ADDENDUM TO THE PS2/MPS DIAGNOSTICS MANUAL
(#9Ø1171-Ø5Ø NC)
DIAGNOSTIC REVISIONS Ø4Ø1

December 9, 198Ø

Customer Engineering Department
Evans & Sutherland Computer Corporation
58Ø Arapeen Drive
Salt Lake City, Utah 841Ø8

All information contained herein together with all drawings, diagrams and specifications herein or attendant hereto are, and remain, the property of Evans & Sutherland Computer Corporation. Many of the intellectual and technical concepts described herein are properietary to Evans & Sutherland and may be covered by U.S. and Foreign Patents or Patents Pending or are protected as trade secrets. Any dissemination of this information or reproduction of this material for commercial or other purposes other than the express purpose for which it has been made available is strictly forbidden unless prior written permission is obtained from Evans & Sutherland Computer Corporation.

Evans & Sutherland Computer Corporation assumes no responsibility for any errors that may appear in this manual. The information in this document is subject to change without notice.

TABLE OF CONTENTS

1. INTRODUCTION

This addendum lists all diagnostic revisions since November 1979. The release in which the change first occurred (Ø312, Ø313, or Ø4Ø1) is also indicated.

1.1. CHANGES IN STANDARD OPERATION

The "S" command was added, which is similar to "X", but diagnostics which support loop on error will terminate after the first error eligible for looping (Ø312). The "R" command for repetition was added (Ø4Ø1). For example, "R2X" would call each phase once, then loop back and call each phase again, in comparison to "P2X" which would call phase 1 twice, then phase 2 twice. The "R" command is useful for running a diagnostic overnight, and ensures that all phases will be called, whereas "P" with a large count might spend all night in phase 1.

1.2. GENERAL CHANGES IN VISUAL TESTS

The following changes apply to QSDØ2Ø.SØ6, QSDØ26.SØ4, QSDØ27.SØ4, QSDØ28.SØ6, RSDØØ5.SØ3, RSDØØ6.SØ4, RSDØØ7.SØ4, RSDØØ8.SØ4, and RSDØØ9.SØ5 (Ø312). Phases have been rearranged so that a simple "X" command invokes all autorefresh phases, but no DMA phases. A reset will occur at the end of the test, killing autorefresh. DMA phases must be explicitly requested with the "D" command. If a single autorefresh phase is requested with "D", in most cases the test pattern will continue to be autorefreshed after the program has stopped. In case of uncertainty, use the "H" command for help. These tests were also changed to relinquish the CPU while killing time in autorefresh phases. The QSDØ2Ø raster was changed to full screen. QSDØ2Ø, RSDØ13, and RSDØ14 allow the user to override any portion of all line generator status commands. The writeup in this addendum for RSDØ13 describes this feature in detail.

2. NEW DIAGNOSTIC AND TEST PROGRAMS

Listed below are the new diagnostic and test programs. The description of these programs are listed in Addendum A.

2.1. QSDØ21.SØ2 (Ø4Ø1) INTERACTIVE BLACK AND WHITE TUNING TEST

The most useful black and white test patterns may be interactively selected at the keyboard, without the need for restarting the program.

2.2. QSDØ34.SØ3 (Ø4Ø1) LINE GENERATOR FIFO TEST

Version SØ1 (Ø312) of the line generator FIFO test was not correctly detecting FIFO full in Phase 3. The format of error messages was also revised.

2.3. QSDØ35.SØ2 (Ø4Ø1) LINE GENERATOR INTERACTIVE TEST

Version SØ2 incorporates a picture system reset at termination to allow the line generator and RTI to resume normal operation.

2.4. QSDØ36.SØ2 (Ø4Ø1) LINE GENERATOR STATE MACHINES TEST

Error messages were enhanced over version SØ1 (Ø312).

2.5. QSDØ37.SØ2 (Ø4Ø1) CHARACTER GENERATOR STATE MACHINES TEST

Error messages were enhanced over version SØ1 (Ø312).

2.6. QSDØ4Ø.SØ2 (Ø4Ø1) INTERACTIVE CSM TUNING TEST

Like QSDØ21, but for use with the Color Shadow Mask (CSM) display.

2.7. QSD117.SØ2 (Ø312) MULTI USER REFRESH CONTROLLER INTERRUPTS TEST

Changes were made in interrupt handling over version SØ1, which are transparent to the user. This diagnostic is the same as QSD113 phase 13, except that interrupts to the host CPU are enabled, and occurrence of (only) system interrupts is verified. This test does not run under mapped RSX-11M.

2.8. QSD14Ø.SØ1 AND QSD141.SØ1 (Ø4Ø1) EXTENDED MEMORY DIAGNOSTICS

These diagnostics test the extended memory control logic, initialization, error correction and detection, address mapping, and data storage.

2.9. QSD151.SØ1 (Ø4Ø1) "FAST" DMA TEST

This is the same as QSDØØ1 except that shorter test data sequences are used, and QSDØØ1 phases 7 through 12 are eliminated as they cannot readily be implemented in a multi-user operating system. This diagnostic replaces QSDØØ1 on the VAX.

2.1Ø. RSDØ13.SØ3 AND RSDØ14.SØ3 (Ø4Ø1) CSM VISUAL TEST

The user may specify any subfields in LG status commands by modifying PSTB locations. There are new test patterns in RSDØ14.

3. OTHER REVISIONS

Listed in the subsections below are other revisions and restrictions.

3.1. PROGRAM REVISIONS

3.1.1. QSDØØØ.SØ3 (Ø4Ø1) Direct I/O Interface Diagnostic

Speedup by elimination of the data sequence Ø through 177777 (all 32K combinations) unless explicitly requested by modification of P1ØØ. A shorter data sequence is normally used.

3.1.2. QSDØØ2.SØ2 (Ø312) Memory Diagnostic

Minor changes in operator prompting; also changed to relinquish the CPU while killing 6Ø seconds.

3.1.4. QSDØØ3.SØ2 (Ø4Ø1) MAP Statux, Maintenance, and Match Tests

Elimination of spurious occurrences of an error message in phase 1 (index = 134).

3.1.5. QSDØ28.SØ6 (Ø4Ø1) Character RAM Visual Test

DMA phases (4 and 5) were not putting up a picture in version SØ5.

3.1.6. QSD1ØØ.SØ2 (Ø312) Interrupt Diagnostic

New interrupt handling for compatibility with the VMS Operating System. All phases now terminate. Phases 3 through 5 terminate when no interrupt occurs on the device in question within a timeout period. The timeout interval may be modified through P1ØØ.

3.1.7. QSD1ØØ.SØ4 (Ø4Ø1) Interrupt Diagnostic

Elimination of a bug in phase 1 which first surfaced with VMS 2.Ø. Changes in operator prompting in phase 1. As before, this test does not run under mapped RSX-11M.

3.1.8. QSD1Ø2.SØ2 (Ø4Ø1) Data Tablet Diagnostic

New help message. Phases 2 and 3 are not valid on a Multi Picture System.

3.1.9. QSD113.SØ3 (Ø4Ø1) Multi User RFC Microcontroller Diagnostic

Phase 12 and timeout bugs eliminated.

3.1.1Ø. QSD116.SØ2 (Ø312) Multi User RFC Device Control Diagnostic

Phase 1 bug eliminated.

3.1.11. QSD12Ø.SØ5 (Ø4Ø1) Memory Test Using Map Active I/O

Extended memory was incorporated, and several bugs eliminated.

3.2. ACCEPTANCE TEST PROGRAM REVISIONS

3.2.1. RSDØØ5.SØ3 (Ø312) MAP Finite State Machines Demo

This test now uses autorefresh and runs under mapped RSX-11M. The background grid is drawn at lower intensity.

3.2.2. RSDØØ6.SØ4 (Ø312) Refresh Rates Test

The order of the phases was reversed.

3.2.3. RSDØ1Ø.SØ3 (Ø4Ø1) MAP Tranformations Demo

The "Update Rate" function was replaced by "memory walk" function which, when enabled, periodically relocates data buffers throughout Extended or Non-extended Picture System Memory. Several bugs were eliminated, including cursor overflow. P11Ø = line generator speed. P111 non-zero for high resolution data tablet. Memory update method was optimized. See Appendix A for functional description of this test.

3.2.4. RSDØ11.SØ4 (Ø4Ø1) MAP Transformations Demo

Same changes as RSDØ1Ø, excluding those concerned with the data tablet. Light pen handling changed to accommodate the "memory walk" function. As in RSDØ1Ø, P11Ø = line generator speed, to allow half-speed with a CSM. See Appendix A for functional description of this test.

3.2.5. RSDØ12.SØ2 (Ø4Ø1) Peripherals Test

Improved help message.

3.3. QSDDT REVISIONS

3.3.1. QSDDT.SØ5 (Ø312) Interactive Diagnostic Program

New "Y" command to sense interrupts (not in mapped RSX). New "!" command to search memory for a bit pattern is the converse of the "?" command. These commands use P1ØØ as a test mask. If P1ØØ is non-zero, any zero-bits are "don't-care" positions. Problems in the "C" address mode were eliminated.

3.4. REMOTE TERMINAL INTERFACE RESTRICTIONS

Use of the RTI as a terminal on the host computer for operation of picture system diagnostics is restricted. The following programs do not operate properly in this mode: DMA phases of visual tests such as QSDØ2Ø, QSDØ26, QSDØ27, QSDØ28, RSDØØ5 and others; line generator diagnostics QSDØ34 through QSDØ37; refresh controller diagnostics QSD112, and possibly QSD113 and QSD116; and light pen diagnostic QSD1Ø8 phases 4 through 7.

3.5. OTHER RESTRICTIONS AND KNOWN PROBLEMS

Data tablet diagnostic QSD1Ø2 phases 2 and 3 will report errors when the "free run" switch on the tablet controller card is in its normal setting for a Multi Picture System (195181-1ØØ IC 42 switch 2 open). Some diagnostics, such as QSDØØØ, QSDØ11, QSD1ØØ, QSD1Ø8, and particularly QSDØØ2, run much more slowly on the VAX than on the PDP-11. For diagnostic operation on the VAX, see Appendix F of "Getting Started with the Multi Picture

System and the VAX/VMS Operating System." In QSD034.S03 phase 1, error number 2 (PGYBUS), the received data is incorrectly reported in the error message. The printed data was actually received on the PGXBUS, although an error did occur on the PGYBUS. In some instances QSD001 has been distributed as version S02, which should have been renamed QSD151 version S01. QSD001.S01 remains the correct version of the diagnostic. QSD001 is no longer available on the VAX, but is replaced by QSD151. Neither QSD001 nor QSD151 is available under mapped RSX-11M. QSD003 does not declare its version number which is now S02. RSD000 through RSD002 and RSD004 have been distributed to a few mapped RSX-11M systems, but will not run. The error message, "DMA error; R5 = XXXXXX" will occur, indicating a software error, i.e. that a DMA was attempted under mapped RSX.

Following is a summary of RTI, mapped RSX, and some other restrictions:

DIAGNOSTIC	RUNS UNDER MAPPED RSX	MAY BE USED WITH RTI	OTHER RESTRICTIONS
QSD001.S01	NO		NO VMS
QSD017.S02	NO		
QSD018.S04	NO		
QSD020.S06	SOME PHASES	SOME PHASES	
QSD026.S04	SOME PHASES	SOME PHASES	
QSD027.S04	SOME PHASES	SOME PHASES	
QSD028.S06	SOME PHASES	SOME PHASES	
QSD031.S01			PS2 ONLY
QSD033.S02	NO	NO	
QSD034.S03		NO	
QSD035.S02		NO	
QSD036.S02		NO	
QSD037.S02		NO	
QSD100.S04	NO	NO	
QSD102.S02			PS2 ONLY

(continued)

DIAGNOSTIC	RUNS UNDER	MAY BE USED	OTHER
	MAPPED RSX	WITH RTI	RESTRICTIONS
QSD108.S07		NO	
QSD111.S07			MPS ONLY
QSD112.S02		NO	MPS ONLY
QSD113.S04		NO	MPS ONLY
QSD114.S01			MPS ONLY
QSD115.S01			MPS ONLY
QSD116.S02		NO	MPS ONLY
QSD117.S02	NO		MPS ONLY
QSD151.S01	NO		
RSD000.S03	NO	NO	
RSD001.S02	NO	NO	
RSD002.S03	NO	NO	
RSD003.S02			PS2 ONLY
RSD004.S02	NO	NO	
RSD001.S04	SOME PHASES	SOME PHASES	

APPENDIX A. DESCRIPTION OF NEW DIAGNOSTIC AND TEST PROGRAMS

A.1. INTERACTIVE BLACK AND WHITE TUNING PROGRAM QSDØ21.SØ2

A.1.1. General Description

This is a test-pattern generation program to facilitate tuning of picture system black and white displays. It consists of one phase. The standard diagnostic commands D, P, L, C, and S have no meaning in this program. The user is allowed to select any of the following patterns for display in autorefresh mode:

- 1 - Circles at slow speed
- 2 - Big "X" for phase adjustment
- 3 - Boxes at slow speed
- 4 - Boxes at half speed
- 5 - Boxes at full speed
- 6 - Character set
- 7 - Color bars

The user may also issue two other commands to keyboard input:

- Minus 1 - Disable descriptive messages
- Minus 2 - Reset the picture system and halt

A.1.2. Diagnostic Procedures

A.1.2.1. Phase 1

a. Set the pattern number to 1. Go to step (d).

b. Get an octal number from the operator's terminal. If the number is minus 2, reset the picture system and stop. If the number is minus 1, set a flag to disable messages, and go to step (c). If the number is greater than zero, go to step (d).

c. This step is reached only if the operator inputs a minus 1, zero, or carriage return (which is the same as zero). Increment the pattern number. If the pattern number is greater than 7, set it equal to 1.

d. Reset the picture system. Initialize the refresh controller. If messages are enabled, print a description of the next test pattern.

e. Load picture sytem memory, starting at location zero, with the refresh data for the next test pattern. Start autorefresh, and go to step (b).

A.2. LINE GENERATOR FIFO DIAGNOSTIC QSD034.S03

A.2.1. General Description

This diagnostic checks the operation of the line generator FIFO which is located on the 195219-100 card. There are four phases and seven possible errors. Phase 1 verifies that the FIFO can accept four line generator commands. Phase 2 verifies that the FIFO keeps the commands in the same order in which they are entered. Phase 3 verifies that FIFULL does get asserted when the FIFO is full. Phase 4 is functionally similar to phase 2, however the data passed to the line generator is character data.

NOTE

> This diagnostic may not be run when the remote terminal interface is in use.

