
000-277		WRITE 0
000-277	CHECK 0,	WRITE 1
000-277	CHECK 1,	WRITE 0
277-000	CHECK 0,	WRITE 1
277-000	CHECK 1,	WRITE 0
277-000	CHECK 0,	WRITE 1
		PURGE (loop on error)
100-377		WRITE 0
100-377	CHECK 0,	WRITE 1
100-377	CHECK 1,	WRITE 0
377-100	CHECK 0,	WRITE 1
377-100	CHECK 1,	WRITE 0
377-100	CHECK 0,	WRITE 1

b.

OPERATION	DEFINITION	REMARKS
WRITE 0	L0, L1, L2, L3.	Load 4 patterns.
CHECK 0, WRITE 1	H0, H1, H2, H3 L4, L5, L6, L7.	Hit 4 patterns. Load 4 complement patterns.
CHECK 1, WRITE 0	H4, H5, H6, H7 L0, L1, L2, L3.	Hit 4 complement patterns. Load 4 patterns.

NOTES: Li means load pattern i.
Hi means check hit on pattern i.
Requires 8 unique patterns for each of 3 passes. Physical addessess are shown in table 12-7c.

Table 13-7c. Physical Addresses for Tag RAM Test

SET NO.	0	1	2	3	0	1	2	3
PATTERN NO.	0	1	2	3	4	5	6	7
PASS 1	0030000	0031000	0032000	0034000	0037400	0036400	0035400	0033400
TEST BITS	8	9	10	11	8	9	10	11
PASS 2	0030000	0031000	0032000	00034000	0140000	0141000	0142000	0144000
TEST BTIS	12	13	14	15	12	13	14	15
PASS 3	0030000	0031000	0032000	0034000	3630000	3631000	3632000	3634000
TEST BITS	16	17	18	19	16	17	18	19

The test is designed so that the error message can usually be analyzed to determine which RAM is bad. For example, if the error message indicates pass 3 and physical address 0034000 or 3634000, then the RAM for set 3, bits 16-19, is suspect.

Table 13-8. Valid RAM Write-Complement-After-Read Test

a.

INDEX	OPERATION AT EACH INDEX
	PURGE (loop on error)
000-277	INITIALIZE
	PURGE
000-277	CHECK 0, WRITE 1
000-277	CHECK 1
	PURGE (loop on error)
277-000	INITIALIZE
	PURGE
277-000	CHECK 0, WRITE 1
277-000	CHECK 1
	PURGE (loop on error)
100-377	INITIALIZE
	PURGE
100-377	CHECK 0, WRITE 1
100-377	CHECK 1
	PURGE (loop on error)
377-100	INITIALIZE
	PURGE
377-100	CHECK 0, WRITE 1
377-100	CHECK 1

b.

OPERATION	DEFINITION	REMARKS
INITIALIZE	L4, L5, L6, L7	Load 4 patterns.
CHECK 0, WRITE 1	M4, M5, M6, M7 L0, L1, L2, L3	Check miss on 4 patterns. Load 4 new patterns
Check 1	H0, H1, H2, H3	Check hit on 4 new patterns.

NOTES: Li means load pattern i. Mi means check miss on pattern i. Hi means check hit on pattern i. 8 unique patterns (tag and data).

13.5 OPERATING PROCEDURES

The MAINTAIN III Test Executive must be loaded before the Cache test program will correctly operate. Standard subroutines in the Test Executive are called by the Cache test program.

a. Load the Test Executive, which includes the binary object tape loader, per the procedure outlined in this manual.

b. Ready the Cache test object program for loading.

c. Press Key L on the Teletype keyboard, followed by a period. This commands the Test Executive to load the program and begin execution of the test.

The following standard sense switch options will be used during execution of the Cache test program:

Switch	Set	Reset
SS1	Suppress error printouts	Print error messages
SS2*	Halt on error	No halt on errors
SS3	Return to previous parameter request	Continue test

* SS2 can be used to loop on an error halt, or to continue the test.

After the halt:

a. To continue to the next error halt, keep SS2 set and press START on the computer.

b. To loop on the error condition, reset SS2 and press START on the computer. Looping will continue until sense switch 2 is set. If the error condition clears up, looping will still continue until SS2 is set.

The '←' character may be used to delete the previous character typed. The ' \ ' character may be used to delete the current line being entered. If any parameter or character is input which is unacceptable to the test program, the message 'INVALID' is output and the request is repeated.

The Cache Test Program begins by identifying itself and requesting a map of memory. The program then requests the number of the subtest to be run. The user enters the subtest number followed by a period. The program then requests the number of cycles the subtest is to be run. The user enters the number of cycles followed by a period. If a period and no number is entered, the subtest will run until terminated by toggling sense switch 3. Termination of each subtest is indicated by a request for a new subtest number. Toggling sense switch 3 while the program is waiting for a new subtest number returns control to the start of the program. Entering a period instead of a subtest number causes an exit to the MAINTAIN EXEC.

The following is an example of a test run. Operator entries are underlined, comments in parentheses.

```
THIS IS THE V75 TEST EXECUTIVE
MEMORY SIZE IS 32K
L. (Load the Cache Test Program)
V70 CACHE MEMORY TEST
ENTER CACHE/MAP DEVICE ADDRESS . (Default value of 046)
CONFIGURE MEMORY  Y = YES, PERIOD = NOY (must be configured once)
ENTER MEMORY CONFIGURATION IN DECIMAL K
FROM 0. TO 63.
FROM . (Period-only, terminates prompting)
ENTER SUBTEST NUMBER 1.
CYCLES = 1.
TAG COMPARE TEST
ENTER SUBTEST NUMBER
```

The progress of the Tag Comparison Test and Tag RAM Test can be observed on the computer control panel. The tag bit position is displayed in the panel lights, beginning with bit 8 (bit zero of the Tag field) and progressing through bit 15 to bit 3 (bit 11 of the Tag field). The Tag RAM Test advances four bits at a time, the Tag Comparison Test one bit at a time.

13.6 SUMMARY OF TEST PARAMETER REQUESTS

V70 CACHE MEMORY TEST

```
ENTER CACHE/MAP DEVICE ADDRESS n.
```

Where 'n' is either 46 or 56. (Default is 46). The test will also accept non-standard device addresses, leaving the burden of their correctness with the test operator.

```
ENTER SUBTEST NO. n.
```

Where 'n' is any decimal number 1 through 18. A period-only causes exit to the MAINTAIN EXEC.

```
CYCLES = n.
```

Where 'n' is any decimal number 1 through 32768. This input refers to the number of subtest iterations requested. A period without decimal number is a request for continuous testing.

```
ENTER MEMORY CONFIGURATION IN DECIMAL K
FROM n.   TO n.
FROM n.   TO n.
FROM .
```

Where 'n' is any decimal number 0 through 1023. Entering a period without a number terminates the prompting.

13.7 ERROR MESSAGE FORMATS

The Error-Reporting Format for all tests is:

```
ERa PHYSICAL PAGE NO. xxxx

ERa TEST b ADDR. c EXPECT d ACTUAL e
```

Where:

- a is the ERROR-CODE
- b is the SUBTEST NUMBER
- c is the MEMORY ADDRESS
- d is the EXPECTED DATA
- e is the ACTUAL DATA

Some tests may not have applicable data. In those cases the space will be blank. When applicable, the physical page number where the error occurred, will be typed. Set and Tag fields are added for subtests Tag Comparison and Tag RAM.

```
ER a TEST b ADDR.c EXPECT d ACTUAL e ST f TG g
```

where:

- f is the set value (0-3)
- g is the TAG bit (Tag Comparison Test) or group of 4 Tag bits (Tag RAM)

13.8 ERROR CODES

Code	Test or Routine	Test No.	Description
ER01	Valid RAM Tag Comparison	1,6,7	A read miss should have occurred, but did not
ER02	Several	1,2,4,6,7	A read hit should have occurred, but did not.
ER03	LMR	1,8,7	A word previously loaded by megamap was incorrect, when it was read back by megamap.
ER04	Load	1,2,3,4,5,6,7	Cache did not contain the word expected on a cache load with memory present.
ER05	Load	1,2,3,4,5,6,7	Cache did not contain a zero value on a cache load with memory not present.
ER06	WRWDS4	7,8	With cache disabled, a data word was changed by storing it and reading it back.
ER07	Write/Read Bytes	7,9	With cache disabled, a data byte was changed by storing it and reading it back.
ER08	Unique address 1 Unique address 2	7,10,11	With cache disabled, a data word containing its own address, was changed by storing it and reading it back.
ER09	Read word hit	7,12	A read hit should have occurred, but did not.
ER10	Read word miss	7,13	Incorrect memory data word following a read miss.
ER11	Write word hit	7,14	Incorrect memory data word following a write hit.
ER12	Write word hit	7,14	Incorrect cache data word following a write hit.

Code	Test or Routine	Test No.	Description
ER13	Write word miss	7,15	Incorrect memory data following a write miss.
ER14	Write word miss	7,15	Incorrect cache data following a write miss.
ER15	Write bytes	7,16	Byte write miss error.
ER16	Write bytes	7,16	Byte write hit error (memory data).
ER17	Write bytes	7,16	Byte write hit error (cache data).
ER18	Read word miss	7,13	Incorrect cache data word following a read hit.
ER19	Data loop 1	7,17	Incorrect cache data word following back-to-back read hits.
ER20	Data loop 1	7,17	Incorrect memory data word following a write hit.
ER21	Data loop 1	7,17	Incorrect cache data word following write hit.
ER22	Data loop 2	7,17	Incorrect cache data word following a read hit.
ER23	Data loop 3	7,17	Incorrect processor data word following a read hit.
ER24	Data loop 3	7,17	Incorrect cache data word following a read hit.
ER25	Data loop 4	7,17	Incorrect memory data word following a write miss.
ER26	Data loop 5	7,17	Incorrect word 0 in memory following back-to-back write hits of word 0 in first word and word 1 in second word.
ER27	Data loop 5	7,17	Correct word 1 in memory following back-to-back write hits of word 0 in first word and word 1 in second word.
ER28	Data loop 5	7,17	Incorrect word 0 in cache following back-to-back write hits of word 0 in first word and word 1 in second word.
ER29	Data loop 5	7,17	Incorrect word 1 in cache following back-to-back write hits of word 0 in first word and word 1 in second word.
ER30	Data loop 6	7,17	Incorrect word 1 in memory following back-to-back write hits of word 0 in second word and word 1 in first word.

Code	Test or Routine	Test No.	Description
ER31	Data loop 6	7,17	Incorrect word 0 in memory following back-to-back write hits of word 0 in second word and word 1 in first word.
ER32	Data loop 6	7,17	Incorrect word 1 in cache following back-to-back write hits of word 0 in second word and word 1 in first word.
ER33	Data loop 6	7,17	Incorrect word 0 in cache following back-to-back write hits of word 0 in second word and word 1 in first word.
ER34	Control logic	7,18	Cache data change during megamap operation (SET 0).
ER35	Control logic	7,18	Cache data change during megamap operation (SET 1).
ER36	Control logic	7,18	Cache data change during megamap operation (SET 2).
ER37	Control logic	7,18	Cache data change during megamap operation (SET 3).
ER38	Control logic	7,18	Cache data change of map word during megamap operation.
ER39	Control logic	7,18	Incorrect map word read back after loading (cache enabled).

SECTION 14
MEGAMAP PROGRAM

The Megamap Test Program tests the memory map and megamap for V70 series computers. It is comprised of the following six modules, five of them subtests.

The subtests:

a. Miscellaneous Register Test

b. Map Register Test

c. Executive States Access and Map Select Test

d. Memory Test

e. Memory Protect Test

The sixth component:

f. Map Utility Routines

The Megamap Test Program operates with the MAINTAIN III test executive, residing with it in the first 12K of physical memory.

The user employs a Teletype unit to operate the program. He may specify the memory map configuration at the start of the test by implementing a certain initialization procedure.

Section 14.2 provides a summary of the test hardware configuration.

Errors normally are reported by error messages output to the Teletype printer. Section 14.3 details specific error messages and the nature of error reporting control as determined by sense switch selections.

Definitions of certain terms used in this section:

Logical Address	An address in a logical memory area.
Logical Memory	A set of memory locations used by the programmmer. Logical memory may or may not have contiguous physical memory locations.
Map Numbers	Numbers 0 through 15, assigned to the maps used by the operating system and the various users. The numbers are determined by four key bits originating from either the BIC, PMA, map key register or processor (using writable control store microprogramming).
Mapping	The process of translating a logical memory address to a physical memory address.
Page	A 512-word block of physical memory.
Physical Address	An address in physical memory.
Physical Memory	Random access memory defined by hardware.
Privileged Instruction	Any instruction that causes a memory protection violation when in the user mode (e.g., halt and I/O instructions). The halt instruction is only permitted in the inactive mode.
Swapping	The process of moving data between main and auxiliary memory in order to multiplex the use of main memory.

Information of related interest is provided by the V70 Series Memory Map Manual and the V70 Series Megamap Manual.

14.1 DESCRIPTIONS OF MEGAMAP COMPONENTS

14.1.2 Map Register Test

This component validates basic communication with the memory map board. All states of the following registers are set and verified: K, VE, and EMS.

Each state of the specified register is established via a Write Key Control (function 1) output instruction. Verification is made by examining the contents of the respective bits through a READ STATUS input.

This component is executed with the memory map disabled.

14.1.1 Miscellaneous Register Test

This component tests the 1024 map registers and the DMC/DMA logic of the memory map used for loading and reading back the register contents.

There are five subtests in this component:

1. Unique Address and Unique Address Complement Test.
2. Alternate Ones (012525) Test.
3. Alternate Ones (005252) Test.
4. Worst-Case Register Chip Test. (Walking one and walking zero).
5. Selective register addressing test.

This component is executed with the memory map off (disabled).

14.1.3 Executive States Access and Map Select Test

This component validates the fetch and store operations for the four Executive Mode States, and verifies that user maps can be correctly transferred to and executed from.

14.1.4 Memory Test

The Memory Test component tests all memory specified during initialization. The Memory Test consists of up to nine user specified subtests executed in ascending numerical sequence. The nine subtests are:

1. Unique Address and Complement Unique Address Test
2. Binary Address Test
3. All Zeroes and Complement Zeroes Test
4. All Ones and Complement Ones Test
5. Checkerboard and Complement Checkerboard Test
6. Worst-Case Pattern and Complement Worst-Case Pattern Test
7. Adjacent Cell Disturbance Test
8. N-squared Test
9. Galloping Diagonal Test

Memory Error is enabled with an EXC 045 command. If the Memory Error option is attached, the program receives an interrupt on a Memory Error. If a Memory Error is detected, the error and associated address are typed (if sense switch setting allows).

A special subtest will test the map swapping control bits of a Map 1. The change and usage bits are tested to verify that they are properly set and reset.

The change bit indicates that a page has been written into since the bit was last reset.

The usage bit indicates that a page has been accessed (written or read) since the bit was last reset; the bit is reset by the hardware whenever the map register is read by the processor to facilitate scanning and counting the frequency of use.

14.1.5 Memory Protect Test

This component verifies that the memory map option prevents specified operations from taking place. The following violations are applicable in both the executive mode and user mode:

1. Halt
2. Write into Read Operand Only Location
3. Write into Read Only Location
4. Jump into Read Operand Only Location
5. Write into Unassigned Location
6. Read an Unassigned Location
7. Fetch an Instruction from Read Operand Only Location
8. I/O Data Transfer which would cause a Write or Unassigned Error

The following is valid in the executive mode, but is a user mode violation:

I/O Instruction Execution

An error is reported on a first-come, first serve basis. When a violation is detected, the instruction in process is allowed to complete. If a write error is detected, the memory write cycle is changed to a read cycle to prevent modification of memory. Upon completion of the instruction in process, the CPU is interrupted.

The interrupt processing routine can obtain the contents of the "program counter plus one" where the memory protect error occurred by setting up the map to read the instruction address register via the FUNC 2 instruction, then reading the register in via an input data transfer instruction (CIA for example). (On the V77-4xx computer, the Instruction Address is found in location 062, and is equal to the program counter.)

For an I/O data transfer error, additional error information is contained in map registers EK, UA, and PD.

For an unassigned error interrupt, the unassigned virtual address is contained in the map unassigned address error register, which can be obtained via an input data transfer.

14.1.6 Map Utility Routines

This component provides a number of utility routines to aid the user in troubleshooting and establishing memory map conditions for the running of other test programs.

The following routines are offered.

- Dump a map's contents. User can specify 1 of 16 maps for dumping to the TTY; starting map address and number of words (1 to 64) can be specified.
- Change the contents of a map register. User specifies map number, register number, and new value.
- Load Map. Reads a BLD formatted paper tape containing map contents into memory, then transfers contents to user specified map.
- Set External BIC key register. User specifies BIC device address and key value.
- Set External BTC key register. User specifies BTC device address and key value.
- Set Memory Map key register. User specifies key value.
- Switch to User Mode. Causes the memory map to switch from executive mode to user mode.
- Cache Enable.
- Cache Disable.
- Cache Purge (V70 through V76 only).
- Test Program Initialization.
- Return to MAINTAIN III Executive.
- Enable ERCC Interrupt
- Disable ERCC Interrupt

14.2 HARDWARE SUMMARY

The Megamap Test minimum hardware configuration is as follows:

a. A V70 series computer with 12K of memory contiguously addressable from 0.

b. The Megamap option for V70 - V76, or the V77 with Map (1024K).

c. 33/35 ASR Teletype

d. An Object Media Loading Device (Paper Tape Reader, Card Reader, Mag Tape Drive)

Optionally, the program can support the following:

e. Cache

f. Memory Error

14.3 OPERATING PROCEDURES

The Megamap Test Program operates in conjunction with the Test Executive Program, which supplies all common I/O routines and other standard MAINTAIN III routines.

Megamap is loaded as follows:

> By the procedure outlined in section 2, load and execute the Test Executive.
>
> Position the Megamp Test Program in the Object Media Input Device.
>
> Press key L on the Teletype, followed by a period, thus commanding the Test Executive to load the tape.

Sense switch options are listed in the following table:

Switch	Set	Reset
SS1	Suppress error printouts	Print error messages
SS2	Halt on error; loop or continue*	Do not halt on error
SS3	Terminate test and return to start of test	Continue test

* SS2 can be used to loop on an error following an error halt, or to continue the test following the halt:

a. To continue to the next error halt, keep SS2 set and press START on the computer.

b. To loop on the error, reset SS2 and press START on the computer. Looping will continue until SS2 is set, then the program continues in the halt on error mode.

