TEXAS INSTRUMENTS

Improving Man's Effectiveness Through Electronics

Model 990 Computer Diagnostics Handbook

MANUAL NO. 945400-9701
ORIGINAL ISSUE 1 JANUARY 1977
REVISED 1 AUGUST 1978
INCLUDES
CHANGE 1 . . . . . 15 JANUARY 1979

Digital Systems Division

LIST OF EFFECTIVE PAGES

INSERT LATEST CHANGED PAGES DESTROY SUPERSEDED PAGES

Note: The portion of the text affected by the changes is indicated by a vertical bar in the outer margins of the page.

Model 990 Computer Diagnostics Handbook (945400-9701)

Original Issue . . . . . 1 January 1977
Revised . . . . . 1 August 1978 (ECN 003451)
Change 1 . . . . . 15 January 1979 (ECN 009919)

Total number of pages in this publication is 350 consisting of the following:

PAGE NO.	CHANGE NO.	PAGE NO.	CHANGE NO.	PAGE NO.	CHANGE NO.
Cover	1	10-1 - 10-4	1	21-2 - 21-9	0
Effective Pages	1	11-1	1	21-10 - 21-14	1
iii - xxiv	1	11-2	0	22-1	1
1-1 - 1-2	1	12-1 - 12-8	1	22-2 - 22-4	0
2-1 - 2-2	0	13-1	1	22-5 - 22-8	1
3-1 - 3-2	1	13-2	0	23-1 - 23-4	1
3-3	0	14-1	1	24-1	1
3-4	1	14-2 - 14-4	0	24-2 - 24-6	0
3-5 - 3-6	0	15-1 - 15-2	1	25-1	1
3-7	1	15-3	0	25-2 - 25-4	0
3-8 - 3-11	0	15-4	1	26-1	1
3-12	1	15-5 - 15-8	1	26-2 - 26-6	0
3-13 - 3-16	0	16-1	1	27-1 - 27-2	1
3-17	1	16-2	0	27-3	0
3-18	0	17-1	1	27-4	1
4-1 - 4-8	0	17-2	0	27-5 - 27-7	0
5-1 - 5-2	0	17-3 - 17-4	1	27-8 - 27-10	1
6-1 - 6-2	0	18-1	1	28-1	1
7-1	1	18-2	0	28-2	0
7-2 - 7-4	0	19-1	1	29-1	1
8-1	1	19-2 - 19-8	0	29-2	0
8-2 - 8-4	0	19-9 - 19-12	1	29-3	1
9-1	1	20-1	1	29-4	0
9-2 - 9-4	0	20-2 - 20-6	0	30-1	1
9-5 - 9-6	1	21-1	1	30-2	0

LIST OF EFFECTIVE PAGES

INSERT LATEST CHANGED PAGES DESTROY SUPERSEDED PAGES

Note: The portion of the text affected by the changes is indicated by a vertical bar in the outer margins of the page.

Model 990 Computer Diagnostics Handbook (945400-9701)

PAGE NO.	CHANGE NO.
30-3	1
30-4	0
30-5 - 30-8	1
31-1	1
31-2	0
31-3 - 31-4	1
32-1 - 32-4	1
33-1 - 33-6	1
34-1	1
34-2	0
34-3	1
34-4 - 34-6	0
35-1	1
35-2 - 35-8	0
36-1	1
36-2 - 36-5	0
36-6	1
36-7 - 36-10	0
37-1 - 37-2	1
38-1	1
38-2 - 38-6	0
39-1 - 39-2	1
39-3 - 39-4	0
40-1 - 40-4	0
41-1 - 41-4	0
42-1 - 42-6	1
43-1 - 43-14	1
44-1 - 44-12	1
45-1 - 45-8	1
46-1 - 46-4	1
Appendix A Div	0
A-1 - A-8	0
Appendix B Div	0
B-1	1
B-2	0
B-3 - B-4	1
B-5 - B-6	0
B-7 - B-10	1
B-11 - B-12	0
Appendix C Div	0
C-1 - C-4	0
Appendix D Div	0
D-1	1
D-2 - D-4	0
Appendix E Div	0
E-1 - E-2	1
Alphabetical Index Div.	0
Index-1	0
Index-2 - Index-10	1
User's Response	1
Business Reply	1
Cover Blank	0
Cover	0

PREFACE

This manual describes the diagnostic programs for Texas Instruments 990 Computers and associated peripheral devices. The handbook is intended as a reference for systems analysts, field-service engineers, factory-service personnel, and other users of the diagnostics programs and is designed for use with the applicable maintenance manual.

The organization of the manual consists of 46 sections and 5 appendixes described briefly below:

I	Diagnostics Overview – Provides a general description and lists the available diagnostic programs.
II	Standalone Tests – Describes how to use the standalone diagnostics tests, including selection of options and the general I/O formats.
III	Diagnostic Operational Control System (DOCS) Tests – Describes how to use any of the DOCS tests and includes a description of the available general verbs and general I/O formats.
IV	Loading Procedures – Provides detailed instructions for the loading of the DOCS and any diagnostic from the supported load media.
V - XLVI	Individual Test Descriptions – Provides a detailed description of each of the diagnostics tests, including initialization instructions, particular options, unique verbs (DOCS tests only), system requirements, and interpretation of error messages. Refer to the Table of Contents for the list of individual tests.
A	Modifying Test Code – Details information on how to modify test object code in case the target system is nonstandard.
B	Maintenance Unit – Describes the use of the Texas Instruments 990 maintenance unit.
C	Programmer Panel – Describes how to use the 990 programmer panel.
D	Reading Linked Maps – Describes how to interpret the load map listing (LML) supplied with each test.
E	Part Numbers – Lists the part numbers associated with each diagnostic test.

The following manuals contain maintenance and troubleshooting information for the 990 computers and peripherals.

The following are field service manuals:

Title	Part Number
Model 990/10 Computer System Field Maintenance Manual	945402-9701
Model 990/4 Computer System Field Maintenance Manual	945401-9701
Model 990 Computer Peripheral Equipment Field Maintenance Manual	945419-9701

The following are maintenance and/or installation and operation manuals:

Title	Part Number
Model 990/10 Computer System Depot Maintenance Manual	945404-9701
Model 990/4 Computer System Depot Maintenance Manual	945403-9701
Model 990 Computer Family Maintenance Drawings, *Volume I – Processors* *Volume II – Peripherals*	 945421-9701 945421-9702
Silent 700® Electronic Data Terminal Models 732/733 ASR/KSR Maintenance Manual	960129-9701
Model 990 Computer Model 743 KSR Data Terminal Installation and Operation	943462-9701
Model 990 Computer DS990 System Installation and Operation Manual	946284-9701
Model 820 KSR Installation and Operation Manual	2250454-9701
Model 990 Computer Model 913A CRT Display Terminal Depot Maintenance Manual	945406-9701
Technical Manual, Model 306 Printer (plus Model 306C Addendum)	974993-9701
Technical Manual, Model 500 Printer (plus Model 588 Addendum)	974998-9701
Model 990 Computer Communications Interface Module Depot Maintenance Manual	945410-9701
Model 990 Computer Communications System Installation and Operation	945409-9701
Model 990 Computer Model FD800 Floppy Disc Controller Depot Maintenance Manual	945418-9701
Model 990 Computer PROM Programming Module Depot Maintenance	945405-9701

Title	Part Number
Model 990 Computer Models DS31/DS32 Disc Controller Depot Maintenance Manual	945414-9701
Maintenance Manual for Series 30 Disc Drive	961684-9701
Product Description Manual for Model 029 Power Supply (Disc Drive Power Supply)	961684-9702
Model 990 Computer 16 Input/16 Output EIA Data Module Depot Maintenance Manual	945415-9701
Model 990 Computer 16 Input/16 Output TTL Data Module Depot Maintenance Manual	945407-9701
Model 990 Computer TTY/EIA Terminal Interface Module Depot Maintenance Manual	945408-9701
Model 990 Computer Analog/Digital Converter Module Digital/Analog Converter Module Depot Maintenance Manual	944775-9701
Model 990 Computer Automatic Calling Unit Depot Maintenance Manual	946225-9701
Model 990 Computer Pulse/Tone Automatic Call Unit and External Automatic Call Unit Interface Installation and Operation	945425-9701
Model 990 Computer Model 911 Video Display Terminal Depot Maintenance Manual	945424-9701
Model 990 Computer TMS 9900/9980A Emulator and Buffer Modules Depot Maintenance Manual	946239-9701
Model 990 Computer AMPL Logic State Trace Data Module Depot Maintenance Manual	946242-9701
Model 990 Computer AMPL Emulator and Buffer Modules TMS 9900-1, SBP 9900A, TMS 9980A, and TMS 9981 Installation and Operation Manual	946278-9701
Model 990 Computer AMPL TMS 9940 Emulator and Buffer Modules Installation and Operation Manual	946279-9701
Model 990 Computer 32 Input/Transition Detection Module, 32-Bit Output Data Module, and Digital Input/Output Termination Panel Installation and Operation	946269-9701
Card Reader Operation and Maintenance Manual	974912-9701
Model 990 Computer 5MT/6MT Serial Interface Module Installation and Operation	946267-9701

Title	Part Number
990 Computer FD850 TILINE Floppy Disk System Installation and Operation	2261886-9701
990 Computer Terminal Interface Installation and Operation	946240-9701

The following manuals contain information about operation of the programmer panel:

Title	Part Number
Model 990/4 Computer System Hardware Reference Manual	945251-9701
Model 990/10 Computer System Hardware Reference Manual	945417-9701

The following manuals contain information about the operation of diagnostics loading (input) devices:

Title	Part Number
Model 990 Computer Model 733ASR/KSR Data Terminal Installation and Operation	945259-9701
Model 990 Computer Model 804 Card Reader Installation and Operation	945262-9701
Model 990 Computer Model DS31/32 Disc System Installation and Operation	945260-9701
Model 990 Computer Model FD800 Floppy Disc Installation and Operation	945253-9701
Model 990 Computer Model DS25/DS50 Disk Systems Installation and Operation	946231-9701
Model 990 Computer Model DS200 Disk System Installation and Operation	949615-9701

The following manuals contain information about alternative interactive or output devices:

Title	Part Number
Model 990 Computer Model 913 CRT Display Terminal Installation and Operation	943457-9701
Model 990 Computer Model 911 CRT Display Terminal Installation and Operation	945423-9701
Model 990 Computer Models 306 and 588 Line Printers Installation and Operation	945261-9701
Model 990 Computer Model 810 Printer Installation and Operation Manual	939460-9701

The following manual contains information about the link editor SDSLNK.

Title	Part Number
Model 990/10 Computer Program Development System Operation Guide	945256-9701

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

APPENDIXES

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF TABLES (Continued)

LIST OF TABLES (Continued)

SECTION I

DIAGNOSTICS OVERVIEW

1.1 INTRODUCTION

Texas Instruments offers extensive diagnostic programs for the 990 computers and associated peripherals. Since, even at the fastest known memory speed, an exhaustive test (one which tests each combination of bits) would require thousands of years of machine time, Texas Instruments has instead written tests to check data flow and control logic with bit patterns that most often generate errors. More exhaustive testing can only be done with special test equipment (such as TI's ATS 960). The diagnostic programs test every data path accessible to software and only the most critical data combinations to minimize run-time.

There are two general categories of tests, standalone tests and Diagnostic Operational Control System (DOCS) tests, depending on the type of hardware being tested. DOCS is essentially an operator-oriented test monitor used when a test requires operator interaction. Standalone tests have minimal I/O requirements and do extensive repetitive testing not suited for operator interaction. Table 1-1 lists the available diagnostic programs.

Brief operating procedures for most of the diagnostics are included in this manual. For those diagnostics that are not included, the appropriate "Program Description" document can be obtained which contains all necessary information.

1.2 EQUIPMENT REQUIRED

Each test has certain minimum hardware requirements in addition to the unit under test. In general, these requirements may be satisfied by any system that includes:

1. A 990 computer with sufficient memory. The 990/10 tests generally require less than 8K of memory. Most 990/4 tests require less than 4K memory.

2. A device to load the test into the 990 computer, such as a 733 ASR kit with cassettes, an 804 card reader kit, 990 floppy disk kit, any of the 990 hard disk kits, or the 990 maintenance unit, which includes a 733 ASR cassette drive.

3. A device to output error messages. All tests are able to output error messages to the 733 ASR and to the programmer panel. All DOCS tests are able to output error messages to the 306, 588, or 810 line printers. The 990 maintenance units includes a 990 programmer panel.

NOTE

Most tests are available in several versions. Be sure to select the version that corresponds with the hardware system under test. The programmer panel version of the DOCS tests does not support either the 733 ASR, 820 KSR, 913 VDT, or 911 VDT as a controlling device.

4. For interactive tests, the system must include either a programmer panel, 733 ASR/KSR terminal, or 990 maintenance unit for operator inputs. For DOCS tests, a 913 VDT, a 911 VDT, or an 820 KSR can be used.

Table 1-1. Diagnostic Tests

Test	Section	Type	Part Number	Description
ACUTST	31	DOCS	2250109	Automatic Call Unit Test
ADCHK	29	DOCS	2250110	A to D Converter Test
• AU10	7	Standalone	0997753	Arithmetic Unit Test – 990/10 only
• AU04	8	Standalone	0997754	Arithmetic Unit Test – 990/4 only
• AU04TEST	8	Standalone	0937983	Arithmetic Unit Test – 990/4 only
• CRCOMM	22	DOCS	2250100	Communications Interface Module Test
CRDRDR	41	DOCS	2250140	Model 804 Card Reader Test
CRT911	26	DOCS	2250101	911 CRT Controller and VDT Test
CRT913	14	DOCS	2250128	913 CRT Controller and VDT Test
CRUEXP	23	DOCS	2250111	CRU Expansion Chassis Interface Test
DACHK	28	DOCS	2250112	D to A Converter Test
• DDFLOP	42	DOCS	2250139	TILINE Floppy Disk Test
• DS10PD	34	DOCS	2250113	DS10 Disk Test
DSKM3X	15	DOCS	2250102	DS31/DS32 Disk Test
DSKTRI	21	DOCS	2250114	DS25, DS50, DS200 Disk Test
• EMU900	43	DOCS	2250229	TMS 9900/9980 Emulator and Buffer Test
• EMU940	44	DOCS	2250228	TMS 9940 Emulator and Buffer Test
EMUTST	36	DOCS	2250132	Emulator Board Test
EROMBT	25	DOCS	2250133	EPROM Memory Board Test
EXTACU	45	DOCS	2250158	External Autocall Unit Interface Test
FIVMOD	40	DOCS	2250134	5MT/6MT Interface Module Test
FLPDSK	27	DOCS	2250105	Floppy Disk Test
FLPTST	32	DOCS	2250117	2230/2260 Fast Line Printer Test
IO16	16	DOCS	2250121	16 I/O TTL Module Test
INPMOD	38	DOCS	2250120	32-Input/Transition Detection Module Test
LOCLIN	30	DOCS	2250122	Local Multidrop Module Test
• LP810	33	DOCS	2250106	810 Line Printer Test
• LPTEST	13	DOCS	2250132	990 Line Printer Test
• MAPTST	20	Standalone	0937757	Map Logic Test – 990/10 only
MEMPRT	11	DOCS	2250124	Memory Protect Logic Test – 990/4 only
OUTMOD	39	DOCS	2250125	32-Output Data Module Test
• PROMPG	18	DOCS	2250118	PROM Programmer Test
RAM04	10	DOCS	2250107	Random Access Memory Test – 990/4 only
• RAM10	9	DOCS	2250119	Random Access Memory Test – 990/10 only
RMTEIA	46	DOCS	2250242	TTY/EIA Remote Terminal Test
RMTFLP	37	DOCS	2250108	Remote Diskette System Loopback Test
• TAPTST	19	DOCS	2250126	979A Transport and Interface Test
TILCOU	24	DOCS	2250129	TILINE Coupler Logic Test
TRACE	35	DOCS	2250130	Trace Module Test
• TST733	12	DOCS	2250250	733 ASR/KSR Data Terminal Test
• TTYEIA	17	DOCS	2250131	Full Duplex TTY/EIA Interface Module Test

SECTION II

STANDALONE TESTS

2.1 STANDALONE TESTS

Standalone tests do not run under control of the DOCS (although they can be loaded by the DOCS and can return control to the DOCS after execution). These tests output messages to a 733 ASR or KSR terminal, a 913 or 911 Video Terminal and to a programmer panel.

2.2 OPERATION

Standalone tests require no operator inputs. When loaded into the computer standalone tests immediately start execution. When the system includes an output device, output messages are generated in the following form:

TEST NAME VERSION NUMBER

HEADER MESSAGES

(If any errors occur)
ERROR MESSAGES

COMPLETION MESSAGE

LOOP COUNT

At the end of the test, the loop count is output to the programmer panel and the test starts over. If the test was loaded by the DOCS, control is returned to the DOCS after one complete execution of the test.