A.2.2 Diagnostic Procedures

A.2.2.1. Phase 1 Data Input Verification

a. Do steps (b) through (g) 20_{10} times using the following octal data: 0, 1, 2, 4, 10, 20, 40, 100, 200, 400, 1000, 2000, 4000, 10000, 20000, 40000, 100000, 177777, 52525, 125252.

b. Reset the line generator, then halt it by writing 10_8 then 2 into the PGSR. This leaves the mode bits in PGSR = 0, which gates the FIFO onto the PGXBUS and PGYBUS.

c. Write eight words equal to the current test data into the line generator passive input port. This should fill the FIFO.

d. Read the PGXBUS and compare it with the current test data.

e. Read the PGYBUS and compare it with the current test data.

f. If the PGXBUS data was incorrect, output the following message:

1: FIFO ERROR; DATA EXPECTED = XXXXXX DATA RECEIVED = XXXXXX

g. If the PGYBUS data was incorrect, output the following message:

2: FIFO ERROR; DATA EXPECTED = XXXXXX DATA RECEIVED = XXXXXX

A.2.2.2. Phase 2 Data Sequence Verification

a. Send the following eight words to the FIFO: ØØØ2ØØ, 177777, ØØØ6ØØ, ØØØØØØ, ØØØ2ØØ, 125252, ØØØ6ØØ, 52525.

b. Check the first six words out of the FIFO to verify that they are equivalent to the first six words sent to the FIFO. If not, output one or both of the following messages:

3: FIFO ERROR; DATA EXPECTED=XXXXXX DATA RECEIVED=XXXXXX

4: FIFO ERROR; DATA EXPECTED=XXXXXX DATA RECEIVED=XXXXXX

c. Send the following two words to the FIFO: ØØØ2ØØ, 177777.

d. Read two words out of the FIFO and check to see if they are equivalent to the last two words sent in step (a). If they are not equivalent, output one or both of the messages from step (b).

e. Read read two more words out of the FIFO. This time check to see if they are equivalent to the words sent in step (c). If not, output one or both of the messages from step (b).

f. Again, read still two more words out of the FIFO. By now the FIFO should be empty and zeros should be received. If not, output one or both of the messages from step (b).

A.2.2.3. Phase 3 FIFO Full Test

a. Reset and halt the line generator. Perform steps (b) and (c) nine times.

b. If the direct I/O interface is not ready (DIOREADY in the IOST register is not set), output the following message:

5: FIFO FULL OCCURED PREMATURELY: CNT = N

c. Write one word of data into the line generator passive input port.

d. Again, check DIOREADY, which should now (after nine repetitions of steps [b] and [c]) be low, meaning the FIFO is full. If the DIO interface is ready, output the following error message:

   5: FIFO FULL DID NOT OCCUR

A.2.2.4. Phase 4 Character Data sequence Verification

a. This phase is similar to phase 2, except that character data is used. This allows the left FIFO register and PGXBUS to be checked more thoroughly than in phase 2. Repeat steps (b) through (e) 4 times using 0, 77577, 52525, and then 25052 as the test data.

b. Write eight words equal to the current test data into the line generator passive input port.

c. Check the first six words of data out of the FIFO.

d. If the PGXBUS is in any instance unequal to the current test data, output the following message:

   6: FIFO ERROR; DATA EXPECTED=XXXXXX DATA RECEIVED=XXXXXX

e. If the PGYBUS is in any instance unequal to the current test data, output the following message:

   7: FIFO ERROR; DATA EXPECTED=XXXXXX DATA RECEIVED=XXXXXX

A.2.3. Error Analysis

ERROR #	EXPLANATION
1	The X-word of the line generator command was incorrect as it was read out of the FIFO. This could be due to one or more of the following four reasons: 1. The data loaded into the FIFO was incorrect. 2. The FIFO registers are malfunctioned. 3. The PGXBUS is malfunctioned, and/or 4. The maintenance mode of the line generator is malfunctioned.
2	This is the same as error 1 except that it was the Y-word of the FIFO that was wrong.

	Therefore, if a bus is malfunctioned, it would be the PGYBUS instead of the PGXBUS.
3, 6	Interpret this as you would error 1. However, suspect the FIFO counters also.
4, 7	This should be the same as error 2, except that the FIFO counter is suspect.
5	The signal FIFOFULL is to be suspected here. Also check the FIFO counters.

A.3. PICTURE GENERATOR STATE MACHINE SEQUENCE TEST QSDØ35.SØ2

A.3.1. General Description

This diagnostic is designed as a tool and not a test. Therefore, there are no phases and no errors to be detected. However, the program requires an "X" command to begin execution. Program execution consists of halting and single-stepping the picture generator a specified number of times. Each time the picture generator is single-stepped, the current address of each of the state machines is displayed.

NOTE

This test should not be operated while the remote terminal interface is in use.

A.3.2. Diagnostic Procedures

a. Specify the number of picture generator steps to observe. A null entry causes the program execution to terminate. Any other response is interpreted as an octal number of single steps that the picture generator is to perform. A negative entry is used to specify that the line generator should not be reset, and the absolute value of the entry is the number of single steps requested.

b. The operator next specifies the picture generator command. Seven commands are coded into the program (numbered 1-7), or the operator may create some other command. The following is a list of the commands which have been coded into the program:

- 1. Absolute status command (2ØØ, Ø)
- 2. Relative status command (6ØØ, Ø)
- 3. Move command (13ØØØØ, 17ØØØØ)

- 4. Draw command (170200, 170200)
- 5. Character, four "A"s (40501, 40501)
- 6. Perform control character (300, 0)
- 7. Light pen sensitize (42600, 125250)

A null entry is interpreted as a zero. Any entry greater than 7 means the operator is specifying a command other than the ones listed above. Since a picture generator command consists of two words, an operator-generated command requires two carriage returns to produce one commmand. Four commands can be sent to the picture generator at one time by the operator. Whenever the program receives a distinct command, the command is sent to the picture generator via DIO. When four commands have been entered, the program begins single-stepping the line generator. The operator can overflow the line generator FIFO by using the negative entry option in step (a). Thus, by not allowing the FIFO to empty and entering more line generator commands, the FIFO could be overflowed. Should this occur, the following message will be displayed:

```
DIO NOT READY -LOOPING
```

c. When the program is expecting a picture generator command and receives a null (or zero), it begins the process of single stepping the picture generator the specified number of times. Before single stepping the picture generator, the program prints the following heading:

```
STEP ISM ASM MSM OSM CGSM
```

Where:

STEP is the step number for the line.
ISM is the input state machine current address.
ASM is the arithmetic state machine current address.
MSM is the multiplier state machine current address.
OSM is the output state machine current address.
CGSM is the character generator state machine current address.

Following this header, the picture generator is single-stepped the specified number of times. The states through which each state machine is sequenced are displayed.

Finally, the program loops back to step (a).

A.3.2.2. Program Example

The following is a sample program.

```
>RUN QSDØ35
QSDØ35.SØ2
PICTURE GENERATOR STATE MACHINE SEQUENCE TEST
%
X
RUNNING
HOW MANY STEPS?
1ØØ
  ENTER L.G. COMMAND:
1
Ø — Terminator for L.G. Command
```

```
STEP   ISM   ASM   MSM   OSM   CGSM

 1     Ø     Ø     Ø     Ø     5
 2     1     Ø     Ø     Ø     5
 3     22    Ø     Ø     Ø     5
 4     23    Ø     Ø     Ø     5
 5     25    2     Ø     Ø     5
 6     25    3     Ø     Ø     5
 7     25    3     4     Ø     5
.
.
.
74     Ø     3     5     6     5
75     Ø     3     5     6     5
76     Ø     3     5     6     5
77     Ø     3     5     6     5
1ØØ    Ø     3     5     6     5
HOW MANY STEPS?
1ØØ
  ENTER L.G. COMMAND:
5
6
7
Ø
```

the "CGSM" number
is directly equivalent
to the Rom Address on
the 10522 card

```
STEP   ISM   ASM   MSM   OSM   CGSM

 1     Ø     Ø     Ø     Ø     5
 2     1     Ø     Ø     Ø     5
 3     24    Ø     Ø     Ø     5
 4     21    Ø     Ø     Ø     5
 5     Ø     Ø     Ø     Ø     4
.
.
.
74     Ø     27    22    22    123
75     Ø     25    22    22    121
76     Ø     25    22    23    131
77     Ø     25    22    23    131
1ØØ    Ø     25    22    24    131
```

If instead you had
told it to step 4
then you would print
out up to this point
however the CGSM would
be at its next state
which in this case
would be (4)

```
HOW MANY STEPS?
1ØØ
  ENTER L.G. COMMAND:
3
4
Ø
STEP  ISM  ASM  MSM  OSM  CGSM

 1    Ø     Ø    Ø    Ø    5
 2    1     Ø    Ø    Ø    5
 3    3Ø    Ø    Ø    Ø    5
 4    31    Ø    Ø    Ø    5
 5    Ø     36   Ø    Ø    5
 6    Ø     36   Ø    Ø    5
 7    1     36   Ø    Ø    5
1Ø    36    31   Ø    Ø    5
11    35    37   Ø    Ø    5
12    Ø     4    14   Ø    5
13    Ø     4    14   Ø    5
14    Ø     4    14   Ø    5
15    Ø     34   14   Ø    5
16    Ø     5    14   Ø    5
.
.
.
77    Ø     5    3Ø   24   5
1ØØ   Ø     5    3Ø   24   5
HOW MANY STEPS?
4Ø
  ENTER L.G. COMMAND:
17ØØØØ
17Ø2ØØ
Ø
STEP  ISM  ASM  MSM  OSM  CGSM

 1    Ø     Ø    Ø    Ø    5
 2    1     Ø    Ø    Ø    5
 3    34    Ø    Ø    Ø    5
 4    35    Ø    Ø    Ø    5
 5    Ø     4    Ø    Ø    5
 6    Ø     4    Ø    Ø    5
 7    Ø     4    Ø    Ø    5
1Ø    Ø     34   Ø    Ø    5
11    Ø     5    Ø    Ø    5
.
.
.
37    Ø     5    3Ø   24   5
4Ø    Ø     5    3Ø   24   5
HOW MANY STEPS?
Ø
> (the program has terminated)
```

A.4. LINE GENERATOR STATE MACHINE TEST QSDØ36.SØ2

A.4.1. General Description

This diagnostic checks the line generator state machines. The control ROMs are located on the 195218-1ØØ, 195216-1ØØ, and 195213-1ØØ cards. These are the input control ROM, arithmetic control ROM, the multiplier control ROM, and the output control ROM.

Each state machine is checked individually by the diagnostic, and error messages are generated if an error occurs.

There are four phases to this diagnostic:

- Phase 1 - Input state machine control ROM.
- Phase 2 - Arithmetic state machine control ROM.
- Phase 3 - Multiplier state machine control ROM.
- Phase 4 - Output state machine control ROM.

<u>NOTE</u>

> This program may not be run while the remote terminal interface is in use

A.4.2. Diagnostic Procedures

A.4.2.1. Phase 1 Input State Machine Control ROM

The program steps for performing this test are:

a. Reset the picture system.

b. Reset and halt the line generator.

c. Repeat steps (d) through (g) 14 times, sending the following commands to the line generator:

- 1. Status command; reset character generator, dot mode, absolute coordinates (13ØØ, 1)

- 2. Draw command; draw to Ø, Ø (17ØØØØ, 17ØØØØ)

- 3. Draw command; draw to 2ØØ, Ø (17Ø2ØØ, 17ØØØØ)

- 4. Draw command; move to Ø, Ø (13ØØØØ, 17ØØØØ)

- 5. Move command; move to 2ØØ, Ø (13Ø2ØØ, 17ØØØØ)

- 6. Relative status command; (600,1)
- 7. Move command; move to 0, 0 (130000, 170000)
- 8. Relative status command; (600, 1)
- 9. Move command; move to 200, 0 (130200, 170000)
- 10. Relative status command; (600, 1)
- 11. Draw command; draw to 0, 0 (170000, 170000)
- 12. Relative status command; (600, 1)
- 13. Draw command; draw to 200, 0 (170200, 170000)
- 14. Light pen status; (42600, 125250)
- 15. 2 "delete" character commands; (77577, 1)

d. Verify that the line generator input state machine is in the reset state.

e. Reset and halt the line generator. Write the next two words of line generator command into the line generator passive input port.

f. Verify that the line generator input state machine got to the first state of the expected sequence for the last line generator command written. If not, output the following message:

```
**1:  INPUT STATE MACHINE ERROR**
  UNABLE TO SET TO STATE: XX RECEIVED STATE: XX
  LINE GENERATOR COMMANDS: XXXXXX, XXXXXX
```

g. Once the initial state of the expected sequence has been reached, single-step the line generator in maintenance mode, verifying each state in sequence until the reset state (state 0) is reached. If an error occurs, output the following message:

```
**2:  INPUT STATE MACHINE ERROR**
  START STATE: XX EXPECTED STATE: XX RECEIVED STATE: XX
  LINE GENERATOR COMMANDS: XXXXXX, XXXXXX
```

A.4.2.2. Phase 2 Arithmetic State Machine Control ROM

The program steps for performing this test are:

a. Reset the picture system.

b. Reset and halt the line generator.

c. Repeat steps (d) through (g) 14 times, sending the following commands to the line generator:

   - 1. Status command; reset character generator, dot mode, absolute coordinates (13ØØ, 1)

   - 2. Move command; move to Ø, Ø (13ØØØØ, 17ØØØØ)

   - 3. Draw command; draw to Ø, Ø (17ØØØØ, 17ØØØØ)

   - 4. Status command; relative coordinates (6ØØ, 1)

   - 5. Move command; move to Ø, Ø (13ØØØØ, 17ØØØØ)

   - 6. Status command; relative coordinates (6ØØ, 1)

   - 7. Draw command; draw to Ø, Ø (17ØØØØ, 17ØØØØ)

   - 8. Status command; light pen sensitize, absolute coordinates, segment name (4Ø2ØØ, 1)

d. Verify that the line generator arithmetic state machine is in the reset state.

e. Reset and halt the line generator. Write the next two words of line generator command into the line generator passive input port.

f. Verify that the line generator arithmetic state machine got to the first state of the expected sequence for the last line generator command written. If not, output the following message:

```
**3:  ARITHMETIC STATE MACHINE ERROR**
  UNABLE TO SET TO STATE: XX RECEIVED STATE: XX
  LINE GENERATOR COMMANDS: XXXXXX, XXXXXX
```

g. Once the initial state of the expected sequence has been reached, single-step the line generator in maintenance mode, verifying each state in sequence until the reset state (state Ø) is reached. If an error occurs, output the following message:

```
**4:  ARITHMETIC STATE MACHINE ERROR**
  START STATE: XX EXPECTED STATE: XX RECEIVED STATE: XX
  LINE GENERATOR COMMANDS: XXXXXX, XXXXXX
```

A.4.2.3. Phase 3 Multiplier State Machine Control ROM

The program steps for performing this test are:

a. Reset the picture system.

b. Reset and halt the line generator.

c. Repeat steps (d) through (g) 14 times, sending the following commands to the line generator:

- 1. Status command; reset character generator, dot mode, absolute coordinates (13ØØ, 1)

- 2. Move command; move to Ø, Ø (13ØØØØ, 17ØØØØ)

- 3. Draw command; draw to Ø, Ø (17ØØØØ, 17ØØØØ)

- 4. Status command; relative coordinates (6ØØ, 1)

- 5. Move command; move to Ø, Ø (13ØØØØ, 17ØØØØ)

- 6. Status command; relative coordinates (6ØØ, 1)