When loading of the test program is complete, automatic transfer to the start of the program takes place (program starts at 0500), and the following messages are output to the Teletype printer:

```
THIS IS THE MAP/MEMORY TEST
**
```

The double asterisk indicates that the program is waiting for an input directive. Enter a period (.) to initiate execution of the test; or enter one of the utility directives as desired (see section 14.4 for the utility directives). ←

14.3.1 Map Type Selection

The following message is prompted when a period is typed for the first time:

```
.SELECT MAP TYPE:
1=256K, 2=1024K, 3=V77 WITH MAP
```

The user types in the corresponding number and a period to indicate the type of map and system in use. Type 1 and 2 refer to the memory map or megamap used on the V70 through V76 computers, and type 3 is the megamap used with the V77-4xx.

14.3.2 Memory Configuration Selection

The user is requested to specify memory configuration. The appropriate user response is determined by the type of map hardware. After initial selection and a return to this point in the program, a period (.) input will cause the program to use the previously designated memory values. The memory configuration is optionally used with the memory test. The inputs may be in octal or decimal, (octal is preceded with a 0). The following message is output:

```
MEM CONFIG (OCT/DEC) =
```

(256K) Four decimal numbers are input, denoting the amount of physical memory (0 - 64K) on each of the four busses. The numbers are separated by commas and terminated by a period.

Example: (256K, Type 1)

Configure a Map 1 system for 64K of memory on the first two busses and 16K on the third bus. The total memory is 144K, where 'K' is defined as 1024 words.

```
MEM CONFIG (OCT/DEC) = 64,0100,16,0.
```

(1024K) Up to 8 groups of two decimal numbers are input to indicate the range of available physical memory. The first decimal number of each group indicates the low limit of physical memory, the first number is separated from the second number by a dash (-), and the second number indicates the high limit of contiguous physical memory. The groups of physical memory are separated by commas and terminated by a period.

Example: (1024K, MAP TYPE 2 or 3)

Configure a 1024K system for a 24K and 32K to 64K physical memory. Note: The physical memory is not contiguous. The total memory is 56K where 'K' is defined as 1024 words.

```
MEM CONFIG (OCT/DEC) 0-23,040-077.
```

14.3.3 Map Test Sequence Selection

The sequence of tests if requested with the prompt:

```
MAP TEST SEQ =
```

The operator may respond with a period to indicate that all five tests are to be performed in a sequence from 1 to 5, or the operator may respond with one to five inputs designating the desired sequence to execute the test components. The sequence of parameters must be separated by commas and terminated by a period.

The test number corresponding to each program component is indicated by the following:

Test Number	Program Component
1	Miscellaneous Register Test
2	Map Register Test
3	Executive States Access and Map Select Test
4	Memory Test
5	Memory Protect Test

Example 1:

Run all tests in standard sequence.

```
MAP TEST SEQ =.
```

Example 2:

Run tests 1, 5 and 3 in that order.

```
MAP TEST SEQ = 1,5,3.
```

14.3.4 Memory Test Component Selection

The memory test component is completely skipped over until the user has typed in a period (.), or a 4, to include the memory test in the map test seqeunce.

If the memory test is not included, the program asks for the cycle count.

The first time the memory test is included in the Map Test sequence, it will establish the V70 parity error interrupt location, and the worst-case pattern(s). After this information is established, and when future memory tests are executed, the program will skip over these areas and go directly to 4K module(s) to be tested.

Note: If an error is realized in any of the above areas, the utility command "I", initializes the program so that the information can be reestablished.

The memory test component then requests parameters to configure itself with the message:

```
V70 PARITY ERROR INTERRUPT LOCATION
```

Respond with an octal value for the memory error interrupt location, followed by a period. (If memory error on V77-600 ERCC is not implemented, respond with a period).

The memory test than requests the worst-case core memory patterns with the message:

worst-case pattern(s)

The user responds with up to eight octal patterns, separated by commas and terminated by a period. If the user wishes to use the patterns previously input, he should respond with only a period.

The preset tables of worst-case patterns include:

0203, 04001, 024, 0144, 044, and 03000 for core memory; 07777 for a semiconductor memory; and 041, 031, 0500, 0120040, 021, 07, 044, 0242, 040200, 0262, 0102000, and 03400 for ERCC MEMORY.

If the program is initialized with the "I" utility command, the original preset table will be restored.

The subsequent memory tests will begin with the message:

```
4K MODULE(S) TO BE TESTED =
```

To include all memory above 12K, as defined in the MEM CONFIG (OCT/DEC), a comma (,) only should be typed.

To test the same combination as the previous test, a period (.) only should be typed.

For testing specific 4K memory modules: the digits may be entered in either octal or decimal (where octal numbers are always characterized by a leading zero--e.g., 016 is octal sixteen, 16 is decimal sixteen); the dash (-) may be employed to denote contiguous memory areas; entries are to be separated with a comma (,); and a period (.) is used after the last value to terminate the input.

Modules 0, 1, and 2 are not allowed, as the test program resides in the first 12K of memory. During the memory test, the physical memories are mapped into virtual locations 030000 through 037777, and tested 4K at a time.

Example:

```
4K MODULE(S) TO BE TESTED =
3-15,
17,
19,23.
```

This would test memory locations 030000 through 0177777, (12K to 64K), 021000 through 0217777, (68K to 72K), and 0230000 through 0277777, (76K to 96K).

Note: When using a comma (,) for converting the MEM CONFIG to 4K modules, if more than 253 modules are detected, the program will be initialized, and the user must restart and re-enter the MEM CONFIG. The error message is:

```
MEM CONFIG ERROR, RE-ENTER
```

Strings and individual 4K modules may be input, to include up to 253-4K modules, and there is not validity checking with the MEM CONFIG (OCT/DEC) on individual inputs.

Refer to table 14-2 for referencing decimal 4K modules, to octal addresses.

To determine which memory tests to execute, the program asks:

```
MEMORY TESTS TO EXECUTE =
```

Type one of the following:

Response	Definition
(period)	Specifies that tests 1, 3, 4, 5, 7 and 9 are to be run (used to test semiconductor memories).
(comma)	Specifies that test 1, 2, 3, 4, 5, and 6 are to be run (used to test core memories).
n	Execute Test n (input as many n's as required, separated by commas, and terminated by a period).

Test Number	Program Component
1	Unique Address and Complement Address Test
2	Binary Address Test
3	All Zeroes and Complement Zeroes Test
4	All Ones and Complement Ones Test
5	Checkerboard and Complement Checkerboard Test
6	Worst-Case Pattern and Complement Worst-Case Pattern Test
7	Adjacent Cell Disturbance Test
8	N-Squared Test
9	Galloping Diagonal Test

Note that the Cache Memory was disabled when loading the MAINTAIN III Test Executive. A System Reset enables Cache and Disables Megamap. The Utility Commands may be used to control Cache Memory prior to executing tests.

Also, note that the N-Squared Test is extremely time consuming.

14.3.5 Cycle Count Selection and Program Termination

The user specifies the number of cycles of the Map Test Sequence to be performed in response to the following prompt:

```
CYCLES = n.
```

where n = an octal number specifying the number of cycles the test should run. The terminator can be either a period or a comma. A comma should be employed to allow printing of the message, signifying that a cycle of the test has been completed. A period should be employed to suppress printing of the message. A terminator without a number specifies continuous execution of the program. If a comma terminator is used, the following message is output to the TTY printer upon completion of the current test cycle:

```
END CYCLE n
```

where n is the cycle (octal) just completed

Upon completion of the number of cycles specified, or upon abort termination via SS3, the following end-of-test messages are always printed:

```
TEST COMPLETE
TOTAL CYCLES = nn
```

where nn is the total number of cycles (octal) executed.

Following the above message, the program outputs the ** and waits for a new input directive.

14.4 MAP UTILITY ROUTINES OPERATION

Note that the Map Type must be selected before any utility routines may be used, where all numerical parameters must be expressed in octal.

Input Directive	Description
DM m,r,n.	Dump the contents of map to the TTY printer, starting with register r, and continuing for n registers. m is valid for 0 through 017. r is valid for 0 through 077. n is valid for 0 through 077. The complete map can be dumped by entering only the first (M) parameter and a period (.). If r + n is greater than 077, the dump will terminate after reading the last register of the specified map.
CM m,r,x.	Change the contents of register r of map m to the value of x. A comma terminator (instead of the normal period) allows successive registers to be changed. Incrementing past register 077 of the specified map number is not allowed; the routine will automatically terminate if this is attempted.
LM m.	Load map m from a BLD object format paper tape. The contents are read from the paper tape and transferred to register 0 through 77 of the specified map. Physical memory locations 01000 through 01077 must be used for generating such object tapes. The Test Executive utility routines (C and P) can be used to establish the contents of those locations and then be punched onto paper tape. The execution address used when punching the tape **must** be 0502.
BIC a,x.	Set the BIC key register. The key register is set to value x for the BIC with device address a.
BTC a,x.	Set the PMA/BTC key register. The key register is set to value x for the BTC with device address a.
MK x.	Set the memory map key register to the value x.
UMl.	Switch to user map. Effects a switch from the executive map to location l in the user map. The user map as established by the existing contents of the memory map key register is entered.
CD.	Disables Cache
CE.	Enables Cache
CP.	Purges Cache (V70 through V76 system only)
I	Test Program Initialization. This initializes the Test Program so that the user may change the parameters for: (1) map type; (2) memory configuration; (3) parity interrupt location; and (4) worst-case patterns. It also reestablishes the DMA interrupt subtest for map test 5.
E.	Return to MAINTAIN III Test Executive.
PE	Enable V77-600 ERRCC interrupt
PD	Disable V77-600 ERRCC interrupt

14.5 ERROR INDICATIONS

14.5.1 Input Directive Errors

An illegal input directive or parameter limit error will result in a return to the double asterisk position. Upon detection of an error, the input will be aborted and the ** will be output to signify that a startover of the directive is required.

Each of the five test components has a basic error message format. The first field of each error message is a test identifier T1 through T5 as detailed below. For tests 1, 2, and 4, each of the remaining fields are prefaced with L-, where L is a single letter identifying the field: R = Register, M = Map, E = Expected test data, A = Actual test data, P = Physical memory address, and V = Virtual memory address.

14.5.2 Miscellaneous Register Test Errors

An error detected in this component of the test is reported by the following error message:

```
T1 R-rr E-eeeeee A-aaaaaa
```

where rr is one of the following memory map registers: K, VE, or ES. The expected number eeeeee and the actual number aaaaaa give the test values as implied by the names. The number of significant digits for the expected/actual values are: K = 4, VE = 1, and ES = 4.

14.5.3 Map Register Test Errors

An error detected in the component will be reported by the following error message:

```
T2 M-n R-mm E-eeeeee A-aaaaaa
```

where n = 0 through 017, designating the memory map number (key number) in which the error was detected. The Register number mm = 0 through 077, designating the specific register of map n that was found in error. The expected number eeeeee and the actual number aaaaaa are the test values as implied by the names; the values will be in the range of 000000 through 017777 to cover the 13 flip-flops comprising a memory map register.

14.5.4 Executive States Access and Map Selection Test Errors

An error detected in this component of the program will be reported by the following error message (subject to the exception noted below):

```
T3 ERROR = t, n
```

where t = 0 through 3 to identify the executive mode state under test, or t = 4 to identify map selection testing; and n identifies the type of error.

For executive mode testing:

n	Type of Error
1	Operand Fetch was not from Map 0.
2	Operand Fetch was not from Map 1.
3	Operand Store was not into Map 0.
4	Operand Store was not into Map 1.

For map selection testing: n = 1, 2, 4, 010 indicates the map being tested.

Note: If the instructions comprising this test are not fetched from the Executive map (map 0), the results are unpredictable. Store testing assumes that the fetch operations are working correctly.

14.5.5 Memory Test Errors

An error in this component will be reported by the following error messages, typed once per cycle:

```
TEST ADDRESS EXPECTED ACTUAL CYCLE
```

where:

TEST = address
ADDRESS = physical address of word in error
EXPECTED = expected word
ACTUAL = actual word

In Test 2, the location being modified is type out.

When Test 7 (Circulating Bit) is run, the test number will be 00XX4Y. If XX equals 040, then memory is indicated to have not been preloaded with the desired constant (ones or zeros); if XX is from 00 to 021, then the bit is indicated to have been toggled. Y indicates the the Worst Case Pattern used.

T4 M -n P-p,pppppp V-vvvvvv E-eeeeee A-aaaaaa

- The P-address will contain Xs in the low-order positions and has no meaning.
- The V-address will contain the letters C-bit or U-bit to designate the bit in error.
- The E- and A- numbers will be 000000 to 000001.

14.5.6 Memory Protect Test Errors

A controlled error detected in this component will be reported by the following error message:

```
T5  MP ERROR = m,n
```

where m is the map (key) number (either 0 or 5) in which the error occurred, n is an error code--described in table → 14-1.

An interrupt to an incorrect address results in a HALT if it occurs at physical memory locations 020 through 035--which are the valid map error interrupt locations. For each subtest, the valid or expected interrupt location is loaded with a JMP for normal processing; all invalid locations between 020 and 035 are loaded with a halt code corresponding to the address itself. The V77-4xx only uses locations 020 through 021 for non-jump ERRORs, and locations 026 through 027 for jump ERRORs. The entire area is still set to an HLT that is equal to its address, although the correct V77-4xx location is substituted for the function of the V70's location.

The Memory Protect Test is mainly comprised of two routines, run in succession. First run is the DMA Transfer Complete Interrupt Routine. Next run is the Memory Protect Test Error Routine. (However, since the V77-4xx computer does not have DMA, the DMA Transfer Complete Interrupt Routine is bypassed for that particular system.)

In the DMA Transfer Complete Interrupt Routine, the following message is output if the correct interrupt to location 016 is not received:

```
T5 NO DMA TRANSFER COMPLETE INTERRUPT
```

Most V70 series computers are not wired for this interrupt. Interrupt capability is tested once in such systems (flow of control then moves on). The easiest way to deal with systems wired for the DMA interrupt but found incapable of receiving same is to initialize the program and retest with the SS2 set. This will halt on the error. SS2 should then be reset for the loop on error mode. (SS1 allows or suppresses printing.)

The Memory Protect Test Error Routine forces upon entry a Memory Protect Interrupt via I/O or HLT, thus bringing the map into the Masked Executive States with Memory Protect disabled. Upon return from the error routine, the proper map and state designations must be re-established, as determined by the individual subtests.

If the subtest needed STATUS, the RSTS routine returns with the STATUS in the A Register and also saves it in the location labeled LSTS (last STATUS). The error routine also reads the STATUS, mainly to clear it for the next interrupt; this STATUS is saved in the location labeled ML9C in the routine. The contents of LSTS from the subtest are not destroyed. They may easily be examined if running in the Halt On Error mode.

14.5.7 General Considerations

- There is a subroutine labeled MAP OFF to disable the map from any map, state, or mode.
  This was incorporated with the use of the V77, in which, once in the user mode, the only way out is via System Interrupt or System Reset.
  It uses a HLT 070 to force a Privileged Instruction Interrupt, to get to a known map and state. If you get a 070 HLT, it implies that Memory Protect is not working.
  The Halt is bypassed until the first time the system is mapped. This occurs when executing Map Test 3, 4, or 5.

- Map DMA data transfer is done throughout the program. The following error message can be associated with any of the five test components:

  ```
  SENSE DMA BUSY TIME OUT.
  ```

- The following error message is associated with the Load Map Utility and is output if a checksum error is detected upon loading of the paper tape: CHECKSUM ERROR = aaaaaa, where aaaaaaa is address of record in which the checksum was detected.

14.5.8 Memory Parity Errors

The following messages pertain to systems with Memory Parity Interrupts or V77-600 Systems with ERCC Memory Arrays.

14.5.8.1 Memory Parity Interrupts

Memory Parity Error Interrupts or V77-600 ERCC Double Bit Error Interrupts result in the message

PARITY ERROR AT XXXXXX

Where xxxxxx is the address interrupted.

14.5.8.2 V77-600 ERRCC Error

Detection of an ERCC parity error results in the message

PARITY ERROR IN MODULE M PHYSICAL PAGE PPPPPP

Where M designates the ERCC ARRAY module in error and PPPPPP is the physical page on that module in error.

This message is followed by one of the following:

DOUBLE BIT ERROR (if more than one bit error)

PARITY BIT BB COORDINATES RR, CC
(if single bit error)

where:

BB is parity or data bit in error
RR is row coordinate
CC is column coordinate

14.6 LOADING AND EXECUTION OF MAINTAIN III PROGRAMS IN A MEGAMAP SYSTEM

14.6.1 Unmapped Environment

Any existing test program can be executed in a system with the memory map option when the memory map is disabled. When the map option is disabled (a front panel reset does it), the first 32K of physical memory on bus 0 is connected to the processor; all instructions are permitted, and the system operates as if the map board was not present. A test program should be loaded and executed according to the existing operating procedures.

14.6.2 Mapped Environment

Any existing test program can be executed in a memory mapped system by applying the following sequence:

1. Load the Test Executive Program
2. Load the Memory Map Test Program
3. Execute the Memory Map Test Program to verify the mapping environment
4. Use the Change Map utility routine or the Load Map utility routine to establish the contents of a map and equate the desired physical memory to virtual memory.
   Any user map number (01 through 017) can be used. The memory protect facility of the map option will be disabled, that I/O instructions may operate normally.

14.6.3 Example

Execute the MAINTAIN III Shared Memory Test in a virtual map containing the first 8K of physical memory on bus 0, plus the first 24K of memory on bus 1, using MAP TYPE 1.

1. Load the Test Executive Program
2. Use the change (C) utility of the Test Executive and store the following octal words in memory locations 01000 through 01077: 001000 through 001017 and 001200 through 001257.