2.3 STANDALONE OPTIONS

The standalone options available to the operator are either idle on errors, print headers, suppress header printout, print errors, or suppress error printout.

If the idle on errors option is selected and an error occurs, the computer will enter the idle mode after displaying the error on the programmer panel or outputting the error to the output device.

The suppress-header-printout option prevents the output of test and subtest header messages.

The suppress-error-printout option prevents the printout of error messages. Error numbers are still output to the programmer panel; however, unless idle on errors is also selected, the error may be displayed on the programmer panel too briefly for visual observation.

These options can be selected by either patching the object code before loading the test into the 990 (see Appendix A) or by performing the following procedure.

After the test is loaded and starts executing:

1. Halt the computer by pushing the HALT/SIE switch on the programmer front panel.

2. Enter the address of the flag to be modified on the data switches. The address of the flag is the address shown in table 2-1 plus the load point of the test (see Appendix E). Usually the load point will be $A0_{16}$. If loaded by the DOCS, the load point is whatever was specified.

3. Push the MA switch.
4. Push the MDD switch. The contents of the location selected are now displayed.
5. Enter the new value on the data switches.
6. Push the MDE switch.
7. Push the CLR switch
8. Push the ST (ENTER) switch
9. Enter the load bias in the data switches (standard is $A0_{16}$)
10. Push the MA (ENTER) switch
11. Push the MDD switch
12. Push the PC (ENTER) switch
13. Push the RUN switch

The addresses of the three flags are given in table 2-1. The load point of the test must be added to find the memory address.

2.4 RUNNING STANDALONE TESTS USING THE PROGRAMMER PANEL

The standalone tests can be run using only the programmer panel by setting the error print and the header print flags to zero. In this case, the only available option is halt on errors. The only messages output are the loop count and, if an error is found, the error message number. These error messages will be output to the programmer panel.

Table 2-1. Flag Addresses

Relative Address	Flag	Comments
$001E_{16}$	Error Print Flag	Set this flag to zero to suppress error printout
0020_{16}	Header Print Flag	Set this flag to zero to suppress header printout
0032_{16}	Idle on Errors Flag	Set this flag to any nonzero value to cause the compiler to IDLE whenever an error is found

SECTION III

DIAGNOSTIC OPERATIONAL CONTROL SYSTEM (DOCS)

3.1 INTRODUCTION

The DOCS is a small operating system which is used to load and execute 990 diagnostic test modules. As with most operating systems, the DOCS plays the role of support and control for the diagnostic test modules which do the real testing. The DOCS is a command or verb oriented system. This means that the operator tells the DOCS what to do by typing verbs on the interactive device. The verbs are generally two or three characters long and each verb instructs the DOCS to carry out some function. Each diagnostic test module supports a set of verbs related to the piece of equipment the test was designed to evaluate. The DOCS also supports a set of general verbs which are designed to aid the operator in diagnosing problems. The entire set of verbs is called the Diagnostic Control Language (DCL).

The DOCS performs the following major functions:

1. Determines the type of computer on which the DOCS is loaded.
2. Initializes all interrupt and XOP trap locations.
3. Determines the interrupt level of the real time clock.
4. Determines the location and type of interactive device on which the operator is responding.
5. Initializes all system parameters.
6. Loads diagnostic test modules into memory.
7. Optionally passes control to the test module initialization routine (IT verb).
8. Optionally executes all standard parts of the test module (EA verb).
9. Controls the verb execution mechanism.
10. Processes all test module service calls.
11. Processes all DOCS supported verbs.
12. Controls all I/O functions between the test module and the operator.

Three versions of the DOCS are available: the Standard DOCS, the Mini DOCS, and the Front Panel DOCS. The following paragraphs describe the different versions of the DOCS and how the operator interfaces with them.

3.2 STANDARD DOCS

The Standard DOCS is the primary version of the Diagnostic Operational Control System. It has the ability to run on the 990 family of computers and can interface with the operator on most of the

of the standard interactive devices. Both the Mini DOCS and the Front Panel DOCS can be considered subsets of the Standard DOCS.

3.2.1 **SUPPORTED COMPUTER MODELS.** The Standard DOCS will run on all the 990 computers. Currently, these include the 990/4, 990/5, 990/10, and 990/12. Code exists in the DOCS to determine the computer model on which the DOCS is running since interrupt handling and some system parameters depend upon this information.

3.2.2 **SUPPORTED I/O DEVICES.** The supported I/O devices include those peripherals that the operator can use to input data to the DOCS and those which the DOCS can use to output data. The term "interactive device" refers to those terminals that can be used for both inputting data from, and outputting data to the operator. The Standard DOCS currently supports the 733 ASR/KSR, the 820 KSR, the 913 VDT, and the 911 VDT as interactive devices. The Standard DOCS supports the 733 ASR/KSR, 820 KSR, 913 VDT, 911 VDT, and any of the line printers on the TTY/EIA interface as error message devices. The printers include the 306, 588, and 810. The phrase "error message device" refers to those peripherals that optionally can be used to output all error messages to the operator.

The DOCS does not depend on a fixed configuration for any of the I/O devices. The interactive device can be located at any CRU location within the main chassis or expansion chassis. When started, the DOCS code determines the device type and the location at which the operator is responding. Once this is established, the DOCS will ask the operator for any additional information required to handle the interactive and error message devices.

3.2.3 **SUPPORTED LOADING MEDIA.** The Standard DOCS is packaged on a number of load media. Both the Standard DOCS and the diagnostic test module are currently available on cards, cassette, hard disk, and floppy diskette. The DOCS must be loaded into memory first before any of the test modules can be loaded. The procedures that the DOCS uses to load the test modules are given in paragraph 3.2.9.13.

3.2.4 **OPERATOR INTERFACE.** The following paragraphs discuss the operator conventions, the start-up and initialization sequences, and the interrupt handling procedures.

3.2.4.1 **Operator Conventions.** The conventions the operator must use to interface with the Standard DOCS are listed below. These apply to any of the supported interactive terminals.

1. A hyphen (-) is printed when the DOCS is waiting for a response from the operator.

2. All inputs are ended by pressing the RETURN or NEW LINE key.

3. Most requests for operator input are designed with a predefined default value. In some of these requests the default value is printed out. If the operator presses the RETURN or NEW LINE key before entering any other data, the DOCS uses the default value. If instead, the operator enters a new value followed by RETURN or NEW LINE, the DOCS will use this new data and in most cases replace the default with the new value. The next time this particular request is issued, the default will be the data entered the previous time.

4. The RUB OUT key on the 733 ASR/KSR or the left cursor control (←) on the 913 or 911 VDT can be used to void the current operator input and allow reentry of the data. These keys are only effective before the operator has pressed the RETURN or NEW LINE key.

5. All numbers are input and output in hexadecimal. Depending on the type of request, the operator can enter a contiguous string of up to 64 hexadecimal digits. The string of digits should have no embedded blanks and leading zeros are not required. For example, if a memory test requests the ending memory location to be tested and the address is $1FFFF_{16}$, the operator responds as shown in the example below:

   ENDING MEMORY ADDRESS? (DEF=3B3E) – 1FFFF

   For an ending address of 730_{16}, the response is:

   ENDING MEMORY ADDRESS? (DEF=3B3E) – 730

6. Whenever a number is entered, it is checked to make sure it is a legal hexadecimal value. If the number is not a legal hexadecimal value, a message is printed and the operator can reenter the number.

7. All YES/NO questions are answered using the convention 1=YES and 0=NO. Any value other than 0 or 1 will cause an error message. The operator can then reenter the answer.

8. All ASCII data is entered as one contingious string of up to 64 characters.

9. Whenever the DOCS is ready to have a verb mnemonic entered by the operator, it will print:

   VERB ? –

   Any legal verb can be entered. If the operator enters an illegal verb (not in the list of available verbs), an error message is printed and the operator can then enter the correct verb.

10. If the operator presses the at (@) key, the 911 VDT CMD key, or the 913 VDT HELP key, the DOCS will terminate any current activity and return to the question VERB? –.

11. Whenever messages are being output, they can be stopped by pressing any key on the interactive device; the DOCS will return to the question VERB ? –.

12. All DOCS supported verbs must be preceded by a period (e.g., .IS).

3.2.5 START-UP. The Standard DOCS start-up is the process the DOCS goes through from the time it is loaded until it starts asking initialization questions. During this time the DOCS determines the interactive device which the operator wishes to use by scanning CRU base addresses from hexadecimal 0000 through 1F00 for the character 'Y'. The operator should allow sufficient time for loading of the DOCS before entering a Y. When an operator enters Y from a 733 ASR/KSR or a 911/913 VDT located at any CRU base address from 0000 through 1F00, the DOCS recognizes the address and type of interactive device responding, and communicates with it accordingly. This means that the interactive device is not restricted to a predesignated CRU base address, but may be located at any CRU base address within the specified range. Since the DOCS repetitively scans the CRU address range, no time limit exists for operator response.

CRU base address, but may be located at any CRU base address within the specified range. Since the DOCS repetitively scans the CRU address range, no time limit exists for operator response.

3.2.6 **INITIALIZATION.** During initialization, the Standard DOCS asks the operator to input several pieces of information: The interrupt level of the interactive device (the device on which the operator typed Y); the type of device, the CRU base address, and the interrupt level of the error message device (if different than the interactive device); and the system control options. Table 3-1 shows an example of the initialization questions and figure 3-1 presents a flow chart of the start-up, initialization, test module loading, and test module execution procedures.

When the DOCS prints out the request for the interactive device interrupt level, it will also print out a default value. If the default is correct, the operator presses the RETURN or NEW LINE key. Otherwise, the operator enters the correct value.

The next question asked by the Standard DOCS is the type of device selected for error messages (either 733 ASR/KSR, 913 VDT, 911 VDT, or line printer). The printed default value will be that of the interactive device. If the operator responds by pressing the RETURN or NEW LINE key, the interactive device will also serve as the error message device and no further questions are asked by the Standard DOCS. If the operator responds with a different device type, then the Standard DOCS will ask for the CRU base address and interrupt level.

Four system control options are available. The first option (E) controls whether or not error messages are printed (see table 3-1). The second option (H) controls the printing of header messages. The third (N) controls whether or not error message numbers are printed, and the fourth (I) controls the idling of the computer if an error occurs. When the Standard DOCS outputs the request for control options, the operator should respond by taking the default or by typing a string of one or more single character options. Each option should be separated from the next by a comma and no imbedded blanks should be included. The control options can be entered in any order and the inclusion of an option means that the DOCS will print messages or idle on errors. The options can be changed at any time by using the Initialize Control Options (.IP) verb.

The operator can exit the DOCS initialization phase at any time by pressing the @ key. However, the DOCS will output the following warning message:

***** WARNING *****

INITIALIZATION NOT COMPLETE-----INTERRUPT & XOP TRAPS HAVE BEEN RESET

Table 3-1. Standard DOCS Initialization Example

```
♦♦♦ DOCS      VERSION # = 01/79 ♦C       TI 990/04 ♦♦♦
♦♦♦ USE "RETURN" OR "NEW LINE" TO TERMINATE INPUT ♦♦♦
♦♦♦ USE "<--"(LEFT ARROW) OR "RUB OUT" TO ERASE INPUT ♦♦♦

733 INT LEVEL? DEF= 06   -
ERROR MSG DEVICE(0=733 1=913 2=911 3=LP)? DEF= 0  -

ENTER CONTROL OPTIONS SEPARATED BY COMMAS:
(E =ERR MSG'S, H =HDR MSG'S, N =ERR #'S, I =IDLE ON ERR'S)
(DEF= E,H,N   )  -
```

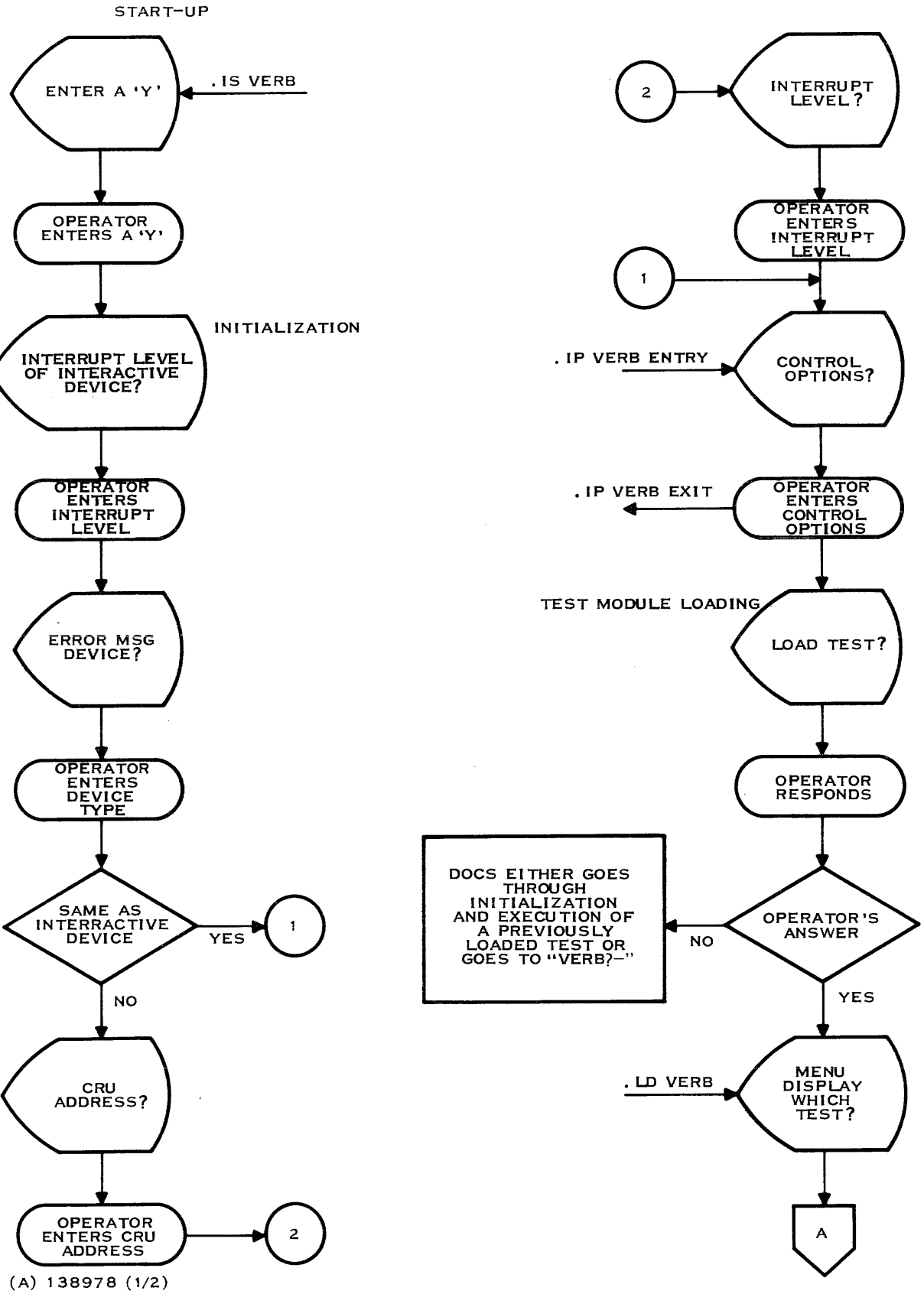


Figure 3-1. Standard DOCS Flow Chart for Start-Up, Initialization, Test Module Loading, and Test Module Execution (Sheet 1 of 2)