- 7. Draw command; draw to Ø, Ø (17ØØØØ, 17ØØØØ)

- 8. Status command; light pen sensitize, absolute coordinates, (4Ø2ØØ, 1)

d. Verify that the line generator multiplier state machine is in the reset state.

e. Reset and halt the line generator. Write the next two words of line generator command into the line generator passive input port.

f. Verify that the line generator multiplier state machine got to the first state of the expected sequence for the last line generator command written. If not, output the following message:

```
**5: MULTIPLIER STATE MACHINE ERROR**
  UNABLE TO SET TO STATE: XX RECEIVED STATE: XX
  LINE GENERATOR COMMANDS: XXXXXX, XXXXXX
```

g. Once the initial state of the expected sequence has been reached, single-step the line generator in maintenance mode, verifying each state in sequence until the reset state (state Ø) is reached. If an error occurs, output the following message:

```
**6:  MULTIPLIER STATE MACHINE ERROR**
  START STATE: XX EXPECTED STATE: XX RECEIVED STATE: XX
  LINE GENERATOR COMMANDS: XXXXXX, XXXXXX
```

A.4.2.4. Phase 4 Output State Machine Control ROM

The program steps for performing this test are:

a. Reset the picture system.

b. Reset and halt the line generator.

c. Repeat steps (d) through (g) 14 times, sending the following commands to the line generator:

- 1. Status command; reset character generator absolute coordinates (6ØØ, 1)

- 2. Move command; move to 377, 372 (13Ø377, 17Ø372)

- 3. Draw command; draw to Ø, Ø (17ØØØØ, 17ØØØØ)

- 4. Status command; absolute coordinates (6ØØ, 1)

- 5. Draw command; draw to Ø, Ø (17ØØØØ, 17ØØØØ)

- 6. Draw command; draw to Ø, 2 (17ØØØØ, 17ØØØ2)

- 7. Move command; move to 377, Ø (13Ø377, 17ØØØØ)

- 8. Status command; reset character generator, dot mode, absolute coordinates (13ØØ, 1)

- 9. Move command; move to Ø, 2 (13ØØØØ, 17ØØØ2)

- 1Ø. Draw command; draw to Ø, Ø (17ØØØØ, 17ØØØ2)

- 11. Draw command; draw to Ø, 2 (17ØØØØ, 17ØØØ2)

- 12. Draw command; draw to Ø, Ø (17ØØØØ, 17ØØØØ)

- 13. Staus command; light pen sensitize, absolute coordinate, segment name 1 (4Ø2ØØ, 1)

d. Verify that the line generator output state machine is in the reset state.

e. Reset and halt the line generator. Write the next two words of line generator command into the line generator passive input port.

f. Verify that the line generator output state machine got to the first state of the expected sequence for the last line generator command written. If not, output the following message:

```
**7: OUTPUT STATE MACHINE ERROR**
  UNABLE TO SET TO STATE: XX RECEIVED STATE: XX
  LINE GENERATOR COMMANDS: XXXXXX, XXXXXX
```

g. Once the initial state of the expected sequence has been reached, single-step the line generator in maintenance mode, verifying each state in sequence until the reset state (state Ø) is reached. If an error occurs, output the following message:

```
**1Ø: OUTPUT STATE MACHINE ERROR**
  START STATE: XX EXPECTED STATE: XX RECEIVED STATE: XX
  LINE GENERATOR COMMANDS: XXXXXX, XXXXXX
```

A.4.3. Error Analysis

Odd numbered errors (1, 3, 5, 7) indicate the state machine was unable to get to the first state of an expected sequence. Possible reasons for this failure are:

- A control ROM which is malfunctioning or not up to current revision level.

- A malfunctioning address line to or from the ROM.

- A malfunction in the PGXBUS or PGYBUS.

Even numbered errors (2, 4, 6, 1Ø) indicate the state machine did reach the initial state of an expected sequence, but thereafter failed to reach an expected state. If no odd numbered errors occur, an obsolete or malfunctioning ROM would be the most probable cause. Even numbered error messages take the following form:

```
N: MMMMMM STATE MACHINE ERROR
START STATE: xx EXPECTED STATE: yy RECEIVED STATE: zz
LINE GENERATOR COMMANDS: cccccc, cccccc
```

The initial state of the expected sequence, which was reached by the state machine, was "xx." The expected state, the state actually reached, and the data written into the line generator passive input port, are reported as previously indicated .

A.5. CHARACTER GENERATOR STATE MACHINE DIAGNOSTIC QSDØ37.SØ2

A.5.1. General Description

This diagnostic checks the operation of the character generator state machine (located on the 195223 card). There is only one phase and two possible errors. The program feeds character generator commands to the picture generator. After each command (four characters) has been written into the line generator FIFO, the line generator is single-stepped and the state of the character generator is compared against a table of expected states. Should there be any differences between what is expected and the actual state of the character generator an error is reported. The reported states are directly equivalent to the ROM addresses on the 195223 card.

NOTE

> This diagnostic assumes that diagnostics QSDØ34 and QSDØ36 have been run without any errors. This program may not be run while the remote terminal interface is in use.

A.5.2. Diagnostic Procedures

A.5.2.1. Phase 1 Character Generator State Macchine Control ROM

a. Reset the picture system

b. Do steps (c) through (i) five times with the following commands being sent to the picture generator:

- 1. 13426,774ØØ (start transfer =26, end transfer = 27, nul = Ø, del = 177)

- 2. 6Ø12,1Ø415 (line feed = 12, form feed = 14, carriage return = 15, DC1 = 21)

- 3. 1443Ø,16Ø32 (superscript = 3Ø, reset superscript = 31, subscript = 32, set italic = 34)

- 4. 5Ø44Ø,17Ø35 (space = 4Ø, Q = 121, reset italic = 35, cursor = 36)

- 5. 45Ø3,4Ø (C = 1Ø3, horizontal tab = 11, space = 4Ø, nul = Ø)

c. Reset (PGSR := $1Ø_8$) and halt (PGSR :=2) the picture generator.

d. Send the next command (from the previously mentioned list) to the passive input port of the picture generator.

e. Check to verify that the character generator is in reset state (state 5).

f. Issue single-step to picture generator (PGSR := 3).

g. Place picture generator in maintenance mode with the character generator state machine current address multiplexed onto the X-Bus (PGSR := 366).

h. Compare the present state of the character generator (character generator current address) with the expected state. If there is a difference, report an error.

i. If current state is reset state (state 5), go to step (c), otherwise go to step (f).

A.5.3. Errors

There are two errors which could occur during the running of this diagnostic; the character generator could fail to initialize to the correct state (fail to start processing the character command properly), or the character generator could fail to follow the correct sequence of states once a particular commmand had begun processing (not step to the correct state in sequence). These errors are reported as follows:

```
**1:  CHAR GEN STATE MACHINE ERROR**
  UNABLE TO SET TO STATE: XXX RECEIVED STATE: XXX
  CHARACTER GENERATOR COMMANDS: XXXXXX, XXXXXX
```

-or-

```
**2:  CHAR GEN STATE MACHINE ERROR**
  START STATE: XXX EXPECTED STATE: XXX RECEIVED STATE: XXX
  CHARACTER GENERATOR COMMANDS: XXXXXX, XXXXXX
```

A.5.4. Error Analysis

Error index = 1 may indicate any of the following:

o The ROM is out of date or malfunctioning (this would probably be the case if only a few states are in error).

o One or more of the address lines to the ROM are "hung" in either a high or a low state.

- One or more of the "next address" lines from the ROM are "hung" in either a high or a low static state.

- The X-Bus lines are "hung" in either a high or a low static state.

- The information sent to the character generator is incorrect. This could be checked by the FIFO diagnostic (QSDØ34) and the line generator state machine diagnostic, phases 1 & 2 (QSDØ36). QSDØ34 would be used to check the FIFO and QSDØ36, Phases 1 & 2 would be used to check the input state machine of the picture generator.

Error index = 2 may indicate any of the above failures. This error could also be caused by a 195223 card that is not at ECO level A3.

A.6. INTERACTIVE COLOR SHADOW DISPLAY TUNING PROGRAM QSDØ4Ø.SØ2

A.6.1. General Description

This is a test-pattern generation program to facilitate tuning of picture system color shadow mask displays. It consists of one phase. The standard diagnostic commands D, P, L, C, and S have no meaning in this program. The user is allowed to select any of the following patterns for display in autorefresh mode:

- 1 - Raster for brightness and line width adjustment
- 2 - Dot pattern for convergence
- 3 - Squares and text for end-point match
- 4 - Color bars
- 5 - Crosshatch
- 6 - Secondary colors

The user may also issue two other commands by keyboard input:

- Minus 1 - Disable descriptive messages
- Minus 2 - Reset the picture system and halt

A.6.1.1. User Options

The user may modify PSTB locations to control the following options. The interpretation of these options is the same as in RSDØ13 and RSDØ14.

- P1ØØ - Refresh rate (default 6Ø Hz)

- P1Ø1 - Line generator speed

- P11Ø through P113 - Full line generator status specification. See the functional description of RSDØ13 for details.

A.6.2. Diagnostic Procedures

A.6.2.1. Phase 1

a. Set the pattern number to 1. Go to step (d).

b. Get an octal number from the operator's terminal. If the number is minus 2, reset the picture system and stop. If the number is minus 1, set a flag to disable messages, and go to step (c). If the number is greater than zero, go to step (d).

c. This step is reached only if the operator input a minus 1, zero, or carriage return, which is the same as zero. Increment the pattern number. If the pattern number is greater than 6, set it equal to 1.

d. Reset the picture system. Initialize the refresh controller. If messages are enabled, print a description of the next test pattern.

e. Load picture system memory, starting at location zero, with the refresh data for the next test pattern. Start autorefresh, and go to step (b).

A.7. MULTI-USER REFRESH CONTROLLER INTERRUPT DIAGNOSTIC QSD117.SØ2

A.7.1. General Description

Successful execution of this diagnostic indicates that the multi-use refresh controller can interrupt the host computer. This diagnostic does not run under mapped RSX-11M.

This diagnostic consists of one phase:

1. Multi-User Refresh Controller Interrupt Test

A.7.2. Diagnostic Procedure

A.7.2.1. Phase 1 Interrupt Test

a. Reset the picture system.

b. Write a refresh table containing $32_{1Ø}$ task "pointers" equal to 177775, the diagnostic interrupt command.

c. Connect all 4 picture system interrupts. Set PSIE in the IOST register, and RFSTOP IE in the SYSIE register.

d. Repeat steps (e) through (g) 128 times.

e. Acknowledge the last refresh controller interrupt request by writing a "1" into RFSTOP REQ in the SYSREQ register.

f. Wait for a picture system interrupt. If no interrupt occurs within the timeout interval, disconnect interrupts, print the following message, and exit the phase:

```
PHASE 1 INTERRUPT ERROR
```

g. If any interrupt other than a system interrupt occurs, disconnect interrupts, print the following message, and exit the phase:

```
UNEXPECTED INTERRUPT ERROR; MASK =XX
```

h. Disconnect interrupts, and exit the phase.

A.7.3. Error Analysis

This test is very similar to phase 13_8 of QSD113, except that interrupts to the host cpu are actually enabled as described in step (c).

```
PHASE 1 INTERRUPT ERROR
```

The meaning of this error message is that an expected system interrupt (RFSTOP) did not occur.

If QSD1ØØ phase 1 and QSD113 Phase 13_8 run without error, there is probably a failure in the circuitry which enables the interrupt request to the system interrupt.

```
UNEXPECTED INTERRUPT ERROR; MASK = XX
```

Bits in the displayed mask value correspond to picture system interrupts as follows:

o Ø: real time clock interrupt
o 1: system interrupt
o 2: device interrup
o 3: DMA interrupt

The expected mask value is 2, corresponding to system interrupt only.

A.8. DYNAMIC MEMORY TEST USING MAP ACTIVE I/O FOR EXTENDED AND NON-EXTENDED MEMORY QSD120.S05

A.8.1. General Descriptiom

This picture system memory test is designed to detect dynamic failures which may not be detected by QSD002. It runs considerably faster than QSD002, and offers various user options, including: (a) user specification of memory test page range, (b) automatic determination of available memory, (c) user selection of DIO or DMA data transfer, and (d) user specification of data pattern and other options.

The test consists of three phases:

- Phase 1 - Address data
- Phase 2 - Zero/ones test (5 passes)
- Phase 3 - Random data (5 Passes)

A.8.2. Diagnostic Procedure

The following description pertains to all three phases, with the exception of information which is specific to each phase, as described in steps (r) through (t).

a. First time only: the diagnostic determines if the system is an extended or non-extended memory system. This is done by loading all the mapping registers with "0" and writing "177777" and "52525" into locations 0 and 1 respectively. Memory management is then enabled and a "0" is written into location 177001. If this occurs and extended memory system with memory management is enabled, location 177001 should map to physical location 1. Location 1 is checked to see if it has changed. If it has, the user will be told there is extended memory. If location 1 did not change, the user will be told there is non-extended memory.

b. If P101 is altered, announce the user specified memory range. Otherwise, determine the available memory, and announce the detected range. The available memory is determined as follows for non-extended memory: write "0" and "52525" into locations 0 and 1 (respectively) via DIO, then read and verify location 0. Write "40000" and "52525" into locations 40000 and 40001, and verify location 40000. Proceed in this fashion with locations 100000 and 140000. The first error determines the amount of available memory, always starting at 0 and extending to a maximum of 177377. For extended memory, the memory mapping registers are set to point to the first of each card. They are loaded with "0" for card 0, "200$_8$" (octal) for card 1 and so on until

"1600$_8$" for card 7. After the mapping registers are loaded, memory management is enabled and virtual memory location 0 and 1 are written with "177777" and "52525", respectively. They are then read back via DIO and verified.

The smallest increment that may be tested is 16K. The 16K page range will be reported to the user. Page 0 will be the start and page 3 is the maximum for non-extended and page 17$_8$ is maximum for extended memory. P100 and P101 can specify the beginning and ending 16K page, respectively.

c. First time only: if P77 equals "0", use DIO for block data transfers; otherwise, use DMA. Set a flag and announce whether DIO or DMA is being used

d. First time only: if P76 is non-zero, interrogate the user for options to specify random key, block increment, user data sequence, or whether or not error correction should be enabled on extended memory systems.