   (Bit 9 = 1 of each word sets the pages for full write/read access. Bit 7 = 1 of the second sequence selects the first 24K increment of bus 1).
3. Use the Punch Paper Tape (P) utility and generate the paper tape of the map via the TTY. The directive must be: P1000, 1077, 502.
4. Load the Memory Map Test Program and let it execute for one cycle:

```
THIS IS THE MAP/MEMORY TEST
**
.SELECT MAP TYPE:
1=256K MAP, 2=1024K MAP, 3=V77
   WITH MAP 1.
MEM CONFIG (OCT/DEC) = 32,32,0,0.
  (assumes 32K on bus 0, and 1,
   and no memory on bus 2 or 3.)
MAP TEST SEQ =
V70 PARITY ERROR INTERRUPT LOCATION.

WORST CASE PATTERN(S)

4K MODULE(S) TO BE TESTED =

MEMORY TESTS TO EXECUTE =,
CYCLES = 1,
TEST COMPLETE
TOTAL CYCLES = 000001

**
```

5. Load the paper tape to establish Map 2.

```
LM2.
**
```

6. Set the map key register for map 2.

```
MK2.
**
```

7. Switch to Map 2 which will contain the Test Executive in the first 8K of bus 0.

```
UM14000.
```

8. Press START to start up the Test Executive.

```
THIS IS THE V70/620 TEST EXECUTIVE
MEMORY SIZE IS 32K
```

9. Load and Execute the memory test using the standard procedures.

14.7 MEMORY PROTECTION ERROR DESCRIPTIONS

Descriptions of memory protection errors are listed in table 14.1.

Subtests 1 through 20 and 26 through 31 are applicable to all map types. Subtests 21 through 25 are skipped by V77-4xx computers. Subtests 32 through 52 are applicable only to the V77-4xx. Subtests 43 through 52 follow subtest 31. Subtests 32 through 42 are out of sequence as extensions of the first 31 subtests.

The second interrupt location refers to the V77-4xx computer only, if different than the first location indicated.

Table 14-1. Memory Protection Errors

Error Code n	Test Description	Expected Result
1	Execute a Halt Instruction	Interrupt to location 020
2	Execute an I/O instruction (OAR)	Map 0: No interrupt Map 5: Interrupt to location 022/020
3	Write into a Read Operand Only location	Interrupt to location 024/020
4	Determine if write operation of subtest 3 was changed to a read cycle to prevent memory alteration.	No change in contents of location where the write was directed.
5	Determine if the Instruction Address is tracking the Program Counter	V70's Instruction Address Register = P+1 V77 Instruciton Address location 062 = P
6	Disable memory protection and do a write protect violation.	No interrupt.
7	Determine if write operation of subtest 6 was inhibited.	Content of write location was not changed.
10	Re-enable memory protection and write into a Read Only location.	Interrupt to location 024/020
11	Determine if write operation of subtest 10 was changed to a read cycle to prevent memory alteration.	No change in contents of location where the write was directed.
12	Jump to a Read Operand Only location.	Interrupt to location 026
13	Read an Unassigned location	Interrupt to location 030/020
14	Determine if the read in an unassigned location puts the address into the unassigned register, for V70 through V76 computers; the tracking register, for V77-4xx computers.	UA/TR contains the virtual address of the location where the unassigned error occurred.
15	Write into an Unassigned location.	Interrupt location 030/020
16	Determine if the Write in an unassigned location puts the address into the unassigned register, for V70 through V76 computers; the tracking register, for V77-4xx computers.	UA/TR contains the virtual address of the location where the unassigned error occurred.

(continued)

Table 14-1. Memory Protection Errors (continued)

Error Code n	Test Description	Expected Result
17	Determine if the write operation of subtest 15 was changed to a read cycle to prevent memory alteration.	No change in contents of location where the write was directed.
20	Fetch instruction from Read Operand Only location.	Interrupt to location 032/020
(21-25)	Not used by V77-4xx	
21	DMA output data to an unused map. Transfer to be from locations in an Unassigned page.	Interrupt to location 034/020
22	Check the DMA error stop (SEN #1) and the Reset Map DMA transfer (EXC2 #4) control logic.	Upon entry from subtest 21, Sense DMA Busy - No and Sense DMA Error Stop - Yes, EXC2 #4 resets the error stop sense status.
23	Read the status register and check the EK, UE, and PD fields.	EK = 0, UE = 1, and PD = 1. *See note following subtest 24.
24	DMA input data. Read part of a map and store into a Read Only page.	Interrupt to location 034/020
25	Read the status register and check the EK, UE, and PD fields.	EK = 0*, UE = 0, and PD = 1.
		* The EK field in normal system operation would contain the user key from the BIC or BTC which caused the error. This test uses the Executive's (Map 0) facility to DMA I/O data to the maps for testing this memory protect feature.
26	Set Executive map to state 2. Switch to map 5 and execute a HLT instruction. Determine Executive Mode State when interrupt occurs by doing an executive state operand fetch.	Interrupt to location 020. Executive (Map 0) state = 0.
27	Execute EXC 545 to clear the Executive Mode State Mask. Determine the Executive Mode State.	Executive Map State = 2.

(continued)

Table 14-1. Memory Protection Errors (continued)

Error Code n	Test Description	Expected Result
30	Read (LDAE) locations 077777 and 050000 which are in Unassigned pages.	Interrupt to location 030/020 on both reads. The Unassigned/Tracking Address Register contains 077777 the first pass and 050000 the second, upon detection of the error interrupts.
31	Execute Halts from virtual memory locations 077776, 077777, and 0.	Interrupt to location 020 for each halt. The Instruction Address Register/ Location will equal P+1/P for each halt location tested. The halt at 077777 will set the instruction address equal to 0 for the V70 series computers.
The remaining subtests except number 41 are used only with V77-4xx computers		
32	Check status for privileged instruction after Halt interrupt from subtest 1.	Status MP = 00, AS = 0
33	Check status for privileged instruction after I/O interrupt from subtest 2, Map 5.	Status MP = 00, AS = 0.
34	Check status for Write Protect after Write Protect Interrupt from Subtest 3, using Read Operand Only location.	Status MP = 10, AS = 0.
35	Check tracking register for virtual address violated in subtest 3.	Tracking register equal address of MP violation.
36	Check status for Write Protect after Write Protect Interrupt from subtest 10, using a Read Only location.	Status MP = 10, AS = 0
37	Check status for Unassigned error from subtest 13.	Status MP = 11, AS = 0
40	Check status for Unassigned error from subtest 15.	Status MP = 11, AS = 0
41	ALL - Jump to Unassigned location. Interrupt and check Instruction Address. This follows subtest 17.	Interrupt, and Instruction Address Register/Location = Address of JMP Instructions.
42	Check status for Instruction Fetch error from subtest 20.	Status MP = 01, AS = 0

(continued)

Table 14-1. Memory Protection Errors (continued)

Error Code n	Test Description	Expected Result
43	Check status AS bit = 1 after second interrupt and not reading status on first interrupt. This follows subtest 31.	Status AS = 1
44	Check status A, U, and EMM bits for map active, and mask not being cleared.	Status A = 1 EMM = 1, U = 1 for Map 5 and U = 0 for Map 0.
45	Check status EMM bit after executing Clear Executive Mask and Set Executive Mask commands	Status EMM = 0 after EXC2 0546 and EMM = 1 after EXC2 0446, plus interrupt.
46	Check status for no transition armed from subtest 45	Status TS = 0
(47 through 52 are checked Map 0 only)		
47	Check status for transition to user armed, prior to interrupt.	Status TS = 11
50	Check status for transition to inactive armed, prior to interrupt.	Status TS = 10
51	Check status for active with map off.	Status A = 0
52	Check EXC2 0746 command for sending status on next input. Status check includes A, U, EMM, EMS, AS, and K fields, after the interrupt.	Status A = 1 U, EMM, EMS, AS, and K = 0.

14.8 4K MEMORY MODULES IN TERMS OF OCTAL ADDRESSES

Table 14-2 gives the decimal representation of 4K memory modules.

Table 14-2. Decimal Representation

MODULE	MEMORY ADDRESSES	MODULE	MEMORY ADDRESSES
3	000030000-000037777	46	000560000-000567777
4	000040000-000047777	47	000570000-000577777
5	000050000-000057777	48	000600000-000607777
6	000060000-000067777	49	000610000-000617777
7	000070000-000077777	50	000620000-000627777
8	000100000-000107777	51	000630000-000637777
9	000110000-000117777	52	000640000-000647777
10	000120000-000127777	53	000650000-000657777
11	000130000-000137777	54	000660000-000667777
12	000140000-000147777	55	000670000-000677777
13	000150000-000157777	56	000700000-000707777
14	000160000-000167777	57	000710000-000717777
15	000170000-000177777	58	000720000-000727777
16	000200000-000207777	59	000730000-000737777
17	000210000-000217777	60	000740000-000747777
18	000220000-000227777	61	000750000-000757777
19	000230000-000237777	62	000760000-000767777
20	000240000-000247777	63	000770000-000777777
21	000250000-000257777	64	001000000-001007777
22	000260000-000267777	65	001010000-001017777
23	000270000-000277777	66	001020000-001027777
24	000300000-000307777	67	001030000-001037777
25	000310000-000317777	68	001040000-001047777
26	000320000-000327777	69	001050000-001057777
27	000330000-000337777	70	001060000-001067777
28	000340000-000347777	71	001070000-001077777
29	000350000-000357777	72	001100000-001107777
30	000360000-000367777	73	001110000-001117777
31	000370000-000377777	74	001120000-001127777
32	000400000-000407777	75	001130000-001137777
33	000410000-000417777	76	001140000-001147777
34	000420000-000427777	77	001150000-001157777
35	000430000-000437777	78	001160000-001167777
36	000440000-000447777	79	001170000-001177777
37	000450000-000457777	80	001200000-001207777
38	000460000-000467777	81	001210000-001217777
39	000470000-000477777	82	001220000-001227777
40	000500000-000507777	83	001230000-001237777
41	000510000-000517777	84	001240000-001247777
42	000520000-000527777	85	001250000-001257777
43	000530000-000537777	86	001260000-001267777
44	000540000-000547777	87	001270000-001277777
45	000550000-000557777	88	001300000-001307777

(continued)

Table 14-2. Decimal Representation (continued)

MODULE	MEMORY ADDRESSES	MODULE	MEMORY ADDRESSES
89	001310000-001317777	132	002040000-002047777
90	001320000-001327777	133	002050000-002057777
91	001330000-001337777	134	002060000-002067777
92	001340000-001347777	135	002070000-002077777
93	001350000-001357777	136	002100000-002107777
94	001360000-001367777	137	002110000-002117777
95	001370000-001377777	138	002120000-002127777
96	001400000-001407777	139	002130000-002137777
97	001410000-001417777	140	002140000-002147777
98	001420000-001427777	141	002150000-002157777
99	001430000-001437777	142	002160000-002167777
100	001440000-001447777	143	002170000-002177777
101	001450000-001457777	144	002200000-002207777
102	001460000-001467777	145	002210000-002217777
103	001470000-001477777	146	002220000-002227777
104	001500000-001507777	147	002230000-002237777
105	001510000-001517777	148	002240000-002247777
106	001520000-001527777	149	002250000-002257777
107	001530000-001537777	150	002260000-002267777
108	001540000-001547777	151	002270000-002277777
109	001550000-001557777	152	002300000-002307777
110	001560000-001567777	153	002310000-002317777
111	001570000-001577777	154	002320000-002327777
112	001600000-001607777	155	002330000-002337777
113	001610000-001617777	156	002340000-002347777
114	001620000-001627777	157	002350000-002357777
115	001630000-001637777	158	002360000-002367777
116	001640000-001647777	159	002370000-002377777
117	001650000-001657777	160	002400000-002407777
118	001660000-001667777	161	002410000-002417777
119	001670000-001677777	162	002420000-002427777
120	001700000-001707777	163	002430000-002437777
121	001710000-001717777	164	002440000-002447777
122	001720000-001727777	165	002450000-002457777
123	001730000-001737777	166	002460000-002467777
124	001740000-001747777	167	002470000-002477777
125	001750000-001757777	168	002500000-002507777
126	001760000-001767777	169	002510000-002517777
127	001770000-001777777	170	002520000-002527777
128	002000000-002007777	171	002530000-002537777
129	002010000-002017777	172	002540000-002547777
130	002020000-002027777	173	002550000-002557777
131	002030000-002037777	174	002560000-002567777

(continued)

Table 14-2. Decimal Representation (continued)

MODULE	MEMORY ADDRESSES
175	002570000-002577777
176	002600000-002607777
177	002610000-002617777
178	002620000-002627777
179	002630000-002637777
180	002640000-002647777
181	002650000-002657777
182	002660000-002667777
183	002670000-002677777
184	002700000-002707777
185	002710000-002717777
186	002720000-002727777
187	002730000-002737777
188	002740000-002747777
189	002750000-002757777
190	002760000-002767777
191	002770000-002777777
192	003000000-003007777
193	003010000-003017777
194	003020000-003027777
195	003030000-003037777
196	003040000-003047777
197	003050000-003057777
198	003060000-003067777
199	003070000-003077777
200	003100000-003107777
201	003110000-003117777
202	003120000-003127777
203	003130000-003137777
204	003140000-003147777
205	003150000-003157777
206	003160000-003167777
207	003170000-003177777
208	003200000-003207777
209	003210000-003217777
210	003220000-003227777
211	003230000-003237777
212	003240000-003247777
213	003250000-003257777
214	003260000-003267777
215	003270000-003277777
216	003300000-003307777
217	003310000-003317777
218	003320000-003327777
219	003330000-003337777
220	003340000-003347777
221	003350000-003357777
222	003360000-003367777
223	003370000-003377777
224	003400000-003407777
225	003410000-003417777
226	003420000-003427777
227	003430000-003437777
228	003440000-003447777
229	003450000-003457777
230	003460000-003467777
231	003470000-003477777
232	003500000-003507777
233	003510000-003517777
234	003520000-003527777
235	003530000-003537777
236	003540000-003547777
237	003550000-003557777
238	003560000-003567777
239	003570000-003577777
240	003600000-003607777
241	003610000-003617777
242	003620000-003627777
243	003630000-003637777
244	003640000-003647777
245	003650000-003657777
246	003660000-003667777
247	003670000-003677777
248	003700000-003707777
249	003710000-003717777
250	003720000-003727777
251	003730000-003737777
252	003740000-003747777
253	003750000-003757777
254	003760000-003767777
255	003770000-003777777

SECTION 15
V77-800 MICRODIAGNOSTIC TEST PROGRAM

The V77-800 Microdiagnostic Test Program of the MAINTAIN III tests all microinstruction controls of the V77-800 central processor. A description of the central processor and the microinstruction control fields is contained in the V77-800 Processor Functional Analysis and Servicing Manual. The diagnostic program verifies the operational status of the central processor control logic and assists in locating faults. All available Writable Control Store (WCS) sizes (up to eight pages) can be used during the test. Provisions are also incorporated for the use of the Floating Point Processor (FPP).

Since operation of the processor, WCS, and FPP are so interrelated, some of the tested functions are WCS and FPP functions. Whether data is being tested in the processor, WCS or FPP is essentially transparent to the user.

The diagnostic test program is normally provided on a punched paper tape suitable for loading from a teletypewriter or high speed paper-tape reader. Other media are available (e.g., card deck, magnetic tape). The program is normally loaded and executed via virtual console (CRT or TTY) keyboard commands.

The test program operates with the MAINTAIN III Test Executive Program, both residing in the first 12K of main memory. The test executive supplies all common I/O routines and other standard MAINTAIN III routines. Since the test executive is the software interface, it must be loaded and operational before the microdiagnostic test program can be loaded. Loading procedures are outlined in Section 2.

The standard sense switch option controlled by the supervisor component is not used. Sense switch options are presented in Section 15.3.

Tests are run in sequence on a single selected page. Error data is displayed when errors are detected on the selected page. Sense switch settings or operator intervention for an uncontrollable interrupt will alter the test sequence.

Errors are reported via error messages which are displayed or printed at the operator's station. All available information is provided prior to a program jump to the test executive. Re-entry points to the test program are provided for continuation of testing.

15.1 HARDWARE REQUIREMENTS

The microdiagnostic test minimum hardware configuration is as follows:

a. V77-800 computer with at least two 512-word pages of WCS and 64K of main memory.

b. A virtual console (CRT or TTY terminal).

c. A program loading device such as a paper-tape reader, card reader, or magnetic tape drive.

d. Floating point processor (optional).

15.2 DESCRIPTION OF TEST COMPONENTS

The diagnostic program consists of two components:

a. Supervisor

b. Test

15.2.1 Supervisor

The supervisor component provides the user interface, loads the microinstruction tests into WCS, and controls execution of the program and the individual tests. The operator specifies the following data at the beginning of the test:

a. The number of WCS pages available.

b. Whether or not the FPP option is to be tested.

c. The page of WCS which is to be used for execution of the major portion of the test.

During the test after an error has been detected, the operator can specify:

a. Test number to execute after an error condition.

b. Page to dump if contents of WCS are desired.

c. Number of microwords to dump if WCS is being dumped.

d. Microinstruction WCS address and bit contents if an instruction is to be changed.

The major segments of the supervisor component are:

SEGMENT TITLE	MNEMONIC
Program Entry	START
Initialization	INIT
Test load and execution	TL1
Test restart	NPTST
Error data display	ERRDSP
WCS dump	WCSDMP
Microinstruction alteration	ALTMI

15.2.1.1 Program Entry

The program entry segment provides five entry points which the operator can use to access the various segments of the supervisor component. The entry points and their purpose are:

ENTRY ADDRESS	PURPOSE
G0500	Normal program entry to execute test series. Jump to INIT.
G0502	Alternate program entry to restart specific test after error detection. Jump to NPTST.
G0504	Program entry to display all available data and return to test executive after error detection has halted program execution. Jump to ERRDSP.
G0506	Program entry to dump a portion of WCS and return to test executive. Jump to WCSDMP.
G0510	Program entry to enter a microinstruction into WCS and restart current test. Jump to ALTMI.

15.2.1.2 Initialization

The initialization segment is used to begin the test program. Jump and mark instructions are stored at the various interrupt locations. The operator is requested to enter the number of WCS pages available and to enter Y or N to indicate if FPP is available. A read/write is made to all pages of WCS up to and including the number of pages indicated as available by the operator. Test 1 is loaded in each page of WCS which was successfully read and written into. All available pages are listed by the computer. The operator is requested to enter the page number in which the tests are to be run. The test page number is inserted as part of the address into all microcode data words. Execution of test 1 is then started.

15.2.1.3 Test Load and Execution

The test load and execution segment displays the test number of the current test, registers R0 through R7 are set to 0 and the test is executed. Upon return from the test, all registers are saved and the values are compared against constants to determine whether or not the test was successful. For an unsuccessful test, all available status information and the contents of all registers are displayed. The expected contents are also displayed. The next test is then loaded into WCS and executed. When the last test (test 31) has been executed, a program completed message is displayed.