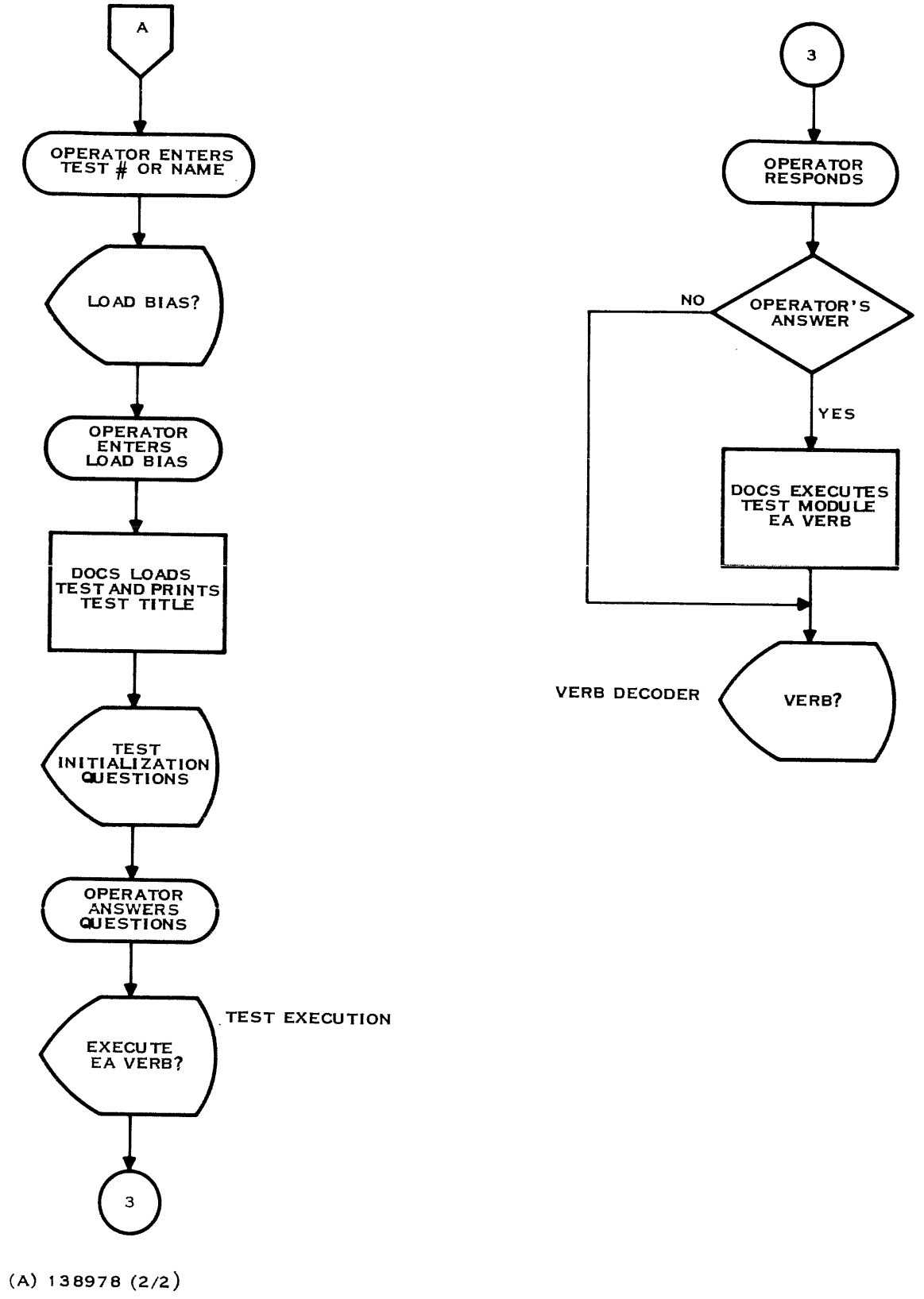


Figure 3-1. Standard DOCS Flow Chart for Start-Up, Initialization, Test Module Loading, and Test Module Execution (Sheet 2 of 2)

3.2.7 **TEST MODULE LOADING AND EXECUTION.** After initialization, code in the Standard DOCS leads the operator through loading and execution of a test module. However, the operator can load and execute the test module by entering the correct set of verbs. Table 3-2 gives an example of the messages output during this phase.

The operator is first asked if a test should be loaded. The operator will answer with a "0" (NO) if a test has already been loaded, or if the operator wishes to run the DOCS by itself. The Standard DOCS will either execute the Initialize Test (IT) verb of the previously loaded test module (this verb does the test module initialization) or will go to the question VERB?. If the operator answers 1 (YES), the Standard DOCS will lead the operator through the test module load procedure. The format of this procedure depends on the available load media from which the DOCS was loaded. The various formats are given in paragraph 3.2.9.13 which describes the Load Diagnostic verb (.LD). In all of these procedures, the DOCS will ask for the location where the test module is to be loaded. The default printed is the first location following the end of the DOCS. Care must be taken when specifying the load point so that the DOCS does not attempt to load a test module on top of itself. If, however, the test module must be loaded in the same area that contains the DOCS, have the DOCS load itself and specify a load bias that will move the DOCS to a different memory area. Once the DOCS has been relocated, the test module can be loaded.

After loading the test module, the Standard DOCS will print the title of the test module and execute the IT verb. The operator will be asked a number of questions related to specific information needed to execute the test correctly.

When the test module has been initialized, the Standard DOCS will ask if the EA verb should be executed. This verb executes all of the standard test module subtests. If the operator answers 0 (NO), the DOCS goes to the question VERB?. Otherwise, the EA verb is executed and followed by the question VERB?.

NOTE

Once the DOCS has been loaded, it does not have to be reloaded everytime a new test module is loaded. The DOCS remains in memory and can load any number of test modules by repeated execution of the .LD verb. However, the DOCS can interface with only one test module at a time. If a test module is loaded and this is followed by an execution of the .LD verb, all communication with the old test module is lost.

Table 3-2. Standard DOCS Test Module Loading and Execution Example

LOAD TEST? (DEF=0) - 1

(The DOCS will display a menu for test module selection or prompt the operator on loading instructions.)

LOAD BIAS? (DEF = 203C) – <RETURN>

16 I/O MODULE VERSION 04/78 *D

16 I/O CRU BASE DEFAULT = 0060 – <RETURN>

16 I/O MODEL NUMBER DEFAULT = 01 – <RETURN>

16 I/O INTERRUPT LEVEL DEFAULT = OE – <RETURN>

EXECUTE EA VERB? (DEF=1) – <RETURN>

(The DOCS will execute the test module's EA verb.)

VERB? –

3.2.8 **INTERRUPT HANDLING.** The Standard DOCS is designed to handle all interrupts except for those specifically being handled by a test module. During the DOCS's start-up and initialization phases, interrupt vectors are set up for power up (level 0); power fail (level 1); system error (level 2); the real time clock (levels 5, 7, or 15); the interactive device; and the error message device. All other interrupt vectors are set to trap to code which handles unused interrupts. If one of the unused interrupts occurs, the following message is output and the operator has the option of re-setting it or not:

UNEXPECTED INT LEVEL = XX, RESET INTERRUPT? (DEF=1) –

If the operator responds with a 1 (YES), the DOCS executes a RSET instruction and returns to the start-up phase to reinitialize the system. Otherwise, the DOCS goes to VERB?. Many of the test modules will take over an interrupt trap. This is usually done during the test module initialization. If the .IS, .LD, or IT verbs are executed after initialization has completed, the DOCS will restore the interrupt vectors to their original values before the verb is executed. Care must be exercised by the operator if any of the verbs are terminated by typing the at (@) key. The operator must realize that the interrupt vectors have been restored and that their contents may not be valid for proper execution of the test module.

3.2.9 **STANDARD DOCS VERBS.** The verbs supported by the Standard DOCS are described in the paragraphs below. These are general utility verbs designed to aid the operator in diagnosing problems and to load test modules. All DOCS verbs are preceded by a period (.) to distinguish them from the verbs supplied by the test module. Table 3-3 is a summarized list of the Standard DOCS verbs.

Table 3-3. Standard DOCS Verbs

Mnemonic	Meaning
.IS	Initialize system
.IP	Initialize control options
.PV	Print available verbs (both DOCS and test module)
.DM	Dump memory (capable of mapping on 990/10 and 990/12)
.MM	Modify memory (unmapped)
.MV	Move memory (unmapped)
.MI	Memory initialize (capable of mapping on 990/10 and 990/12)
.MS	Memory search (unmapped)
.CW	Read/Write CRU word (looping option)
.AD	Add hexadecimal data
.GO	Go to user specified location
.CS	Clear interactive VDT screen
.LD	Load diagnostic test module

3.2.9.1 .IS – Initialize System

Format VERB? – .IS

Action This sends the DOCS back to start-up with the prompt

ENTER A 'Y'

3.2.9.2 .IP – Initialize Control Options

Format VERB? .IP

ENTER CONTROL OPTIONS SEPARATED BY COMMAS:

(E=ERR MSG'S, H = HDR MSG's, N = ERR #'s, I = IDLE ON ERR's)

(DEF = E, H, N) – X, X, . . .

Action This allows the operator to modify the system control options.

3.2.9.3 .PV – Print Available Verbs

Format VERB? – .PV

DOCS VERBS

.XX .XX .XX .XX XX

TEST VERBS

XX XX . . . XX

Action The PV verb prints out a list of all of the verbs that are defined. The verb names such as .DM, .MM, and others are printed. If no test is currently loaded, no test verbs will be output.

3.2.9.4 .DM – Dump Memory

Format VERB? – .DM

ADDR – AAAAAA

Action The .DM verb can be used to dump words of memory to the selected error message device. AAAAAA can be up to a 21 bit, byte level memory address from which the dumping starts. If mapping is available, data can be printed from above $FFFF_{16}$. Otherwise, the address will wrap around and the data will come from low memory.

NOTE

Eight lines of hexadecimal data and ASCII interpretation will be output to the error message device followed by the prompt RETURN TO CON'T. If the operator presses the RETURN key, eight more lines are output. This continues until the operator presses the at (@) key.

Output Press - @ - TO STOP

ADDR 0 2 4 6 8 A C E

013000 1234 1234 1234 1234 1234 1234 1234 1234 <ASCII interpretation>

(Repeated lines omitted)

013070 9876 9876 - - - - - - - - - - - - - - - - <ASCII interpretation>

RETURN TO CON'T

3.2.9.5 .MM – Modify Memory

Format VERB? – .MM

ADDR – XXXX

AAAA DDDD – YYYY

•

•

•

AAAA DDDD – @

Action The operator enters the address (XXXX) and the data at that address is displayed (DDDD). The operator can then modify the data or press the RETURN key to go to the next location. To end the verb, the operator presses the at (@) key. This verb will not handle addresses above $FBFC_{16}$ (the verb is not mapped). The TILINE Peripheral Control Space ($F800_{16}$ – $FBFC_{16}$) can be modified with this verb.

3.2.9.6 .MV – Move Memory

Format VERB? – .MV

FROM – XXXX TO – YYYY # - ZZZZ

Action Move ZZZZ words from address XXXX to address YYYY. This verb will not handle addresses above $FBFC_{16}$.

3.2.9.7 .MI – Memory Initialize

Format

VERB? – .MI

ADDR – AAAAAA # – NNNN DATA – DDDD

Action

Starting at the address specified by the operator (AAAAAA can be up to a 21 bit, byte level address), this verb will initialize NNNN words of memory with the specified data (DDDD). If mapping is available, the data can be stored in locations above $FFFF_{16}$. Otherwise, the addresses will wrap around and the data will be stored in low memory. This verb will not initialize the TILINE Peripheral Control Space (not available on the 990/4) but will initialize the RAM which is shadowed by it.

3.2.9.8 .MS – Memory Search

Format

VERB? – .MS

COND*EQ NE GT LT * –XX

ADDR – AAAA # – NNNN DATA – DDDD

Action

Search memory starting at ADDR for # words or until the condition being searched is found.

EQ means search until DATA = word in memory

NE means search until DATA ≠ word in memory

GT means search until DATA > word in memory

LT means search until DATA < word in memory

If the condition is found, print out the address of the word and the data. If the condition is not found, print NOT FOUND message. For example, to search memory from location 00F0 to 01FF (10F words) for a data word which is logically greater than CCEC, enter the following:

VERB? – .MS

COND*EQ NE GT LT* –GT

ADDR -F0 # - 10F, DATA - CCEC

3.2.9.9 .CW – CRU WRITE

Format

VERB? – .CW

CRU BASE – BBBB = XXXX

BITS – B DATA – DDDD LOOP ON WRITE? – L

Action — When the operator specifies the CRU base, the verb prints the data (XXXX) present at that base. The operator can then write from zero to 16 bits (B) of data (DDDD) to that CRU base. He can also loop on this write by answering 1 (YES) on the loop question (L). To terminate the looping, the operator presses the at (@) key.

3.2.9.10 .AD – Add Hexadecimal Data

Format

VERB? – .AD
A(DDDDDDDD) – XXXX OPER (+) – B(DDDDDDDD) –
XXXX = RRRRRRRR

Action — The .AD verb can be used to add two hexadecimal numbers together. The default value for A is displayed and can be changed. The default value for the operator, a plus (+) or minus (-) sign, is displayed and can be changed. The operator can specify if the B value is to be added or subtracted (two's complement) from the A value. The default value for B is then displayed and the operator can change it. The sum of A ± B is then displayed (RRRR).

3.2.9.11 .GO – Go To

Format

VERB? – .GO

ADDR – XXXX

Action — This verb branches to the address input by the operator. A BL (branch and link) instruction is used to branch to the specified location. If the code at that location executes an RT(B *R11) instruction, control returns to the DOCS. This verb does not handle addresses above $FBFD_{16}$.

3.2.9.12 .CS – Clear Interactive VDT Screen

Format — VERB? – .CS

Action — This verb is used to clear the interactive VDT screen and moves the cursor to the upper left hand corner. If the interactive device is a 733 ASR/KSR, this verb causes a carriage return and line feed.

3.2.9.13 .LD – Load Diagnostic

Format — VERB? – .LD

Action — This verb is used to load diagnostic test modules into memory so they can be executed. Diagnostics are available on a number of load media and the operator prompting messages output by the .LD verb depend on the media being used. For example if the DOCS has been loaded from a hard disk media, it will prompt the operator differently than if it were loaded from a cassette media. The reason for this is that when the DOCS is originally placed on a media, a loader for that media is also

installed. This means that the DOCS loaded from one media cannot load diagnostic test modules from another media. The exception to this is the unit record media (cassette, cards, MDDS, MDU). The loading protocols for the various media are given below:

Sample Output – Hard Disk Media (DS10, DS25, DS31, DS50, DS200)

```
TYPE: @ TO INTERRUPT DISPLAY

00)  CRCOMM    01)  CRDRDR    02)  FLPDSK    03)  OUTMOD
04)  DS10PD    05)  FIVMOD    06)  ADCHK     07)  CRUEXP
08)  INPMOD    09)  IO16      0A)  EMUTST    0B)  EROMBT
0C)  DSKM3X    0D)  RAM10     0E)  PROMPG    0F)  RAM04
10)  AU10      11)  MNDOCS    12)  ACUTST    13)  DACHK
14)  DOCS      15)  LP810     16)  DSKTR1    17)  LPTEST
18)  FLPTST

TYPE: DIAG. # OR FILE NAME? –

LOAD BIAS? (DEF=275C) –
```

When the load media is a hard disk, the DOCS will display a menu of the diagnostics resident on the media. The operator is then asked to type the number of the diagnostic to be loaded as shown in the menu or the diagnostic name itself (e.g. the operator could type 15 or LP810). When the interactive device is a VDT, an additional message is output following the menu – TYPE 'RETURN' FOR ADDITIONAL FILES. Since there may be more diagnostics on the media than can be displayed on the VDT screen, the operator can press the RETURN key and DOCS will display another screen of menu. Continued typing of the RETURN key will cause additional files to be displayed or the original menu will be displayed again.

Once the diagnostic is chosen, the operator is asked to specify the load point for the test module (the default printed is the location immediately following the DOCS). If the operator specifies a load bias less than the end of the DOCS, the following warning message is printed:

```
*** WARNING *** THIS MAY OVERLAY DOCS.

DOCS MAY HAVE TO BE RELOADED TO UPPER MEMORY

ARE YOU SURE? (DEF=0) –
```

If the operator responds with a 1 (YES), the DOCS will attempt to load the test module. Otherwise the operator is again asked for the load point.

After the load bias has been specified, the DOCS will load the test module and tell the operator the ending location of the module.

Sample Output – Single Density Floppy Diskette

The operator is first prompted with the following message:

```
INSERT THE CORRECT DISKETTE & PRESS 'RETURN' –
```

Following this, the procedure is exactly the same as described for the hard disk media.

Sample Output – Cards

WHICH LOADER? (0=CARD, 1=CASS, 2=MDU, 3=MDDS) (DEF=2) – 0

LOAD BIAS? (DEF=2586) –

LOAD DEVICE CRU BASE? (DEF=0040) –

PLACE CARD DECK IN CARD READER & READY.

PRESS 'RETURN' WHEN UNIT IS READY –

Sample Output – Cassette

WHICH LOADER? (0=CARD, 1=CASS, 2=MDU, 3=MDDS) (DEF=2) – 1

LOAD BIAS? (DEF=2586) –

LOAD DEVICE CRU BASE? (DEF=0000) –

PLACE CASSETTE IN TRANSPORT & POSITION AT LOAD POINT.

PRESS 'RETURN' WHEN UNIT IS READY –

Sample Output – MDU (Maintenance Diagnostic Unit)

WHICH LOADER? (0=CARD, 1=CASS, 2=MDU, 3=MDDS) (DEF=2) – 2

LOAD BIAS? (DEF=2586) –

PLACE CASSETTE IN MDU, PUSH RESET & REWIND.

PUSH LOAD BUTTON ON MDU BOX WHEN UNIT IS READY

Sample Output – MDDS (Factory Download System)

WHICH LOADER? (0=CARD, 1=CASS, 2=MDU, 3=MDDS) (DEF=2) – 3

LOAD BIAS? (DEF=2586) –

LOAD DEVICE CRU BASE? (DEF=0000) –

DIAL TEST # ON CONTROL BOX & PUSH LOAD BUTTON.

3.3 MINI DOCS

Mini DOCS is designed to take up as little memory space as possible while still supporting the major features of the Standard DOCS. Many of the system options and verbs have been deleted to achieve this size reduction. The Standard DOCS requires from about 2500_{16} to 2700_{16} words of memory depending on the type of loader being used. The Mini DOCS requires from 1900_{16} to $1B00_{16}$ words of memory. The Mini DOCS is available on all supported media and can be loaded directly or by the Standard DOCS.