NOTE

If error correction is disabled the error LEDs will be on during the entire diagnostic.

e. First time each phase: reset the picture system and announce the phase.

f. Load the starting and ending address of of the 16K block to be tested.

g. Load the source data buffer with data for this phase and this pass, as described in steps (r) through (t).

h. Output the source data buffer to the first memory block (normally 0 through 776) via DIO or DMA.

i. If DMA is disabled, go to step (l).

j. Start the data from the first block into a destination buffer via DMA, no wait. There are two destination buffers, and input data is double-buffered to allow overlapping of input data transfer and verification, provided DMA is enabled.

k. Start the MAP transferring data from the first memory block to the second block.

l. Wait for MAP done, reset the MAP, and initialize the MAP to read data from block n via MAP active input, and write the identical data in block n+1 via MAP active output. Start the MAP.

m. Start the data from block n into a destination buffer via DIO or DMA. If DMA is enabled, do not wait for completion. If DMA is disabled, of course, the program is occupied controlling the DIO data transfer.

n. Compare the last block read in with the source data block. The verification subroutine is described in steps (u) through (aa).

o. If an error has occurred, go to step (h), interpreting "first memory block" as the block following the one in which an error was discovered.

p. If the last block has not been verified, increment n and go to step (l).

q. If there are more passes in this phase, go to step (e); otherwise, exit the phase.

r. The entire diagnostic can be run with the error correction disabled on an extended memory system. See the user option for more details. Important! If error correction is disabled, the error lights will come on and the test will check only the lower 16 bits on extended memory. Phase one consists of one pass using address data or a user selected data pattern. Address data is not actually the address of each location, but data intended to confirm that all address lines are functioning. The first location in the block contains Ø, and subsequent locations are incremented by $4Ø1_8$. The block size is 776_8 and the last location of the block contains "177375". The second block normally begins with 776_8 throught $1ØØØ_8$ loaded with "Ø", "4Ø1", "1ØØ2". The distance between blocks is controlled by the block increment parameter which must be greater than or equal to the block size, and is normally 776_8. The data is organized in this way so that, for example, locations Ø and 1ØØØ, which differ in address by one bit, will not contain the same data in either 8-bit slice. Alternatively, if the user specifies, for example, a data sequence of length 3 consisting of "52525", "125252", "177777", the sequence is replicated throughout the length of the source data block.

s. Phase two consists of zero/one data in 5 passes. The data sequences are of length 2 consisting of :

- Pass 1 - Ø, Ø
- Pass 2 - 125252, 52525
- Pass 3 - 177777, 177777
- Pass 4 - 52525, 125252
- Pass 5 - Ø, Ø

t. Phase three consists of 5 passes using new random data for each pass. Normally the random key is initialized to zero, but the user may specify some other initial value.

u. Steps (u) through (aa) describe the verification subroutine. As input data is double-buffered, so is verification. Select the current input buffer for verification, and toggle the selector.

v. Step through the source buffer and destination buffer, comparing expected and received values. If an error occurs, go to step (w). Exit the verification subroutine when done.

w. Set the error flag, and compute the picture system address corresponding the the erroneous data in the destination buffer. Read the picture system memory location again via DIO, and compare with the expected data in the source buffer. If the data matches set error index x = 1 and go to step (z).

x. Read the source location for the MAP active data transfer (e.g., if an error occurs in location 776 the data was transferred by the MAP from location Ø to location 776, so check location Ø). If this data agrees with the expected data, set index x = 2 and go to step (z) (this is the normal index for memory failures).

y. Set index = 3.

z. Print the following error message:

MEMORY ERROR; PSADR = nnnnnn EXPT = nnnn RECD = nnnn BLK INCREMENT = nnn INDEX = x

aa. Go to step (v) to continue checking the block.

A.8.3. User Options

a. If P77 is non-zero, DMA is enabled. Default P77 = Ø.

b. If P1Ø1 is minus 1, the program determines the amount of available memory. Note that a hard memory failure will cause incorrect determination of memory size. The user should verify the amount of available memory. If P1Ø1 is not minus 1, P1ØØ is the inclusive lower bound and P1Ø1 is the exclusive upper bound of the area of memory tested. Default P1Ø1 is minus 1.

c. If P76 is non-zero, the program will interrogate the user after the "X" command to allow specifying the random key for phase 3 ("R"), the block increment parameter for all phases ("I") the data sequence for phase 1 ("D"), or the error correction mode for extended memory systems ("M"). Default P76 is "Ø." The error-correction mode is enabled or disabled. A "Ø" after the "M" command will disable error correction and a "1" will enable error correction. Default will be error correction enabled. For more information, see program steps (r) through (t).

A.8.4. Error Analysis

There is one error message with three possible error indices:

o INDEX = 1 The data stored in memory was found to be correct using a single-word read via DIO. However, a previous block read via DIO or DMA incurred an error. This may indicate a transient memory output or picture system interface problem.

o INDEX = 2 This index indicates the most probable type of error, a normal memory failure. To relate address and bit to an IC on a 195143 card, see the prognosis for error number 3 in QSDØØ2.

o INDEX = 3 Data was stored in a memory location and verified once, and subsequently went bad. Again, see the prognosis for error number 3 in QSDØØ2.

A.9. PS2 EXTENDED MEMORY TEST PART I QSD14Ø.SØ1

A.9.1 General Description

This diagnostic is used for testing the Picture System Extended Memory. The amount of memory may vary from 32K to 256K words, and memory is broken up into 16K word blocks called segments. Memory is therefore specified by segment numbers which go from Ø to 17_8. Error looping is supported on all phases except phases 1 and 3. In these phases, if an error message can be printed more than once, the error looping routine in INIT is called to check for termination. The diagnostic consists of six phases:

1. Memory initialization test (run time: 25 sec.)
2. Memory address/data test (run time: 1Ø sec.)
3. Mapping registers test (run time: 1Ø sec.)
4. DIO/DMA mapping test (run time: 5 sec.)
5. Memory data bits test (run time: 4 min.)
6. Memory error correction bits test (run time: 4 min.)

NOTE

> All run times listed above are approximate under an RSX operating system and under normal load conditions.
>
> The single- and double-bit error status bits are cleared at the completion of each phase.

A.9.2. Start-Up Sequence

If no segment numbers have been specified (see option section), the program will automatically size memory and print the following message:

```
MEMORY AUTO SIZED
```

NOTE

> Hardware problems may cause invalid results during memory auto sizing.

The program will then print the following message which shows the start and end 16 K segment numbers as generated by either memory auto sizing or optional manual entry:

```
MEMORY SEGMENT RANGE = XX TO XX
```

A.9.3. Options

All options are specified by entries in the PSTB table.

- LOCATION 76 - extended memory registers base address. default is Ø for a base address of 1774ØØ
- LOCATION 77 - if Ø do not use DMA or interrupts (defaults). If minus 1 use DMA and interrupts
- LOCATION 1ØØ - start segment number (Ø-17)
- LOCATION 1Ø1 - end segment number (Ø-17)

NOTE

> If location 1ØØ is minus 1, memory auto sizing is enabled. The start and end segment numbers must be in the range of Ø to 17 (octal). Also the start segment number must be less than or equal to the end segment number.

If the above is not true, the following message will be printed:

```
INVALID SEGMENT NUMBER
```

A.9.4. Diagnostic Procedure

A.9.4.1. Phase 1 Memory Initialization Test

a. This test checks the memory initialization sequence. First the status register is read and checked to see if all the bits initialize correctly. If not, the following message is output:

```
1A:  STATUS REG INITIALIZATION ERR;  EXPT = XXXXXX, RECD =
     XXXXXX
```

b. Next a "1" is written into the INITMEM bit location of the status register and then the MEMREADY bit is checked to see if it went to "Ø". If it did not, the following message is output:

```
1B:  MEMREADY BIT DID NOT CLEAR
```

c. After 5 seconds the MEMREADY bit is checked to see if it is set again; if it is not the following message is output:

1C: MEMREADY BIT DID NOT RESET

d. Next the MMENBL bit is set in the status register and then checked. If it is not set the following message is output:

1D: MMENBL BIT DID NOT SET

e. All memory locations are then checked to see if they were initialized to "Ø." If not the following message is output:

1E: MEMORY DID NOT INITIALIZE CORRECTLY

f. Next, MMENBL bit is cleared and checked. If it did not clear, the following message is output:

1F: MMENBL BIT DID NOT CLEAR

g. Finally, location Ø is set to "Ø" and "177777$_8$ is written to location 177ØØØ. Location Ø is read and checked to see that it did not change. If it did the following message is output:

1G: PAGE 254 MAPPED TO PAGE Ø

A.9.4.2. Phase 2 Memory Address/Data Test

a. This test is an address and data check, with its main purpose to check the address registers of port A and port B. First a "1" is written into the port B disabled bit of the status register. The bit is verified to see if it was set; if not, the following message is output:

2A: B PORT DISABLE BIT CAN NOT BE SET

b. Next the address value is written into each memory location of the lower 16K words of memory, and then memory is read and checked. If an error occurs, the following message is output:

2B: A PORT ADDRESS ERR; SEGMENT ADDR = XXXXX, EXPT = XXXXXX, RECD = XXXXXXX

c. The B port disable bit is then cleared and checked. If the bit does not clear, the following message is output:

2C: B PORT DISABLE BIT CAN NOT BE CLEARED

d. This phase is repeated to check the B port while A port is disabled.

A.9.4.3. Phase 3 Memory Management Mapping Register Test

All mapping registers are first loaded with "Ø" and then read and checked. This test is repeated for values "1" through "7777_8".

If an error occurs the following message is output:

```
3:  MAPPING REG ERR;  REG ADDR = XXXX, EXPT = XXXX, RECD =
    XXXX
```

A.9.4.4. Phase 4 DIO/DMA Mapping Test

This test checks that the mapping registers for the DIO and DMA map correctly into memory. The MMENBL bit is set in the status register during this phase and cleared upon completion of the phase. Each 256-word page address is written into word Ø of each 256-word page using a DIO transfer with the DIO mapping register being set to each 256-word page number before the transfer. The 256-word page numbers are read and then checked using a DMA transfer. However, the DMA mapping register is only changed for each 16K block increment. If the DMA/interrupt mode is not enabled (see option section), the DIO mapping register is used instead of the DMA mapping register and the following message is output:

```
USING DIO INSTEAD OF DMA
```

If an error occurs, the following message is output:

```
4:  DIO/DMA MAPPING ERROR;  PAGE = XXXX, EXPT = XXXX, RECD =
    XXXX
```

A.9.4.5. Phase 5 Memory Data Bits Data Test

a. First, the diagnostic mode in the status register is set to "2" (read/write data bits with error correction disabled) along with the MMENBL bit, and then verified. If it will not set to "2", the following message is output:

```
5A:  CANNOT SET DIAGNOSTIC MODE TO 2
```

b. One write-then-read pass is done with the data as next described. The first location in the block contains "Ø" and subsequent locations are incremented by 4Ø1$_8$. The block size is 776_8, and the last location of the block contains 177375. The second block begins with 775 through 1ØØØ loaded with "Ø", "4Ø1", "1ØØ2". The data is organized in this way so that, for example, locations Ø and 1ØØØ, which differ in address by one bit, will not contain the same data in either 8-bit slice.

c. 4 more passes are done with zero/one data. The data sequences are of 2 word lengths consisting of:

- Pass 2 - Ø, Ø
- Pass 3 - 125252, 52525
- Pass 4 - 52525, 125252
- Pass 5 - 177777, 177777

d. The next 3 test passes use randכm data with new random data for each pass. If any errors כccur, the following message is printed:

```
5B:  MEMORY ERROR:  SEGMENT NUM = X, SEGMENT ADDR = XXXXXX,
     EXPT = XXXXXX, RECD = XXXXXX
```

e. The mode in the status register is set to "Ø" and verified. If the mode is not reset to zero, the following error message is output.

```
5C:  CANNOT RESET DIAGNOSTIC MODE TO ZERO
```

A.9.4.6. Phase 6 Memory Error Correction Bits Data Test

a. First, the mode in the status register is set to 3 (read/write error correction bits with error correction disabled) along with the MMENBL bit and then verified. If it will not set to 3, the following message is output:

```
6A:  CANNOT SET DIAGNOSTIC MODE TO 3
```

b. One write then read pass is done with the memory location receiving the lower 6 bits of a block address of size 776.

c. 4 more passes are done with zero/one data. The data sequences are of 2 word lengths consisting of:

- Pass 2 - Ø, Ø
- Pass 3 - 25, 52
- Pass 4 - 52, 25
- Pass 5 - 77, 77

d. The next 3 passes test the memory by using random data with new random data for each pass. If any errors occur, the following message is output:

```
6B:  MEMORY ERROR: SEGMENT NUM = X, SEGMENT ADDR = XXXXXX,
     EXPT = XXXXXX, RECD = XXXXXX
```

e. The mode in the status register is set to "Ø" and verified. If the mode is not reset to "Ø", the following message is output:

6C: CANNOT RESET DIAGNOSTIC MODE TO ZERO

A.9.5. Error Analysis

The following prognosis is made under the assumption that each phase is run in order.

ERROR # EXPLANATION

1A There is a problem in the status register or the logic which drives the register. If the double- or single-bit error latches will not clear before the start of this phase, bit 15 or 14, respectively will be on. Other phases will help to isolate other problems.

1B The MEMREADY bit in the status register did not clear. Check U43 on the 195442 card to verify that the INITMEM bit can be written. Check the memory initialization state machine for correct operation. Check U73 on the 195442 card. Also check the RAM decode logic U2Ø, U3Ø, U4Ø, and U7Ø on the 195443 card to see if the RDMEMSTA and WRTHMEMSTA signals are generated properly.

1C The MEMREADY bit in the status register did not reset. Check the logic listed in error message 1B.

1D The MMENBL bit in the status register cannot be set. Check U61 andd U72 on the 195442 card.

1E All the memory locations did not initialize to "Ø." Further phases will isolate the problems.

1F The MMENBL bit in the status register cannot be cleared. Check U61 and U72 on the 195442 card.

1G The virtual to physical address logic is at fault. Check U53, U73, and U74 on the 195443 card.

2A The indicated port-disable bit cannot be set in the status register. Check U61 and U62 on the 195442 card.

2B An error has occurred while using the specified port. The memory-timing generator or the specified port-state machine may be at fault.

Check the data paths on the 195444 card. Check the specified memory card. Check the addressing logic on the 195443 card.

2C The indicated port-disable bit cannot be cleared in the status register. Check U61 and U62 on the 195442 card.

3 The mapping registers can not be loaded correctly. Check the mapping register U10, U11, U13, U30, U31, and U33 on the 195441 card. Check the PSBUS interface logic U14, U24, U34, and U43 on the 195441 card. Also check U20 on the 195443 card to see if the GATERAMOUT and *WRTRAM signals are generated correctly.

4 The virtual to phsical addressing logic for the DIO and DMA is not working. Check the memory management RAM U10, U11, U13, U30, U31, and U33 on the 195441 card. Also check the active-channel encoder U44 on the 195441 card.

5A The diagnostic bits cannot be set in the status register. Check U61 and U72 on the 195442 card.

5B The memory card indicated is probably at fault. If a single bit is failing during only an odd or only during an even address, Table 1 will indicate the bad RAM chip on the 195445 card. However, if a single bit is failing during both odd and even addresses, check the A and B input register, and the output register on the 195445 card.

5C The diagnostic bits cannot be cleared in the status register. Check U61 and U72 on the 195442 card.

6A The diagnostic bits cannot be set in the status register. Check U61 and U72 on the 195442 card.

6B The memory card indicated is probably at fault. If a single bit is failing during only an odd or only during an even address, Table 1 will indicate the bad RAM chip on the 195445 Card. However, if a single bit is failing during both odd and even addresses, check the A and B input register, and the output register on the 195445 card.