15.2.1.4 Test Restart

The test restart segment provides an entry point from the test executive after an error condition has caused a jump to the executive. The operator selects any test number from 2 to 31 as the test to restart the test run. Parameters for the selected test are set up by the computer and a program jump is made to the normal routine for setting up that test.

15.2.1.5 Error Data Display

Error data display segment provides an entry point from the test executive to display status and register information. A program return is made to the test executive after information is displayed.

15.2.1.6 WCS Dump

WCS dump segment transfers a selected area of WCS to the virtual console for display. The operator enters the WCS page number and the number of microinstructions to be dumped. The requested data is then dumped to the virtual console and a program return is made to the test executive.

15.2.1.7 Microinstruction Alteration

The microinstruction alteration segment allows the operator to enter one microinstruction into WCS. The operator enters the WCS address and the three word microinstruction. The microinstruction is written into WCS and the program returns to the test executive. The data word in the program corresponding to the changed microinstruction should also be changed if the test is to be resumed with a jump to test restart (NPTST).

15.2.2 Test

The test component consists of a series of firmware tests (microinstruction programs) which are loaded into WCS and executed one at a time. When a test fails to return the correct result, (error detected), all available status and register information is displayed on the operator console. The result which should have been returned is also displayed. Any unexpected interrupt (equipment maulfunction) also causes status and register information to be displayed or allows its display by the operator.

The series of tests are designed to build on one another and test all firmware capabilities during their execution in WCS. Some microcode functions dealing with I/O to various devices or with specific V70/620 instructions are not included in this test but are tested in the instruction tests (Section 3).

Return is made to the MAINTAIN III test executive upon completion of a series of tests or after an unexplained

interrupt. Re-entry points are provided by the supervisor component for continued testing or display of information.

Each instruction test consists of one or more microinstructions. Each test is automatically loaded into WCS at the time it is to be executed and can be loaded into any page of WCS that the operator specifies. Unused portions of the WCS page are filled with an unconditional jump to an error routine. Each test is for a specific function and the tests are sequentially numbered starting with test 1.

The microinstruction tests are listed in Table 15-1.

Table 15-1. Microinstruction Tests

TEST NUMBER	PURPOSE
1	Page jump. Allows for dumping contents of the processor's working-storage registers S1 through S6. Also provides common entry, error, and exit code for all tests.
2	Register swap. Loads registers S1 through S6 with contents of incremented programming registers R0 through R7 then reloads R registers from the S registers.
3	Register file. Tests the class 1 register file function macros and accumulator I/O.
4	ALU B-input. Moves source data of the ALU B-input to the R0 through R7 registers.
5	BS and RF fields. The various uses of the BS and RF fields are tested.
6	Miscellaneous BS field functions. The various uses of the BS field are tested.
7	Left shift. Left shift operations specified by the SRC field are tested.
10	Right shift. Right shift operations specified by the SRC field are tested.
11	Left/right shift 0. Tests the left and right shift 0 instructions.
12	RFSD 1 function. Tests the use of the RFSD field, when it contains a 1, for single sources.
13	RFSD 1 function. Tests the use of the RFSD field, when it contains a 1, for multiple sources.
14	RFSD 2 function. Tests the use of the RFSD field when it contains a 2.
15	RFSD 2 or 3 functions. Tests the use of the RFSD field when it contains 2 or 3.
16	JC functions. Tests various JC field functions by alternating contents of ALU.
17	Miscellaneous functions. Tests miscellaneous JC and SP field functions.
20	Flag functions of JC and SP fields. Tests flag functions by using various functions of the JC and SP fields.
21	Operand fetch. Tests operand fetch from memory.
22	Operand fetch. Tests operand fetch by using indexed and indirect addressing.
23	Operand store. Tests operand store into memory.
24	Operand store. Tests indexed operand store into cache.
25	Stack function. Tests operation of the stack function.
26	Field selection function. Tests operation of the field selection microinstructions.
27	Read/write to TTY/CRT controller. Tests read/ write to TTY/CRT controller with asterisk and bell output.
30	Cache status. Tests the cache status.
31	FPP. Tests the FPP addition and subtraction operation.

15.3 OPERATING PROCEDURES

The MAINTAIN III test executive must be loaded before the microdiagnostic test program can be loaded. Operating procedures for the V77-800 control panel and virtual console are contained in the V77-800 Computer Operations Manual.

Load the test executive in accordance with the procedures outlined in Section 2.

If paper tape is used:

a. Place the microdiagnostic program tape in the paper tape reader.

b. Position the tape to any place within the leader area past the test part number punched in the leader.

c. Enter an L on the virtual console.

d. Enter a period on the virtual console.

If magnetic tape is used:

a. Consult the MAINTAIN III Usage Description Bulletin for the file number of the V77-800 microdiagnostic (microinstruction diagnostic) test.

b. Position tape to that file number by using MAINTAIN III tape commands. The commands are listed in Section 2.

c. Load the test program by using the tape commands listed in Section 2.2.2.1.

The tape loads and a program identification message is displayed.

When loading of the test program is complete, automatic transfer to the start of the program takes place (address 0500).

Sense switch settings can alter the test programs as follows:

a. Sense Switch 1 (SS1). Setting SS1 when the first TEST message is displayed causes a program halt after an error message display. Reset SS1 to continue with next test.

b. Sense Switch 2 (SS2). Setting SS2 suppresses display of error message and causes program to loop on error. Program check of SS2 is made prior to check of SS1. Set SS2 only after program halt on error (SS1 SET) or no error data will be displayed.

c. Sense Switch 3 (SS3). Setting SS3 causes program exit to test executive. Program check of SS3 is made after check of SS2 and SS1.

15.3.1 Program Start

While operating the microdiagnostic test program, all input and output messages are controlled via the virtual console. In the following procedures, responses to be initiated by the operator are underlined R where R indicates a response is to be made. Responses from the computer are not underlined.

The following messages are output to the virtual console:

V77-800 MICRO INSTRUCTION TEST
NUMBER WCS PAGES AVAILABLE? (OCTAL NUMBER)
R.

Enter the number of pages of WCS installed on the computer as an octal number followed by a period. The computer checks to see if the page is available. Entries other than those in the following list can cause unpredictable results. Should a non-valid entry be made, it may be necessary to reload the test executive and the microdiagnostic programs.

The only valid entries are:

OCTAL NUMBER	WCS PAGE
01	8
02	9
03	10
04	11
05	12
06	13
07	14
10	15

The next message to be output is:

FPP AVAILABLE ? (Y/N)
R.

Enter a Y if the FPP option is installed or an N if it is not installed. Entry of an X terminates the program and causes a jump to the test executive. Any other response causes the computer to output INVALID and repeat the message. If N is selected, test 31 is not run.

The computer then outputs:

AVAILABLE PAGES

XX
XX
XX

Where XX is the octal number of each WCS page up to and including the number of pages that were previously entered as available. The numbers indicate the pages that can be successfully written into and read from during subsequent tests. If the NUMBER OF WCS PAGES AVAILABLE entry is higher than what is physically available, the computer will only list the number of pages installed.

15.3.2 Program Execution

Once the initial program entries are made, the computer outputs the message:

PAGE TO EXECUTE TEST 1 IN? (OCTAL NUMBER)
R.

Enter the single WCS page number in which the test is to be conducted. The octal entry is taken from the list following the AVAILABLE PAGES message and is terminated with a period. The period causes an automatic program start.

The following messages will then be output as each test is successfully executed:

TEST NUMBER XX START
TEST NUMBER XX FINISH

Where XX indicates the octal test number (1 through 31). If FPP is not available, test 31 will not be run.

When the first START message has been displayed, SS1 can be set to halt the test sequence after an error has caused status information to be displayed. Reset of SS1 causes the program to continue with the next test.

The only test that requires operator intervention is test 27. When an asterisk (*) is output and an audible tone is generated, the operator must depress the space bar on the virtual console to allow the testing to continue.

Upon completion of all tests the following message is output:

MICRO DIAGNOSTIC TEST COMPLETED

The program then exits back to the test executive (address 0).

15.3.3 Program Continuation

The test program can be restarted by an operator entry of G0500 at the virtual console. The program will restart with the program identification message (Section 15.3.1).

15.3.4 Utility Routines

Utility routines are used by the operator to perform various functions related to the test in progress. They are basically used as an aid in troubleshooting and establishing conditions for tests. The program exits to the test executive upon completion of the routine.

15.3.4.1 WCS Dump

An operator entry of address G0506 initiates a program that allows the operator to transfer an area of WCS to the virtual console for display. The following message is output:

PAGE TO DUMP ? (OCTAL NUMBER)
R.

Enter the octal number of the page followed by a period.

The next output message is:

NUMBER OF MICRO WORDS ? (OCTAL NUMBER)
R.

Enter an octal number (0000 through 0777), followed by a period, to indicate the number of words to be transferred. The requested words are transferred to the virtual console. The format is one microword per line beginning with word 0 and ending with the word that corresponds to the entered number.

15.3.4.2 Input to WCS

An operator entry of address G0510 initiates a program that allows the operator to write a microinstruction into WCS from the virtual console. The following message is output:

MICRO INSTRUCTION - WCS ADDRESS ? (OCTAL NUMBER)
R.

Enter the octal number (WCS address) of the microinstruction followed by a period. Only the word address is required since the computer already knows the page address from the PAGE TO DUMP entry.

The computer then outputs the following message:

BITS 47-32 R.
BITS 31-16 R.
BITS 15-00 R.

A single line of the output message is printed. The program then waits for an operator response. Enter an octal number representing the corresponding bits requested. Terminate the entry with a period. Repeat the procedure as each line is printed out. The three entries (segments) make up the microinstruction to be entered.

The program returns to the test executive after writing the instruction in WCS.

If the test is to be rerun by using a jump to the beginning of the test program (G0500) or an individual test (G0502), the equivalent memory location of the test must be changed in memory. All tests are loaded into WCS from the data words contained in the test program.

15.4 ERROR MESSAGES

Following the detection of an error, the program sends out an error message. The various error messages are outlined in this section.

15.4.1 Input Errors

An illegal input causes the message INVALID to be output. The message is followed by the original message that requested the input.

15.4.2 Invalid Test Data

Each microinstruction test returns specific values in registers R0 through R7 upon completion and exit from a test. If the values returned do not match the expected values an error message is generated. If SS1 is set, the program halts until SS1 is reset. After SS1 is set, the program continues to the next test.

The output error message is:

TEST NUMBER XX ERROR
STATUS INFORMATION
PROCESSOR ST (status word)
CACHE STATUS (status word)
TRACKING REGISTER (contents)
FPP STATUS (status word)
REGISTER CONTENTS
R0 (actual and expected contents)
R1 (actual and expected contents)
R2 (actual and expected contents)
R3 (actual and expected contents)
R4 (actual and expected contents)
R5 (actual and expected contents)
R6 (actual and expected contents)
R7 (actual and expected contents)

The XX is the test number in which the error occurred. The parenthesis indicate what the computer outputs in addition to the standard information.

15.4.3 Unexpected Interrupts

If an unexpected interrupt (error detected) occurs, a program jump to a standard interrupt location occurs. One of the following messages, with the interrupt location, is displayed:

COMMON INTERRUPT (interrupt location)

STEP INTERRUPT (interrupt location)

FPP INTERRUPT (interrupt location)

The message is followed with the error message described in Section 15.4.2.

The SS1 option can be used to halt the program at this point. When SS1 is reset the program exits to the test executive.

15.4.4 Unexplained Interrupts

When an unexplained interrupt or halt occurs, no error message is displayed. An operator entry of address G0504 initiates a program that causes the error message described in Section 15.4.2 to be output. The program then exits to the test executive.

15.4.5 Test Recovery

If the test program goes into an unexpected halt or loop while testing WCS, the system can be reset and the test started again. To restart the test which failed, re-enter the test program with an entry of address G0502. The following message is displayed:

TEST NUMBER TO EXECUTE ? (OCTAL NUMBER)
R.

Enter the octal number of the test to be executed followed by a period. The entry restarts the test program at the beginning of the selected test. Do not enter test 1 at this point as only a test program restart can be used to set up test 1.

Before the selected test is started, the following message is displayed:

PAGE TO EXECUTE TEST IN ? (OCTAL NUMBER)
R.

Enter the octal number followed by a period, of the WCS page in which the test is to be executed. Tests then continue from this point.

SECTION 16
V77-800 CACHE MEMORY TEST PROGRAM

The V77-800 Cache Memory Test Program of the MAINTAIN III tests the cache memory of the SPERRY UNIVAC V77-800 Computer, F3078-0x. A description of the cache memory is contained in the V77-800 Processor Functional Analysis and Servicing Manual.

Cache memory and supporting logic increase processing speed in V77-800 computers. Execution speed is increased by saving data and instructions from main memory and storing the information in fast access cache RAM. Data and instructions are then fetched from cache memory at a greater speed than is possible from main memory.

The Cache test program verifies the operational status of the cache memory and assists in locating faults. The merit of the test program lies chiefly in its power to detect a high percentage of non-operational gates which experience has shown are a very high proportion of all faults. A non-operational gate is characterized by an input or output of a logic element being at a constant level (logic one or zero). Once detected, hardware faults can be isolated to specific components with the aid of one or more tests.

Three subtests (unique word address test, unique double-word address test, and unique byte address test) verify the operation of map functions for single-word, double-word, and byte accessing of main memory. The tests are performed with cache disabled. The subtests are necessary to isolate specific failures to the cache board.

Three other subtests (valid RAM test, tag RAM test, and data RAM test) verify the operation of cache RAM and are used to isolate faults to a specific RAM.

The remaining subtests verify that cache is correctly updated during accessing of the main memory and that data is not changed by the cache operations.

Functionally, cache is verified for single-word, double-word, and byte operations. The cache test does not verify all the hardware on the cache board. Some of the tested cache operations utilize hardware on other boards, particularly the mapping and control board. Instruction fetches pass through cache logic, thus, any program testing cache assumes that cache hardware is working well enough for the program to run. The program thoroughly tests cache functions but is of limited use as a diagnostic tool for hardware faults associated with an instruction fetch.

The cache memory test program consists of the following four test components:

a. Initialization

b. Executive

c. Preliminary subtest

d. Cache subtests.

The preliminary subtests assume that the complete instruction set has been verified by the MAINTAIN III Test Executive. Cache subtests assume the successful completion of the preliminary subtests. Each test component has various error messages which are displayed or printed at the operator's station.

The nature of error reporting control is determined by sense switch selection. The sense switch functions differ slightly from standard MAINTAIN III settings due to cache logic constraints.

16.1 HARDWARE REQUIREMENTS

The cache memory test minimum hardware configuration is as follows:

a. V77-800 computer with cache memory

b. One virtual console

c. One program loading device such as a paper-tape reader, card reader, or magnetic tape drive.

16.2 DESCRIPTION OF TEST COMPONENTS

The initialization component defines mapping and control RAM contents. The executive component prompts the user for subtest numbers and allows the user to call individual subtests.

The subtests, described in the following sections, systematically test most of the cache hardware and some map operations. From a functional point of view, the cache is almost transparent to software and has virtually no functions to test. However, there are thousands of possible faults that will cause some program to either fail or run slower than normal.

In the cache command repertoire, the cache enable and disable commands allow a test program to:

a. Load a word into cache,

b. change the associated memory word without changing cache,

c. check for a "hit" by reading the word,

d. and check that the word came from cache and not from memory.

A "miss" can be detected in a similar manner.

The twelve subtests available to the user are:

Subtest Number	Subtest Name	Cache Status
1	Burn-in	Disabled/enabled
2	Unique word address	Disabled
3	Unique double word address	Disabled
4	Unique byte address	Disabled
5	Valid RAM	Enabled
6	Tag RAM	Enabled
7	Memory disabled	Enabled
8	Data RAM	Enabled
9	Cache hit	Enabled
10	Cache miss	Enabled
11	Cache double word	Enabled
12	Cache byte	Enabled

Tests 2, 3, and 4 are preliminary subtests which verify that each location in main memory is accessible and that there are no constant bits at each location. The tests are performed with cache disabled. Cache is disabled by the test program.

After main memory and map operations are verified, the various cache components are tested with cache enabled. Cache is enabled by the test program.

Tests for valid, tag, and data RAMs verify that RAMs are addressed correctly and that there are no constant bits. The remaining cache subtests verify the correct operation of cache hit and miss logic, byte, and double word I/O. The three cache RAM tests must run successfully prior to executing the remaining tests in order for the latter to be used as an effective diagnostic tool.

Two pairs of routines are required for each subtest to read or write with cache enabled. One pair is located in the upper index locations of cache while the lower index locations are tested. The other pair of routines is located in the lower locations while the upper locations are tested.

Cache subtests are conducted with cache enabled.

16.2.1 Burn-in Test

The burn-in test causes all subtests to be executed. It executes tests with cache disabled as well as tests with cache enabled.

16.2.2 Unique Word Address Test

Unique word address test verifies that each word of main memory can be addressed by single precision instructions. The test checks the address path through the memory mapping and control board by testing the ability to latch an address. The test also verifies that each bit of each addressed word can be loaded with a zero or a one by checking for constant bits in the data paths of each address.

16.2.3 Unique Double-word Address Test

Unique double-word address test verifies that each word of main memory can be addressed by double precision instructions. The procedure is identical to the unique word address test except that double word loads and stores are used and two addresses are counted up each time. The test is performed for double precision data words beginning on odd and even boundries.

16.2.4 Unique Byte Address Test

Unique byte address test validates main memory operation and byte-addressing logic without cache prior to proceeding with the cache enabled tests. With cache disabled, a left byte is written into main memory. The word is read and compared for left byte changes only. A right byte is then written, read, and compared for right byte changes only. The data read from memory should be identical to the data written into memory.

16.2.5 Valid RAM Test

Valid RAM test checks valid RAM bits and addressing. The test verifies that cache is correctly purged when cache is disabled and when the cache purge command is executed. Two passes through cache RAM are necessary to detect possible unique indexing errors.

16.2.6 Tag RAM Test

Tag RAM test checks that every tag RAM bit can be set to a one or a zero and that every location is addressed correctly.