3.3.1 **MINI DOCS INITIALIZATION**: Start-up for the Mini DOCS is exactly the same as for the Standard DOCS. However, the only question asked during the initialization phase of the Mini DOCS is the interrupt level of the interactive device. The error message device will be defaulted to the interactive device and the Control Options will be defaulted to print error messages, headers and error numbers (E, H, N).

3.3.2 **MINI DOCS VERBS.** The only verbs supported by the Mini DOCS are .IS, .PV and .LD. These are described in paragraphs 3.2.9.1, 3.2.9.3 and 3.2.9.13, respectively.

3.4 FRONT PANEL DOCS

The front panel version of the DOCS is used when the only available interface with the operator is the programmer panel. Although the programmer panel is a very restricted type of communication link between the operator and the DOCS, a mechanism allows test modules to be loaded and executed. This mechanism also allows for interaction and error information reporting. As far as the operator is concerned, there are no available verbs. Internally, however, the Front Panel DOCS will execute a predefined set of verbs which accomplish the functions just mentioned. A brief discussion of the 990 programmer panel is given in Appendix C.

3.4.1 **SUPPORTED LOADING MEDIA.** The Front Panel DOCS is available on all media except the hard disk type (DS10, DS25, DS31, DS50, DS200). The Front Panel DOCS also can load test modules from the media on which it is available. The procedures for loading test modules are identical to those described in paragraph 3.2.9.13, except that the manner in which the questions are answered is different. This will be discussed in the following paragraphs.

3.4.2 **OPERATOR CONVENTIONS.** The conventions used by the operator to communicate with the Front Panel DOCS are listed below:

1. The DOCS uses the programmer panel data, FAULT, and IDLE lights to output information to the operator. All data is output as hexadecimal numbers.

2. The operator uses the HALT, RUN, and data switches to input information to the DOCS. All data is entered as hexadecimal numbers.

3. Whenever the DOCS or the test module requires input from the operator, the default value of the parameter will be displayed in the programmer panel data lights. The computer then will idle (the IDLE light will be lit), waiting for an operator response. If the operator presses the HALT and RUN switches, the DOCS uses the default value. To enter a new value, only the HALT switch will be pressed. This will cause the program counter value to be displayed on the front panel data lights. The operator now should enter, via the data switches, the new value for the parameter and press the RUN switch. The DOCS then will obtain the parameter value entered on the data switches and will continue execution.

4. All YES/NO questions are answered with a 1 for YES and 0 for NO.

5. If an error occurs in a test, the error message number will be displayed in the right byte of the front panel data lights, and FF will be displayed in the left byte. The FAULT light will also be lit. All documentation in this handbook showing error messages will give a corresponding error number.

6. An error interrupt (either level 2 or an unexpected level) will be communicated by displaying $FFFX_{16}$ in the data lights. The X will be the interrupt level. The computer then will idle.

7. The Front Panel DOCS may experience certain internal error conditions. These errors are communicated to the operator by displaying $FXXX_{16}$ in the data lights. The computer then will idle. XXX corresponds to the internal error codes that are listed below:

Value	Error
F000	Illegal service call made to the DOCS.
F001	Unexpected XOP occurred.
F002	Loaded Test Module has no EA verb.

8. When the computer enters an idle state (with the IDLE light lit), it can be restarted by pressing the HALT and RUN switches.

3.4.3 **INITIALIZATION.** The Front Panel DOCS asks one initialization question: IDLE ON ERRORS? –. If the operator answers with a 1 (YES), the computer will idle on error. If the operator answers with 0 (NO), the computer will continue testing after displaying the error number and setting the FAULT light.

3.4.4 **TEST MODULE LOADING AND EXECUTION.** After the operator has responded to the DOCS initialization question, the Front Panel DOCS will execute the test module loading procedures. The procedures are identical to those described in paragraph 3.2.9.13 with the specific questions depending on the load media being used.

If, for example, the floppy diskette is the load media in use, the first question asked by the Front Panel DOCS is whether the operator has inserted the diskette containing the desired test module. When the proper diskette has been inserted, the operator responds with a 1 (YES). The next question asks for the diagnostic file number. This information can be found on the diskette label. All files contained on the diskette are listed on the label along with their corresponding numbers. The next question asks for the load bias. If the operator responds with an address that may cause the DOCS to be overlayed, an additional question will ask the operator ARE YOU SURE?. If the operator enters a 0 (NO), the load bias question will be repeated. Otherwise, the DOCS will attempt to load the test module.

When the Front Panel DOCS has completed the load successfully, the operator will be asked the test module's initialization questions. These questions are identical to those asked when the Standard DOCS is used and are listed in each section that describes a DOCS type test module. The only difference is that the questions are answered through the programmers panel.

The Front Panel DOCS handles the execution of a test module by repetitively executing the test module EA verb. This verb executes all of the standard subtests of the module and if no errors are encountered, the computer will go into idle with a 0000 displayed in the programmer panel. At this point, the DOCS is asking the question LOAD TEST? –. If the operator responds with a 1 (YES), DOCS will return to the loading procedures for loading a new test module. If the operator responds with a 0 (NO), the DOCS will display the loop count and execute one more pass of the EA verb.

NOTE

If the .EA verb contains interactive subtests, additional questions will be asked before the LOAD TEST? – question. The operator should familiarize himself with the test module operation to prevent confusion over which questions are being asked.

Figure 3-2 shows a flow chart for the overall operation of the Front Panel DOCS.

3.5 DOCS RESTART PROCEDURE

There are a number of situations where the operator may need to restart a diagnostic. For example, he may have completed a programmer panel version of a test and needs to run it again on a different unit without having to reload the test. In another case, he may need to test another unit using the standard-version but cannot get control back to VERB? – by entering @. For these and other situations, the quickest way back to VERB? – is by using the information in the Power-Up Interrupt Vector established by the initialization code. The following procedures can be used:

1. Press HALT and CLR.
2. Enter LOAD POINT + 30_{16} in the data switches.
3. Press ENTER PC and RUN.

The programmer panel version will start by asking the question IDLE ON ERRORS? –. The standard version starts running at VERB? –.

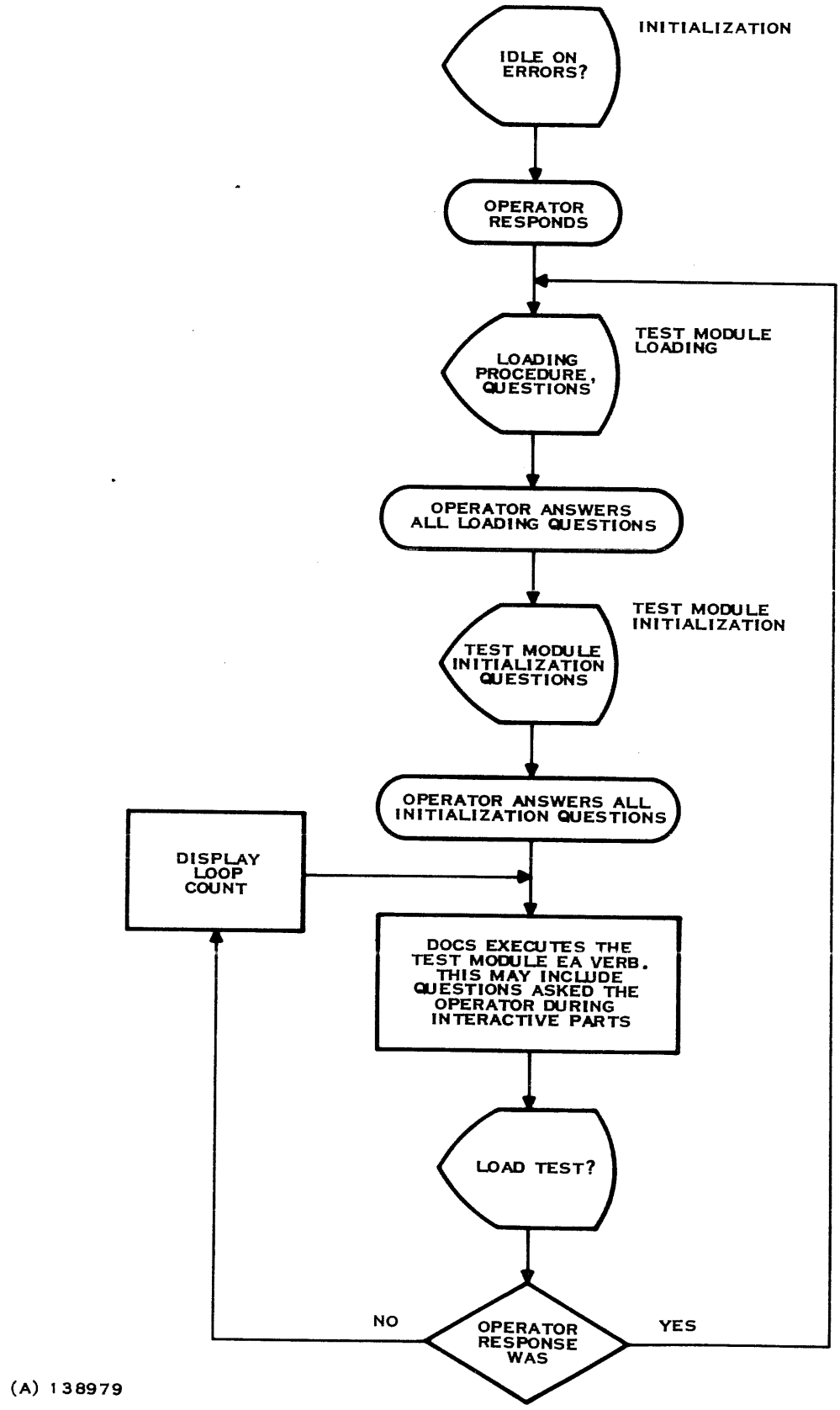


Figure 3-2. Front Panel DOCS Flow Chart

SECTION IV

LOADING PROCEDURES

4.1 INTRODUCTION

This section contains the procedures required to load either the Diagnostic Operational Control System (DOCS) or Standalone tests into the 990 computer and start their execution. Diagnostics are available on a variety of load media and each of these is discussed in the following paragraphs.

4.2 LOAD MEDIA AND PACKAGING

Diagnostics are available on four general media: cassette, cards, hard disks, and floppy diskettes (see figure 4-1). The hard disk media include the DS10, DS25, DS31, DS50, and DS200 disk packs. The floppy media is a single-density, single-sided diskette type. The diagnostics have been packaged according to load media and the part numbers for each of these kits are shown in Appendix E.

Standalone diagnostics (e.g., AU10 and AU04) are packaged on all media. On the unit record media (cards and cassettes) they can be loaded directly using the procedures given below. They can also be loaded and executed by the DOCS using the procedures described in Section III. The DOCS also has the ability to load old VERBPK type diagnostics. After execution, however, VERBPK tests will not return control to the DOCS, and the DOCS must be reloaded.

The cassette kit consists of a set of cassettes where each cassette contains two diagnostics: one on side A and one on side B. In addition, one of the cassettes in the kit contains three versions of the DOCS: the Standard DOCS on side A, and the Mini DOCS and the Front Panel DOCS on side B.

The card kit includes a card deck for each DOCS version: the DOCS test module and Standalone test.

Each disk kit contains all of the diagnostics on one pack. These media are controlled by the Standard DOCS. When booted, the DOCS will be loaded and will start execution. The procedures described in Section III must then be used to load any of the DOCS test modules or Standalone tests. The Mini DOCS resides on each of these media and can be loaded by the Standard DOCS. There is no Front Panel version of the DOCS on this media. All disk kits are DX10 file compatible.

The floppy kit consists of a set of diskettes where each diskette contains a number of diagnostics. Also, each diskette (except FPDOCS) contains a copy of, and is controlled by, the Standard DOCS which is loaded and executed when the diskette is booted. One of the diskettes (FPDOCS) is controlled by the Front Panel version of the DOCS. When this diskette is booted, the Front Panel DOCS is loaded and executed. One of the diskettes contains a copy of the Mini DOCS and a floppy copy utility. Once DOCS is loaded, the procedures described in Section III should be followed to load the DOCS test modules or Standalone tests. All diskettes are TX990 file compatible.

NOTE

The DOCS loaded from one media cannot, in general, load test modules from another media. However, when loaded from disk or diskette, the DOCS can load test modules from the card or cassette media. When loaded from card or cassette media the DOCS can load test modules from either of these media.

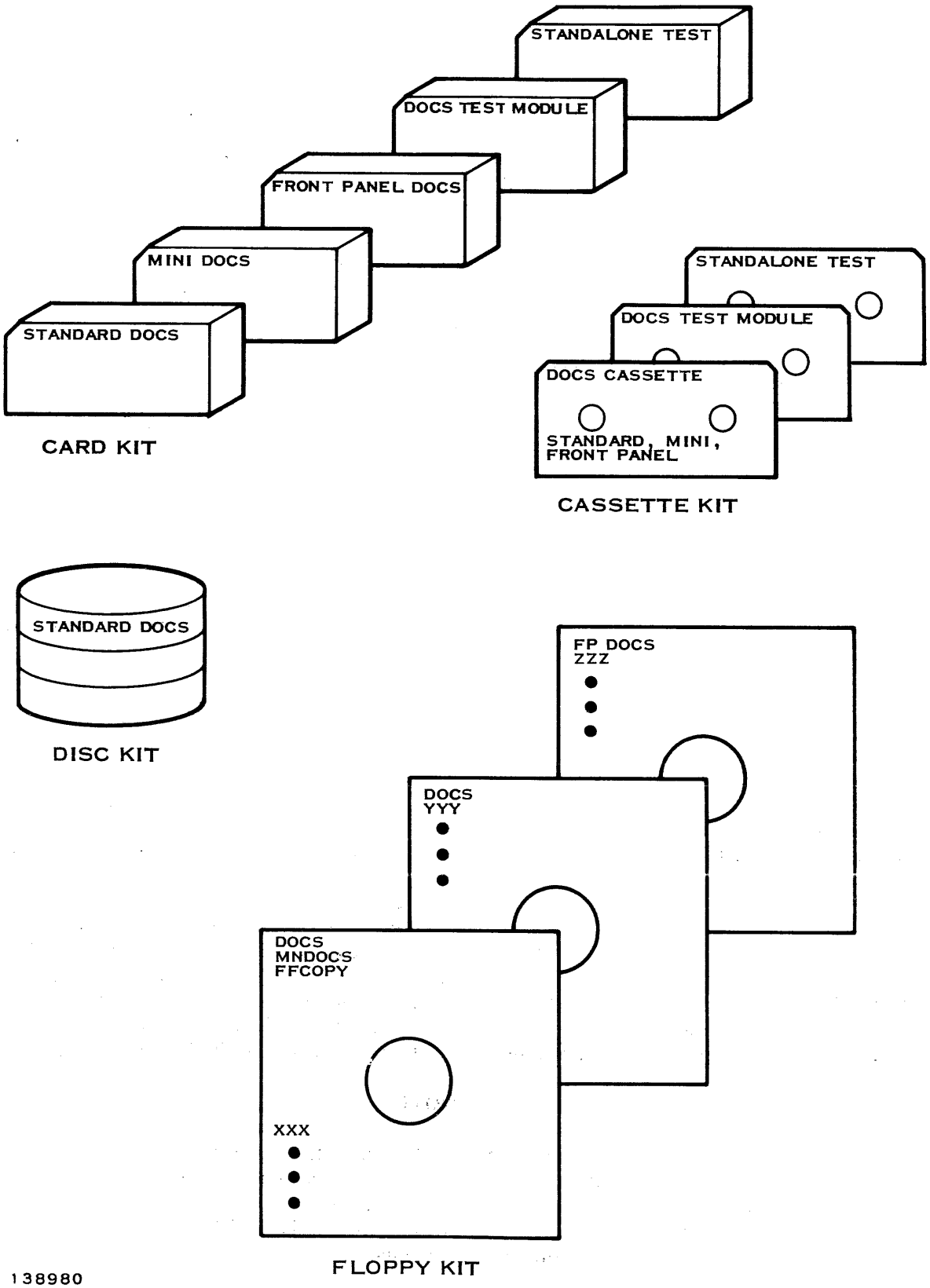


Figure 4-1. Diagnostic Media Kits

4.3 LOADING FROM CASSETTE ON THE 733 ASR

DOCS or Standalone tests can be loaded from cassette on the 733 ASR using the following procedure. The 733 ASR is expected to be located at CRU address 0 and interrupt level 6.

1. Select the correct side of the cassette (A or B).
2. Place the cassette on either cassette unit of the 733 ASR and place that unit in the playback mode and position it to the beginning of the correct file.
3. Press the HALT/SIE switch.
4. Press the CLR switch and the 8 switch to change the displayed value to 0080_{16}.
5. Press the MA switch.
6. Press the CLR switch to change the displayed value to 0000_{16}.
7. Press the MDE switch.
8. If the test is to be loaded at the default ($A0_{16}$) go to step 10.
9. To change the load point, press CLR and enter 92_{16} in the display. Then press MA, CLR, and enter the new load point in the display. Press MDE.
10. Press the RSET switch.
11. Press the LOAD switch.