NOTE

Bit locations Ø through 5 in the error message data values correspond to actual data bits 16 through 21, respectively.

6C The diagnostic bits cannot be cleared in the status register. Check U61 and U672 on the 195442 card.

Actual	Chip Location	
Bit	Even	Odd
Location	Address	Address
Ø	U28	U39
1	U5Ø	U61
2	U29	U4Ø
3	U51	U62
4	U3Ø	U41
5	U52	U63
6	U31	U42
7	U53	U64
8	U32	U43
9	U54	U65
1Ø	U33	U44
11	U55	U66
12	U34	U45
13	U56	U67
14	U35	U46
15	U57	U68
16	U36	U47
17	U58	U69
18	U37	U48
19	U59	U7Ø
2Ø	U38	U49
21	U6Ø	U71

TABLE 1. RAM Bit To IC Location Table

A.1Ø. PS2 Extended Memory Test Part II QSDØ141.SØ1

A.1Ø.1. General Description

This diagnostic is used for testing the Picture System Extended Memory. The amount of memory may vary from 32 K to 256 K words and memory is broken up into 16 K word blocks called segments. Memory is therefore specified by segment numbers which go from Ø to 17_8. In the case where the same error message can be printed more than once, the code calls the error looping routine in INIT for a termination check. The diagnostic consists of seven phases:

1. Error correction code generation test (run time: 2 min. 55 sec.)
2. Single-bit error detection and correction test (run time: 4Ø sec.)
3. Double-bit error detection test (run time: 5 sec.)
4. Memory test with error correction (run time: 3 min. 3Ø sec.)
5. MAP mapping test (run time: 5 sec)
6. Memory test with error correction using the map (run time: 3 min. 5 sec.)
7. Refresh controller mapping test (run time: 5 sec.)

NOTE

> All run times listed above are approximate under an RSX operating system and under normal load conditions.
>
> The single- and double-bit error status bits are cleared at the completion of each phase.

A.1Ø.2 Start up Sequence

If no segment numbers have been specified (see option section), the program will automatically size memory and print the following message:

```
MEMORY AUTO SIZED
```

NOTE

> Hardware problems may cause invalid results during memory auto sizing.

The program will then print the following message which shows the start and end segment numbers as generated by either memory auto sizing or optional manual entry:

```
MEMORY PAGE RANGE = XX TO XX
```

A.1Ø.3. Options

All options are specified by entries in the PSTB table.

- LOCATION 76 - extended memory registers base address default is Ø for a base address of 1774ØØ
- LOCATION 77 - if Ø do not use DMA or interrupts (default). If - 1 use DMA and interrupts
- LOCATION 1ØØ - start segment number (Ø-17)
- LOCATION 1Ø1 - end segment number (Ø-17)

NOTE

> If location 1ØØ is minus 1, memory auto sizing is enabled. The start and end segment numbers must be in the range of Ø to 17_8. Also the start segment number must be less than or equal to the end segment number.

If the above is not true, the following message will be printed:

```
INVALID SEGMENT ADDRESS
```

A.1Ø.4. Diagnostic Procedure

A.1Ø.4.1. Phase 1 Error Correction Code Generation Test

a. The test writes a data pattern into memory and reads the error-correction bits to see if they are correct. This phase uses $1ØØØ_8$ words of memory starting at location Ø. First a block of data patterns is written into memory with error correction enabled. Then the error-correction bits are read back with error correction off (diagnostic mode 3) and checked to see if they are correct.

b. All possible bit patterns are checked and if an error occurs the following message is output:

```
1:   ECC GENERATION ERROR; PATTERN SENT = XXXXXX, EXPT = XX,
     RECD = XX
```

A.1Ø.4.2. Phase 2 Single Bit- Error Detection and Correction Test

a. This phase writes a data pattern into memory which has a single-bit error in the data bits. It then reads the data and verifies that the error is detected and corrected. It also checks for double-bit errors. This phase uses $1ØØØ_8$ words of memory starting at location Ø. First a block of $1ØØØ_8$ data patterns is written into memory with error correction enabled. Next, another block of the same $1ØØØ_8$ data patterns, but with single-bit errors, is written into the memory data bits with error correction disabled (diagnostic mode 2). Therefore, the error-correction bits are not changed. Finally each word is read with error correction enabled.

b. If the single bit error is not detected from a test pattern, the following message is output:

```
2A:  SINGLE BIT ERROR NOT DETECTED; CORRECT PATTERN =
     XXXXXX, ERROR PATTERN = XXXXXX
```

c. If the test pattern was not corrected, the following message is output:

```
2B:  SINGLE BIT ERROR NOT CORRECTED; CORRECT PATTERN =
     XXXXXX, ERROR PATTERN = XXXXXX, RECD PATTERN = XXXXXX
```

d. If a double-bit error is detected, the following message is output:

```
2C:  UNEXPECTED DOUBLE BIT ERROR; CORRECT PATTERN = XXXXXX,
     ERROR PATTERN =XXXXXX
```

A.1Ø.4.3. Phase 3 Double Bit Error Detection Test

a. This phase writes a data pattern into memory which has a double-bit error in the data bits. It then reads the data and verifies that the error is detected. This phase uses $48Ø_{1Ø}$ words of memory starting at location Ø. First, a block of data is written into memory with error correction enabled. Next a block of the same data , but with double-bit errors is written into the memory data bits with error correction disabled (diagnostic mode 2). Therefore the error-correction bits are not changed. Finally each memory word is read with error correction enabled and the program verifies that a double-bit error was detected. If it was not, the following message is output:

```
3A:  DOUBLE BIT ERROR NOT DETECTED; CORRECT PATTERN =
     XXXXXX, ERROR PATTERN = XXXXXX
```

b. If a single-bit error is detected, the following message is output:

```
3B:  UNEXPECTED SINGLE BIT ERROR; CORRECT PATTERN = XXXXXX,
     ERROR PATTERN = XXXXXX
```

c. The DBINTENBL (double-bit error interrupt enable) bit is set in the memory status register and then checked to see if it was set. If not, the following error message is output:

```
3C:  CANNOT SET DBINTENBL BIT
```

d. If the DMA/interrupt option is enabled (see option section) the phase continues at step (e). Otherwise the phase skips to step (f), and the following message is output:

```
NOT CHECKING DOUBLE BIT SYSTEM INTERRUPT
```

e. A memory location which has previously been set to a double-bit error pattern is read and a check is made to see if a PS system interrupt occurred. If not, the following message is output:

```
3D:  DID NOT RECEIVE SYSTEM INTERRUPT
```

f. The DBINTENBL bit is then cleared and verified. If it did not clear, the following message is output:

```
3E:  CANNOT CLEAR DBINTENBL BIT
```

g. If the DMA/interrupt enable option is not set, the phase exits. Otherwise, the same memory location with a double-bit error is read, and a check for a PS system interrupt is made. If it did not occur, the phase exits. If it did occur, the following message is output:

```
3F:  RECEIVED UNEXPECTED SYSTEM INTERRUPT
```

A.1Ø.4.4. Phase 4 Memory Test with Error Correction

a. This phase does a check of memory with the error correction on. The MMENBL bit is set in the status register at the beginning of this phase and cleared upon completion. The following patterns are used:

   - Pass 1 - 2 word block of Ø, Ø

   - Pass 2 - 2 word block of 125252, 52525

   - Pass 3 - 2 word block of 52525, 125252

   - Pass 4 - 2 word block of 177777, 177777

o Passes 5 through 7 - random data with new random data for each pass

b. This phase writes all of memory for each test pattern in blocks of 1ØØØ$_8$ words.

c. Memory is then read in blocks of 1ØØØ$_8$ words. If neither a single- or double-bit error occurs, the phase skips to step (d). Otherwise the following message is output:

```
4A:  BLOCK READ ERROR; DBS = X, SBS =X, SEGMENT NUMBER = XX,
     SEGMENT BLOCK ADDR = XXXXXX TO XXXXXX
```

Then each memory location is read and checked for single- and double-bit errors and the received value is checked against the expected value. After a block is checked, step (d) is skipped and the next block read. If an error occurs, the following message is output:

```
4B:  MEMORY ERROR; DBS =X, SBS = X, SEGMENT NUM = XX SEGMENT
     ADDR = XXXXX, EXPT = XXXXXX, RECD = XXXXXX
```

DBS and SBS give the double and single-bit error status. If X is "1", an error occurred. If a single-bit error occurred, the received value will be a corrected value.

d. The block just read from memory is checked to see if all the values are correct. If not, the above message is output.

NOTE

The single- and double-bit error flags will be zero.

e. Blocks are read and checked until all of memory has been verified.

A.1Ø.4.5. Phase 5 MAP Mapping Test

a. This test checks that the mapping registers for the MAP input and output ports address correctly into memory. The MMENBL bit is set in the status register at the start of this test and then cleared upon completion. The MAP input is set to one 256-word page address and the MAP output is set to another page. The lower word in the page pointed to by the MAP output is set to zero, and the lower word in the page pointed to by the MAP input is set to 125252. The MAP then does a transfer from its input to output and the output word is checked. If an error occurs the following message is output:

```
5A:  MAP MAPPING ERROR; MAP INPUT PAGE = XXXX, MAP OUTPUT
     PAGE = XXXX
```

b. This procedure is repeated to check all available 256-word pages of memory.

c. If the MAP does not respond when first accessed, the following message is output and the diagnostic exits:

```
5B:  MAP TIMEOUT
```

A.1Ø.4.6. Phase 6 Memory Test with Error Correction Using the MAP

a. This test is a repeat of the data patterns used in phase 4; however, the MAP is used to transfer the blocks of data. The MMENBL bit is set in the status register at the beginning of this phase and cleared upon completion.

b. This phase writes all of memory in blocks of $4ØØ_8$ words. If a single- or double-bit error occurs during a MAP transfer, the following message is output:

```
6A:  MAP XFER ERROR; DBS = X, SBS = X
```

c. Memory is then read in blocks of $4ØØ_8$ words. If neither a single- or double-bit error occurs during this block read, the phase skips to step (d). Otherwise the following message is output:

```
6B:  BLOCK READ ERROR; DBS = X, SBS = X, SEGMENT NUMBER =
     XX, SEGMENT BLOCK ADDR = XXXXXX TO XXXXXX
```

Then each memory location is read and checked for single- and double-bit errors. Also, the received value is checked against the expected value. After a block is checked, step

(d) is skipped and the next block read. If an error occurs, the following message is output:

```
6C:  MEMORY ERROR; DBS = X, SBS = X, SEGMENT NUM = XX
     SEGMENT ADDR = XXXXX, EXPT = XXXXXX, RECD = XXXXXX
```

DBS and SBS give the double- and single-bit error status. If X is "1", an error occurred. If a single-bit error occurred, the received value will be corrected.

d. The block just read is checked to see if all the values are valid. If not the above message is output.

NOTE

The single- and double-bit eror flags will be zero.

e. All blocks are read and checked until all of memory is verified.

f. If the MAP does not respond when first accessed, the following message is output:

```
6D:  MAP TIMEOUT
```

A.1Ø.4.7. Phase 7 Refresh Controller Mapping Test

a. This phase sets the MMENBL bit in the status register during the test and clears it upon completion. The phase first verifies that when memory mapping is enabled, page $254_{1Ø}$ will map into page Ø. If it does not, the following message is output:

```
7A:  PAGE 254 DOES NOT MAP INTO PAGE Ø
```

b. A program which uses the refresh controller device commands is written into page $254_{1Ø}$. This program will output the current page number of the refresh controller mapping register into location $2ØØ_8$ of the page pointed to by this register. This location is then checked using the DIO. If there is an error, the following message is output:

```
7B:  REF CNTRL MAPPING ERROR; PAGE = XXXX
```

c. If the refresh controller does not respond the first time it is accessed, the following message is output and the diagnostic exits:

```
7C:  REF CNTRL TIME OUT
```

A.1Ø.5. Error Analysis

The following prognosis is made under the assumption that all phases in QSD14Ø have passed, and that all phases are run in order.

ERROR #	EXPLANATION
1	The error-code generator is not working. Check U42, U43, U52, U53, U72, and U73 on the 195444 card. Also the diagnosic multiplexer might be at fault. Check U54, U63, and U74 on the 195444 card.
2A	Error-detection logic is not working. Check U1Ø, U2Ø, U21, U4Ø, U41, and U51 on the 195444 card to see if the correct parity is generated. Check U6Ø, U61, and U7Ø on the 195444 card to see if the error type is decoded and latched. Also check the status register IC U62 on the 195442 card.
2B	The error-correction logic is at fault. Check U11, U12, U13, U3Ø, U31, U32, and U33 on the 195444 card.
2C	Error-detection logic is not working. Check the logic listed for error message 3A.
3A	Error-detection logic is not working. Check U1Ø, U2Ø, U21, U4Ø, U41, and U51 on the 195444 card to see if correct parity is generated. Check U5Ø, U6Ø, U61, U7Ø, and U71 to see if the type is decoded and latched correctly. Also check the status register IC U62 on the 195442 card.
3B	Error-detection logic is not working. Check the logic listed for error message 3A.
3C	Cannot set the DBINTENBL bit in the status register. Check U61 and U62 on the 195442 card.
3D	Cannot generate a PS system interrupt. Check U7Ø on the 195442 card. Also QSD1ØØ should be run to check the PS system interrupt.
3E	Cannot clear the DBINTENBL bit in the status register. Check U61 and U62 on the 195442 card.
3F	A PS system interrupt is generated even though the DBINTENBL bit is not set. Check U7Ø on the 195442 card.

4A An error occurred while reading a block of memory. Messages 4B will normally be printed at the same time. The block of memory which was read may be marginal.

4B The indicated memory location is marginal. Previous phases and QSD14Ø should be rerun to isolate the problem.

5A The virtual to physical addressing logic for the MAP is not working. Check the memory management RAM U1Ø, U11, U13, U3Ø, U31, and U33 on the 195441 card. Also check the active-channel encoder U44 on the 195441 card.

5B The MAP is not working. Run the MAP diagnostics to isolate the problem.

6A An error occurred while the MAP was transferring a block of data from the lower 4ØØ$_8$ words of memory to another location. Message 6C will normally be printed at the same time. The memory card at card address Ø may be marginal.

6B An error occurred while reading a block of memory. Message 6C will normally be printed at the same time. The specified memory block is probably marginal.

6C The specified memory location is marginal. Previous phases and QSD14Ø should be rerun to isolate the problem.

7A The virtual to physical addressing logic is at fault. Check U53, U73, and U74 on the 195443 Card.

7B The virtual to physical addressing logic for the refresh controller is not working. Check the memory management RAM U1Ø, U11, U13, U31, and U33 on the 195441 card. Also check the active-channel encode U4 on the 195441 card.

7C The refresh controller is not working. Run the refresh controller diagnostics to isolate the problem.

A.11. MPS FAST DMA DIAGNOSTIC QSD151.SØ1

A.11.1. General Description

Successful execution of this diagnostic indicates that the DMA portion of the CPU/Picture System interface is operational. It verifies the following:

a. All DMA modes of operation (active output, active input, and passive input).

b. Correct DMAPSA operation (excluding the validity of the "PSADD" lines on the PSBUS).

c. Correct DMA response to a busy PSBUS device.

g. Execution of simultaneous DIO and DMA block data transfers to and from picture system memory.