16.2.7 Memory Disabled Test

Memory disabled test verifies that memory can be disabled during cache operations and that a program can be executed with memory disabled. Since memory is disabled during each of the cache tests, a memory disabled test should be executed prior to initiating any other test with cache enabled.

16.2.8 Data RAM Test

Data RAM test checks that every data RAM bit can be set to a one or zero. It then checks that every location is addressed correctly.

16.2.9 Cache Word Hit Test

Cache word hit test verifies that cache logic operates correctly during a read or write hit with cache enabled. The test verifies that data received by the processor during a read hit comes from cache and not from memory. The test also verifies that both cache and memory are properly updated during a write hit. Testing is performed on each available memory page.

16.2.10 Cache Word Miss Test

Cache word miss test verifies that cache logic operates correctly during a read or write miss with cache enabled. The test verifies that only data received by the processor during a read miss comes from memory and not from cache. The test also verifies that data written during a write miss updates main memory but not cache RAM. This test is performed on each available memory page.

16.2.11 Cache Double Word Test

Cache double word test verifies that double word transfers between cache and memory are performed correctly.

The test is performed for double precision data words starting on odd and even addresses. Tests are performed on all available pages of memory.

16.2.12 Cache Byte Test

Cache byte test checks the ability to read and write bytes with cache enabled. The test verifies that cache is correctly updated during byte read hits and misses and during byte write hits and that cache is not changed by a byte write miss. This test is performed on each available page.

16.3 OPERATING PROCEDURES

The MAINTAIN III test executive must be loaded before the cache memory test program can be loaded. Operating procedures for the V77-800 control panel and virtual console are contained in the V77-800 Operations Manual.

Load the test executive in accordance with the procedures outlined in Section 2.

For paper tape systems, the procedure for loading the test program is as follows:

a. Place the cache memory test program tape in the paper tape reader.

b. Position the tape within the leader area between the test part number and the start of the program.

c. Enter an L on the virtual console.

d. Enter a period on the virtual console.

For magnetic tape systems, the procedure for loading is as follows:

a. Consult the MAINTAIN III Usage Description Bulletin for the file number of the V77-800 cache memory test.

b. Position tape to that file number by using MAINTAIN III tape commands (Section 2).

c. Load the test program by using the tape commands listed in Section 2.2.2.1.

The tape loads and a program identification message is displayed.

When loading of the test program is complete, automatic transfer to the start of the program takes place (address 0500).

Sense switch settings can alter the test program as follows:

SENSE SWITCH	SET	RESET
SS1	Suppress error messages	Print error messages
SS2	Loop on error	Continue program
SS3*	Terminate test	Continue testing

* Setting SS3 during the parameter input sequence causes the program to exit to the MAINTAIN III EXECUTIVE.

When entering parameters during program initialization, the back slash (\) character can be used to delete the current line. If any operator entered parameter or character is unacceptable to the test program, the single word message INVALID is displayed. The program request is then repeated.

While operating the cache memory test, all input and output messages are via the virtual console. In the following procedures, responses to be initiated by the operator are underlined R where R indicates a response to be made. Responses from the computer are not underlined.

The test must be restarted if a power failure occurs.

16.3.1 Program Start

The cache test program begins by identifying itself. The identification message is:

V77-800 CACHE MEMORY DIAGNOSTIC PROGRAM

16.3.2 Test Number, Page, and Index Selection

Immediately following the identification message the following test request message is displayed:

TEST R

Enter the number of the subtest to be executed followed by a period or a comma. The subtests are listed in Section 16.2. The user can enter a "?" at anytime during test selection input to review the available cache tests.

If the user terminates the entry with a period, the following logical pages are tested:

Logical Page	Physical Page	Logical Page	Physical Page
00000	0100	03677	0177
00001	0101	03676	0176
00002	0102	03675	0175
00004	0104	03673	0173
00010	0110	03667	0167
00020	0120	03657	0157
00040	0140	03637	0137
00200	0026	03477	0033
00400	0027	03277	0034
01000	0030	02677	0035
02000	0026	01677	0036
02425	0125	01252	0152
01463	0163	02214	0114
00607	0107	03070	0170
03417	0117	00260	0160
00077	0037	03600	0032

Terminating the entry with a comma causes the program to prompt the user with:

PAGE? R

Enter a specific logical test page. Terminate the entry with a period or comma. A period causes all possible indexes to be tested.

Terminating the PAGE? entry with a comma causes the program to prompt the user with:

INDEX? R.

Enter the index to be tested. Testing is limited to the index specified.

16.3.3 Cycle Count Selection

After the test number, page, and index have been selected, the program requests the number of test cycles to run.

The cycle count prompt is:

CYCLES? R.

Enter a decimal number to indicate the number of test cycles to be executed. Terminate the entry with a comma or a period.

Terminating the entry with a comma causes the following message to be displayed at the end of each test cycle:

TEST
TOTAL CYCLES COMPLETED c

where:

TEST
Is the name of the completed test.

c
Is the number of completed test cycles.

If the entry is terminated with a period, the cycle count message is not printed.

For continuous testing, enter a period with no numeric input. Testing will continue until terminated by toggling sense switch 3.

16.3.4 Test Termination

Termination of each subtest is indicated by a TEST? prompt.

To continue testing, enter a new subtest number.

Setting sense switch 3 causes the program to complete the current test and prompt RESET SENSE SWITCH 3.

Testing is terminated by toggling sense switch 3 while the program is waiting for a new subtest number. Control is returned to the MAINTAIN III test executive.

16.4 ERROR MESSAGES

Following the detection of an error, the program displays an error message. The various error messages are outlined in this section.

16.4.1 Error Message Format

The error reporting format for all tests is:

ERa TEST b ADDR (c)d e EXPECTED f
TAG g ACTUAL h

where:

a
Is the error code.

b
Is the subtest number.

c
Is the logical page address.

d
Is the physical page address.

e
Is the physical line address.

f
Is the actual value.

g
Is the expected value.

h
Is the valid bit and tag.

Some tests may not have applicable data. In those cases, the error message contains blanks.

Both words of a double precision value in tests 3 and 11 are displayed in the error message.

16.4.2 Error Codes

The error codes that could appear in the error message are listed in Table 16-1.

Table 16-1. Error Codes

Error Code	Test No.	Description
ER01	2,3,4,7 9,10,11	Memory fault, read-after-write error. Data stored not equal to data read.
ER02	2,3,4	Memory (from ER01) or addressing. Actual word not equal to expected word.
ER07	7	Memory disable. Memory altered during write with memory disabled.
ER09	7	Memory disable. Non-zero value read during read miss with memory disabled.
ER10	8	Data RAM, read-after-write error. Cache data not updated during write.
ER11	8	Data RAM (from ER10) or indexing. Cache data RAM unique addressing error.
ER12	6	Tag RAM or indexing, read-after-write error. Read did not update tag RAM.
ER13	6	Tag RAM or indexing (from ER12). Actual tag not equal to expected tag.
ER15	5	Valid RAM. Valid bit set.
ER17	5	Valid RAM or cache purge. Valid bit not set.
ER19	9,11,12	Hit logic. Read hit data did not come from cache.
ER20	9	Hit logic. Write hit did not update cache data.
ER21	9	Hit logic. Write hit did not update main memory.
ER22	6,9,12	Hit logic. Cache miss should have been a cache hit.
ER23	10,12	Miss logic. Write miss did not update main memory.
ER24	10,12	Miss logic. Read miss data did not come from memory.
ER25	10,12	Miss logic. Tag not updated for read miss.
ER27	10,12	Miss logic. Write miss did not update cache data.
ER31	12	Read-after-write error. Write miss did not update cache tag RAM.

16.4.3 Special Messages

The first portion of the memory disabled test (subtest 7) executes a program with memory, disabled. If the program fails to produce expected results, the following message is displayed:

a EXECUTION ERRORS, b CACHE

where:

a
Is the number of improperly executed instructions.

b
Is either the message UPPER or LOWER.

SECTION 17

V77-800 MAP/MEMORY TEST PROGRAM

The V77-800 Map/Memory Test Program of the MAINTAIN III tests the main memory system of the V77-800 Computer, F3078-0X. A description of the memory system is contained in the V77-800 Memory System Functional Analysis and Servicing Manual. The memory program verifies the operational status of the computer memory and assists in locating faults. All available memory sizes (64K through 1024K) can be tested. Memory size (configuration) is specified at the start of the program. The program assumes that the first 32K of memory and the cache have been tested.

Since operation of the cache and mapping and control boards are so interrelated, some of the tested functions are cache functions. Whether data comes from the cache memory or main memory is essentially transparent to the user.

Format of the memory test program is normally a punched paper tape for loading from a teletypewriter or high speed paper-tape reader. Other media are available (e.g., card deck, magnetic tape). The program is normally loaded and executed via virtual console (TTY or CRT) keyboard commands.

The test program operates with the MAINTAIN III Test Executive Program both residing in the first 16K of main memory. Test executive supplies all common I/O routines and other standard MAINTAIN III routines. Since the test executive is the software interface, it must be loaded and operational before the map/memory test can be loaded. Test executive loading procedures are outlined in Section 2.

The complete map test component should be conducted prior to attempting the memory test component. Operation of the memory test component assumes that the memory map is functional. Memory test component uses the mapping facilities to map each physical 16K words of memory to be tested into its logical second 16K memory space.

Each test component has various error messages which are displayed or printed at the operator's station. The nature of error reporting control is determined by sense switch selection.

17.1 HARDWARE REQUIREMENTS

The map/memory test minimum hardware configuration is as follows:

a. V77-800 computer with 64K of memory contiguously addressable from memory location 0.

b. A virtual console (CRT or TTY terminal)

c. A program loading device such as a paper tape reader, card reader, or magnetic tape drive.

17.2 DESCRIPTION OF TEST COMPONENTS

The map/memory test program consists of four components:

a. Test monitor

b. Utility routines

c. Map test

d. Memory test

17.2.1 Test Monitor

Test monitor is used by the operator to establish the configuration of the physical memory to be tested. The user enters the size of the main memory in use. The size can be entered as either an octal or decimal notation which represents a block of memory.

The test monitor is entered when the test program is initially loaded. The only other time that it is displayed is when a restart (utility routine command I) has been initiated.

17.2.2 Utility Routines

Utility routines are used by the operator to perform various functions related to the main memory system, cache and the test environments. They are basically used as an aid in troubleshooting and establishing main memory and cache conditions for the running of test programs. Capabilities are provided for the operator to load map contents, set key bits, and switch to user mode, allowing tests to be executed in different parts of memory.

17.2.3 Map Test

Map test is used to test the map functions and some of the cache functions. The following subtests are available:

SUBTEST NUMBER	SUBTEST NAME
1	Static register
2	Map RAM
3	Executive mode state and user user key selection
4	Executive key
5	Extended indexing
6	64K mode
7	Memory protect
8	Memory

When prompted, the operator can enter the number or numbers of the subtests to be executed. The entered numbers should be separated by a comma and terminated with a period. A total of eight entries in any sequence is permitted. Regardless of entry sequence, the program will sequentially run from low to high subtest number. Duplicate entries are permitted but serve no useful function as the map test component will only cycle through each subtest once. To repeat tests, utilize the cycle function described in the operating procedures, Section 17.3.4.

If the operator responds with a period or comma (no numeric entry) all subtests are run.

Subtest number 8 is not actually a subtest of the map test component, it is a "transfer link" to the memory test component. If either a period or an 8 is entered, the program will cycle through the map test component, run any tests that have been requested and proceed to the memory test component.

17.2.3.1 Static Register Subtest

Various static cache and map registers are tested by using the processor status word, cache status word and memory map programmed I/O instructions. The tested registers are:

a. Executive key

b. User Key

c. Executive mode state

d. Extended index

Utility routine S is used to print out the cache and processor status words. The interpretation of the words is contained in the utility routine description.

17.2.3.2 Map RAM Subtest

The memory mapping and control board map array is tested as a memory. The thirteen 1K by 1-bit RAMs form 1024 13-bit words. Map RAM subtest performs the following tests:

TEST NUMBER	TEST PERFORMED
1	Unique address
2	Unique complement address
3	Write/Read test patterns
	a. All zeros
	b. All ones
	c. Alternate
	d. Reverse alternate
4	Walk a one through zeros
5	Walk a zero through ones
6	Map word counter
7	Map address register
10	Map memory address register

17.2.3.3 Executive Mode State and User Key Selection Subtest

The functioning of the four executive mode and masked mode states are tested. The test also checks each user key bit by using key 1, 2, 4 and 8. The modes and origins are:

EXECUTIVE MODE STATE	ORIGIN		
	INSTRUCT'N FETCH	OPERAND FETCH	OPERAND STORE
0	Executive	Executive	Executive
1	Executive	Executive	User
2	Executive	User	Executive
3	Executive	User	User

Tests performed and functions tested are:

TEST	FUNCTION TESTED
0	Test for executive mode
1	Fetch executive mode 0 key 0
2	Fetch executive mode 1 key 0
3	Fetch executive mode 2 key 1
4	Fetch executive mode 3 key 1
5	Store executive mode 0 key 0
6	Store executive mode 1 key 1
7	Store executive mode 1 key 1
10	Store executive mode 2 key 0
11	Store executive mode 2 key 0
12	Store executive mode 3 key 1
13	Fetch with executive mask set
14	Store with executive mask set
15	Fetch user mode key 1
16	Fetch user mode key 2
17	Fetch user mode key 4
20	Fetch user mode key 8
21	Load map from virtual memory
22	Store byte executive mode 1 key 1
23	Store byte executive mode 1 key 1
24	Store byte executive mode 2 key 1
25	Store byte executive mode 2 key 1
26	Store byte block executive mode 1 key 1
27	Store byte block executive mode 1 key 1

17.2.3.4 Executive Key Subtest

Executive key bits are tested by verifying the executive keys (1, 2, 4 and 8). Testing is similar to that in subtest 3 except that executive key bits are used in place of user key bits. Tests performed and functions tested are:

TEST	FUNCTION TESTED
1	Fetch executive key 1
2	Store executive key 1
3	Fetch executive key 2
4	Store executive key 2
5	Fetch executive key 4
6	Store executive key 4
7	Fetch executive key 8
10	Store executive key 8

17.2.3.5 Extended Indexing Subtest

The extended indexing subtest cycles through all seven indexing registers by using the four user keys (1, 2, 4 and 8). Unique test data are fetched in the user mode with extended indexing enabled. After all of the extended registers have been tested by a load instruction, the store operation is checked by using register 3 key 1.

17.2.3.6 64K Mode Subtest

Operation of the 64K mode is tested in the three map states:

MAP STATE	TEST FUNCTION
Inactive	Indirect bit becomes address bit 15.
Active Executive	Indirect bit is ORed into key bit 0 to form an odd/even pair of keys.
User	Same action as active executive.

17.2.3.7 Memory Protection Subtest

Memory map's memory protection functions are tested. The four access control states are set into each of four pages. The four states are:

ACCESS CONTROL STATE	DEFINITION
0	Unassigned
1	Full access
2	Read operand only
3	Read only

To test for a write error, the program verifies that the data was not changed. Write function is changed to a read function by hardware. The memory protection errors that are tested:

1. Halt
2. Write into read operand only
3. Write into read only
4. Write into unassigned location
5. Read into unassigned location
6. Jump into read operand only
7. Fetch instruction from read operand only
8. Issue privileged I/O instruction

The following executive call jump instructions are also tested for no interrupt occurrence:

1. JSR373
2. JSR404
3. JSR406

An operational error listing is contained in Section 17.4.

17.2.4 Memory Test

Memory test is used to test the memory array module functions of the main memory system. Memory is tested in 16K word increments starting with the second 16K memory array increment. The first 16K increment contains the test executive and the map/memory test programs, consequently, the increment is not tested by this program. The first 32K of memory should already have been tested by using the memory test program contained in Section 4.

The memory test is divided into six subtests. Any or all of the subtests can be run on any 16K memory increment. There are 64 possible 16K memory increments in a 1024K word memory system. The increments are numbered 0 through 63. Section 17.3.3 contains a table that converts the octal physical address to an octal or decimal 16K increment number to be used in this test. The subtests and functions tested are:

SUBTEST NUMBER	SUBTEST NAME	FUNCTIONS TESTED
1	Unique Address	a. Store addresses b. Test address store complement c. Test complement store address d. Test address
2	Ones and Zeros Patterns	a. Store zeros b. Test zeros store ones c. Test ones store zeros d. Test zeros e. Store ones f. Test ones store zeros g. Test zeros store ones h. Test ones
3	Alternate Patterns	a. Pattern 1 = 0125252 b. Pattern 2 = 052525 c. Store zeros d. Store pattern 1 e. Test pattern 1 store pattern 2 f. Test pattern 2 store pattern 1 g. Test pattern 1 h. Store zeros i. Store pattern 2 j. Test pattern 2 store pattern 1 k. Test pattern 1 store pattern 2 l. Test pattern 2

SUBTEST NUMBER	SUBTEST NAME	FUNCTIONS TESTED
4	Walk One	a. Test pattern all zeros except one location which is set to ones b. Test pattern c. Increment and store pattern d. Test pattern e. Continue increment, store, and test until ones pattern "walks" through all 16K locations
5	Walk Zero	Same as walk one except "walk" zeros
6	Unique Module	a. Store module (being tested) number in module under test b. Test unique module number after all other tests completed

17.3 OPERATING PROCEDURES

The MAINTAIN III test executive must be loaded before the map/memory test program can be loaded. Operating procedures for the V77-800 control panel and virtual console are contained in the V77-800 operations manual.

Prior to operating the map/memory test program, ensure that the first 32K of main memory and the cache have been tested (Sections 4 and 15). After completing the tests, load the map/memory test program into main memory.

Load the test executive in accordance with the procedures outlined in Section 2.

For paper tape systems, the procedure for loading the test program is as follows:

a. Place the map/memory program tape in the paper tape reader.

b. Position the tape within the leader area between the test part number and the start of the program.

c. Enter an L on the virtual console.

d. Enter a period on the virtual console.

For magnetic tape systems, the procedure for loading is as follows:

a. Consult the MAINTAIN III Usage Description Bulletin for the file number of the V77-800 Map/Memory Test.

b. Position tape to the file number by using MAINTAIN III tape commands. The commands are listed in Section 2.

c. Load the test program by using the tape commands listed in Section 2.2.2.1.

The tape loads and a program identification message is displayed.

When loading of the test program is complete, automatic transfer to the start of the program takes place (address 0500).