    The cassette will now load into the 990's memory.

4.4 LOADING FROM CASSETTE ON THE MAINTENANCE DIAGNOSTIC UNIT (MDU)

DOCS or Standalone tests can be loaded from cassette using the MDU. A detailed description of the MDU can be found in Appendix D.

1. Select the correct side of the cassette.
2. Place the cassette in the transport and press the MDU RESET button followed by pressing the REWIND button.
3. Press the HALT/SIE switch on the MDU front panel.
4. Press the LOAD button on the MDU.

The cassette will now load into the 990's memory at $A0_{16}$. If multiple files are on the same side of the cassette, repeat steps 3 and 4 until the correct file has been loaded.

4.5 LOADING FROM CARDS

The following procedure describes how to load DOCS or Standalone tests from cards. The card reader (Model 804) is expected to be located at CRU address 40_{16} and interrupt level 4.

1. Place the object deck in the input hopper of the card reader, and place the card reader in the ready state.
2. Press the HALT/SIE switch.

3. Press the CLR switch and the 8 switch to set the displayed value to 0080_{16}.

4. Press the ENTER MA switch.

5. Press the MDE switch.

6. To change the load point, perform step 9 in the 733 ASR loading procedure (see paragraph 4.3).

7. Press RSET switch.

8. Press the LOAD switch.

Steps 3, 4, and 5 place the value 0080_{16} into memory address 0080_{16}. Any nonzero value (except FFFF) placed in that address has the same effect. However, using some other nonzero value requires more steps.

The cards will now load into the 990's memory and the code will begin executing.

4.6 LOADING DOCS FROM DISK

The steps below should be followed to load the DOCS from any of the hard disk media.

1. Acquire a DS10, DS25, DS31, DS50, or DS200 diagnostic disk with the 990 diagnostics installed.

2. Insert the disk pack into the drive at the system disk location ($F800_{16}$, unit 0). Bring the drive up to speed and leave it in WRITE PROTECT mode.

NOTE

The removable pack on the DS10 drive can be used as the system disk by connecting a jumper plug at the rear of the drive.

3. Perform a standard DX10 load:

   a. If the 990 has a disk boot ROM, press the HALT and LOAD switches. If a programmer panel is not available, cycle system power.

   b. If a disk boot ROM is not available, acquire a DX10 boot card deck or cassette and load using these media.

NOTE

If a DX10 3.X disk boot ROM is installed, the disk can be booted from a nonstandard location. If the locations in table 4-1 contain the indicated data, it is a DX10 3.X disk boot ROM. Locations 82_{16} and 84_{16} should be modified to reflect the proper locations if the disk is nonstandard. A one in bits 4, 5, 6, or 7 at location 84_{16} indicates disk unit 0, 1, 2, or 3, respectively.

Table 4-1. DX10 Disk Boot ROM Locations

Location	Data	Meaning
80_{16}	$FFFF_{16}$	TILINE Device
82_{16}	$F800_{16}$	TILINE Address
84_{16}	0800_{16}	Unit Select

4.7 LOADING DOCS FROM FLOPPY DISKETTE

The floppy diagnostic media is controlled by the DOCS. This means that the first program loaded when the diskette is booted is the DOCS. The following procedures describe how to load the DOCS from the floppy media.

4.7.1 STANDARD LOADING. The following steps describe loading with standard ROMs and with a standard CRU base address:

1. Acquire the diagnostic floppy kit.
2. Insert the appropriate diskette into any of four possible drives.
3. Press the HALT and LOAD switches. If a programmer panel is not available, cycle system power.

4.7.2 NONSTANDARD LOADING. If the unit does not have a floppy ROM or the floppy controller is in a nonstandard location, perform the following steps:

1. Acquire the diagnostic floppy kit.
2. Acquire the nonstandard diagnostic floppy loader on cards or cassette (part number 937921-0001).
3. Insert the appropriate diskette into any of the four possible drives.
4. Load the nonstandard loader from cassette or cards into the computer.
5. If the floppy controller is at the standard location (CRU base 80_{16} or $A0_{16}$), the DOCS should load and execute. If note, an error code of $AAAA_{16}$ is output to the programmer panel, and the computer idles.
6. If the floppy controller is at a nonstandard CRU location:
   a. Press the HALT switch.
   b. Enter the CRU base address of the floppy controller into the ROM's register 9 (typically at location 92_{16}).
   c. Press the RUN switch.

NOTE

If error conditions occur during the loading of the DOCS from the floppy, the loader routine (TXBOOTDK) will output the error code to the programmer panel. The error codes are listed in table 4-2.

Table 4-2. DOCS Error Conditions

Error Code	Meaning	Issued By
00_{16}	No Fault	TXBOOTDK
11_{16}	ID check error	TXBOOTDK
12_{16}	No address mark found	TXBOOTDK
15_{16}	Data check error	TXBOOTDK
17_{16}	Deleted sector detected	TXBOOTDK
19_{16}	Not ready	TXBOOTDK
$1A_{16}$	Write protected	TXBOOTDK
$1B_{16}$	Equipment check error	TXBOOTDK
$1C_{16}$	Invalid track or sector	TXBOOTDK
$1D_{16}$	Seek error or ID not found	TXBOOTDK
61_{16}	Nonrelocatable code	TXBOOTDK
62_{16}	Checksum error	TXBOOTDK
63_{16}	File not found	TXBOOTDK

4.7.3 **FLOPPY COPY UTILITY.** The Floppy-to-Floppy copy utility program (FFCOPY) is contained on the first diskette of the Floppy Diagnostic Kit. The utility will copy any good diskette to another good diskette where the output diskette does not have to be preformatted.

FFCOPY runs under the control of the DOCS. Therefore, the DOCS must be loaded first using the procedures discussed in the previous paragraphs. Once the DOCS is loaded, FFCOPY can be loaded and executed using the procedures given in Section III. Three verbs are available in FFCOPY: IT, EA, and FD.

4.7.3.1 **IT Verb.** The IT verb is used to initialize the utility parameters. When executed, the verb will ask the following questions:

ENTER FLOPPY DISK CRU BASE, DEFAULT = 0080

ENTER FLOPPY DISK INTER LEVEL, DEFAULT = 07

COPY FROM (SOURCE) UNIT # (DEFAULT = 0)?

NOTE: THE UNIT NUMBERS ARE (0, 1, 2, 3)

COPY TO (DESTINATION) UNIT # (DEFAULT = 1)?

4.7.3.2 **EA Verb.** The EA verb is used to execute the formatting and copy utility. When the utility starts executing, it will output the following prompting and completion messages:

```
**********************************************************************

INSERT (ORIGINAL) DISKETTE INTO UNIT # 0 AND

INSERT (COPY) DISKETTE TO BE FORMATED INTO UNIT # 1

AND PUSH –RETURN– KEY –

FINISHED FORMATTING OUTPUT DISKETTE WITH 0000 RETRYS

COPY COMPLETE WITH 0000 RETRYS

**********************************************************************
```

After the copy is completed (this will take about 2.5 minutes), the copy utility will loop back to the beginning and start over. Therefore, multiple copies can be made without entering the EA verb every time.

4.7.3.3 **FD Verb.** The FD verb can be used to format a diskette only. When executed, the verb will output those prompting and completion messages related to formatting as shown in the EA verb output, paragraph 4.7.3.2.

4.7.3.4 **Error Messages.** The following error messages may be output by the utility if error conditions occur. If an error condition does occur, any formatting or copying activities will be aborted.

Number	Message
10_{16}	DATA READ .NE. DATA WRITTEN
20_{16}	UNABLE TO FORMAT FLOPPY
30_{16}	TIMED OUT WHILE X-FERRING DATA
40_{16}	TIMED OUT AFTER ISSUING COMMAND
50_{16}	WITHOUT INTERRUPT
60_{16}	**STATUS ERROR** COMMAND ISSUE = XXXX STATUS = YYYY
70_{16}	UNEXPECTED FLOPPY DISK INTERRUPT AT LOC X

SECTION V

(To be supplied later.)

SECTION VI

(To be supplied later.)

SECTION VII

990/10 ARITHMETIC UNIT TEST (AU10)

7.1 INTRODUCTION

The AU10 tests the logic on the AU1 board in the 990/10 minicomputer. AU10 operates as a standalone test and consists of five subtests: (1) real-time clock test, (2) Level 2 interrupt test, (3) register test, (4) ALU test, and (5) microcode test.

The primary logic tested by these tests is on the AU1 board. The AU2 board is tested only in that the functioning of the AU2 board is required to support the test operation. The only major area of the 990/10 not tested is the CRU logic. This omission is due to the inability of software to have access sufficient to ensure correct operation. This test also makes no attempt to test map logic, as it is designed to test both mapped and unmapped systems. The map test is used to test the AU logic associated with the mapping function.

7.1.1 TEST 1 – REAL-TIME CLOCK TEST. This test tests the proper functioning of the real-time clock and clock interrupts.

7.1.2 TEST 2 – LEVEL TWO INTERRUPT TEST. This test will test the Level 2 interrupt.

7.1.3 TEST 3 – REGISTER TEST. This test is designed to test the internal registers on the 990/10 minicomputer. The tested registers are the PC, WP, ST, DA, and DD.

7.1.4 TEST 4 – ALU TEST. This test tests the ALUs and the data paths associated with ALU functions.

7.1.5 TEST 5 – MICROCODE TEST. This test exercises all microcode in the 990/10 minicomputer.

7.2 OPERATOR INTERFACE

AU10 is a standalone test. For a detailed explanation of standalone tests, refer to section II in this manual.

This test offers only the standard standalone test operator options: (1) idle on errors, (2) print headers, and (3) print errors. Consult section II on standalone tests for instructions in selecting these options.

7.3 SYSTEM REQUIREMENTS

The minimum hardware configuration suggested for AU10 includes (1) a 990/10 minicomputer with 8K words of memory, (2) output device and (3) an appropriate Loading device.

7.4 ERROR MESSAGES

The following paragraphs describe the error messages output by AU10.

7.4.1 ERROR MESSAGES (TESTS 1-5). Error numbers and messages for AU10 tests are shown in table 7-1.

Table 7-1. AU10 Error Numbers and Messages (Tests 1-5)

Error Number Hexadecimal	Message
1	***ERROR*** – THE REAL TIME CLOCK FAILS TO INTERRUPT
2	***ERROR*** – LEVEL 2 INTERRUPT TEST
3	***ERROR*** – THE REGISTER TEST HAS AN ERROR.
4	***ERROR*** – ACU ERROR
5	***ERROR*** – MICROCODE ERROR
30	BAD INT
31	PARITY ERROR
32	ILLEGAL OP
33	PRIVILEGED ERROR INT
34	TILINE TIMEOUT
35	MAP ERROR INTERRUPT LATCHED (BYTE) ADDR XXXXXX
36	UNEXPECTED INT LEVEL = XX
37	***ERROR*** – THE PC FOR THE CURRENT TEST IS OUT OF THE ALLOWABLE ADDRESSING AREA
38	***ERROR*** – RTC DID NOT OCCUR IN WINDOW
62	INTERRUPT ERROR

7.4.2 **MISCELLANEOUS ERROR MESSAGES.** The following message indicates how to trace the cause of the error.

REGISTER 11 IN THE FOLLOWING PRINTOUT CONTAINS THE RELATIVE

ADDRESS OF THE BRANCH TO THE ERROR ROUTINE.

The following message tells where the branch to print the messages took place. It will usually be useful in tracing errors.

PGM LISTING ADDR: XXXX

This message is a dump of all registers in use at the time of the error.

WORK POINTER PC & STATUS AT TIME OF ERROR WP = XXXX PC = XXXX

ST = XXXX WORKSPACE XXXX XXXX XXXX XXXX XXXX XXXX XXXX
XXXX XXXX XXXX XXXX XXXX XXXX XXXX
XXXX XXXX

The following message indicates that an error occurred in the subtest.

ERROR RETURN FROM SUBTEST

The following message is displayed when the real-time clock (RTC) error check detects an out-of-boundary condition.

ERRORTHE PC FOR THE CURRENT TEST IS OUT OF THE ALLOWABLE

ADDRESSING AREA

ADDRESS OF PRIOR AND CURRENT INTERRUPT

CURRENT WP PC ST

PRIOR WP PC ST

7.4.3 HEADER MESSAGES.

Start-test message:

AU10 990/10 ARITHMETIC UNIT TEST VERSION XX/XX/XX

NOTE

The version is the release data MM/DD/YY.

Start-subtest messages:

START SUBTEST 1 – REALTIME CLOCK TEST

START SUBTEST 2 – LEVEL 2 INTERRUPT TEST

START SUBTEST 3 – REGISTER TEST

START SUBTEST 4 – ALU TEST

START SUBTEST 5 – MICROCODE TEST

End-of-subtest message:

SUBTEST COMPLETE

Count of test loops completed:

LOOPS COMPLETED = XXXX

End-of-test message:

TEST COMPLETE

SECTION VIII

990/4 ARITHMETIC UNIT TEST (AU04 and AU04TEST)

8.1 INTRODUCTION

The 990/4 Arithmetic Unit (AU) tests consist of two diagnostics: the AU04 (requiring 4K words of memory) and the AU04TEST (requiring 8K words of memory). Both of these diagnostics test the arithmetic unit of the 990/4 microcomputer.

8.1.1 **AU04.** The AU04 diagnostic consists of two subsets: (1) Instruction Execution test, and (2) Clock Interrupt test.

8.1.1.1 **Test 1 – Instruction Execution Test.** This test executes each arithmetic unit instruction and verifies the result and status.

8.1.1.2 **Test 2 – Clock Interrupt Test.** This test verifies the proper occurrence of clock interrupts at level 5 using the CKON and CKOF instructions.

8.1.2 **AU04TEST.** The AU04TEST diagnostic consists of five subtests: (1) Real-time clock test, (2) Interrupt and XOP test, (3) Register test, (4) Arithmetic Logic Unit (ALU) test, and (5) Microcode test.

8.1.2.1 **Test 1 – Real-Time Clock Test.** This test checks the functioning of the real-time clock and the clock interrupts.

8.1.2.2 **Test 2 – Interrupt and XOP Test.** This test performs extensive testing of the AU interrupt handling and masking capabilities at interrupt level 5. The XOP capability is also tested.

8.1.2.3 **Test 3 – Register Test.** This test directly checks the PC, WP, and ST registers and indirectly checks the DA and DD registers.

8.1.2.4 **Test 4 – ALU Test.** This test checks the ALUs and the data paths associated with ALU functions.

8.1.2.5 **Test 5 – Microcode Test.** This test exercises all microcode in the 990/4 microcomputer.

8.2 OPERATOR INTERFACE

AU04 and AU04TEST are standalone tests. For a detailed explanation of standalone tests, refer to section II in this manual. AU04TEST includes a more accurate clock test than AU04 and also exceeds 4K words in size.

These tests offer only the standard standalone test operator options: (1) idle on errors, (2) print headers, (3) print errors. Consult section II on standalone tests for instructions in selecting these options.

8.3 SYSTEM REQUIREMENTS

The minimum hardware configuration suggested for AU04 includes: (1) a 990/4 microcomputer with 4K words of memory, (2) output device, and (3) an appropriate device to load the diagnostics. AU04TEST requires 8K words of memory.

8.4 ERROR MESSAGES AND NUMBERS

Table 8-1 lists the messages output by AU04 except for the individual instruction errors. Table 8-2 lists the messages output by AU04 for individual instruction errors.

Table 8-1. AU04 Error Messages and Numbers

Number	Message	Explanation
none	990/04 ARITHMETIC UNIT TEST VERSION #XX/XX/XX	Test start message XX/XX/XX is the release date of the test
none	NO ERRORS	No errors occurred during test execution
*none	RTC DID NOT OCCUR IN WINDOW	Error during clock test
none	AU TEST COMPLETED	Test complete
N	ERROR: XXXX INSTRUCTION PGM LISTING ADDR: AAAA	Error during instruction test. N and XXXX are the index number and name of the instruction given in Table 5-1. AAAA is the relative address of the error in the program.
*1	****ERROR****– THE REAL TIME CLOCK FAILS TO INTERRUPT	Error during clock test
2	***ERROR***	Interrupt and XOP test
3	***ERROR***	The register test has an error
4	***ERROR***	ALU error
5	***ERROR***	Microcode error
30	***ERROR***	RTC did not occur in window
31	UNEXPECTED INT LEVEL XX	none
32	***ERROR***	The PC for the current test is out of the allowable area
62	INTERRUPT ERROR	none
64	PARITY ERROR	Memory parity error
69	UNEXPECTED INT LEVEL=XX	Unexpected external interrupt at level XX
70	POWER FAILURE OCCURRED TEST RESTARTED	Power failure/power-up interrupt sequence
0	ERROR WP=XXXX PC=XXXX ST=XXXX WORKSPACE=XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX	Workspace pointer, program counter, status, and workspace at time of error.

