V70/620 MAINTAIN III

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 Varian Data Machines, Publications Department, 2722 Michelson Drive, P.O. Box E, Irvine, California, 92664.

See addenda at the back of this manual

varian data machines / a varian subsidiary
2722 michelson drive/p.o. box c-19504/irvine/california/92713

varian data machines

98 A 9952 071

OCTOBER 1975

FOREWORD

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

RELATED DOCUMENTATION

The computer handbook provides data about capabilities and use as well as a brief guide to the other software for the system. The maintenance manuals are provided for every computer and each of the internal options and peripheral controllers. These manuals include information about the operation, installation, 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

FOREWORD

SECTION 1
SYSTEM OVERVIEW

SECTION 2
TEST EXECUTIVE PROGRAM

SECTION 3
INSTRUCTION TESTS

SECTION 4
MEMORY TEST PROGRAM

SECTION 5
TELETYPE TEST PROGRAM

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

APPENDIX A
CONSOLE OPERATION

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1
SYSTEM OVERVIEW

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

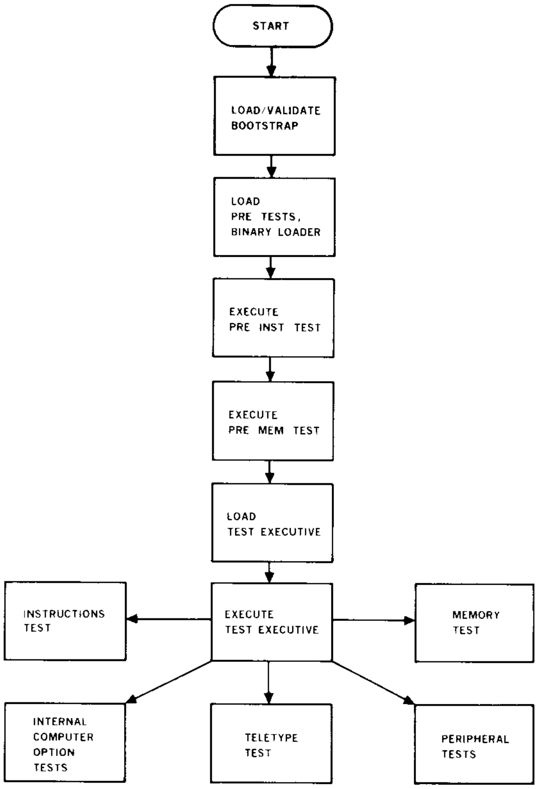
The test programs in this manual cover only the computers and internal options. Refer to the Bibliography for a list of Software Performance Specifications (SPS) covering peripheral test programs that operate in the MAINTAIN III Test Program System.

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 Varian-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.

1.3 BIBLIOGRAPHY

The following list of specifications cover MAINTAIN III Test Programs for current product-line peripherals.

SPS	Test Program
89A0180	Card Reader Test
89A0199	Card Punch Test
89A0247	Magnetic-Tape Test
89A0239	Moving-Head Disc Test (70-750x)
89A0207	Moving-Head Disc Test (70-76xx)
89A0194	Drum/Fixed-Head Disc Test (70-770x)
89A0189	Paper-Tape Reader/Punch and BIC Test
89A0174	Line Printer Test
89A0228	Universal Asynchronous Serial Controller Test
89A0274	Memory Map Test
89A0249	Varian 73 WCS Test
89A0313	Floating-point unit Test
89A0278	Moving-Head Disc Test (70-751x)
89A0202	Relay Contact I/O Module Test
89A0303	Statos 31 Basic System Test
89A0308	Statos 31 Option System Test
89A0323	Statos 33 Basic Test
89A0258	5402 DAMC Test
89A0275	5401 DAMC Test

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

After bringing up computer power:

a. Enter step mode.

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

c. Press RESET to initialize the computer control circuits.

d. Initialize the TTY by setting control to LOCAL (off-line), typing CONTROL, D, T, and Q, and returning to on-line. Position the test program tape in the reader with the first nonblank binary frame at the reading station. Set the reader control level to STOP.

   If using the high-speed tape reader, set the LOAD/RUN switch to LOAD and position the tape in the reader with the first nonblank binary frame at the reading station. Set the LOAD/RUN switch to RUN.

(The Varian part number is punched in the leader portion of the object tape (e.g., 92U0106 013x in the test executive tape, where x indicates the revision level). Position the tape in the reader past this area.)

If the computer is equipped with Teletype or HSPT automatic bootstrap loader (ABL), press RESET, place the computer in run mode, and press BOOT. Proceed to step k.