<u>NOTE</u>

> This diagnostic assumes successful execution of QSDØØØ. Phases 1 through 6 do not require picture system memory. Phases 7 & 1Ø$_{8}$ require the first 1ØØØ$_{1Ø}$ words of picture system memory and assume they can be written and read correctly, via the DIO.

This diagnostic is very similar to QSDØØ1 except that QSDØØ1 phases 7 through 12$_{8}$ are omitted. Phases 7 and 1Ø$_{8}$ of this diagnostic correspond to QSDØØ1 phases 13$_{8}$ and 14$_{8}$.

The diagnostic consists of the following eight phases (listed in octal):

- Phase 1 - DMAPSA Write/Read/PSRESET
- Phase 2 - DMAPSA Increment
- Phase 3 - Device BUSY/DMARESET/DMAPSA Non-Increment
- Phase 4 - DMA Active Output to DMAPSA
- Phase 5 - DMA Active Input from DMAPSA
- Phase 6 - DMA Passive dInput from DIO
- Phase 7 - DMA Block transfers OUT/IN
- Phase 1Ø$_{8}$ - Simultaneous DIO & Block Transfers

A.11.1.1. Abbreviated Data Sequence

QSDØØ1 uses the lengthy test data sequence consisting of Ø incremented through 177777. This program uses a much shorter data sequence which should normally be adequate for diagnostic purposes. The sequence is generated by a table and a complex algorithm, and will not be fully described in this writeup. The user may specify the long data sequence by modifying P1ØØ to be non-zero.

A.11.2. Diagnostic Procedures

A.11.2.1. Phase 1 DMAPSA

This phase verifies that all bits of the DMAPSA can be set, cleared, and reset. The program steps for performing this test are:

a. Check all bits with the test data sequence.

b. Check all bits with $1ØØØ_{1Ø}$ random data values.

c. If step (a) or (b) fails, the following message is displayed:

```
1A:  DMAPSA W/R ERROR; DATA EXP=XXXXXX DATA REC=XXXXXX
```

e. Set all bits, PSRESET, check for all zero.

f. If step (e) fails, the following message is displayed:

```
1B:  DMAPSA PSRESET ERROR; DATA EXP=Ø DATA REC=XXXXXX
```

g. Repeat step (e) several times.

h. Stop. This phase is complete.

A.11.2.2. Phase 2 DMAPSA Increment

This phase verifies that the DMAPSA wll increment after DMA active/input operations, providing it was not loaded with a passive port address prior to the operation. The program steps for performing this test are:

a. Initiate a single DMA active output transfer (DWAWC=177777; DMABA=address of a host memory location containing zero; set GO in the IOST).

b. Verify that the DMAPSA did increment.

c. If step (b) fails, the following message is displayed:

```
2:  DMAPSA INCREMENT ERROR; DATA EXP=XXXXXX DATA REC=XXXXXX
```

d. Repeat steps (a) and (b) with DMAPSA = Ø through 177776 (except for DMAPSA PSBUS address).

e. Stop. This phase is complete.

A.11.2.3. Phase 3 Device Busy/DMARESET/DMAPSA Non-Increment

This phase verifies that (1) the DMA will not access a busy passive port, (2) that DMARESET will release the DMA from a "hung condition", and (3) the DMAPSA will not increment after DMA active output/input operations when it contains a passive port address prior to the operation. The programs steps for performing this test are:

a. PSRESET, DMAPIP should be busy.

b. Initiate a single DMA active output transfer to the DMAPIP (DMAWC=177777; DMABA=address of a host memory location containing zero; set GO in the IOST).

c. Timeout, verify that DMAREADY is not set (the DMA should be "hung" since it accessed a busy passive port).

d. If step (c) fails, the following message is displayed:

```
3A:  ERROR; DMAREADY SET AFTER BUSY DMAPIP ACCESS
```

e. DMARESET and verify that within a timeout period DMAREADY is set.

f. If step (e) fails the following message is displayed:

```
3B:  ERROR; DMAREADY CLR AFTER DMARESET
```

g. Initiate a single DMA active output transfer (DMAWC=177777; DMABA=address of a host memory location containing zero; set GO in the IOST).

h. DMARESET in case the DMA is "hung".

i. Verify that the DMAPSA did not increment.

j. If step (i) fails, the following message is displayed:

```
3C:  DMAPSA NON-INCREMENT ERROR; DATA EXP=XXXXXX DATA
REC=XXXXXX
```

k. Repeat steps (g) through (i) several times with DMAPSA=17777Ø through 177777.

l. Stop. This phase is complete.

A.11.2.4. Phase 4 DMA Active Output to DMAPSA

This phase verifies the DMA active output mode of operation by doing single DMA transfers to the DMAPSA. The program steps for performing this test are:

a. Initiate a single DMA active output transfer to the DMAPSA (DMAWC=177777; DMABA=address of a host memory location containing zero; set GO in the IOST).

b. Check the DMAPSA for correct contents.

c. If step (c) fails, the following message is displayed:

   4: DMA ACTIVE OUTPUT TO DMAPSA ERROR; DATA EXP=XXXXXX DATA REC=XXXXXX

d. Repeat steps (a) and (b) several times with the test data sequence (even numbers).

e. Repeat steps (a) and (b) several times with random data (even numbers).

f. Stop. This phase is complete.

A.11.2.5. Phase 5 DMA Active Input From DMAPSA

This phase verifies the DMA active input mode of operation by doing a single DMA transfer of the contents of the DMAPSA to CPU memory. The program steps for performing this test are:

a. Clear a host memory location.

b. Load the DMAPSA with its PSBUS address.

c. Initiate a single DMA active input transfer to a host memory location. (DMAWC=177777; DMABA=host memory location address; set DMAIN and GO bits in the IOST).

d. Check the host memory location for DMAPSA PSBUS address.

e. If step (d) fails, the following message is displayed:

   5: DMA ACTIVE INPUT FROM DMAPSA ERROR: DATA EXP=XXXXXX DATA REC=XXXXXX.

f. Repeat steps (a), (b), (c), and (d) several times.

g. Stop. This phase is complete.

A.11.2.6. Phase 6 DMA Passive Input From DIO

This phase verifies the DMA passive input mode of operation by directing DIO data transfers to the DMA passive input port (DMAPIP). The program steps for performing this test are:

a. Initiate a single DMA passive input transfer (DMAWC=177777; DMABA-host memory location address; set PASSIVE, DMAIN, and GO bits in the IOST).

b. Do a DIO data transfer to the DMAPIP.

c. Check the host memory location for correct data.

d. If step (c) fails, the following message is displayed:

6: DMA PASSIVE INPUT FROM DIO ERROR; DATA EXP=XXXXXX DATA REC=XXXXXX

e. Repeat steps (a), (b), and (c) several times with random data.

f. Stop. This phase is complete.

A.11.2.7. Phase 7 DMA Block Transfers Out/In

This phase verifies that the DMA can transfer blocks of data to and from picture system memory. The program steps for performing this test are:

a. Initiate a $5ØØ_{1Ø}$ word DMA active output transfer to picture system memory locations Ø through 499 (DMAWC=177Ø14; DMAPA=host memory "data block out" starting address; set GO in the IOST).

b. When the output transfer completes, initiate a $5ØØ_{1Ø}$ word DMA active input transfer from picture system memory locations Ø through 499. (DMAWC=177Ø14; DMABA=host memory "data Block in" starting address; set DMAIN and GO bits in th IOST).

c. When the input transfer completes, verify that the block of data sent to picture system memory is identical to the block of data received from picture system memory.

d. If steps (c) fails, the following message is displayed:

NNN; 13: DMA BLOCK TRANSFER ERROR; DATA EXP=XXXXXX DATA REC=XXXXXX

Where: NNN = the octal data transfer number (O≤NNN≤763)

e. Repeat steps (a), (b), and (c) several times with blocks of random data.

f. Stop. This phase is complete.

A.11.3. Phase 10 Simultaneous DIO and DMA Block Transfers

This phase verifies that simultaneous DMA and DIO data transfers can be performed. The program steps for performing this test are:

NOTE

> Under a multi-user operating system, system loading and overhead may preclude concurrent DIO and DMA transfers.

a. Initiate a 500_{10} word DMA active output transfer to picture system memory locations 500_{10} through 999 (DMAWC=177014; DMABA=host memory "data block out" starting address; set GO in the IOST).

b. Without waiting for the DMA transfer to finish, initiate a 500_{10} word DIO transfer to picture system memory, locations 0 through 499.

c. When both DMA and DIO transfers are complete, initiate a 500_{10} word DMA active input transfer from picture system memory locations 0 through 499 (DMAWC=177014; DMABA=host memory "data block in" starting address; set DMAIN and GO bits in the IOST).

d. Without waiting for the DMA transfer to finish, initiate a 500_{10} dio transfer from picture system memory locations 500_{10} through 999.

e. When both DMA and DIO transfers are complete, verify that the 1000_{10} words sent to picture system memory are identical to the 1000_{10} words received from picture system memory.

f. If step (e) fails, the following message is displayed:

```
NNNN; 14:  DIO OR DMA BLOCK TRANSFER ERROR; DATA EXP=XXXXXX
DATA REC=XXXXXX
```

Where: NNNN = the octal data transfer number (0≤NNNN≤1747).

g. Repeat steps (a), (b), (c), (d), and (e) several times with blocks of random data.

h. Stop. This phase is complete.

A.11.4. Error Analysis

ERROR #	PHASE	EXPALNATION
1A	1	This error indicates that one or more bits of the DMAPSA register (located on the 195106-100 card) could not be either written or read correctly. The problem could be the data path to the register, register control logic, the register itself, or the data path from the register to the PSBUS.
1B	1	This error indicates that the DMAPSA register (located on the 195106-100 card) could not be reset to zero by PSRESET.
2	2	This error indicates that the DMAPSA register (located on the 195106-100 card) did not increment correctly after a DMA active output operation when it did not contain a passive port address prior to the operation. The problem may be the DMA active output control logic on the 195131-100 card or the logic which generates the *INCDMAPSA signal on the 195106-100 card.
3A	3	This error indicates that the DMREADY bit (located in the IOST register on the 195131-100 card), was set after the DMA tried to access the DMAPIP, which should have caused the DMA to "hang" (i.e., DMREADY should have been clear).
3B	3	This error indicates that a DMARESET would not release the "hung" condition the DMA was in due to trying to access a busy passive port (DMAPIP). DMARESET should have caused the DMAREADY bit (located in the IOST register on the 1915131-100 card) to be set.
3C	3	This error indicates that the DMAPSA register (located on the 195106-100 card) did not retain the passive port address it contained prior to a DMA active output operation. The signal *INCDMAPSA (195106-100 card) should not occur.

4	4	This error indicates that the DMAPSA register (located on the 1951Ø6-1ØØ card) did not contain correct data after a DMA active output transfer to the register. Bit zero should always be loaded with zero so the entire register will not be affected by the *INCDMAPSA signal at the end of the transfer. Bit zero is not checked as part of a correct result. The problem is likely to be the DMA data path to the register.
5	5	This error indicates that correct data was not received when the DMAPSA register (located on the 1951Ø6-1ØØ card) was used as the source of data during a DMA active input transfer. The data received should be equal to the DMAPSA PSBUS address. The problem may be the DMA active input control logic on the 195131-1ØØ card or the DMA data path from PSUB to CPU.
6	6	This error indicates that correct data was not received when DIO data transfers were directed to the DMA passive input port (DMAPIP). The problem may be the DMAPIP address response logic which generates AEPASADTI (1951Ø6-1ØØ card) control logic, and which responds to either AEPASDAT1, or else the DMA data path rom PSBUS to CPU.
13_8	7	This error indicates that there was a data error during a DMA block transfer, either to or from picture system memory. If many bits are in error, it is ususally an indication of a timing problem within the DMA control logic or the data path. Board swapping (1951Ø5-1ØØ, 1951Ø6-1ØØ, 195131-1ØØ) is the fastest way to solve the problem since it is very hard to loop on a 5ØØ word data transfer.
14_8	$1Ø_8$	This error indicates that there was a data error during simulataneous DIO or DMA data transfer to or from picture system memory. It usually indicates an arbitration or timing problem within DIO/DMA control logic or DIO/DMA data path usage. Board swapping (1951Ø5-1ØØ, 1951Ø6-1ØØ, 195131-1ØØ) is the fastest way to solve the problem if many bits are in error or if no failure pattern can be established.

A.12. RSDØ1Ø OR RSDØ11 MATRIX TRANSFORMATIONS AND PICTURE MEMORY TEST (PHASE 3 OR 4)

This test visually demonstrates the following Matrix Arithmetic Processor (MAP) functions: clipping, perspective, viewport mapping, depth cueing, translation, rotation, scaling, and hit testing.

This test also confirms that double buffered autorefresh and MAP active input/active output are functioning. Visual patterns may be created to check line-generator short-vector adjustment. Phases 3 and 4 display a two-dimensional grid composed of 2065_{10} short vectors. It also tests extended or non-extended memory under dynamic operating conditions.

The input device to control this test may be selected as follows:

- RSDØ1Ø - Phase 3 - Tablet
- RSDØ1Ø - Phase 4 - 16 function switches
- RSDØ11 - Phase 3 - Light pen
- RSDØ11 - Phase 4 - 32 function buttons

The available functions correspond to function switches or areas of the menu from left to right as follow:

(1) - Terminate (HALT)
(2-4) - Unused
(5) - Enable/disable hit testing (HIT TEST)
(6) - Enable/disable relocation in picture memory (MEM WALK)
(7) - Change the W scale factor (SCALE W)
(8) - Change the Z scale factor (SCALE Z)
(9) - Change the Y scale factor (SCALE X)
(1Ø) - Change the X scale factor (SCALE X)
(11) - Rotate about the Z axis (ROT Z)
(12) - Rotate about the Y axis (ROT Y)
(13) - Rotate about the X axis (ROT X)
(14) - Translate in Z (TRAN Z)
(15) - Translate in Y (TRAN Y)
(16) - Translate in X (TRAN X)

This program functions by program device polling (rather than refresh controller device polling and interrupts). It does not utilize DMA or interrupts.

A.13. RSDØ13.SØ3 COLOR DISPLAY TEST

A.13.1. GENERAL DESCRIPTION

This program generates visual test patterns which may be used to adjust any color display, but is primarily designed for the Evans & Sutherland Color Shadow Mask Display. The program consists of seven phases. The first six phases cause various test patterns to be displayed in autorefresh mode. The seven phases are described below.

- Phase 1 (brightness/line width) - the test pattern displayed in this phase is a white raster consisting of 512_{10} horizontal vectors.

- Phase 2 (convergence, dots) - this phase displays a pattern of white dots, 15_{10} columns by 14_{10} rows.

- Phase 3 (convergence, random lines) - this phase displays 200_8 random vectors, or the number of random vectors up to 500_8 specified by P77. These vectors are white.

- Phase 4 (phase match) - this phase places a large star along the display's diagonals. Each vector is drawn twice, one directly over the other.