Sense switch settings can alter the test program as follows:

SENSE SWITCH	SET POSITION	RESET POSITION
1	Suppress error message	Print error message
2*	Before error halt: Halt on error	Before error halt: Do not halt on error Print error and continue
	After error halt: Continue testing	After error halt: Loop on error
3	Terminate test and return to start of test	Continue testing

* SENSE switch 2 can be used to continue the test following an error halt or to loop on the error:

a. To continue to the next error halt, leave the SENSE switch SET and depress R on the virtual console.

b. To loop on the error, RESET the SENSE switch and depress R on the virtual console. Looping will continue until the switch is SET, the program then continues in the "halt on error" mode until the next error halt.

When loading of the Test Program is complete, automatic transfer to the start of the program takes place.

While operating the map/memory test program, all input and output messages are via the virtual console. In the following procedures, responses to be initiated by the operator are underlined R where R indicates a response is to be made. Responses from the computer are not underlined.

17.3.1 Test Monitor Selection

After the map/memory test program is loaded, the following message is output:

THIS IS THE V77-800 MAP/MEMORY DIAGNOSTIC PROGRAM
MEMORY CONFIGURATION (OCT/DEC) =
FROM R. TO R.

Enter the physical size of the available memory to be tested. The size can be entered as either an octal or decimal notation with K understood. To distinguish an octal from a decimal entry, preceed the octal entry with a 0.

Each FROM R. TO R. entry must be contiguous memory, in 16K increments, with the initial operator response a 0. Enter a period to terminate the response. The computer will output TO and await a second response.

As an example, assume that the main memory system has two 128K memory array modules separated by 128K. The entries would be:

OCTAL ENTRY	DECIMAL ENTRY
FROM 0. TO 0177.	FROM 0. TO 127.
FROM 0400. TO 0577.	FROM 256. TO 383.
FROM .	FROM .

The entry of a period after FROM without a numeric entry terminates the operation. The configuration is then monitored during the test to ensure that the operator does not attempt to test a memory module that is not available. The following message is output to indicate the configuration message has been completed:

MM**
R

At this point, any one of the map utility routines listed below can be entered. When the test program is first loaded, both the cache and parity interrupts are disabled. Tests will run in the mode (cache or parity enable/disable) the system is currently in. Cache and parity enable/disable can be operator controlled by utility routine commands CD, CE, PD, or PE. The operator must use the utility routines to enable these functions if required. After utility routines have been entered or if no utility routines are to be used, enter a T. The T. will cause the program to move into the map test component.

INPUT DIRECTIVE	DESCRIPTION
BIC a,x.	Set key register of BIC device address a to value x.
CD	Cache disable. The cache remains disabled until CE is entered.
CE	Cache enable
CM m,r,x.	Change the contents of register r of map m to the value of x. A comma terminator, instead of a period allows successive registers to be changed. Incrementing past register 077 is not allowed, the routine will automatically terminate when attempted.
CP	Cache purge
DM m,r,n.	Dump the contents of map m to the operator's console starting with register r and continuing for n registers. Values for m are 0-017, for r are 0-077, and for n are 0-077. The complete map can be dumped by entering only the first m parameter followed by a period. If r + n is greater than 077, the dump will terminate after reading the last register of the specified map.
E	Return to MAINTAIN III Test Executive by doing a jump to location 0.
I	Initialize test program. Initializes the test program so that the user may change the parameters for memory configuration. Resets all test pointers.
LM m,x.	Load map key m starting with value x. The value is then incremented and stored in the next map register until all 64 words have been stored.
Mkx.	User map key is set to value x.
PD	Parity disable
PE	Parity enable
S*	Print current processor and cache status words.
T	Enter map test sequence routine.
UMi.	Switch from executive mode to user mode at location i. Map key and contents must be set up prior to entering this command.

* The bit definitions of the words are:

a. Processor status word using read command 102045 or write command 103045.

BIT	MNEMONIC	MEANING
0	EXKO	Executive key bit 0
1	EXK1	Executive key bit 1
2	EXK2	Executive key bit 2
3	EXK3	Executive key bit 3
4	CA	Cache active
5	ENXIX	Enable extended index mode
6	PARO	Parity code bit 0
7	PAR1	Parity code bit 1
8	FPPUFL	Enable FPP interrupt on underflow
9	FPPINT	Enable FPP interrupt on overflow
10	DM	Disable memory
11	STEP	Step (trace) interrupt
12	RTC	Enable real time clock
13	HALT	System halted
14	M64K	Enable 64 K mode
15	INDFL	Overflow status bit

The combination of bit 6 (PAR0) and 7 (PAR1) above are further defined as:

PAR1	PAR0	MEANING
0	0	Parity disabled
0	1	Enable double error interrupt only
1	0	System processing parity error interrupt
1	1	Enable all parity error interrupts

b. Cache status word using read command 102145.

BIT	MNEMONIC	MEANING
0	UMK0	User mode key bit 0
1	UMK1	User mode key bit 1
2	UMK2	User mode key bit 2
3	UMK3	User mode key bit 3
4	EMM	Executive mask mode not
5	EMS0	Executive mode state 0
6	EMS1	Executive mode state 1
7	EMS2	Executive mode state 2
8	EMS3	Executive mode state 3
9	MPEN	Memory protect enabled
10	TSO	Transition code 0
11	TS1	Transition code 1
12	EMMLI	Executive mask mode last interrupt
13	USMLI	User mode last interrupt
14	MP0	Memory protect code 0
15	MP1	Memory protect code 1

Some of the status bits are combined to give the meanings listed below:

EMS	FETCH	STORE
0	Map 0	Map 0
1	Map 0	Map N
2	Map N	Map 0
3	Map N	Map N

TS1	TS0	MEANING
0	0	No transition armed
0	1	TAE armed (transition to executive)
1	0	TIN armed (transition to inactive)
1	1	TEU armed (transition to user)

MP1	MP0	MEANING
0	0	Privileged instruction
0	1	Instruction fetch
1	0	Write protected
1	1	Unassigned

c. Cache status word using write command 103145.

BIT	MNEMONIC	MEANING
0	UMK0	User mode key bit 0
1	UMK1	User mode key bit 1
2	UMK2	User mode key bit 2
3	UMK3	User mode key bit 3
4	EMM	Executive mask mode not
5	EMS0	Executive mode state 0
6	EMS1	Executive mode state 1
7	EMS2	Executive mode state 2
8	EMS3	Executive mode state 3
9	MPEN	Enable memory protect
10	(spare)	
11	TIN	Transition to inactive armed
12	TAE	Transition to executive armed
13	TEU	Transition to user armed
14	CLRMP	Clear memory protect status
15	(spare)	

17.3.2 Map Test Selection

The program moves into the map test component when a T. is entered in response to the MM** prompt. The computer responds with the message:

MAP TEST SEQUENCE R.

Enter one or more (up to eight total) of the following subtest numbers as a response:

SUBTEST NUMBER	SUBTEST NAME
1	Static register
2	Map RAM
3	Executive mode state and user key selection
4	Executive key
5	Extended indexing
6	64K mode
7	Memory protection
8	Memory

If the operator responds with a period or comma (no numeric entry) all subtests are run.

During an initial test, all subtests should be run prior to entering the memory test component. Operator responses are separated by a comma and a period terminates the operation. When a period or an 8. is entered, the program moves into the memory test component.

17.3.3 Memory Test Selection

In the memory test component, the computer responds with the message:

16K MEMORY MODULE(S) TO BE TESTED = R

Enter one of the following responses:

RESPONSE	DEFINITION
,	Test all modules (except 0) entered in the configuration statement (test monitor component).
.	Test modules entered in previous memory module statement (use on retest only).
n	Test specific 16K memory module. Module numbers are:

MODULE NUMBER DECIMAL	MODULE NUMBER OCTAL	PHYSICAL ADDRESS BLOCK
0	0	0-0037777
1	01	0040000-0077777
2	02	0100000-0137777
3	03	0140000-0177777
4	04	0200000-0237777
5	05	0240000-0277777
6	06	0300000-0337777
7	07	0340000-0377777
8	010	0400000-0437777
9	011	0440000-0477777
10	012	0500000-0537777
11	013	0540000-0577777
12	014	0600000-0637777
13	015	0640000-0677777
14	016	0700000-0737777
15	017	0740000-0777777
16	020	1000000-1037777
17	021	1040000-1077777
18	022	1100000-1137777
19	023	1140000-1177777
20	024	1200000-1237777
21	025	1240000-1277777
22	026	1300000-1337777
23	027	1340000-1377777
24	030	1400000-1427777
25	031	1440000-1477777
26	032	1500000-1537777
27	033	1540000-1577777
28	034	1600000-1637777
29	035	1640000-1677777
30	036	1700000-1737777
31	037	1740000-1777777
32	040	2000000-2037777
33	041	2040000-2077777
34	042	2100000-2137777
35	043	2140000-2177777
36	044	2200000-2237777
37	045	2240000-2277777
38	046	2300000-2337777
39	047	2340000-2377777
40	050	2400000-2437777
41	051	2440000-2477777
42	052	2500000-2537777
43	053	2540000-2577777
44	054	2600000-2637777
45	055	2640000-2677777
46	056	2700000-2737777
47	057	2740000-2777777
48	060	3000000-3037777
49	061	3040000-3077777
50	062	3100000-3137777
51	063	3140000-3177777
52	064	3200000-3237777
53	065	3240000-3277777
54	066	3300000-3337777
55	067	3340000-3377777
56	070	3400000-3437777
57	071	3440000-3477777
58	072	3500000-3537777
59	073	3540000-3577777
60	074	3600000-3637777
61	075	3640000-3677777
62	076	3700000-3737777
63	077	3740000-3777777

A comma will cause the program to test all 16K memory modules that are contained in the configuration statement of the test monitor component. Module 0 which contains the test executive and the map/memory test programs will not be tested.

If a period is entered on an initial test, the program will default to a comma entry. After initial selection and a return to this point in the program, a period input will cause the program to use the previously designated module numbers. If more than one test is anticipated, use a cycle count entry (Section 17.3.4) rather than this entry.

For testing specific 16K memory modules, enter the decimal or octal number of the module as designated in the table of module numbers above. The numbers are separated by commas and terminated by a period. Entries can be in any order, the program sequentially tests from low to high numbered module. A two-number entry with the two

numbers separated by a dash (-) can be used to indicate a continuous memory area. The first number indicates the low limit and the second number indicates the high limit of memory. For example, to test physical address area 0740000-1377777 enter:

16K MEMORY MODULE(S) TO BE TESTED = 15-23.

The program responds with the message:

MEMORY TESTS TO EXECUTE = R.

Enter one of the following responses:

RESPONSE	DEFINITION
. or ,	Run subtests 1, 2, 3 and 6 (subtests 4 and 5 are extremely long and are not normally run unless required).
n	Run subtest n. Input as many n's as required, separated by commas and terminated by a period. The available subtests are:

SUBTEST NUMBER	SUBTEST NAME
1	Unique address
2	Ones and zeros pattern
3	Alternate patterns
4	Walk one
5	Walk zero
6	Unique module

The program will output parity error messages.

17.3.4 Cycle Count Selection

The user specifies the number of cycles the program is to perform in response to the following message:

CYCLES = R.

One of the following responses can be made by the operator:

RESPONSE	DEFINITION
. or ,	Continuously execute test program
n,	Execute program for n (decimal entry) cycles. Print END CYCLE n, where n signifies the cycle number of the test that has been completed.
n.	Execute program for n (decimal entry) cycles. Suppress END CYCLE n message.

Upon completion of the number of cycles specified or abort termination via SENSE switch 3, the following end-of-test messages are always printed:

TEST COMPLETE
TOTAL CYCLES = n
MM**

where n is the number of test cycles completed and MM** indicates the computer is awaiting a new directive.

17.4 ERROR MESSAGES

Following the detection of an error, the program sends out an error message. The various error messages are outlined in this section.

17.4.1 Global

Global messages can occur at any time and are not test related. They are:

DATA BIT

DOUBLE BIT ERROR

INVALID FLOATING POINT INTERRUPT

INVALID MEMORY PROTECT INTERRUPT

INSTRUCTION ADDRESS =
OPERAND ADDRESS =
PROCESSOR STATUS =
CACHE STATUS =
REGISTERS A,B,X =

INVALID REAL TIME CLOCK INTERRUPT

INVALID TRACE INTERRUPT

MAP BUSY TIMEOUT

NO SYNDROME MATCH FOUND

PARITY BIT

PARITY ERROR

BOARD =
CHIP ROW =

PARITY ERROR-NO VALID BIT SET

POWER RESTORED

17.4.2 Static Register

An error detected in the static register subtest (map subtest 1) is reported by one of the following messages:

TEST 1 PROCESSOR STATUS ERROR
TEST 1 CACHE STATUS ERROR
TEST 1 MAP STATUS ERROR
TEST 1 INTERRUPT EXIT ERROR

Along with the error message, the error condition is reported under:

EXPECTED ACTUAL

17.4.3 Map RAM

An error detected in the map RAM subtest (map subtest 2) is reported by the following message:

TEST 2 MAP RAM ERROR
SUBTEST WORD EXPECTED ACTUAL

Refer to Section 17.2.3.2 for a translation of the subtest numbers. The error condition is printed with the subtest number.

17.4.4 Executive Mode State and User Key Selection

An error detected in the executive mode state and user key selection subtest (map subtest 3) is reported by the following message:

TEST 3 mapping error code

The possible error codes and meanings are:

ERROR CODE	MEANING
0	Executive mode not working
1	Executive mode state 0 fetch failed
2	Executive mode state 1 fetch failed
3	Executive mode state 2 fetch failed
4	Executive mode state 3 fetch failed
5	Executive mode state 0 store failed
6	Executive mode state 1 store failed
7	Executive mode state 1 store failed
10	Executive mode state 2 store failed
11	Executive mode state 2 store failed
12	Executive mode state 3 store failed
13	Executive mask state fetch failed
14	Executive mask state store failed
15	User mode key 1 fetch failed
16	User mode key 2 fetch failed
17	User mode key 4 fetch failed
20	User mode key 8 fetch failed
21	Virtual memory map loading failed
22	Executive mode state 1 store byte failed
23	Executive mode state 1 store byte failed
24	Executive mode state 2 store byte failed
25	Executive mode state 2 store byte failed
26	Executive mode state 1 store byte block failed
27	Executive mode state 1 store byte block failed

17.4.5 Executive Key

An error detected in the executive key subtest (map subtest 4) is reported by the following message:

TEST 4 EXEC KEY ERROR CODE

The possible error codes and meanings are:

ERROR CODES	MEANING
1	Key 1 fetch failed
2	Key 1 store failed
3	Key 2 fetch failed
4	Key 2 store failed
5	Key 4 fetch failed
6	Key 4 store failed
7	Key 8 fetch failed
10	Key 8 store failed

17.4.6 Extended Indexing

An error detected in the extended indexing subtest (map subtest 5) is reported by the following message:

TEST 5 EXTENDED INDEX ERROR

The register and key in error are reported under:

REGISTER KEY

17.4.7 64K Mode

An error detected in the 64K mode subtest (map subtest 6) is reported by the following message:

TEST 6 64 K MODE ERROR CODE

The possible error codes and meanings are:

ERROR CODE	MEANING
1	Executive mode failed
2	64 K mode failed-map inactive
3	64 K mode failed-executive mode
4	64 K mode failed-user mode

17.4.8 Memory Protection

An error detected in the memory protection subtest (map subtest 7) is reported by the following message:

TEST 7 MEMORY PROTECT ERROR
KEY CODE

The possible error codes and meanings are:

ERROR CODE	MEANING
1	Halt was executed
2	Status error on halt interrupt
3	No interrupt on I/O instruction
4	Status error on I/O instruction interrupt
5	Memory Protection interrupt on I/O instruction executive mode
6	No interrupt on write into read operand only
7	Status error on write protect interrupt
10	Write instruction altered memory
11	Tracking register error
12	Location 062 error (instruction address)
13	Invalid interrupt memory protection off
14	Write instruction altered memory
15	No interrupt on write into read only
16	Status error on write protect interrupt
17	Write instruction altered memory
20	Tracking register error
21	Location 062 error (instruction address)
22	Halt (jump to read operand only executed)
23	Status error on instruction fetch
24	No interrupt on read unassigned location
25	Status error on unassigned
26	Tracking register error
27	Location 062 error (instruction address)
30	No interrupt on write into unassigned
31	Status error on unassigned
32	Write instruction altered memory
33	Halt (jump to unassigned executed)
34	Status error on unassigned
35	Location 062 error (instruction address)
36	Status error on instruction fetch
37	Executive mode state 2 error
40	Executive mask mode not set after memory protection interrupt
41	Tracking register error
42	No unassigned interrupt
43	Tracking register error
44	No unassigned interrupt
45	Executive mask mode not set after JSR
46	Invalid memory protect interrupt after JSR
47	Transition code error executive to user
50	Transition code error executive to inactive
51	Location 024 error (processor status word)
52	Trace instruction did not interrupt
53	Trace executed but did not interrupt
54	Trace interrupt but did not execute
55	Old cache status error
56	New processor status word error
57	Not in user mode on last interrupt

Each memory protection test is run in key 0 then in key 5. Some of the tests are repeated but the error codes do not necessarily have the same meaning. As an aid in troubleshooting errors, use Table 17-1.