If using the card reader, place the executive object card deck in the card reader and make ready.

If using magnetic-tape, place the MAINTAIN III test object tape on the appropriate tape drive unit at proper density and proceed to step e through j.

e. Set the REPEAT switch.

f. Load 054000 in the instruction register.

g. Load 000200 in the P register.

h. Manually load the appropriate bootstrap routine listed in table 2-5, 2-6, 2-7, or 2-8 into the A register.

i. Load 000212 in the P register; 007000 in the X register; and zero in the A, B, and instruction registers.

j. Reset REPEAT, press RESET, and, in run mode, press START or RUN.

k. If using the TTY, set the reader lever 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. Teletype Paper-Tape Bootstrap Routine

```
 Address   Instruction          Symbolic Coding
           Code

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

* = where

Z = Device address, (normally 01)

Table 2-6. High-Speed Paper-Tape Bootstrap Routine

```
 Address     Instruction                    Symbolic Coding
             Code
                               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       000213
 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 2-7. Magnetic-Tape Bootstrap Routine

```
 Address     Instructions                   Symbolic Coding
             Code
                               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 2-8. Card Bootstrap Routine

```
 Address    Instruction                Symbolic Coding
            Code

                            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 system the message will be:

```
THIS IS THE V75 TEST EXECUTIVE

MEMORY SIZE IS nk
```

where n indicates memory size (for example 8 or 12, or multiples of 4). The program then waits for control statement input.

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.

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.	Initialze 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.

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.	Load and execute program on drive d, controller c.
Px,y,z,dc.	(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.

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

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 2-9).

Table 2-9. 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 system, if the test executive does not print:

```
THIS IS THE V70/620 TEST EXECUTIVE

MEMORY SIZE IS nK
```

or, if on a V75 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
```

Correctly re-input the erroneous 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 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 computer.

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

INSTRUCTION TESTS

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:

INSTRUCTION TESTS

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 system.

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 tape leader (part 1, 2, or 3) in the TTY with leader at the reading station.

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	620/f with standard instruction set
4	620/f with optional instructions/70 system
7	V75 computer

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 through 8, 11
620/i or 620/L with optional instructions	1 through .11
620/f with standard instruction set	1 through 9, 11, 12, 13
620/f with optional instructions/70 system	1 through 13
V75 systems	1 through 13 14 through 22

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), when applicable (section 2.1.1.1), 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^{0}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

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. 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 of memory (maximum, 32K), a 33/35 ASR TTY (Varian 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 consists of five basic subtests. Part 2 consists of six basic subtests.

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 ones will be stored in a given address. Then 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.

The worst case pattern masks for the memory stack along with a respective error number is indicated below:

Error Number	Worst Case Pattern Mask	
	Bits set	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)

Part 1 of the memory test program tests those areas of the first 8K memory module that are not tested by part 2 (i.e., addresses 000000, 000001, 000040 through 000043, 000400 through 000621, and 003550 through 007755). It contains a modified test executive; the standard test executive program (section 2) must be reloaded before further tests can be executed.

Part 2 of the memory test program verifies the operation of all of memory or of specific 8K modules. If module 0 is specified, only addresses 000002 through 000037, 000044 through 000077, 000120 through 000377, and 000622 through 003547 are tested.

Memory parity is enabled at the beginning of part 2 of the memory-test program. If the memory-parity option is included in the system a parity error interrupts the program. The parity-error types for 620 series computers, are instruction, address, operand, and trap. For V70 series computers, only a parity error can occur.

Because the memory test does not otherwise test interrupts, a trap parity error indicates a parity hardware malfunction. Parity errors produce error messages indicating the type of error and the associated memory address (section 4.3).

The memory test program does not test the memory-parity option; it merely reports the occurance of a memory parity interrupt.

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
2	Before error halt: Halt on error After error halt: Continue testing	 Bypass error halt 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/f or V70 series computer, pressing the INT (interrupt) switch returns control to the test executive.

4.2 EXECUTING THE MEMORY TEST PROGRAMS

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

```
MEMORY TEST.
```

For Part 2 of the memory test program, 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 three parameters).

The following message refers to the V70 parity error interrupt:

```
V70 PARITY ERROR INTERRUPT
LOCATION (620=0) =
```

If the computer is a 620 series, respond with 0; then, the 620 parity errors are retained. If the computer is a V70 series, respond with the even address of the parity error locations. (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 following paragraphs describe further operation of both Part 1 and Part 2 of the memory test program.

The memory test program validates the operator responses described above and 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 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.

4.3 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.

Following detection of a parity error, the computer halts regardless of SENSE switch settings, and 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 =
```

continueo

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 (Varian-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). Varian 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				b7 b6 b5 →	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b4	b3	b2	b1	COLUMN → / ROW ↓	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

TELETYPE TEST PROGRAM

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 (Varian-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:

TELETYPE TEST PROGRAM

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
```

Table 6-1. Typical PF/R Service Routine *(continued)*

```
      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 restrictions. 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

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.