*Included in AU04TEST only.

Table 8-2. AU04 Instruction Error Numbers

Error Number DEC	Error Number HEX	Instruction	Error Number DEC	Error Number HEX	Instruction
1	01	C			
2	02	CB	32	20	SRC
3	03	MOV	33	21	LI
4	04	MOVB	34	22	AI
5	05	A	35	23	ANDI
6	06	AB	36	24	ORI
7	07	S	37	25	CI
8	08	SB	38	26	STWP
9	09	SOC	39	27	LWPI
10	0A	SOCB	40	28	CLR
11	0B	SZC	41	29	NEG
12	0C	SZCB	42	2A	INV
13	0D	JMP	43	2B	INC
14	0E	JLT	44	2C	INCT
15	0F	JLE	45	2D	DEC
16	10	JEQ	46	2E	DECT
17	11	JHE	47	2F	SETO
18	12	JGT	48	30	SWPB
19	13	JNE	49	31	ABS
20	14	JL	50	32	B
21	15	JH	51	33	BL
22	16	JOP	52	34	BLWP
23	17	JNC	53	35	X
24	18	JOC	54	36	RTWP
25	19	JNO	55	37	RSET
26	1A	COC	56	38	CKON
27	1B	CZC	57	39	CKOF
28	1C	XOR	58	3A	IDLE
29	1D	SRA	59	3B	DIV
30	1E	SLA	60	3C	MPY
31	1F	SRL	61	3D	XOP

SECTION IX

990/10 RANDOM-ACCESS MEMORY TEST (RAM10)

9.1 INTRODUCTION

The RAM10 tests memory in the 990/10 minicomputer. Default tests addresses are from 1 word beyond the end of the program to FFFE. These defaults may be modified by the operator. If the program fails to execute properly, then the area of memory in which the program resides may be faulty. In this case, the object program should be reloaded to another location in memory. Do not test the area of memory in which the diagnostic program resides.

NOTE

Since the diagnostics are interrupt driven and since the interrupt vectors occupy the low 64 words of memory, the diagnostics should not be used to check the low 64 words (80_{16} Bytes) of memory.

RAM10 consists of 6 subtests. The first three subtests (1 through 3) locate most memory failures. Subtests 4 and 5 locate marginal failures. Subtests 4 and 5 are long-running tests and should be run in as small an area as possible. Subtest 4 requires 30 to 40 hours to run in 32K of memory. Due to the long execution time in this test, the program will blink the panel lights. Test 5 requires 35 minutes. Test 6 is very short, and is used to test ECC logic for 16K chips.

9.1.1 **TEST 1 – MARCHING ONES AND ZEROS.** This test consists of a minimal test of the addressing, reading and writing functions of the memory in the I/O state.

9.1.2 **TEST 2 – WRITE ONE THROUGH ZEROS BACKGROUND.** This is a memory test for cell-to-cell shorts; it also performs a basic test of the memory refresh logic.

9.1.3 **TEST 3 – REFRESH DISTURB.** This test checks the refresh circuits and the ability of the memory cells to maintain the proper state between refresh cycles.

9.1.4 **TEST 4 – GALLOPING ONES AND ZEROS WRITE – WRITE.** This test checks all possible writes followed by reads at different locations.

9.1.5 **TEST 5 – MOVING INVERSIONS READ – WRITE – READ.** This test reads each word and verifies the pattern, then modifies a single bit and reads the word to verify the new pattern.

9.1.6 **TEST 6 – ECC PATTERN TEST.** This test checks and verifies the Electronic Checking and Correcting (ECC) logic for 16K RAM.

9.2 OPERATOR INTERFACE

RAM10 runs under the DOCS on the 990/10 minicomputer. Refer to Section III for a detailed explanation of the DOCS versions.

9.2.1 **AVAILABLE VERBS.** In addition to the verbs supplied in the DOCS, RAM10 also supplies the verbs listed in table 9-1.

9.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by the DOCS, RAM10 also asks for the beginning and ending addresses for the memory test.

First the computer outputs the entire address:

```
XXXXXXXX
```

and then a prompt for the operator to modify the address:

```
XXXXXXXX-
```

Table 9-1. RAM10 Verbs

Verb	Function
IT	Initialize test
E1	Execute test 1
E2	Execute test 2
E3	Execute test 3
E4	Execute test 4
E5	Execute test 5
E6	Execute test 6
EA	Execute tests 1, 2, and 3
L1	Loop on test 1
L2	Loop on test 2
L3	Loop on test 3
L4	Loop on test 4
L5	Loop on test 5
L6	Loop on test 6
LA	Loop on tests 1, 2, and 3
PE	Print Errors
RM	Read Memory
SL	Scope Loop

NOTE

The ending address must be at least 10_{16} words before the end of memory for the test to run correctly.

For example, to change the address 000F2351 to the address 000E3421, the following transaction would take place:

```
ADDRESS DEFAULT = 000F2351 - 000E3421
```

The Scope Loop Test (SL) that asks for a data location and a word of data to write to that location, continuously writes the data to the location until the @ sign is entered.

The RM verb is used to read a section of memory (addresses are entered the same as the IT verb). Parity errors are found as memory is read.

Table 9-2 lists the initialization questions asked by RAM10. The IT verb simply repeats these questions.

9.2.3 **PROGRAMMER-PANEL INITIALIZATION.** In addition to asking the question IDLE ON ERRORS? the programmer-panel version of RAM10 asks the questions listed in table 9-2. Notice that the addresses must be entered in two parts (upper half followed by lower half).

Table 9-2. RAM10 Initialization Questions

Question	Comment
TYPE OF TEST DESIRED? (0 = 16K RAM ECC ON OR 4K RAM; 1 = 16K RAM ECC OFF, UNCORRECTED DATA; 2 = UNCORRECTED DATA REPLACING ECC FIELD) DEFAULT = 00	User specifies test option.
ENTER ADDRESS OF TILINE PERIPHERAL CONTROL SPACE, ADDRESS DEFAULT = FB00	User specifies address of 16K controller.
INPUT BEGINNING ADDRESS DEFAULT = XXXXXXXX–	None
INPUT ENDING ADDRESS DEFAULT = XXXXXXXX–	None

9.3 SYSTEM REQUIREMENTS

The RAM10 diagnostics will run on any 990/10 minicomputer system with an appropriate loading device and an interactive device.

The memory size required by RAM10 depends on the DOCS version used. This will vary between 8K to 12K words. To be able to check out the entire memory requires twice as much memory as required by the test itself. This would make the total memory requirements between 16K and 24K words.

9.4 ERROR MESSAGES AND NUMBERS

Table 9-3 lists the messages output by RAM10.

Table 9-3. RAM10 Error Messages and Numbers

Number	Message	Explanation
none	16K RAM CONTROLLER PRESENT	Tests are run on 16K RAM
none	RAM10 RANDOM ACCESS MEMORY TEST VERSION MM/YY *X	Header message
none	INPUT ADDRESS YOU WISH TO WRITE TO – XXXX XX–	Address for scope loop
none	INPUT BEGINNING ADDRESS DEFAULT = XXXXXX–	Start-address message
none	INPUT DATA YOU WISH TO WRITE – XXXX	Data for scope loop
none	INPUT ENDING ADDRESS DEFAULT = XXXXXX–	Ending address message
none	ERROR COUNT = XXXX	Total number of errors
none	MEMORY UNDER TEST >XXXXXXXX TO >XXXXXXXX	Address test area
none	START TEXT XX	Test-number message
none	SUBTEST COMPLETE	Subtest-ending message
none	LOOP COUNT = XXXX	Loop count at end of test
none	TEST COMPLETE	Test-ending message
none	RESTART INITIATED	Power failure, going through IS verb to start test over
none	POWER RESTORED	Level 0 interrupt occurred
none	TO EXIT TEST ENTER @ START SCOPE LOOP TEST	Scope loop test starting message
none	INPUT ADDRESS YOU WISH TO WRITE TO – XXXXXX	Address of word to run scope loop on
none	INPUT DATA YOU WISH TO WRITE – XXXX	Data for scope loop to write
none	ENTER ADDRESS TO START READING FROM, DEFAULT = XXXXXX	Starting address for RM verb
none	ENTER ENDING ADDRESS, DEFAULT = XXXXXX	Ending address for RM verb
none	START READING MEMORY FOR PARITY ERRORS FROM XXXXXX TO XXXXXX	Starting message for RM verb
none	MEMORY READ COMPLETED, NUMBER OF ERRORS = XX	Ending message for RM verb
1	ACTUAL DATA IS XXXX EXPECTED DATA IS XXXX	Unexpected data

Table 9-3. RAM10 Error Messages and Numbers (Continued)

Number	Message	Explanation
2	BIT IN ERROR IS XXXX	Multiple line message showing cause of error
3	PARITY ERR	Memory parity error
4	LOCATION OF 16K RAM CHIP FAILURE BIT = XXXX ROW = XXXX	Error in 16K RAM chip
5	TILINE TIMEOUT OCCURRED	TILINE timeout
6	ONE BIT ERROR DID NOT OCCUR AT BIT 4	Expected error did not occur
7	MEMORY BITS ACTUAL DATA XXXXXX EXPECTED DATA XXXXXX	Unexpected data
8	ONE BIT ERROR EXPECTED AT BIT 4 OCCURRED AT XX	Expected error not correct
9	16K RAM CONTROLLER NOT FOUND SUBTEST ABORTED	No 16K controller
A	THE ADDRESS GIVEN IS WITHIN THE PROGRAM. PLEASE PICK AN ADDRESS OUTSIDE OF THE PROGRAM.	Address given is inside program bounds
B	**ERROR ONE OR BOTH OF THE INPUT ADDRESS IS INVALID	Address given is inside program bounds
C	ERROR ADDRESS IS XXXXXXXX	Location at which error occurred
D	AN ERROR NOT CAUSED BY ECC LOGIC HAS OCCURRED. USE TEST 1-5 TO ISOLATE ERROR.	While running test an error not caused by ECC logic occurred

SECTION X

990/4 RANDOM ACCESS MEMORY TEST (RAM04)

10.1 INTRODUCTION

The RAM04 tests memory in the 990/4 microcomputer. Memory size to be tested may range up to the 32K maximum allowed. Default test addresses are from one word beyond the end of the program to $3FFE_{16}$. These defaults may be modified by the operator. If the program fails to operate properly, then the area of memory in which the diagnostics reside may be faulty. In this case, the object program should be reloaded to another location in memory. Do not test the area of memory in which the diagnostic program resides.

NOTE

Since the diagnostics use XOP's and are interrupt driven and since the interrupt vectors occupy the low 64 words of memory, this test does not allow the low 64 words (0080 bytes) to be tested.

RAM04 consists of 6 subtests and a scope loop test. The first three subtests (1 through 3) are short running and locate most memory failures. The next two (4 and 5) are long running and locate marginal failures. Because subtests 4 and 5 are long running, requiring three to four hours each to run in 32K of memory, they should be run in as small an area as possible. To indicate they are running, subtests 4 and 5 blink the panel lights. Subtest 6 only locates parity errors in memory. In addition to these six subtests there is a Scope Loop test.

10.1.1 TEST 1 – MARCHING ONES AND ZEROS. The test consists of a minimal test of the addressing, reading and writing functions of the memory in the I/O state.

10.1.2 TEST 2 – WRITE ONE THROUGH ZEROS BACKGROUND. This is a memory test for cell-to-cell shorts; it also performs a basic test of the memory refresh logic.

10.1.3 TEST 3 – REFRESH DISTURB. This test checks the refresh circuits and the ability of the memory cells to maintain the proper state between refresh cycles.

10.1.4 TEST 4 – GALLOPING ONES AND ZEROS WRITE – WRITE. This test checks all possible writes followed by reads at different locations.

10.1.5 TEST 5 – MOVING INVERSIONS READ – WRITE – READ. This test reads each word and verifies the pattern, then modifies a single bit and reads the word to verify the new pattern.

10.1.6 PARITY TEST. This test reads memory and prints a message if a parity interrupt occurs.

10.1.7 SCOPE LOOP TEST. This test performs a continuous write of given data to a given address.

10.2 OPERATOR INTERFACE

RAM04 runs under the DOCS on the 990/4 microcomputer. Refer to Section III for a detailed explanation of the DOCS versions.

10.2.1 **AVAILABLE VERBS.** RAM04 supports the verbs listed in table 10-1.

10.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by DOCS, RAM04 also asks for the beginning and ending addresses for the memory test. Table 10-2 lists the configuration questions asked by RAM04. The IT verb simply repeats these questions.

NOTE

The ending address must be at least 10_{16} words before the end of memory or the test may not run correctly.

10.2.3 **PROGRAMMER-PANEL INITIALIZATION.** In addition to asking IDLE ON ERRORS? the programmer-panel version of RAM04 asks the questions shown in table 10-2.

Table 10-1. RAM04 Verbs

Verb	Function
IT	Initialize test (specify memory bounds)
E1	Execute subtest 1
E2	Execute subtest 2
E3	Execute subtest 3
E4	Execute subtest 4
E5	Execute subtest 5
E6	Execute subtest 6
EA	Execute subtests 1, 2, and 3
L1	Loop on subtest 1
L2	Loop on subtest 2
L3	Loop on subtest 3
L4	Loop on subtest 4
L5	Loop on subtest 5
LA	Loop on subtests 1, 2, and 3
PE	Print errors
SL	Scope loop

Table 10-2. RAM04 Initialization Questions

Question	Comment
INPUT BEGINNING ADDRESS MUST BE GREATER THAN 007F, DEFAULT = XXXX –	Operator enters starting memory address for test (4 hexadecimal characters).
INPUT ENDING ADDRESS DEFAULT = XXXX –	Operator enters 4 hexadecimal characters for ending address.

10.3 SYSTEM REQUIREMENTS

The RAM04 diagnostics will run on any 990/4 microcomputer system. The system must have an appropriate loading device and interactive device. The memory requirements for RAM04 will depend on the DOCS version used and will vary between 6K and 8K words. With manipulation of the relative positions of RAM04 and DOCS the entire memory can be tested with a minimum of 8K to 10K words of memory. Again the variation is due to the version of DOCS.

10.4 ERROR MESSAGES AND NUMBERS

Table 10-3 lists the messages output by RAM04.

Table 10-3. RAM04 Error Messages and Numbers

Number	Message	Explanation
none	RAM04 RANDOM ACCESS MEMORY TEST VERSION MM/YY *X	Header message
none	INPUT BEGINNING ADDRESS MUST BE GREATER THAN 007F, DEFAULT = XXXX –	Start-address message
none	INPUT ENDING ADDRESS DEFAULT = XXXX –	Ending-address message
none	MEMORY UNDER TEST XXXX TO XXXX	Address test area
none	ERROR COUNT = XXXX	Number of errors occurring during any loop test or EA test
none	START TEST XX	Test-number message
none	SUBTEST COMPLETE	Test-ending message
none	LOOP COUNT = XXXX	Loop count at end of test
none	TEST COMPLETE	Before beginning new test
none	POWER RESTORED	Level 0 interrupt occurred
none	RESTART INITIATED	After power restore or faulty test retry

Table 10-3. RAM04 Error Messages and Numbers (Continued)

Number	Message	Explanation
none	INPUT ADDRESS YOU WISH TO WRITE TO – XXXX	Scope loop message
none	INPUT DATA YOU WISH TO WRITE – XXXX	Scope loop message
0	SOFT FAILURE	Unable to determine bit in error
none	BIT IN ERROR IS XXXX ACTUAL DATA IS XXXX EXPECTED DATA IS XXXX	Multiple-line message showing cause of error
0	ERROR ADDRESS IS XXXX	Location at which error occurred
1	DATA COMPARISON ERROR	Data in memory not same as test data but parity correct
2	PARITY ERROR	Parity error interrupt occurred
3	***WARNING*** ADDRESS IS OUT OF TEST AREA BE CAREFUL NOT TO DESTORY PART OF DOCS RAM04 OR THE XOP AND INTER-RUPT VECTORS	Scope loop warning message
4	**ERROR** ONE OR BOTH OF THE INPUT ADDRESSES IS INVALID, YOU MUST ENTER THE BEGINNING AND ENDING ADDRESSES OF THE AREA OF MEMORY TO BE TESTED. THE ADDRESSES MUST NOT BE LESS THAN 256 WORDS APART. BOTH ADDRESSES MUST BE ABOVE OR BELOW THE PROGRAM AND MUST NOT OVERLAP INTO THE PROGRAM AREA.	An error in one or both of the addresses for the memory test (E1-E5)
5	**ERROR** TEST AREA IS IN DOCS	An error in one or both of the addresses for the memory tests (E1-E6)

SECTION XI

990/4 MEMORY PROTECT LOGIC TEST (MEMPRT)

11.1 INTRODUCTION

The MEMPRT tests the memory protect logic on a 990/4 memory expansion board. MEMPRT consists of four subtests as discussed in paragraphs 11.1.1 through 11.1.4.