Table 17-1. Operational Error Listing

TEST OPERATION	TEST KEY	ERROR CODE	PROBABLE CAUSE
1. Issue halt	0,5	1	Machine halted
	0,5	2	Wrong interrupt status
2. Issue I/O Instruction	5	3	Did not get interrupt
	5	4	Wrong status on interrupt
	0	5	Should not get interrupt
3. Write into read operand only	0,5	6	Did not get interrupt
	0,5	7	Wrong status on interrupt
	0,5	10	Write altered memory
	0,5	11	Tracking register in error
	0,5	12	Location 062 in error
	0	13	Should not get interrupt
	0	14	Write altered memory
	0,5	51	Location 024 PSW in error
	0,5	55	Old cache status error
	0,5	56	New processor status word error

Table 17-1. Operational Error Listing (Continued)

TEST OPERATION	TEST KEY	ERROR CODE	PROBABLE CAUSE
4. Write into read only	0,5	15	Did not get interrupt
	0,5	16	Wrong status on interrupt
	0,5	17	Write altered memory
	0,5	20	Tracking register in error
	0,5	21	Location 062 in error
	0,5	55	Old cache status error
	0,5	56	New processor status word error
5. Jump to read operand only	0,5	22	Machine halts
	0,5	23	Wrong status on interrupt
6. Read from unassigned location	0,5	24	Did not get interrupt
	0,5	25	Wrong interrupt status
	0,5	26	Tracking register in error
	0,5	27	Location 062 in error
7. Store into unassigned location	0,5	30	Did not get interrupt
	0,5	31	Wrong status on interrupt
	0,5	32	Write altered memory
8. Jump to unassigned location	0,5	33	Wrong interrupt on jump
	0,5	34	Wrong status on interrupt
	0,5	35	Location 062 in error
9. Instruction fetch from read operand only	0,5	36	Wrong status on interrupt
10. Check executive mode state 2	5	37	Executive mode state 2 not working
11. Issue halt	5	40	Not in executive mask mode after interrupt
	5	57	Not in user mode on last interrupt
12. Read unassigned location	0	41,43	Tracking register in error
	0	42,44	Did not get interrupt
13. Issue executive calls	5	45	Not in executive mask mode after JSR
	5	46	Executive call caused interrupt
14. Set executive to user mode then halt	0	47	Wrong transition code
15. Set executive to inactive then halt	0	50	Wrong transition code
16. Issue trace	5	52	Trace instruction did not cause interrupt
	5	53	Trace caused jump but no interrupt
	5	54	Trace instruction did not execute

17.4.9 Memory

An error detected in the memory test is reported by the following message:

TEST 8 MEMORY ERROR
TEST MODULE WORD EXPECTED ACTUAL CYCLE

In the second line of the error message is the memory test number and the error condition that was detected. Refer to Section 17.2.4 for a translation of the memory test numbers.

SECTION 18

V77-800 WRITABLE CONTROL STORE TEST PROGRAM

The V77-800 Writable Control Store Test Propgram of the MAINTAIN III tests the Writable Control Store (WCS) of the SPERRY UNIVAC V77-800 Computer, F3078-0x. A description of the writable control store is contained in the V77-800 Writable Control Store Functional Analysis and Servicing Manual.

The WCS test is normally loaded and executed via virtual console (CRT or TTY) keyboard commands. The test program operates with the MAINTAIN III Test Executive Program. Test executive supplies all common I/O routines and other standard MAINTAIN III routines. Since the test executive is the software interface, it must be loaded and operational before the WCS test program can be loaded.

The writable control store test verifies the operational status of the WCS and assists in locating faults. Three subtests verify that each WCS memory location can be uniquely accessed and that all bit cells are operative. Eight utility routines enhance the ability to isolate specific WCS hardware faults.

The test verifies WCS memory by first accessing the memory through the I/O bus and then by executing a microprogram in the memory. The user may supplement the program tests by loading a microprogram into the WCS and then executing it.

The WCS test program consists of the following components:

a. Unique address test

b. Grouped bit memory test

c. Execution test

d. Burn-in test

e. Utility routines

Each test component has various error messages which are displayed or printed at the operator's station. The nature of error reporting control is determined by sense switch selection.

18.1 HARDWARE REQUIREMENTS

The WCS test minimum hardware configuration is as follows:

a. V77-800 computer with writable control store

b. Virtual console (CRT or TTY terminal)

c. Program loading device such as a paper tape reader, card reader, or magnetic tape device.

18.2 DESCRIPTION OF TEST COMPONENTS

This section contains a brief description of each WCS test component.

18.2.1 Unique Address Test

The unique address test verifies that each WCS memory location can be accessed and that all memory cells are operative. The test accesses WCS memory through the I/O bus. The test program writes a unique data pattern into each WCS location and then reads each location to verify the address for the data. The contents of each memory location are complemented and a second read is performed to validate the complemented data pattern.

18.2.2 Grouped Bit Memory Test

The grouped bit memory test is designed to detect constant state (shorted) bits within each WCS memory location. The test accesses memory through the I/O bus. Certain bit patterns are written into each location and then read to verify the contents of that location. The logical complement of each pattern is also tested. The bit pattern is:

052525	**0177777**	**0125252**	(Initial user-
007777	**0007777**	**0007777**	defined pattern)
052525	**0052525**	**0052525**	
031463	**0031463**	**0031463**	
016161	**0143434**	**0070707**	
007417	**0007417**	**0007417**	
001760	**0037403**	**0170077**	
000377	**0000377**	**0000377**	
000000	**0177777**	**0000000**	
000000	**0000377**	**0177777**	
000000	**0000000**	**0000000**	

The initial user-defined pattern is a program input but can be changed to suit the needs of the user. The pattern can be changed through use of a utility routine (P command).

18.2.3 Execution Test

The execution test verifies that WCS memory can be accessed at the speed in which the WCS can execute its functions. A microprogram consisting of an increment A register and an unconditional jump to the next memory location is written into the WCS memory and executed. The result is compared to an expected value. An error message is output if the resultant value does not compare.

18.2.4 Burn-in Test

The burn-in test executes the preceding tests in the following order:

a. Unique address test

b. Grouped bit memory test

c. Execution test.

18.2.5 Utility Routines

Utility routines can be called from the WCS test program executive. The routines permit the user to:

a. change the user-defined bit pattern,

b. return to the MAINTAIN III test executive,

c. examine or change the contents of WCS memory,

d. execute a microprogram,

e. save and restore the contents of WCS memory,

f. and to initialize WCS with the user defined pattern.

A list of available options is displayed when the user enters a "?".

18.3 OPERATING PROCEDURES

The MAINTAIN III test executive must be loaded before the writable control store test program can be loaded. Operating procedures for the V77-800 control panel and virtual console are contained in the V77-800 operations manual.

Load the test executive in accordance with the procedures outlined in Section 2.

For paper tape systems, the procedure for loading the test program is as follows:

a. Place the writable control store program tape in the paper tape reader.

b. Position the tape within the leader area between the test part number and the start of the program.

c. Enter an L on the virtual console.

d. Enter a period on the virtual console.

For magnetic tape systems, the procedure for loading is as follows:

a. Consult the MAINTAIN III Usage Description Bulletin for the file number of the V77-800 writable control store test.

b. Position the tape to that file number by using MAINTAIN III tape commands. The commands are listed in Section 2.

c. Load the test program by using the tape commands listed in Section 2.2.2.1.

The tape loads and a program identification message is displayed.

When loading of the test program is complete, automatic transfer to the start of the program takes place (address 0500). Sense switch settings can alter the test program as follows:

SENSE SWITCH	SET POSITION	RESET POSITION
1	Suppress error message	Print error message
2*	Before error halt: Halt on error	Before error halt: Do not halt on error Print error and continue
	After error halt: Continue testing	After error halt: Loop on error
3	Terminate test and Return to test selection	Continue testing

*SENSE switch 2 can be used to continue the test following an error halt or to loop on the error:

a. To continue to the next error halt, leave the SENSE switch SET and depress R on the virtual console.

b. To loop on the error, RESET the SENSE switch and depress R on the virtual console. Looping will continue until the switch is SET, the program then continues in the "halt on error" mode until the next error halt. If the error condition clears, looping continues until SENSE switch is set.

While operating the WCS test program, all input and output messages are via the virtual console. In the following procedures, responses to be initiated by the operator are underlined R where R indicates a response is to be made. Responses from the computer are not underlined.

The test program can be started by loading octal address 0500 into the P register and typing R on the virtual console. To restart at the beginning of the test initialization, load octal address 0502.

A number can be entered in octal, decimal or hexadecimal form. An entry preceded by a D is treated as a decimal entry. An initial H indicates a hexadecimal entry. All other numeric inputs are treated as octal numbers.

18.3.1 Test Initialization

When the WCS test program is loaded, control is transferred to the initialization routine. The following message is output:

V77-800 WCS TEST
PAGES? R.

Respond with the number of WCS pages available for testing. The response is a single line entry in which the user can specify a single page or a sequence of pages.

WCS pages 0-7 are system firmware and are not to be tested. An entry of 0-7 causes an INVALID message to be displayed.

A single numeric entry (8-11) followed by a period indicates that a single page is to be tested. Whether specified or not, page 8 will always be tested.

To test two or more pages, either of two methods can be used:

a. To test non-sequential pages, terminate the page number entry with a comma. The program requests a second page entry. A period on the second or subsequent entry terminates the test initialization. For example:

PAGES? 8,
PAGES? 10,
PAGES? 11.

b. To test a sequence of pages, enter a start page number followed by a dash (-) and a stop page number. For Example:

PAGE? 8-11.

When a period is entered to terminate the response, the test initialization routine is terminated. The following message is displayed:

PAGES X Y Z ...

where:

X Y Z ...
Is a list of available page numbers in decimal notation.

18.3.2 Test Selection

Test selection is used by the operator to identify the test or utility routine that is to be executed. The test program indicates that it is ready for a test selection command by printing the following message:

TEST R

The user response is any one of the following command mnemonics:

COMMAND MNEMONIC	DEFINITION	TYPE COMMAND
?	Display list of commands	Utility
B	Burn-in test	Test
C	Change WCS memory location	Utility
D	Dump WCS memory	Utility
E	Execution test	Test
H	Display list of commands	Utility
I	Initialize WCS memory with user pattern	Utility
M	Grouped bit memory test	Test
P	Display/change user defined data pattern	Utility
R	Run microprogram in WCS	Utility
S	Save WCS memory	Utility
U	Unique address test	Test
W	Write saved WCS memory	Utility
X	Return to MAINTAIN III executive	Utility

The type command in the preceding list indicates whether the command causes a test or utility routine to be executed.

If all the initialized WCS pages are to be tested, terminate the test command with a period. The test program jumps to the cycle sequence (Section 18.3.4).

Terminating the test command with a comma indicates that only certain addresses are to be tested (Section 18.3.3). At any time that a test command is terminated with a comma, the test program will respond with an ADDRESS? prompt.

The setting of sense switch 3 is tested at the completion of each test or utility routine. Control is returned to the test selection routine when sense switch 3 is set. The program prompts the user with a TEST message and waits for a TEST command.

18.3.3 Address Parameter

The address parameter allows the user to specify an address or a sequence of addresses on which the test or utility routine is to be performed. When the test command is terminated with a comma the following message is dispalyed:

ADDRESS? R.

A single address is specified by entering the desired address followed by a period. A sequence of addresses are specified by entering a start memory location followed by a dash and a stop memory location. The entry is terminated with a period.

18.3.4 Cycle Parameter

The cycle parameter allows the user to specify the number of test cycles to be performed. The following message is displayed:

CYCLES R

The user responds with the number of test cycles to be performed. A 0 indicates that testing is to be continuous.

When the response is terminated with a comma, the number of each completed test cycle is displayed at the end of the test cycle. The following message is displayed:

END CYCLE XXXXXX

where:

X
Indicates the number of the completed test cycle.

The END CYCLE message is suppressed if the CYCLE response is terminated with a period.

18.3.5 Utility Routines

The user can request any of the utility routines (Section 18.3.3) while the program is in the test selection sequence. A complete list of the utility routines is displayed when the user enters a question mark or an H.

At any time that a utility routine command is terminated with a comma the test program will respond with an ADDRESS? prompt. To execute a routine on a sequence of memory addresses after the ADDRESS? prompt:

a. Enter the memory start address,

b. enter a dash,

c. enter a memory stop location,

d. enter a period.

18.3.5.1 Examining/Changing WCS Contents

Contents of WCS memory can be examined and changed by entering the utility routine command:

C,

The program responds with an ADDRESS? prompt.

The program displays the address and data contents of the address specified by the user. Any or all three 16-bit words can be changed or left as is.

A change is made by entering the new data on the same line. If a word is to remain unchanged, enter a comma. Terminating the entry with a period causes the program to accept the user response and return to the test selection sequence. To display the contents of the next memory location, terminate the entry with a comma instead of a period.

Example:

```
C,
ADDRESS?  10016
(010016)  120040 000010 000400 ,,,
(010017)  120000 000010 000400 103054.
C.
(010017)  103054 000010 000400.
```

Location 010016 is examined. No change is made and the user wants to examine the next location. In location 010017, only the first word is changed. To examine location 010017 to see if the change was made, a C. is entered. Examination complete, a period is entered to return to test selection sequence.

18.3.5.2 Dump WCS Memory

The contents of a sequence of WCS memory locations can be displayed by entering:

D,

The program responds with an ADDRESS? prompt. Enter the sequence of memory addresses to be dumped.

The dump is aborted by striking any console key.

18.3.5.3 Initialize WCS Memory

The user defined data pattern (Section 18.2.2) can be written into a sequence of WCS address locations by entering:

I,

The program responds with an ADDRESS? prompt. Enter the sequence of memory addresses to be initialized.

18.3.5.4 Change User-Defined Pattern

The user-defined pattern is changed by entering:

P.

The program responds with the current pattern:

P. 052525 177777 125252

The user then enters a new data pattern on the same line. Each new 16-bit word entry is separated with a comma. If a comma is entered without a new word pattern, the old 16-bit word is retained. The response is terminated with a period.

To retain the same data pattern, omit a numeric value and enter a period.

For example, to change the first two groups, retain the third, and then check the new data pattern:

```
P. 052525 177777 125252 1, 2.
P. 000001 000002 125252.
```

18.3.5.5 Run Microprogram in WCS

User programs can be entered into WCS memory. To execute the user program, enter:

R,

The program will respond with an ADDRESS? prompt. Enter the sequence of memory addresses to be run.

The program then requests an initial value for each of the eight registers R0 through R7 by displaying the message:

```
RX 000000 R
```

where:

X
Is the number of the register.

Enter a value to be used. If the displayed value is not to be changed, omit the numeric value and enter a comma. The program requests a value for the next register. The sequence will continue through register R7. The last entry is terminated with a period. At any time that an entry is terminated with a period, the sequence is terminated and a branch to the specified start address is executed. When the microprogram is completed, the contents of the eight registers are dumped to the console.

For example:

```
R0 000000 ,
R1 000000 1,
R2 000000 2,
R3 000000 3,
R4 000000 4,
R5 000000 5,
R6 000000 6,
R7 000000 7.

003777 000001 000002 000003
000004 000005 000006 000007
```

18.3.5.6 Save WCS Memory

All or a portion of WCS memory can be saved in main memory. To save the entire contents of WCS memory, enter:

S.

To save a portion of WCS memory, enter:

S,

The program responds with an ADDRESS? prompt. Enter the sequence of WCS memory addresses to be saved.

18.3.5.7 Write Saved WCS Data

All or a portion of WCS data saved in main memory can be written into WCS memory. To write all of the saved data into WCS memory, enter:

W.

To write a portion of the saved data, enter:

W,

The program responds with an ADDRESS? prompt. Enter the sequence of WCS memory addresses to be used.

18.3.5.8 Return to MAINTAIN III Executive

Program control can be returned to the MAINTAIN III executive by entering:

X.

18.4 ERROR MESSAGES

Following the detection of an error, the program sends out an error message. The numerics displayed are in octal notation. The various error messages are outlined in this section.

18.4.1 Unique Address Test

A failure detected during the write sequence results in the following two error messages:

```
SEQUENTIAL ADDRESS ERROR AT a
EXPECTED= b c d
ACTUAL= e f g
```

where:

a
Is the WCS address in error (16-bit word).

b c d
Is the expected WCS data (three 16-bit words).

e f g
Is the actual data contents (three 16-bit words).

```
UNIQUE ADDRESS ERROR AT a
EXPECTED= b c d
ACTUAL= e f g
```

where:

a through **g**
Have the same meaning as in the preceding message.

A failure in the WCS address sequencer during input to or output from WCS memory results in the following error message:

SEQUENTIAL ADDRESS ERROR AT
EXPECTED- b (one 16-bit word)
ACTUAL- e (one 16-bit word)

where:

b and **e**
Have the same meaning as in the preceding message.

18.4.2 Grouped Bit Memory Test

The test fails if the 48 bits of data read from WCS memory is not equal to the 48 data bits written to a specific address.

An error occurring during the grouped bit memory test results in the following message:

READ-AFTER-WRITE ERROR AT a
EXPECTED= b c d
ACTUAL= e f g

where:

a through **g**
Have the same meaning as in the preceding test.

18.4.3 Execution Test

Errors detected during the execution test result in the following error message:

EXECUTION TEST ERROR
EXPECTED= a
ACTUAL= b

where:

a and **b**
Have the same meaning as in the preceding tests.

SECTION 19
V77-800 FLOATING-POINT PROCESSOR TEST PROGRAM

The V77-800 Floating-Point Processor Test Program of the MAINTAIN III tests the Floating-Point Processor (FPP), F3053-00, of the SPERRY UNIVAC V77-800 Computer, F3078-OX. A description of the floating-point processor is contained in the V77-800 Floating-Point Processor Functional Analysis and Servicing Manual.

The V77-800 Floating-Point Processor Test Program is referred to as the test program or program in this section. The test program is designed to test all of the FPP instructions which are mainly executed by the floating-point processor. The program includes not only testing of the basic floating-point instruction set but also the processing of the status word when an error is detected.

There are many FPP instructions that are executed in firmware and which are not verified by this program. The complete instruction set is contained in the Floating Point Processor Functional Analysis and Servicing Manual. The test program is designed to test all of the floating-point instructions that are executed by the FPP. Instructions that are tested by this test program are listed in table 19-1.

The FORTRAN firmware is an integral part of the floating-point instruction set and is required for proper execution of the test program. The firmware is resident in the diagnostic and is loaded into writable control store (WCS) at the option of the operator. The firmware code must be resident in WCS before execution of the test program even though only a small portion of the firmware is used during the test cycle. The firmware used is mainly in those areas which decode the instructions and those areas which monitor the FPP and processor status words.

The data used during the test has been precalculated and is resident in the test program. Essentially, all tests are arithmetic calculations with the computed results compared to known results. The data patterns used are such that all data paths of the FPP are exercised.

The FPP test program is normally loaded and executed via virtual console (CRT or TTY) keyboard commands.

The test program operates with the MAINTAIN III test executive program. Test executive supplies all common I/O routines and other standard MAINTAIN III routines. Since the test executive is the software interface, it must be loaded and operational before the test program can be loaded.

Error messages are displayed or printed at the operator's station. The nature of error reporting control is determined by sense switch settings.

19.1 HARDWARE REQUIREMENTS

The FPP test program minimum hardware configuration is as follows:

a. V77-800 Computer with 64K memory, WCS and FPP.

b. FORTRAN firmware.

c. Virtual console (CRT or TTY terminal).

d. Program loading device such as a paper tape reader, card reader, or magnetic tape device.