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 for 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:

POWER-FAILURE/RESTART TEST

```
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
```

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.
1234567866666666
```

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 FRC	
CIB 047	102647		

* 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)

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. After the user inputs this number, the computer issues a carriage return and line feed and types:

```
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.

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 SUMARY 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
```

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.

APPENDIX A

CONSOLE OPERATION:

Differences Between Varian V70 and 620-f computer.

	620/f	Varian 70 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	'ON' Same as 620/f
enables.		
	'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	Varian 70 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	Varian 70 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	Varian 70 Series
	correspond to register bits that contain ones.	
	d. To enter data or instructions in a register: (1) Display the contents of 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) Enter ones by pressing down on the register entry switches corresponding to the bid to be set.	(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 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	(3) Enter ones in the desired bit positions by pressing the appropriate register entry switches. The corresponding indictor lights will turn on.
	(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.	(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	Varian 70 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	Varian 70 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	Varian 70 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.

ADDENDUM 1

V70/620 MAINTAIN III
REFERENCE MANUAL

98 A 9952 071

This addendum contains changes to the MAINTAIN III Reference Manual.

<u>PAGE</u> <u>ACTION</u>

2-13 Add the following section:

2.2.5 <u>Loading AID III</u>

The Varian 70 Series AID III Debugging Program is supplied for use with all V70 series systems and contains the software necessary to facilitate on-line program checkout and correction.

To load AID III from the composite magnetic tape object, perform the following:

a. Position the composite magnetic tape object at load point.

b. Load MAINTAIN III Executive using the bootstrap loader.

c. Using the MAINTAIN III F command, position the composite magnetic tape object to the desired AID III object.

d. Load the AID III object by setting the X register to 007000 and the P register to 000212.

e. The TTY now assumes control with AID III being loaded. For systems without a TTY, set the RUN indicator on and press START.

PAGE ACTION

4-1 Replace the fourth paragraph in the left-hand column with:

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

Replace the paragraph beginning with "Circulating Bit (test 6)" with:

Circulating Bit (test 6). This routine uses the worst case patterns to determine whether all zeros or all ones 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

	Worst Case Pattern Mask	
Error Number	Bits Set	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.

PAGE ACTION

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.

4.2 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 worse case memory patterns with the message:

```
WORSE 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 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).

The memory test program outputs the following message:

```
CYCLES =
```

PAGE ACTION

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
```

PAGE ACTION

The following message is output:

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

This refers to the V70 Parity Error Interrupt. For 620 series computers, respond with 0; then, the 620 parity errors are retained. For 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 worse-case memory patterns with the message:

```
WORSE 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 =
```

PAGE ACTION

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 620 control panel lights momentarily when an error is detected.

PAGE ACTION

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.

ADDENDUM 2

V70/620 MAINTAIN III REFERENCE MANUAL

98 A 9952 071

This addendum contains changes to the MAINTAIN III Reference Manual.

PAGE	ACTION
2-5	Replace the three lines in the right-hand column beginning with "Where n indicates memory size ..." with the following: 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.
3-1	Replace the sentence beginning with "Subtests 14..." with the following: Subtests 14 through 22 are applicable only to the V75 and V77 computers.
3-4	Replace the paragraph beginning with "Subtest 14..." with the following: Subtests 14 through 22 are applicable only to the V75 and V77 computers.
3-5	In the right-hand column, replace all information through the table containing the Computer and Subtests headings with the following: b. Position the Instructions Test Program Object on the Object Input Device.

ISSUED: SEPTEMBER 1976

PAGE	ACTION
3-5	Continued

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	620/f with standard instruction set
4	620/f with optional instructions/70 system
7	V75 computer
10	V77 computer without optional control panel
11	V77 computer with optional control panel

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.

PAGE	ACTION
8-2	At the end of section 8.2.1, add the following: c. On the V77 1. The Internal Interrupt (II)
8-4	In the first sentence of the second paragraph of section 8.6.2.1, add the following type CPU: 2 for V77.
9-2	In section 9.5.1, add the following type CPU: 2 = V77. In the message of item 'a' of section 9.5.1, add: 2 = V77.