- Phase 5 (end point match) - this phase displays a series of white concentric boxes with characters in them. The boxes are drawn from the center of the screen out. The boxes are depth-cued and are drawn counter clockwise. All eight character sizes are displayed.

- Phase 6 - this phase provides a color test of the three primary colors (red, green, and blue) and white. Each color consists of four different groups of eight vertical vectors. The vectors are drawn in each color with a decreasing amount of horizontal separation. All of the vectors are depth-cued.

- Phase 7 - Stop autorefresh

A.13.2. User Options

Following is a list of options which the user may specify by modifying locations in the Picture System Table (PSTB):

- P77 = number of random lines generated in phase 3 (default = 200 octal)
- P100 is the refresh rate, according to the following table:

P100	Refresh Rate
1	120 Hz
2	60 Hz
3	40 Hz
4	30 Hz
etc.	

The default refresh rate is 6Ø Hz for phases 2 through 6, 3Ø Hz for phase 1.

- P1Ø1 - is the line generator speed, according to the following list:

P1Ø1	LG Speed	LG Status Speed Bits
1	full	Ø
2	1/2	1
3	1/4	2
4	1/8	3

The default speed is 1/2 speed.

Although the values of three and four cause 2 and 3 respectively to be placed in the speed bits of the LG status command, there is no actual difference in the line generator speed for these two settings.

- P11Ø-P113 is the full line generator status specification. P1Ø1 allows the user to specify line generator speed only (bits 1 and Ø of status word 1). If more complete specification of line generator status commands is necessary, locations P11Ø through P113 may be modified. If any of these locations are modified, P1Ø1 should be left at its initial value of zero. (In all PSTB locations, a value of zero generally implies the default value.) P112 and P113 serve as a mask, while P11Ø and P111 contain data bits to be incorporated into all line generator status commands which are output to the picture system by their program. For example, P112 = 34 means that the user is overriding (only) bits 4 through 2 in LG status word 1. The contents of bits 4 through 2 in P11Ø would be the new bits used in all status commands. The following settings would specify yellow (hue = 6Ø) at full saturation. (See "PS2/MPS Hardware Reference Manual", Section 2.4.3.6.)

```
P11Ø = 3Ø        (Hue 543 = 6)
P111 = 7Ø        (Hue = Ø; Saturation = 7)
P112 = 34        (override Hue 543)
P113 = 77Ø       (override Hue 21Ø and Sauration)
```

Those bits not overriden assume the value they would normally be given by the test program. If P11Ø is non-zero, then P112 = Ø means the same as P112 = 177777, (i.e., P11Ø constitutes a specification of all 16 bits of LG status word 1). The same relationship applies to P111, P113, and LG status word 2.

A.13.3. Program Procedures

The steps performed by phases 1 through 6 are as follows:

a. If P11Ø through P113 are unmodified, go to step (b). Otherwise, set up line generator status information as specified by P11Ø through P113, and go to step (c).

b. If P1Ø1 has been modified, set up speed information.

c. If this is not the first pass of this phase, go to step (e).

d. Announce the phase, reset the picture system, and initialize the refresh rate according to P1ØØ.

e. Load data buffer with data for the appropriate phase and transfer the data to picture system memory via DIO. Incorporate any user-specified LG status bits.

g. Start autorefresh.

h. Suspend task for 1Ø seconds.

i. Return.

A.14. RSDØ14.SØ3 COLOR DISPLAY TEST, TOO -

A.14.1. General Description

This program is a visual test for any color display, the default being the color shadow mask display. The status words may be changed to operate on the appropriate display. P11Ø through P113 are used to override the default status words (see RSDØ13 for details). All phases use autorefresh. Each phase is described below:

o Phase 1 - color demonstration. This phase shows a series of "color patches." Each patch consists of four "@" signs. There are 512 color patches, representing each color and hue saturation. This phase is drawn at full speed.

o Phase 2 - convergence, dots. .This phase displays a dot pattern made up of $17_{1Ø}$ columns and $13_{1Ø}$ rows. This dot pattern corresponds to the aspect ratio of the display and is drawn at 1/2 speed.

o Phase 3 - convergence, crosshatch. This phase is a square, white grid consisting of $17_{1Ø}$ columns and $13_{1Ø}$ rows. This phase is drawn at 1/2 speed.

- Phase 4 - color lines. A series of horizontal lines is displayed during this phase. Each color and hue combination is represented by a line. This phase is drawn at full speed.

- Phase 5 - primary and secondary colors. The primary and secondary colors and white are demonstrated by this phase. Each consists of 32 horizontal vectors that are depth-cued. This phase is drawn at 1/2 speed.

- Phase 6 - stop autorefresh.

A.14.2. User Options

Following is a list of options which the user may specify by modifying locations in the Picture System Table (PSTB). For more detail, see the description of RSDØ13.

- P77 = refresh rate
- P1Ø1 = line generator speed
- P11Ø through P113 = full line generator status specification

A.14.3. Program Procedures

The steps performed by phases 1 through 5 are as follows:

a. If P11Ø through P113 are unmodified, go to step (b). Otherwise, set up line generator status information as specified by P11Ø through P113, and go to step (c).

b. If P1Ø1 has been modified, set up speed information.

c. If this is not the first pass of this phase, go to step (e).

d. Announce the phase, reset the picture system, and initialize the refresh rate according to P77.

e. Load data buffer with data for the appropriate phase, and transfer the data to picture system memory via DIO. Incorporate any user-specified LG status bits.

g. Start autorefresh.

h. Suspend task for 1Ø seconds.

i. Return.

APPENDIX B. MULTI PICTURE SYSTEM DIAGNOSTICS INSTALLATION UNDER RSX-11M

This appendix describes the procedure for installing Picture System 2 and Multi Picture System diagnostic tasks under mapped or unmapped RSX-11M.

B.1. GENERAL

The picture system diagnostics are distributed as task (TSK) files under UIC [2Ø5,15] for unmapped, and UIC [22Ø,14] for mapped RSX-11M. An unmapped system with diagnostics must be available to the Evans & Sutherland Field Service Engineer during initial installation of the picture system, and on subsequent occasions as required for maintenance.

The diagnostics available under mapped RSX-11M are a subset of those available for the unmapped system, as the mapped diagnostic task may neither initiate Direct Memory Access (DMA) nor enable picture system interrupts. However, many useful operations may be performed with the mapped diagnostics. They are provided to minimize the occasions when the customer's mapped system must be shut down to diagnose the picture system.

The available distribution media are DOS formatted 8ØØ bpi magnetic tape, along with RLØ1, RLØ2, and RKØ5 disk pack as RSX-11M volumes. Indirect command file [22Ø,14]DIAGN.CMD facilitates installation of the diagnostics.

B.2. STORAGE DEVICE AND UIC

The mapped and unmapped tasks require a total of either approximately 48ØØ blocks if transferred from an RSX-11M volume or 82ØØ blocks if copied from magtape. If the system disk does not have sufficient space for storage of the diagnostics, an auxiliary storage device must be selected. In the case of RKØ5 and RLØ1 disk packs, the distribution medium may be treated as the storage device; however, a back-up copy should be maintained.

To avoid name conflicts, the user must select two distinct User Identification Codes (UICs) for storage of the mapped and unmapped tasks. If an attempt is made to run a task under the wrong operating system (such as an unmapped task under the mapped system), the following error message will occur:

```
INS--FILE NOT TASK IMAGE
```

B.3. INSTALLATION OF UNMAPPED DIAGNOSTIC TASKS

The unmapped tasks are task-built to run in partition PAR14K on an unmapped baseline RSX-11M system. Installing them consists simply of using PIP or FLX, which copies them onto the storage device. The command file DIAG.CMD may be used to perform the copy operation, as explained in paragraphs B.4.1 through B.4.1.2. If only unmapped tasks are to be copied, it is equally convenient to perform the copy operation without benefit of DIAGN.CMD. Paragraphs B.3.1 and B.3.2 show how the files may be copied from magtape or disk pack without using DIAGN.CMD.

B.3.1. Copying Unmapped Diagnostic Tasks from Magtape

The following is a sample command sequence which will cause the diagnostic tasks to be read from magtape:

```
>LOA MT:  or >LOA MM:  (if necessary)
>SET /UIC=[uuu,uuu]
>UFD XXn:[uuu,uuu]  (if necessary)
>FLX or >RUN $FLX
FLX>XXn:[uuu,uuu]/RS/CO/BL:58.=MXn:[22Ø,15]*.*/DO
FLX>^Z
```

Where:	XXn: is the selected storage device.
	[uuu,uuu] is the selected UIC for unmapped tasks.
	MXn: is the device code for the drive containing the distribution magtape, typically MTØ: or MMØ:.

The unmapped picture system diagnostics are now ready to run. Operation of the diagnostics is explained in the "PS2/MPS Hardware Diagnostics Manual" (9Ø1171-Ø5Ø, latest version).

B.3.2 Copying Unmapped Diagnostic Tasks from Disk Pack

The following is a sample command sequence which will cause the diagnostic tasks to be read from disk pack:

```
>LOA DK:  or >LOA DL:  (if necessary)
```

```
   >ALL DDn:
   >MOU DDn:/OVR
   >SET /UIC=[uuu,uuu]
   >UFD XXn:[uuu,uuu]   (if necessary)
   >PIP or >RUN $PIP
   PIP>XXn:[uuu,uuu]=DDn:[22Ø,15]*.*
   PIP>^Z
```

Where: XXN: is the selected storage device.
[uuu,uuu] is the selected storage UIC for unmapped tasks.
DDn: is the device code for the drive containing the distribution disk pack, such as DKØ: or DL1:.

The unmapped diagnostic tasks are now ready to run.

B.4. PICTURE SYSTEM MAPPED DIAGNOSTIC TASKS

The picture system interface registers, which normally occupy UNIBUS addresses 76766Ø through 76767Ø, are accessed by the mapped diagnostics through a common block or device partition named PSDEVØ. Normally, PSDEVØ occupies the address range 7676ØØ through 767776. File DIAGN.CMD facilitates generation and installation of the common block and partition. The operation of DIAGN.CMD is described in paragraphs B.4.1 through B.4.1.3. Information pertinent to non-standard usage of PSDEVØ is contained in paragraph B.4.3.

Some mapped diagnostics, such as QSDØ2Ø, contain phases which attempt to initiate a DMA transfer, which is illegal under the mapped system. If such phases are mistakenly executed by the diagnostic user, the task will be terminated after issuing the following error message:

```
   DMA ERROR;  R5=XXXXX
```

B.4.1. Installation of Mapped and Unmapped Tasks and PSDEVØ by DIAGN.CMD

The following is a sample command sequence which will cause the DIAGN.CMD file to execute.

These operations can only be performed under a privileged account:

```
   >LOA DD:  or >LOA MX:  (if necessary)
   >LOA XX:  (if necessary)
   >UFD XXn:[uuu,uuu]  (if necessary)
   >UFD XXn:[mmm,mmm]  (if necessary)
   >SET /UIC=[mmm,mmm]
   >ASN XXn:=SY:  (if necessary)
```

For magtape:

```
   >FLX or >RUN $FLX
   FLX>/RS=MXn:[220,14]DIAGN.CMD/DO
```

For disk pack:

```
   >PIP or RUN $PIP
   PIP>=DDn:[220,14]DIAGN.CMD
```

```
Where: XXn: is the selected storage device.
       [mmm,mmm] is the selected storage UIC for mapped tasks.
       [uuu,uuu] is the selected UIC for unmapped tasks.
       MXn: is the magtape distribution.
       DDn: is the disk pack distribution.
```

The command file may now be started:

```
   @DIAGN
```

B.4.1.1. Specification of Devices and UICs

The command file will ask the user to specify the diagnostics storage device, the distribution device, and the mapped and unmapped storage UICs. A carriage-return, in response to either UIC query, will prevent the associated tasks from being copied.

B.4.1.2. Task Copying

The command file will proceed to run PIP or FLX and copy all necessary files in a manner similar to paragraphs B.3.1 and B.3.2.

B.4.1.3. Installation of PSDEVØ by DIAGN.CMD

Under mapped RSX-11M only, the command file will inquire whether the user wishes to install the PSDEVØ common block at this time. On an affirmative response, "Y", the command file will proceed to specify the partition. It will then assemble, task build, and install PSDEVØ. The command file will then terminate, and the mapped diagnostic tasks should now be operable.

B.4.2. Reinstallation of PSDEVØ

Where xxxx equals 7676 on a machine with 18-bit addresses or 177676 on a machine with 22-bit addresses, the following commands reinstall PSDEVØ:

```
>SET /MAIN=PSDEV0:xxxx:2:DEV
>INS XXn:[mmm,mmm]PSDEV0
```

These commands may be inserted into file SY:[1,2]STARTUP.CMD so that the partition will be automatically installed whenever the mapped system is booted, or VMR may be used to make the partition a permanent part of the system.

B.4.3. PSDEV0 Detailed Information

Before PSDEV0 is task built, the SET/MAIN command, as used in paragraph B.5.2, must be performed.

PSDEV0.MAC, a macro file, contains the following:

```
.TITLE PSDEV0
.IDENT/NC/
.PSECT PSDEV0,D,GBL,OVR
.BLKB 200
.END
```

PSDEV0.CMD, a TKB command file, contains the following:

```
PSDEV0/PI/-HD,,PSDEV0=PSDEV0
/
STACK=0
PAR=PSDEV0
//
```

The base address of the block of five picture system interface registers (normally 767660) as accessed by mapped diagnostic tasks, is derived from two sources of information. The SET/MAIN command contains the physical base address, divided by 1000_8 of the PSDEV0 partition. HSTB, a table which is linked with all diagnostics, contains location H1 (HSTB+2), which must equal octal 160000, plus the offset from the base of PSDEV0 to the first picture system interface register (normally H1=160060). The "PS2/MPS Hardware Diagnostics Manual" explains how H1 may be either patched using ZAP, or modified at runtime with the MODIFY command.

APPENDIX C. USE OF THE MULTI PICTURE SYSTEM HARDWARE DIAGNOSTICS UNDER THE VAX/VMS OPERATING SYSTEM

C.1. GENERAL

All diagnostics for the Multi Picture System are designed to run on-line under the VMS Operating System. Typically, all Multi Picture System harware diagnostics and maintenance may be performed without requiring the VAX to be powered down or timesharing activity to be curtailed.

Multi Picture System Diagnostics utilize diagnostic functions performed by the MPS device driver. Consequently, the MPS device driver must be properly installed before any MPS diagnostics may be run.

When running MPS diagnostics, all picture stations associated with the central graphics processor to be diagnosed must be unused (i.e., unattached and unallocated by any user). When an MPS diagnostic begins, it attempts to allocate the entire graphics processor rather than just a single picture station.

Thus, for example, if a user is running on MPA1: (or has it allocated or is attached to it), and MPS diagnostics are attempted for MPAØ: (different picture station, but same graphics processor), an attach failure will occur. This failure may be reduced by having the user deallocate and stop using all the picture stations of that graphics processor.