11.1.1 **TEST 1 – NULL TEST.** This test checks for memory protect violation or improper interrupts with the upper bound and lower bound addresses equal.

11.1.2 **TEST 2 – UPPER BOUND TEST.** This test checks for memory protect violation or improper interrupts in protected memory above, on, and below the upper bound address.

11.1.3 **TEST 3 – LOWER BOUND TEST.** This test checks for memory protect violation or improper interrupts in protected memory above, on, and below the lower bound address.

11.1.4 **TEST 4 – INTERRUPT TEST.** This test checks the proper operation of interrupts related to memory protect functions.

11.2 OPERATOR INTERFACE

MEMPRT runs under the DOCS on the 990/4 microcomputer. Refer to Section III for a detailed explanation of the DOCS versions.

11.2.1 **AVAILABLE VERBS.** MEMPRT supplies the verbs listed in table 11-1.

Table 11-1. MEMPRT Verbs

Verb	Function
IT	Initialize test
EA	Execute test
LA	Loop on test
PE	Print errors

11.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by the DOCS, MEMPRT asks the initialization questions given in table 11-2. The IT verb simply repeats these questions.

11.2.3 **PROGRAMMER-PANEL INITIALIZATION.** In addition to the question IDLE ON ERRORS? the programmer-panel version of MEMPRT asks the questions shown in table 11-2.

Table 11-2. MEMPRT Initialization Questions

Question	Comment
IS MEMORY PROTECT INTERRUPT CONNECTED?-	If the answer is no, (1=yes, 0=no) the second question will not be asked.
MEMORY PROTECT INTERRUPT LEVEL DEFAULT = 0003 -	Enter memory protect interrupt level.
TOTAL AMOUNT OF MEMORY DEFAULT = 0024 -	Enter the number of 1K blocks of memory.

11.3 SYSTEM REQUIREMENTS

In addition to the hardware required by the DOCS a 4K to 20K memory expansion module with the write protect option must be used.

11.4 ERROR MESSAGES AND NUMBERS

Table 11-3 lists the messages output by MEMPRT.

Table 11-3. MEMPRT Error Messages and Numbers

Number	Message
1	*ERROR* PROTECT VIOLATION WHILE SETTING UNPROTECTED MEMORY TO FFFF
2	*ERROR* PROTECT VIOLATION WHILE WRITING TO UNPROTECTED MEMORY
3	*ERROR* PROTECT VIOLATION DID NOT OCCUR WHEN WRITING TO PROTECTED MEMORY
4	*ERROR* UNPROTECTED MEMORY TEST ADDRESS NOT WRITTEN INTO ITSELF
5	*ERROR* PROTECTED MEMORY WRITTEN INTO
6	*ERROR* PROTECTED NONEXPANSION MEMORY NOT WRITTEN
7	*ERROR* UNEXPECTED MEMORY PROTECT INTERRUPT
8	*ERROR* MEMORY PROTECT INTERRUPT DID NOT OCCUR WHEN PROTECTED MEMORY WRITTEN INTO

SECTION XII

733 ASR/KSR DATA TERMINAL TEST (TST733)

12.1 INTRODUCTION

The TST733 tests the printer and keyboard on a 733 KSR and the cassette units on a 733 ASR. TST733 consists of five subtests as described in paragraphs 12.1.1 through 12.1.5.

12.1.1 **TEST 1 – INTERFACE TEST.** The interface test sends a reset to the unit under test and then checks the status of the interface.

12.1.2 **TEST 2 – PRINTER TEST.** The Printer Test checks the ability of the 733 printer to print each printable character in each of the 80 columns of the page. The printer starts the first line with ASCII 20_{16} (SPACE) then increments the ASCII character for each column of the line. The second line begins with ASCII 21_{16}. Each successive line begins with the next ASCII character. When ASCII 7F (DEL) is reached before the end of the line, the character output is forced back to 20_{16} and the line is completed. When 7F is the first character of the line, all printable characters have been printed in all possible locations. At this point the barber pole pattern is complete and it is ready for the operator to verify that all of the characters are present and that each character is printed in all expected locations.

12.1.3 **TEST 3 – INTERRUPT TEST.** This test saves the contents of the interrupt trap then points the trap to an interrupt routine. The interrupt routine stores the status of the interface (CRU bits 8 through 15) in register 4 of the test workspace. The test clears the interrupts. Register 4 then enables interrupts and checks to determine if an interrupt is still waiting. If an interrupt is received at this time, the test outputs an unexpected interrupt error message. Any unexpected part of the interface status at the time of the interrupt is also reported by error message. The write request interrupt is tested by sending a space character to the printer creating a write request interrupt. If a write request interrupt is not received within 25 milliseconds, the test outputs an error message that no write request was generated. The read request interrupt test requires that the operator enter a character on the keyboard within one minute. If an interrupt is not received within one minute an error message is generated reporting that no read request was generated within one minute. The DATA SET NOT READY interrupt test requires that the operator set the ON LINE/OFF switch to OFF within one minute. If the interrupt is not received within one minute, then an error message is output. The DATA SET READY interrupt test requires that the operator set the ON LINE/OFF switch to ON LINE within one minute. If the interrupt is not received within one minute, then an error message is output. On all interrupts all interface status bits are checked for the expected values and any unexpected status is reported.

If interrupts are not used, the following message will be output.

OPERATOR HAS SPECIFIED UNIT TEST WITHOUT INTERRUPTS – TEST SKIPPED

12.1.4 **TEST 4 – CASSETTE TEST.** This test exercises all functions of the cassette drives. If the unit under test is a 733 KSR, the following message will be output.

OPERATOR HAS SPECIFIED UNIT UNDER TEST WITHOUT CASSETTES – TEST SKIPPED

For normal execution the following messages will be output during execution of each section of the test.

CASSETTE 1

TESTING LOAD OF CASSETTE.

RECORDING TEST PATTERN ON CASSETTE.

TESTING FAST FORWARD TO END OF CASSETTE.

TESTING REWIND OF CASSETTE

LOADING AGAIN ON CASSETTE.

TESTING PLAYBACK ON CASSETTE.

TESTING BLOCK POSITIONING ON CASSETTE.

TEST OF AUTO DEVICE CONTROL OFF/ON

REWINDING AGAIN ON CASSETTE.

END OF TEST ON CASSETTE 1

CASSETTE 2

TESTING LOAD OF CASSETTE.

RECORDING TEST PATTERN ON CASSETTE.

TESTING FAST FORWARD TO END OF CASSETTE.

TESTING REWIND OF CASSETTE

LOADING AGAIN ON CASSETTE.

TESTING PLAYBACK ON CASSETTE.

TESTING BLOCK POSITIONING ON CASSETTE.

TEST OF AUTO DEVICE CONTROL OFF/ON

REWINDING AGAIN ON CASSETTE.

END OF TEST ON CASSETTE 2

CASSETTE COMPATIBILITY TEST.

EXCHANGE TAPES IN CASSETTES 1 AND 2

The test waits for an operator input on the interactive device before proceeding.

LOADING BOTH CASSETTES.

VERIFYING CASSETTE 1

VERIFYING CASSETTE 2

REWINDING BOTH CASSETTES.

12.1.5 **TEST 5 – KEYBOARD TEST.** The Keyboard Test requires that the operator type all printable characters on the keyboard and verify that the character typed is printed. When the return key is depressed the entire string is printed on the 733 as it was entered.

KEYBOARD TEST **** TYPE CHARACTERS ON THE

KEYBOARD AND VERIFY THEY ARE PRINTED ON THE

PRINTER. AN AUTOMATIC CARRIAGE RETURN AND

LINE FEED WILL HAPPEN EVERY EIGHTY CHARACTERS

DEPRESS RETURN KEY TO EXIT TEXT

12.2 OPERATOR INTERFACE

TST733 runs under the control of DOCS on a 990 computer. Refer to Section III of this manual for a detailed description of DOCS.

12.2.1 **AVAILABLE VERBS.** In addition to the verbs supplied by the DOCS package, TST733 provides the verbs listed in table 12-1.

Table 12-1. TST733 Verbs

Verb	Function
IT	Initialize Test
EA	Execute all Tests
ET	Execute Selected Test
LT	Loop on Selected Test
TT	Execute Tape Test

12.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by DOCS, TST733 asks the questions listed in table 12-2.

Table 12-2. TST733 Initialization Questions

Question	Comment
ENTER 733 INTERFACE CRU BASE, DEFAULT = 0000	None
DO YOU WANT TO TEST INTERRUPTS? DEFAULT = 01	When the answer is no (0), the next question will be omitted
ENTER 733 INTERFACE INTERRUPT LEVEL, DEFAULT = 06	None
IS THE UNIT UNDER TEST A 733 ASR ? DEF = 01	When the answer is yes (01), the following message will be output
INSERT 2 SCRATCH CASSETTES	This is a reminder to prevent writing to cassettes inadvertently left in the tape transports.

12.2.3 **PROGRAMMER-PANEL INITIALIZATION**. The programmer-panel version of TST733 asks:

IDLE ON ERRORS?

followed by the questions in table 12-2.

12.3 SYSTEM REQUIREMENTS

The TST733 will run on a 990 Computer with 16K bytes of memory and a Model 733 ASR Data Terminal kit, Part Number 0945162-0001 or a Model 733 KSR Data Terminal kit, Part Number 0945161-0001.

12.4 ERROR MESSAGES

TST733 error messages are listed in table 12-3.

Table 12-3. TST733 Error Messages

Error Number	Error Message	Comments
1	INTERFACE XMIT BIT SET	Interface indicates data transmission to data terminal still in progress.
2	INTERFACE TIMING ERROR SET	One or more characters have been lost. Data not taken from the interface by the computer before new data sent by the terminal.
3	INTERFACE WRITE REQUEST SET	Interface is ready to receive data from the computer when it should not be.
4	INTERFACE READ REQUEST SET	Interface is ready to send data to the computer when it should not be.

Table 12-3. TST 733 Error Messages (Continued)

Error Number	Error Message	Comments
5	INTERFACE INTERRUPT BIT SET	Write request, read request or a change in terminal ready status may have set this bit.
6	INTERFACE DATA SET READY BIT SET	
7	ERROR XMIT BIT NOT SET	Failed to indicate transfer in progress between interface and data terminal.
9	INTERFACE WRITE REQUEST NOT SET	Interface failed to indicate that it is ready to receive another character.
10	INTERFACE INTERRUPT BIT NOT SET	Bit failed to set on write request read request or new status of data set ready.
11	INTERFACE TIMING ERROR NOT SET	Interface failed to report data lost during transfer to data terminal.
12	INTERFACE READ REQUEST NOT SET	Failed to indicate to computer that interface has data ready to take.
14	FAILED TO RECEIVE DATA AFTER PLAYBACK ON.	Data not received from cassette.
15	FATAL ERROR, PC=>	
16	PLAYBACK OFF FROM COMPUTER DID NOT STOP INPUT OF CHARACTERS.	
17	BLOCK POSITIONING ERROR.	Cassette failed to advance or reverse to the correct block.
18	AUTO DEVICE CONTROL OFF DID NOT DISABLE PLAYBACK ON.	Automatic device control is not working on cassette.
19	FAILED TO RECEIVE STATUS AFTER STATUS REQUEST	Requested status from terminal and terminal did not reply.
20	INPUT CHARACTER FROM CASSETTE DID NOT MATCH EXPECTED CHARACTER	
0	EXPECTED XX	Continuation to other messages
	RECEIVED XX	Continuation to other messages
21	TERMINAL STATUS NOT STORED	
22	DATA RECEIVED AFTER C/R ON BLOCK FORWARD.	Carriage return is the last character in the block. Check that tape record is in line format.

Table 12-3. TST733 Error Messages (Continued)

Error Number	Error Message	Comments
23	INTERFACE DATA SET READY NOT SET	Data terminal is not on line.
25	REWIND TOOK TOO LONG	1 minute maximum for rewind.
27	WARNING DATA CARRIER DETECT NOT SET	This signal may be required by the device system device service routine. DX10 does require that this bit be set.
28	UNEXPECTED INTERRUPT	Interrupt condition failed to clear.
29	NO WRITE REQUEST INTERRUPT	Interface failed to generate an interrupt after sending a character to the data terminal.
30	NO READ REQUEST WITHIN ONE MINUTE	Operator failed to enter a character or interface did not generate an interrupt when received.
31	NO NEW STATUS FROM DATA TERMINAL NOT READY	Operator failed to set on line switch to off within one minute or action failed to generate an interrupt.
32	NO NEW STATUS FROM DATA TERMINAL READY	Operator failed to set on line switch to on within one minute or action did not generate an interrupt.
33	INTERFACE DATA CARRIER DETECT SET	
34	INTERFACE REVERSE CHANNEL RECEIVE NOT SET	
35	INTERFACE REVERSE CHANNEL RECEIVE SET	
36	PRINTER NOT READY	Unexpected status from data terminal.
37	PRINTER READY	Unexpected status from data terminal.
38	RECORD NOT READY	Unexpected status from data terminal.
39	RECORD READY	Unexpected status from data terminal.
40	CASSETTE 2 NOT AT BEGINNING OR END	Unexpected status from data terminal.
41	CASSETTE 2 AT BEGINNING OR END	Unexpected status from data terminal.

Table 12-3. TST733 Error Messages (Continued)

Error Number	Error Message	Comments
42	CASSETTE 1 NOT AT BEGINNING OR END	Unexpected status from data terminal.
43	CASSETTE 1 AT BEGINNING OR END	Unexpected status from data terminal.
44	NO PLAYBACK ERROR	Unexpected status from data terminal.
45	PLAYBACK ERROR	Unexpected status from data terminal.
46	PLAYBACK NOT READY	Unexpected status from data terminal.
47	PLAYBACK READY	Unexpected status from data terminal.

SECTION XIII

990 LINE PRINTER TEST (LPTEST)

13.1 INTRODUCTION

The LPTEST tests models 101, 102, 301, 306, 500, 501, and 588 line printers on a 990/10 minicomputer or 990/4 microcomputer system. LPTEST consists of three subtests: test 1 is an interface test, test 2 is an interactive test, and test 3 is a sliding pattern test.

13.1.1 TEST 1. This test checks to verify that the line printer interface is working properly.

13.1.2 TEST 2. This test checks the basic and special features of the line printer such as power on/off, form feed, printer bell, line buffering, etc.

13.1.3 TEST 3. This test checks the line printer with a ripple dump test.

13.2 OPERATOR INTERFACE

LPTEST runs under the DOCS (see Section III for a detailed explanation of the DOCS versions).

13.2.1 AVAILABLE VERBS. LPTEST supplies the verbs listed in table 13-1.

13.2.2 STANDARD INITIALIZATION. In addition to the initialization questions required by the DOCS, LPTEST asks the initialization questions listed in table 13-2.

13.2.3 PROGRAMMER-PANEL INITIALIZATION. In addition to the question IDLE ON ERRORS? the programmer-panel version asks the questions listed in table 13-2.

Table 13-1. LPTEST Verbs

Verb	Meaning
IT	Initialize test
E1	Execute test 1
E2	Execute test 2
E3	Execute test 3
EA	Execute all

13.3 SYSTEM REQUIREMENTS

In addition to the hardware required by DOCS, one of the following line printers is required: Centronix model 101, 102, 301, 306, 500, 501 or model 588 line printer kit.

13.4 ERROR MESSAGES AND NUMBERS

Table 13-3 lists the messages output by the LPTEST.

Table 13-2. LPTEST Initialization Questions

Messages	Comment
AC LINE FREQ 50 OR 60HZ - DEFAULT = 60HZ -	Select the line frequency which corresponds to the local ac power line.
PRINTER MODEL (101, 102, 301, 306, 500, 501 or 588) 0306 -	Enter the line printer model number if it differs from the default, otherwise enter a space.
LP UNDER TEST CRU BASE ADDRESS DEFAULT = 0060 -	Enter the CRU base address if it differs from the default, otherwise enter a space.
LP UNDER TEST INT LEVEL DEFAULT = 000E -	Enter the line printer interrupt level if it differs from the default, otherwise enter a space.

Table 13-3. LPTEST Error Messages and Numbers

Number	Message
1	ERROR 1 REQ- 1
2	ERROR 2 DSR = 0
3	ERROR 3 INT = 1
4	ERROR – NO WRQ INT AFTER CHAR
5	ERROR – DSR INT WITH DTR RESET
6	INT ERROR- BAUD RATE CHECK ABORTED
7	INT ERROR – LINE/MIN RATE ABORTED
8	DSR-1 AFTER DESELECT
9	DSR-0 AFTER SELECT – PRESS ON LP
A	ERROR – DSR = 0 AFTER SELECT
B	ERROR – DSR-1 AFTER POWER LOSS
C	ERROR TRANSMITTING DID NOT SET AFTER LDCR
D	ERROR WRQ DID NOT RESET
E	ERROR XMTING DID NOT RESET
F	ERROR – NO DSR INT AFTER CHAR
10	ERROR DSR NOT = 0 AFTER SENDING CHAR
11	ERROR DID NOT GET DSR INT AFTER PRINTING
12	ERROR DSR DID NOT SET AFTER PRINTING

SECTION XIV

913 VIDEO DISPLAY TERMINAL TEST (CRT913)

14.1 INTRODUCTION

The CRT913 tests the keyboard, screen, and controller of the 913 VDT. This test consists of four parts. Part 1 tests the VDT controller by checking terminal ready and V-SYNCH, by sending characters in loop back mode and by checking for errors. Part 2 tests the VDT memory and memory controller by writing several different patterns to memory, then reading them back. Part 3 tests the Cursor addressing by doing various cursor moves and then reading and checking the cursor column and line addresses. Part 4 is an interactive test that verifies correct operation of the screen and keyboard.