19.2 DESCRIPTION OF TEST COMPONENTS

The FPP test contains a single option: whether to execute the test once or in a continuous mode.

The test is divided into two test components.

The first component is the basic instruction set test. This component tests all of the instructions listed in table 19-1. The test uses data patterns which take into account all the data path decisions as dictated by the FPP with the exclusion of fault paths.

Table 19-1. FPP Instructions

MNEMONIC	INSTRUCTION
FNEG	Floating Negate
FSQRTS	Floating Square Root, Single Precision
FSQRTD	Floating Square Root, Double Precision
FSTSD	Floating Store Direct, Single Precision
FSTDD	Floating Store Direct, Double Precision
FLD	Floating Load, Single Precision
FLDD	Floating Load, Double Precision
FAD	Floating Addition, Single Precision
FADD	Floating Addition, Double Precision
FSB	Floating Subtraction, Single Precision
FSBD	Floating Subtraction, Double Precision

FMU	Floating Multiplication, Single Precision
FMUD	Floating Multiplication, Double Precision
FDV	Floating Division, Single Precision
FDVD	Floating Division, Double Precision
FPPS	Read and Clear FPP Status, Single Precision

The second component is the fault isolation test. This component executes the same instructions as the basic instruction test except that the data patterns ensure a fault condition. This test is executed with the fault interrupts off and the FPP status word monitored for errors.

19.3 OPERATING PROCEDURE

The MAINTAIN III test executive must be loaded before the FPP test program can be loaded. Operating procedures for the V77-800 control panel and virtual console are contained in the V77-800 Computer Operations Manual.

Load the test executive in accordance with the procedures outlined in section 2.

For paper tape systems, the procedure for loading the test program is as follows:

a. Place the FPP test program tape in the paper tape reader.

b. Position the tape with the leader area between the test part number and the start of the program.

c. Enter an L on the virtual console.

d. Enter a period on the virtual console.

For magnetic tape systems, the procedure for loading is as follows:

a. Consult the MAINTAIN III Usage Description Bulletin for the test file number of the V77-800 Floating-Point Processor Test.

b. Position the tape to that file number by using MAINTAIN III tape commands listed in section 2.

c. Load the test program by using the tape commands listed in Section 2.2.2.1.

The tape loads and a program identification message is displayed.

When loading of the test program is complete, automatic transfer to the start of the program takes place (address 0500). Sense switch settings can alter the test program as follows:

SENSE SWITCH	SET POSITION	RESET POSITION
1	Suppress error (or result) message	Print error (or result) message
2	Before error detection, halt on error	Before error detection, do not halt on error, print error message and continue
	After error detection, continue testing	After error detection, loop on error
3	Terminate test and return to start of test	Continue testing

Sense switch 1 is used to control the display of error messages.

Sense switch 2 is used to continue the test following an error halt or loop on the detected error.

a. To continue to the next error halt, leave the SENSE switch set and depress R on the virtual console.

b. To loop on the error, reset the SENSE switch and depress R on the virtual console. Looping continues until the switch is set. The program then continues in the "halt on error" mode until the next error is detected.

Sense switch 3 is used to terminate a test being run in the continuous mode.

While operating the test program, all input and output message are via the virtual console. In the following procedures, responses to be initiated by the operator are underlined R where R indicates a response is to be made. Responses from the computer are not underlined.

19.3.1 Program Start

When loading of the test program is complete, a program start is initiated. The following identification message is displayed.

V77-800 MAINTAIN III FLOATING POINT PROCESSOR DIAGNOSTIC

19.3.2 WCS Loading

The program identification message is followed by a query as to whether the firmware should be written into WCS or not. The query message is:

WRITE FIRMWARE TO WCS? (Y/N) R

An entry of N causes the program to jump to the cycle selection routine.

An entry of Y causes the firmware to be loaded into WCS, read back, and the results compared to the original data. A successful read and compare causes a jump to the cycle count selection routine and the following message to be displayed:

SUCCESSFUL WCS READ/WRITE

If an error is encountered while comparing the results of the read with the actual firmware, the following message is displayed:

UNSUCCESSFUL WCS READ/WRITE

RUN V77-800 WCS DIAGNOSTIC

Control is returned to the MAINTAIN III executive so as to allow testing of the WCS.

19.3.3 CYCLE SELECTION

Following entry of the WCS loading option, the program outputs the message:

ENTER T. OR T,C R

Entry of T. causes the program to execute the test once. An entry of T,C causes the test to cycle continuously until sense switch 3 is toggled.

19.3.4 Test Termination

Termination of the test is indicated by the message:

TOTAL NUMBER OF CYCLES n
TOTAL NUMBER OF ERRORS n

where: n indicates the total number.

The total number of errors is indicated even though SENSE switch 1 was set to suppress the error messages.

After the preceding message is displayed, the program advances to the program start routine. The program identification and WCS loading messages are displayed. The program then waits for further operator action.

19.4 ERROR MESSAGES

In order to correctly diagnose any error that can occur during the test, the operator must have a copy of the program listing.

Each instruction test consists of a test code and an associated data table. The data table format is:

```
DATA    (expected results)
DATA    (operand one)
DATA    (operand two)
```

An example of a data table entry from single precision multiplication is:

```
DATA    040662,  052561
DATA    040325,  0125252
DATA    040325,  0125252
```

When an error is detected, the error message is in the following format:

MESSAGE	MEANING
TEST NAME	Name of the instruction test where the error occurred (table 19-1).
ERROR MESSAGE	Error condition that occurred (table 19-2).
EXPECTED RESLTS	Precalculated results of the arithmetic calculation.
ACTUAL RESULTS	Results of arithmetic calculation.
PROGRAM RESULTS	Location of the test code.
DATA ADDRESS	Address of the data that caused the error.
PROCESSOR STATUS WORD	See Table 19-3
FLOATING-POINT STATUS WORD	See Table 19-4

During the fault isolation cycle, the instruction test name is replaced by the test isolation subroutine name.

The processor and floating-point status word bits are defined in Tables 19-3 and 19-4.

A possible error message is given in the following example:

FLOATING SUBTRACTION

COMPARISON FAILURE

EXPECTED RESULTS 052524 052524

ACTUAL RESULTS 000000 000000

PROGRAM LOCATION 002210

DATA ADDRESS 002270

PSW 000000

FSW 000000

Table 19.2. Error Messages

MESSAGE	DEFINITION
COMPARISON FAILURE	A comparison failure occurred between actual and expected results.
EXPONENT OVERFLOW	The test generated a condition to cause an exponent overflow fault. The fault failed to occur or an unexpected exponent overflow occurred.
EXPONENT UNDERFLOW	The test generated a condition to cause an exponent underflow fault. The fault failed to occur or an unexpected underflow occurred.
DIVIDE BY ZERO	The diagnostic generated a condition to cause a divide by zero fault. The fault failed to occur or an unexpected divide by zero fault occurred.
DIVIDE 0/0	Test generated an arithmetic function of 0 divided by 0. The required fault condition did not occur.

Table 19-3. Processor Status Word

BIT	MNEMONIC	DESCRIPTION
0-3	EK	Executive key
4	CACT	Cache is active
5	ENXIX	Enables extended indexing mode
6,7	PAR	Parity interrupt codes: 0 disables, 1 enables double-bit errors only, 2 halts processor on double-bit errors, 3 enables all parity errors
8	FPP UFL	Enables FPP interrupt on underflow
9	OFL INT	Enables interrupt on overflow
10	DIS MEM	Disables main memory (cache diagnostic)
11	STEP	Enables single step mode (for internal use only)
12	RTCEN	Enables real-time clock interrupt
13	HALT	Halts processor operation (for internal use only)
14		64K-word addressing mode
15	OFL	Overflow

Table 19-4. Floating-Point Status Word

BIT	DESCRIPTION
0	Sign
1	Zero
2	Underflow
3	Overflow
4-15	Not used

APPENDIX A
CONTROL PANEL OPERATION

Differences Between Sperry Univac V70 and 620-f computer.

	620/f	V70 Series
Power Switch	'OFF' - All power voltages off.	'OFF' Same as 620/f
	No corresponding position.	'HOLD' - All power voltages off except those required to preserve contents of semiconductor memory.
	'PWR ON' - All power voltage on and console switches enables.	'ON' Same as 620/f
	'PWR ON DISABLE' - All power voltages on and all console switches (except powerswitch) disabled.	'CONSOLE DISABLE' - All power voltages on and all console switches (except power switch and display select) disabled.
STEP/RUN Switch and STEP and RUN Indicators	The 'STEP/RUN' switch locks in either STEP or RUN position.	The 'STEP/RUN' switch is alternate action.
	If computer is in step mode: a. Pressing STEP/RUN switch to RUN position primes the computer to enter the run mode when the START switch is pressed. The step indicator remains on.	If computer is in step mode: a. Pressing STEP/RUN switch primes the computer to enter the run mode when the START switch is pressed. The step indicator is extinguished and the RUN indicator links.
	b. Pressing the START switch executes the instruction in the I register, and fetches the next instruction from the address specified by contents of the P register and places it in the I register. The STEP indicator goes out and the RUN indicator lights	b. Same as 620/f
	If computer is in run mode: a. Lifting STEP/RUN switch to STEP halts computer after completing execution of the current instruction and fetches the	If computer is in run mode: a. Pressing the STEP/RUN switch halts the computer after completing execution of the current instructions and fetches the next instruction and sets it in the I register. The RUN

	620/f	V70 Series
	next instruction and sets it in the I register. The RUN indicator goes out and the STEP indicator lights.	indicator goes out and the STEP indicator lights.
	b. In the computer encounters a HLT instruction the RUN indicator goes out and the STEP indicator lights.	b. If the computer encounters a HLT instruction, a halt loop in entered and the RUN indicator begins to blink.
START Switch	START is a momentary switch. Pressing it with the STEP/RUN switch in the RUN position places the computer in the run mode and starts the program. Pressing the START switch when the STEP/RUN switch is in STEP executes the instructions in the I register (except HLT) and fetches the next instructions from the address specified by the contents of the P register and places it in the I register.	START is a momentary switch. Pressing it with the RUN indicator blinking places the computer in the run mode and starts the program. Pressing the START switch when the STEP indicator is on executes the instructions in the I register (except HLT) and fetches the next instruction from the address specified by the contents of the P register and places it in the I register.
BOOTSTRAP Switch (BOOT)	Bootstrap is a momentary switch permitting loading of the binary load/dump program into memory. It is active with the STEP/RUN switch in the RUN position.	BOOT is a momentary switch permitting loading of the binary load/dump program into memory. It is active with the RUN indicator blinking.
REGISTER Switches (DISPLAY SELECT) Switches and Indicators	Pressing one of the five REGISTER switches selects the designated register (X, B, A, I, or P) for display or entry. Only one register can be selected at a time. Pressing two or more REGISTER switches simultaneously OR's the front panel display, except the I register will display the I register regardless of other selections.	Pressing on of the five DISPLAY SELECT switches selects the designated register (MEM, STATUS, I, P, or REG) for display or entry (except for STATUS). The MEM register is used for entry or display of memory data. The STATUS register displays the computer status including overflow (bit 8). I and P correspond to the same registers as used in the 620/f. REG is for display of any of the computer's 16 general registers as further specified by the REG select

	620/f	V70 Series
	Switches mechanically latch on. Previously selected switch must be turned off before next selection is made.	display. A, B, and X are general registers 0, 1, and 2 respectively. Only one register can be selected at a time. Pressing two or more register switches simultaneously can result in an invalid display. Switches electronically latch on. Indicators above switches designate selected registers. Pressing a new selection automatically cancels the previous selection.
REG SELECT (1 2, 4, 8) CLEAR and INCR Switches and Indicators	These switches correspond to A, B, and X REGISTER switches in use.	Used in conjunction with the REG switch. Used to designate one of 16 general registers for display or entry. The desired register is selected by entering the appropriate binary code via the register select switches. The binary values of each switch are indicated above (8, 4, 2, 1). Switches are momentary. Pressing any of the 4 select switches causes that bit to be set and the corresponding indicator to light. To reset all 4 bits press the CLEAR switch. The INCR switch causes the selected register number to be incremented by one each time the switch is pressed. The binary codes for the A, B, and X registers are: 8 4 2 1 A 0 0 0 0 B 0 0 0 1 X 0 0 1 0
REGISTER Entry Switches and DISPLAY Indicators (DISPL CLR) (LOAD)	a. The 16 indicators display contents of a selected register when the computer is in the step mode.	a. Same as 620/f
	b. To display the contents of a register place the STEP/RUN switch to STEP and press the REGISTER switch for the desired register.	b. To display the contents of a register press the STEP/RUN button (if the RUN indicator is on) and press the desired switch in the DISPLAY SELECT group. If REG is selected set the proper binary code into the REG SELECT group.
	c. The display indicators light when they	c. Same as 620/f.

	620/f	V70 Series
	correspond to register bits that contain ones.	
	d. To enter data or instructions in a register: (1) Display the contents of the register (2) Enter ones by pressing down on the register entry switches corresponding to the bid to be set. (3) Enter zeros in other bits by pulling up in all other register entry switches. The indicator lights do not change when the register entry switches are manipulated they still display the contents (4) When the desired configuration is entered in the register entry switches, press LOAD. This loads the register with the configuration entered on the switches and the indicators change to display this new configuration is the register.	d. To enter data or instructions in a register (except STATUS or REG #3 or #5). (1) Display the contents of the register. (2) Clear the register to all zeros by pressing the DISPL CLR button. All the display indicator lights will go out (except for STATUS or REG #5). (3) Enter ones in the desired bit positions by pressing the appropriate register entry switches. The corresponding indictor lights will turn on. (4) No further action is of the register. necessary as the actual selected register was first cleared then set to the desired configurations in the two preceding steps.
	e. Switches are mechanically latching.	e. Switches are momentary.
LOAD Switch	LOAD is a momentary spring-loaded switch. When the computer is in step mode and a register has been selected, pressing this switch loads the register with the bit configuration entered on the register entry switches.	No corresponding switch is needed as data is directly entered into the selected register.
REPEAT Switch	REPEAT is a toggle switch that is operative in both step and run modes. To repeat an instruction contained in the I reg-	No corresponding switch is needed as a completely different procedure is used in entering or displaying memory data. (Described later).

	620/f	V70 Series
	ister press REPEAT, and then press START. The instruction is executed again and the program counter advanced. However, the contents of the I register remain the same. This switch is used in entering or displaying memory data in sequential locations.	
SENSE Switches (and Indicators)	The three SENSE switches are toggle switches permitting program modification by the operator.	The three SENSE switches are alternate action switches permitting program modification by the operator. The indicators display the current status of the switches. Pressing any sense switch changes the status of the corresponding switch from off to on or on to off.
INT (Interrupt) Switch	INT is a momentary switch used to interrupt the computer. It is functional only when the computer is in the run mode.	INT is a momentary switch used to interrupt the computer. It is functional only when the computer is in the run mode (RUN indicator on constantly).
RESET Switch	RESET is a momentary switch used for initializing control and for stopping I/O operations. Pressing this switch halts the computer and initializes the computer and peripherals. This switch is electrically interlocked with the STEP/RUN switch and is disabled when the latter is in RUN.	RESET is a momentary switch used for initializing control and for stopping I/O operations. Pressing this switch halts the computer and initializies the computer and peripherals. It is not interlocked with the STEP/RUN switch on display.
OVFL (Overflow) Indicator (Bit 8 of STATUS Display)	OVFL lights whenever an overflow exists. This is true when the computer is in either STEP or RUN mode.	Overflow may only be observed with the computer halted (STEP indicator on or RUN indicator blinking.) It may be observed by pressing the STATUS switch noting to state of bit 8. Bit 8 is on whenever an overflow condition exists.
ALARM Indicator	Alarm lights to signal an overheated system.	No corresponding indicator exists.
Loading Sequential Memory Addresses	a. Set STEP/RUN to STEP and press REPEAT	a. Place the computer in the STEP mode by pressing the STEP/RUN switch if the

	620/f	V70 Series
		RUN indicator is on or blinking.
	b. Load the P register with the base address.	b. Load the P register with the base address.
	c. Load into the I register a storage instruction (STA, etc.) with 100 in the M field (relative addressing) and zero in the A field.	c. Select MEM on the display select.
	d. Select the register specified by the storage instruction in step c.	d. Load the console (MEM) register using the data entry and DISPL CLR switches.
	e. Load the selected register using the data entry switches.	e. Press ENTER to enter the data into the memory locations.
	f. Press START to execute the instructions in the I register.	f. Repeat steps e and f until all instructions (or data) are loaded. The next address to be loaded can be observed by displaying the P register.
	g. Repeat steps e and f until all instructions (or data) are loaded. The next address to be loaded can be observed by displaying the P register.	
Displaying Sequential Memory Addresses	a. Place STEP/RUN to STEP and press REPEAT.	a. Place the computer in the STEP mode by pressing the STEP/RUN switch if the RUN indicator is on or blinking.
	b. Load the P register with the base address.	b. Load the P register with the base address.
	c. Load into the I register a loading instruction (LDA, etc.) with 100 in the M field (relative addressing), and zero in the A field.	c. Select MEM in the display select.
	d. Select the register specified by the loading instructions in step c.	d. Press DISPL (Display) once for each memory location to be displayed.
	e. Press START once for each memory location to be displayed.	

	620/f	V70 Series
Executing of a Stored Program (Manually	a. Select step mode and turn off REPEAT.	a. Select step mode by pressing the STEP/RUN switch if the RUN light is on blinking.
	b. Set the P register to the first address of the program.	b. Same as 620/f
	c. Clear the I register.	c. Same as 620/f
	d. Press START	d. Same as 620/f
	e. Press START again to execute the instruction and to load the next instruction with the I register.	e. Same as 620/f
	f. Repeat step e once for each instruction.	f. Same as 620/f
MISCELLANEOUS		a. General register R3 contains zeroes while in the step mode and cannot be manually altered.
		b. General register R5 contains ones while in the step mode and cannot be manually altered.
		c. Pressing the I Display Selection clears the REG select display.
		d. To use the TSA instruction the MEM display selection should be made.
		e. While the computer is running I/O data input or operations to device code octal 77 addresses the console display.
		f. Console display indicators only represent actual register contents while the computer is not in the run mode.