C.2. EXECUTION OF MPS DIAGNOSTICS

MPS diagnostics are executed by:

1. Logging into the area where the diagnostics reside.

2. Loading the MPS device driver (if it is not already loaded).

3. Performing a logical name assignment to direct the MPS diagnostics to the appropriate central graphics processor. The default assignment is to MPAO. If the second graphics processor is to be diagnosed, the ASSIGN command must be entered to cause the device reassignment. For example:

   ```
   $ ASSIGN MPBØ:  MPAØ:
   ```

If the device driver was generated and installed with a device name other than MP, then this device name should be used for the reassignment. For example, if the MPS device name were generated to be XX, then the following command would be required:

```
$ASSIGN XXAØ:  MPAØ:
```

4. To execute an MPS diagnostic simply enter "RUN filename". For example:

   ```
   $ RUN QSDØØØ
   ```

   The MPS diagnostics run in the same manner as under other diagnostic environments. However, many of diagnostics (notably memory and interface tests) run more slowly in the VMS Operating System environment.

5. To abort a diagnostic which is executing, simply enter Control-Y ("^Y") and STOP. This will return control to the terminal and another diagnostic or other program may then be run.

Additional information concerning operation of the diagnostics is contained in the "PS2/MPS Hardware Diagnostics Manual" (9Ø1171-Ø5Ø, latest revision).

APPENDIX D. PICTURE SYSTEM DIAGNOSTICS OPERATION ON THE PDP-11 UNDER RT-11 VERSION Ø3 or Ø4 OPERATING SYSTEM

D.1. GENERAL

RT-11 is a fast, relatively easy to use single-user operating system for the PDP-11 family of computers. Features of the system include low system overhead, fast I/O, and BATCH capability, allowing a sequence of diagnostics to run unattended by the operator.

From the picture sytem diagnostics user's point of view, the most important difference between RT-11 Version Ø4 or Ø3, and Versions Ø2, Ø1, or the E&S Diagnostic Monitor, is that operations which were formerly accomplished with the PIP or UPDATE utility programs are now accomplished with the DIRECTORY, COPY, and DUPLICATE commands. Also DKO:, formerly the name for the RKØ5 device, is the logical name for the default storage volume which is distinct from SY:, the system device. The proper name for the RKØ5 device is RK:.

Distribution media for picture system diagnostics include RKØ5 and RLØ2 disk packs, RXØ1 and RXØ2 diskettes, and SAV program files on magtape. Minimal hardware includes a PDP-11 with 56K bytes of memory and one of the disk drives mentioned above, or else 8ØØ bpi magtape with some other system device containing the RT-11 VØ4 or Ø3 Operating System.

The HELP command lists all monitor commands and, when followed by command name (e.g., HELP COPY), provides information about the specific command. Commands and options may be abbreviated down to the shortest unambiguous beginning portion of the string; for example, DIR for DIRECTORY. If a user is unsure of command syntax, he may type the command name only, omitting parameters. The monitor will, in that case, prompt the user for required information.

NOTE

> In the examples within this section, all user responses are underscored. All command lines are terminated with a carriage return.

D.2. INSTALLATION

For the available disk and diskette media, no installation is required. For magtape, copy the entire contents of the magtape onto the default storage device:

```
   .COPY/SYS MTO:*.* DK:
```

D.3. OPERATOR'S PROCEDURES

D.3.1. Startup

Boot the system device according to instructions obtained from either the customer or the on-site PDP-11 hardware documentation. STARTS.COM is an indirect file which the monitor will execute upon startup, if found. If the file is not found, the display:

```
   >KMON-F-Command file not found
```

error message may be disregarded. Always specify the date if new files are to be created. Also, the time must be specified if BATCH processing is to be used.

D.3.2. Date, Time

Specify the date:

```
   .DAT 15- SEP-8Ø
```

Specify the time:

```
   .TIM 13:1Ø
```

D.3.3. Task Initiation

The default extension for a program file is "SAV". Progam files other than current versions, will have extensions such as "SØ1". To initiate a task, use the RUN command:

```
   .RUN QSDØØ4
```

-or-

```
   .RUN QSDØØ4.SØ1
```

D.3.4. Task Termination

To interrupt an executing diagnostic, key-in CTRL-C ("C"). To restart, key-in START.

D.3.5. File Maintenance

D.3.5.1. File Specification

A file specification consists of physical or logical device name (ddd), file name (nnnnnn), and extension or type (eee). The specifications must be in the following form:

ddd:nnnnnn.eee

Physical Device Names:

o DLØ - RLØ1 or RLØ2 Drive Ø
o DL1 - RLØ1 or RLØ2 Drive 1
o DYn - RXØ2 Drive n
o DXn - RXØ1 Drive n
o RKn - RKØ5 Drive n
o MTn - TU1Ø Magtape n
o MMn - TU16 Magtape n

Logical Device Names:

o SY - system device, or the device which was booted and contains the operating system.

o DKn - default storage volume n.

If the extension is omitted, the system assumes the following defaults:

o SAV - loadable progam or task file.

o BAT - source file containing BATCH Commands.

o CTL - "control" file generated by the BATCH processor during interpretation of a BAT file.

o COM - indirect command file, which may be processed by the monitor as opposed to the BATCH processor.

o SYS - operating system and device handler files.

For example:

```
RK:QSDØØØ.SAV
```

The following "wildcard" operators may be used to specify a group of files which have some common feature:

* - Within the name or extension field, matches anything.
% - Within a name or extension, matches any letter.

Factoring may be used to propagate a device name, file name, or extension across multiple files. For example:

```
   DK:(A,B).C equals DK:A.C,DK:B.C
```

D.3.5.2. Directory

To obtain a directory, type DIR followed by a file specification. For example:

```
   .DIRECTORY SY:
```

-or-

```
   .DIR DK:*.SAV
```

-or-

```
   .DIR DK:*.(SAV,SØ%)
```

-or-

```
   .DIR DK:(QSD%%%,RSD%%%).*
```

D.3.5.3. File Copying

Contrary to earlier versions of RT-11, the input file specification is on the left, and the output file specification is on the right. For example:

```
   .COPY DYØ:*.* DL1:  (from RXØ2 dr Ø to RLØ1 dr 1, all files)
```

-or-

```
   .COPY DK:QSDDT.SAV SY:
```

-or-

```
   .COPY
   From?  DK:*.*
   To ? SY:
```

System files will not be copied unless the system switch is ulitized, e.g.,

```
   .COPY/SYSTEM RKØ:*.* DLØ
```

D.3.5.4. File Deletion

When the DELETE command is used, confirmation is required for each file to be deleted. In response to each query, type "Y" for yes to cause deletion of the file in question.

```
.DELETE QSD13%.*
Files Deleted:
QSD13Ø.SAV  ?  Y
QSD131.SAV  ?  Y
QSD136.SAV  ?  N
```

(Only the first two are deleted.)

D.3.5.5. File Renaming

In the following example, the SAV file is first renamed to SØ2, then the SØ1 file is renamed to SAV:

```
.RENAME QSD111.SAV QSD111.SØ2
.RENAME QDS111.SØ1 QSD111.SAV
```

D.3.6. Making Bootable Copies

D.3.6.1. Input And Output Devices Of The Same Type

For the following example, mount an RKØ5 disk pack containing the diagnostic system in drive Ø, and a scratch disk pack in drive 1. The copy will be a bootable system.

```
.R FORMAT

*RK1:
RK1:/FORMAT- Are you sure?Y
*^C
.INITIALIZE/BADBLOCKS RK1:
RK1:/Init- Are you sure?Y
.COPY/DEVICE RKØ: RK1:  (for RT-11 VØ4 - .COPY/DEVICE/VERIFY)
RK1:/Copy- Are you sure?Y
```

D.3.6.2. Input And Output Devices Of Differing Type

The following procedure is applicable under RT-11V4, but not under earlier versions of the operating systems. For the following example, mount an RLØ1 pack containing the diagnostics in drive Ø, and a scratch RKØ5 pack in drive Ø.

```
.DATE dd-mmm-yy
.R FORMAT
```

```
*RKØ:
RKØ:/FORMAT - Are you sure?Y
*^C
.INITALIZE/BADBLOCKS RKØ:
RK1:/Init - Are you sure?Y
.COPY/SYSTEM/NOLOG/DATE DLØ:*.* RKØ:
.COPY/BOOT RKØ:RT11SJ.SYS RKØ:
```

D.4. BATCH OPERATION

D.4.1. BATCH Startup

If your version of picture system diagnostics supports BATCH operation, the following commands will be in STARTS.COM in order to facilitate BATCH usage:

```
.ASS TT: LOG
.LOAD BA
```

The default extension for a BATCH command file is BAT. To initiate BATCH processing, run BATCH and key-in the name of the desired command file. For example:

```
.R BATCH
*PS2ØØ.BAT
```

A BATCH command file may be created or modified using the TECO.SAV text editor. The third line in the following example represents any desired message, which will be typed out by the BATCH processor. The TIM and 'FF' commands should follow the execution of each diagnostic. Multiple diagnostic commands may be entered on one line, as in MP1ØØ=2ED3P1ØX.

```
$JOB/RT11
        TTYIO
$MES xxxxxxxxxxxxxxxxxxxxxx
.TIM
.'FF'
.RUN QSDxxx
*S
.TIM
.'FF'

(etc.)

.RUN QSDxxx
*MP1ØØ=2ED3P1ØS
.TIM
.'FF'
$EOJ
```

D.4.2. BATCH Termination

To abort a BATCH job, key-in CTRL-C and CTRL-C again.

D.5. RELATED DOCUMENTS

The following are all publications of Digital Equipment Corporation (DEC), and are part of the RT11 VØ4 Software Documentation Package:

RT-11 Documentation Directory
Introduction to RT-11
RT-11 System User's Guide
RT-11 System Message Manual
PDP-11 TECO User's Guide (Text Editor)
RT-11 Pocket Guide
TECO Pocket Guide

APPENDIX E. DIAGNOSTICS OPERATION UNDER RSX-11M

A complete set of PS2/MPS Diagnostics is distributed under UIC [220,15] to run under unmapped RSX-11M. Also, a subset of the PS2 Diagnostics is distributed under UIC [220,14] to run under mapped RSX-11M to avoid, when possible, shutting down a Multi-User system while diagnosing the picture system. The diagnostics and phases of diagnostics which may run under mapped RSX are the ones that do not use DMA or interrupts.

The RSX Distribution of the PS2/MPS Diagnostics requires a minimum of 28K words of PDP-11 memory.

E.1. STARTUP SEQUENCE

When the RSX-11M system device has been bootstrapped, the following sequence should occur, with the operator supplying the time and date and UIC command as indicated by underlining:

```
RSX-11M V3.2 BLXX XXK
>RED XX0:=SYO:
>MOU XX0:XXXXXX
>@[1,2]STARTUP
>*PLEASE ENTER TIME AND DATE (HH:MM DD-MMM-YY) [S]: 13:20
5-DEC-80
>TIM
>13:20:02 05-DEC-80
>@<EOF>
>SET /UIC=[XXX,XXX]
>
```

This system is now prepared to run picture system diagnostics as requested by the operator.

E.2. INITIATION OF TASKS

Tasks are initiated by means of the RUN command, as in the following examples:

```
>RUN QSD001 or
>RUN QSD001.TSK
```

These two examples are equivalent; both examples request the latest version of the task (program) QSDØØ1. To run a task version other than the latest, use a command of the following form:

```
   >RUN QSDØØ1.SØ1
```

E.3. TERMINATION OF TASKS

When the diagnostic is awaiting operator input, control Z ("Z) causes a picture system reset followed by program termination.

Whether or not the diagnostic is awaiting operator input, control-C ("C") followed by "ABORT" causes termination without a picture system reset.

Users who are familiar with RT-11, but unfamiliar with RSX should be wary of entering control-C followed by a "RUN" command. The effect of this will not be to terminate the previously active diagnostic; rather, the attempt will be made to activate the latter diagnostic while the former is still active. This may result in the following error message:

```
   INS -- TASK NAME ALREADY IN USE
```

E.4. MODIFYING PICTURE SYSTEM DEVICE AND INTERRUPT ASSIGNMENTS

This section discusses the procedure for patching or permanently modifying the PS2/MPS Diagnostics device and interrupt assignments under RSX.

Patches are accomplished under RSX by running the "ZAP" utility. The first command after running ZAP names a task file to be modified, for example:

```
   >RUN $ZAP(CR)
   ZAP>QSDØØ1(CR)     (cue character produced by ZAP)
```

Locations in the task image are referenced by a segment number (always "2" in the case of the diagnostics) followed by a colon and the absolute address. The basic ZAP commands required are as follows, where "nnnnnn" designates an absolute address, and "dddddd" designates a 16-bit data value:

```
   2:nnnnnn/      (open and display absolute location nnnnnn)
   dddddd (CR)    (deposit dddddd in the open location)
   X(CR)          (exit; terminate)
```

Beginning locations of the host computer table (H) and the picture system table (P) are as follow:

	Unmapped	Mapped
P Table first entry (Pl):	2:41232	2:1252
H Table first entry (Hl):	2:41464	2:15Ø4

NOTE

> The addresses of these locations are subject to slight change. PØ always contains the address of HØ.

E.5. ARCH2 OPERATION

The following are non-standard features of ARCH2 operation under unmapped RSX-11M (PS2 only):

a. A maximum of 12 objects may be placed on the platform, as opposed to the standard maxmimum of 3Ø.

b. Error messages are not output by the unmapped RSX version of ARCH2. Such messages (e.g., "ERROR Ø DETECTED IN GRAPHICS SUBROUTINE Ø") would typically indicate that the PS2 interface is not functioning properly.

Termination of ARCH2: The system will crash when ARCH2 is aborted, unless the I/O Status Word (IOST) is cleared to disable picture system interrupts prior to abortion of the task, as in the following sequence:

```
>OPEN 16767Ø(CR)
xxxxxx Ø (Altmode)
>ABO(CR)
```

E.6. OPERATION OF MAPPED PICTURE SYSTEM DIAGNOSTIC TASKS

The mapped picture sytem diagnostic tasks communicate with the picture sytem interface registers directly rather than by means of a driver. This creates a possibility of conflicts between diagnostics and user software concurrently communicating with the picture system. To ensure that no other user is now attempting to communicate with the picture system and to prevent access by user software once the diagnostics have begun to execute, one should allocate the picture system, or all constituent work stations in the case of a multi-user refresh controller, by means of the ALL command. When no further use of the diagnostics is required, the picture system or work stations should be deallocated by means of the DEA command. Examples of allocation are as follows:

```
   >ALL PS:
```

-or-

```
   >ALL MP1
```

E.7. RSX REFERENCES

The following manuals by DEC ar϶ recommended as the first recourse for persons who wish to become more familiar with RSX, or who encounter problems which have not been dealt with in the foregoing discussion:

RSX-11M/RSX-11S Documentation Directory - a guide to the available documentation on RSX.

Introduction To RSX-11M - a conceptual introduction.

RSX-11M Operator's Procedures Manual - operational information including MCR (monitor) commands.

RSX-11M Utilities Procedures Manual - concerning the use of ZAP, PIP, and other utilities.