14.2 OPERATOR INTERFACE

CRT913 runs under the DOCS. Refer to Section III for a detailed explanation of the DOCS characteristics. CRT913 requires 16K bytes of memory.

14.2.1 AVAILABLE VERBS. CRT913 supplies the verbs listed in table 14-1.

Table 14-1. CRT913 Verbs

Verb	Function
IT	Initialize test
E1	Execute part 1 – interface test
E2	Execute part 2 – memory test
E3	Execute part 3 – cursor address test
E4	Execute part 4 – interactive test
L1	Loop on part 1
L2	Loop on part 2
L3	Loop on part 3
EA	Execute parts 1-4 in order
LA	Loop on parts 1-3 in order
CU	Cursor move verb
CD	Modify the CRT memory

14.2.2 STANDARD INITIALIZATION. In addition to the initialization questions asked by the DOCS, CRT913 asks the questions in table 14-2.

14.2.3 **PROGRAMMER-PANEL INITIALIZATION.** In addition to the question IDLE ON ERRORS? the programmer-panel version of CRT913 asks the questions in table 14-2.

The diagnostic test is performed after the last question is answered.

If no errors are detected with the halt-on-errors option selected, the diagnostic will enter the interactive section and idle. The operator should then verify that the test LED is lighted. Press the HALT, then the RUN switch to proceed. The test LED should now be off. Before continuing, disconnect any of the VDT cables, press the HALT, then the RUN switch. If the terminal ready bit is not a 1, an error message number will be displayed on the front panel. The operator should reconnect the VDT cables and verify that the VDT screen is blank.

Press the HALT, then the RUN switch to turn off the cursor. Press the HALT, then the RUN switch to display the ripple dump pattern. Press the HALT, then the RUN switch to display the following set of characters on the VDT, one character at a time:

@ – A – B – C – D – E – F – G – H – I – J – K – L – M – N – O –

P – Q – R – S – T – U – V – W – X – Y – Z – [– / –] – Δ – _ –

– ! – " – – $ – % – & – '. – (–) – * – + –, – – – . – / –

0 – 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – : – ; – < – = – > – ? –

Press the HALT, then the RUN switch to proceed to the next character. After the last character is displayed, the diagnostic enters the keyboard test. In this part, the operator should depress the VDT keyboard keys in the order that is displayed on the VDT screen. The character received is checked and an error message number is displayed if the character is not the character expected.

NOTE

To skip this part of the test, depress key number 59 (delete line) on the VDT keyboard. This can be done before any of the first 10 keys have been depressed, but not after.

After key number 59 is pressed, the operator should depress any key to cause an interrupt. If an interrupt does not occur within 30 seconds, an error occurs. The operator should then depress the repeat key and any other key for 4 seconds. A line of characters should be displayed across the top for the screen to verify that the repeat key is working. The beeper should be audible for the repeat key test. Press the HALT switch to end the test.

Table 14-2. CRT913 Initialization Questions

Number	Question
1	CRT UNDER TEST CRU BASE ADDRESS DEFAULT = 00C0 -
2	CRT UNDER TEST INTERRUPT DEFAULT = 000B -

14.3 SYSTEM REQUIREMENTS

In addition to hardware required by the DOCS, a Model 913A Video Display Terminal Kit (Part Number 975070-0013) is required.

14.4 ERROR MESSAGES AND NUMBERS

Table 14-3 lists the messages output by CRT913.

Table 14-3. CRT913 Error Messages and Numbers

Number	Message
1	TERMINAL READY = 1
2	GOT XXXX V-SYNC PULSES IN 100 CLOCK PERIODS
3	KB DATA READY, DID NOT RESET AFTER KB BACK
4	KB DATA READY, DID NOT SET 16 MILLISECONDS AFTER SENDING LOOPBACK CHAR
5	LOOPBACK DATA SENT = XX DATA RECEIVED = XX
6	PARITY ERROR ON CHAR XX
7	CRT MEMORY ERROR
8	CURSOR ERROR
9	PARITY ERROR ON KB TEST
A	DATA COMPARISON ERROR ON KB TEST
B	EXPECTED INT. DID NOT OCCUR
C	UNEXPECTED INT. OCCURRED
D	TERMINAL READY = 0

SECTION XV

MODEL DS31/DS32 DISK AND CONTROLLER TEST (DSKM3X)

15.1 INTRODUCTION

The DSKM3X runs on a 990 minicomputer equipped with a DS31/DS32 disk drive and TILINE® disk controller (TDC). DSKM3X tests the disk drive and TDC and consists of 9 parts:

Part Number	Name
1	Quick controller test
2	Quick disk and controller test
3	Addressing test
4	Controller special tests
5	Memory addressing test
6	Jitter test
7	Independent seek test
8	Media test
9	Interactive test

The default test values of disk TILINE address, disk interrupt level, unit under test, and other available units are presented to the operator at the beginning of the test. These defaults may be changed by the operator, and the changed value will become the new default.

For execution of part 1 a disk unit is not needed because part 1 tests only the controller. If errors occur in this part of the test, there is a high probability the other parts will not run correctly. It takes approximately 11 to 12 minutes to run parts 1 through 5. If errors occur, it will take longer. Because part 3 takes approximately 5 minutes, it is broken into 3 sections.

15.2 OPERATOR INTERFACE

DSKM3X runs under the DOCS (see Section III for a detailed explanation of the DOCS versions).

15.2.1 AVAILABLE VERBS. In addition to the verbs supplied in the DOCS, DSKM3X supplies the verbs listed in table 15-1.

15.2.2 STANDARD INITIALIZATION. In addition to the questions required by DOCS, DSKM3X also asks the questions listed in table 15-2.

®TILINE is a registered trademark for Texas Instruments' High Speed Asynchronous Multiuser Bus.

Table 15-1. DSKM3X Verbs

Verb	Function
IT	Initialize test
EA	Run tests 1, 2, 3, 4, 5, 6, 7, 8, 9 once
LA	Run tests 1, 2, 3, 4, 5, 6, 7, 8, and loop
E1-E9	Run an individual test once
L1-L8	Run an individual test and loop
IC	Issue command
IM	Issue multiple commands
LO	Loop on multiple commands
DC	Display the U.T.C. status
SR	Store registers
CD	Compare two blocks of data
FD	Format the whole disk

Table 15-2. DSKM3X Initialization Questions

Question	Function
DISK TILINE ADDRESS D-F800-	Disk TILINE address
DISK INTERRUPT LEVEL D-0D-	Interrupt level of disk
UNIT TO TEST 00-	Unit to be tested
OTHER AVAILABLE (0-3 TERMINATE WITH FF)FF-	Other units that may be tested

15.2.3 **PROGRAMMER-PANEL INITIALIZATION.** In addition to the question IDLE ON ERRORS? the programmer-panel version will ask the questions listed in table 15-2.

15.3 SYSTEM REQUIREMENTS

The DSKM3X test requires:

- A 990 minicomputer with 12K of memory and TILINE
- A disk kit: TILINE disk, Diablo 31 or 32 disk unit

15.4 ERROR MESSAGES AND NUMBERS

Table 15-3 lists the error messages output by DSKM3X.

Table 15-3. DSKM3X Error Messages and Numbers

Number	Text	Explanation
none	990/10 DIABLO 31/32 DISK DIAGNOSTIC VER MM/YY *X	Header message
none	DISK TILINE ADDRESS D-F800	Disk TILINE address
none	DISK INTERRUPT LEVEL D-0D	Interrupt level of disk
none	UNIT TO TEST 00	Unit to be tested
none	OTHER AVAILABLE (0-3 TERMINATE WITH FF)FF	Other units that may be tested
none	PART 1	Header message
none	PART 1 DONE	Header message
none	ERROR COUNTS DATA = 0000 TIMING = 0000 STATUS = 0000	Header message
none	PART 2	Header message
none	PART 2 DONE	Header message
none	PART 3	Header message
none	PART 3A DONE	Header message
none	PART 3B DONE	Header message
none	PART 3 DONE	Header message
none	PART 4	Header message
none	PART 4 DONE	Header message
none	PART 5	Header message
none	PART 5A DONE	Header message
none	PART 5 DONE	Header message
none	PART 6	Header message
none	WAS PART 5 RUN LAST?	Question in interactive part of test

Table 15-3. DSKM3X Error Messages and Numbers (Continued)

Number	Text	Explanation
none	HAS SYSTEM POWER BEEN CYCLED?	Question in interactive part of test. Action required, turn disk unit off and on
none	PART 6 DONE	Header message
none	PART 7 DONE	Header message
none	PART 8 DONE	Header message
none	PART 9 DONE	Header message
none	LOOP COUNT = XXXX	Loop count at end of test

Part 1 Error Messages

1	STORE REG ERROR
2	ERROR R6=0 STATUS SHOULD BE C0X0 WAS XXXX
3	R7 SHOULD BE A100 WAS XXXX AFTER 10 RESET
4	LOCK OUT NOT SET AFTER 2 READS
5	UTC REG DATA TEST WAS EXP REG XXXX XXXX XX XXXX XXXX XX

Part 2 Error Messages

6	ERROR ON SEEK TRACK ADDR EXP = XXXX REC = XXXX
7	ERROR ON SEEK SECTOR ADDR EXP 0 REC XX
8	ERROR ON SEEK CRC CHAR EXP = XXXX REC = XXXX

Part 3 Error Messages

9	ERROR PART 3A DATA EXP XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
A	ERROR PART 3B TRACK ADDR XXXX DATA REC XXXX XXXX XXXX XXXX XXXX

Table 15-3. DSKM3X Error Messages and Numbers (Continued)

Number	Text
B	PART 3C FORMAT ERROR FORMAT EXPECTED -XXXX FORMAT READ - XXXX
C	PART 3C DATA ERROR DID A READ OF 200 WORDS TO BUFFER AT XXXX XXXX WORDS WERE WRONG
Part 4 Error Messages	
D	ERROR DID COMMAND WITH NO UNIT SEL R7 STATUS EXP A801 REC XXXX
E	ERROR DID COMMAND WITH NO UNIT SEL R7 STATUS EXP A801 REC XXXX
F	ERROR DID STORE REG WITH WC = XX WORDS TRANSFERRED XX
10	ERROR PART 4 IE TEST STATUS EXP = A801 REC = XXXX
11	ERROR PART 4 SE TEST STATUS EXP = A802 REC = XXXX
12	ERROR PART 4 DE TEST STATUS EXP = A840 REC = XXXX
13	ERROR PART 4 TO TEST STATUS EXP = A804 REC = XXXX
14	ERROR PART 4 SI TEST R0 STATUS EXP = 04XXXX REC XXX R7 STATUS EXP = A801 REC = XXXX
15	ERROR PART 4I DID WRITE WITH SC = 0 BUT DATA NOT 0
16	ERROR PART 4 BUSY FLAT SET DATA (MSB SHOULD BE 0) XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
17	TILINE TIMEOUT ERROR R7 STATUS EXP = A820 REC = XXXX
18	RATE ERROR R7 STATUS EXP = A008
19	READ AMP RECOVERY FAILURE R7 STATUS EXP = C800 REC = XXXX

Table 15-3. DSKM3X Error Messages and Numbers (Continued)

Number	Text
Part 5 Error Messages	
2E	PART 5 DATA NOT EQUAL TO ADDRESS ADDRESS = XXXX DATA = XXXX
2F	COMPARISON ERROR IN MAP MEMORY MAP ADDRESS = XXXX MAP BIAS = XXXX TILINE ADDRESS = XXXXXX EXPECTED DATA = XXXX DATA READ = XXXX
Part 7 Error Messages	
30	DATA ERROR EXPECTED DATA >00FF RECEIVED DATA = XXXX

NOTE

Units are selected as follows:

Unit	Sel
0	08
1	04
2	02
3	01

NOTE

The disk command codes and the disk surface designators are located in the same words. The command codes are located in the left bytes and the surface designators are located in the right bytes. The commands and their identifying codes are shown in the following list.

Com Left Byte	Command
00	STORE REGISTERS
01	WRITE FORMAT
02	READ DATA
03	WRITE DATA
04	UNFORMATTED READ
05	UNFORMATTED WRITE
06	INDEPENDENT SEEK
07	RESTORE

Part 8 Error Messages	
1A	PART 8 LOOP ERROR XX ERRORS IN A TRIES

Table 15-3. DSKM3X Error Messages and Numbers (Continued)

Number	Text
Part 8 Error Messages (Cont'd)	
1B	ERROR PART 8 DID FORMAT WITH WC = XXXX SR = XX & DATA = XXXX DATA READ = XXXX FROM CYL = XXXX SURE = XX
Part 9 Error Messages	
1C	ERROR DATA READ XXXX EXP XXXX FROM SURE XX CYL XXXX
1D	ERROR WP TEST DISK STATUS EXP 2000 REC XXXX
Command Issuer Routine Error Messages	
28	*ERROR* THE UTC DID NOT RESPOND IN 20 SEC
29	*ERROR* ALL ATTENTION LINES WERE NOT SET 3 SEC AFTER DOING A SEEK OR RESTORE COMMAND
2A	UNEX DISK INT AT XXXX
Status Checker Routine Error Messages	
2B	NR BIT FOR UNIT XX WAS NOT SET WHILE SEEKING
2C	ERROR ON SEEK OR RESTORE DISK STATUS UTC REG. UNIT X REG 7 = XXXX UNIT = X REG 0 = XXXX DISK STAT COMM SA S/R R A CYL A BYTE C MEM AD SEL CONT STATUS XXXX 0X XX XX XX XXXX XXXX XXXXXX XX XXXX
2D	**STATUS ERROR CONTROLLER STATUS COMP = X ERR = X IDLE = X U.T.C. REG. DISK STAT COMM SA S/R R A CYL A BYTE C MEM AD SEL CONT STATUS XXXX 0X XX XX XX XXXX XXXX XXXXXX XX XXXX DISK STAT COMM SA S/R R A CYL A BYTE C MEM AD SEL CONT STATUS XXXX 0X XX XX XX XXXX XXXX XXXXXX XX XXXX

SECTION XVI

16 I/O TTL MODULE TEST (IO16)

16.1 INTRODUCTION

The IO16 tests the 16 I/O TTL data module (part number 945145). The test consists of two parts. In Part 1, all possible patterns are transmitted, received, and compared for correctness. A test connector is used to loop the output lines to the input lines. In Part 2, modules with an interrupt feature are tested by simulating an external device interrupt using a test connector. Part 2 is only executed for modules with the interrupt feature.

16.2 OPERATOR INTERFACE

IO16 runs under the DOCS (see Section III for a detailed explanation of the DOCS).

16.2.1 AVAILABLE VERBS. IO16 supplies the verbs listed in table 16-1.

16.2.2 STANDARD INITIALIZATION. In addition to the initialization questions required by the DOCS, IO16 also asks the questions listed in table 16-2.

16.2.3 PROGRAMMER-PANEL INITIALIZATION. In addition to the question IDLE ON ERRORS? the programmer-panel version of IO16 asks the questions listed in table 16-2.

16.3 SYSTEM REQUIREMENTS

IO16 requires the unit under test, a 16 I/O TTL module, in addition to the hardware required by the DOCS. Also required is a 16 I/O TTL Data Module test connector for the model under test (part number 946760).

16.4 ERROR MESSAGES

Table 16-3 lists the messages output by IO16.

Table 16-1. IO16 Verbs

Verb	Function
IT	Initialize test
E1	Execute Part 1, pattern test
E2	Execute Part 2, interrupt test
EA	Execute Parts 1 and 2
L1	Loop on Part 1
L2	Loop on Part 2
LA	Loop on Parts 1 and 2

Table 16-2. IO16 Initialization Questions

Number	Message	Explanation
1	16 I/O CRU BASE DEFAULT = 0060 -	The CRU base is determined by the chassis slot in which the module is placed.
2	16 I/O MODEL NUMBER DEFAULT = 01-	The model number is determined by jumper wiring on the board.
3	16 I/O INTERRUPT LEVEL DEFAULT =0E-	(Only for models 2, 3, 5, and 6 that have the interrupt feature.) The interrupt level is determined by back panel wiring.

NOTE

The last question is asked only if the answer to the previous question is 2, 3, 5 or 6.

Table 16-3. IO16 Messages

Number	Message	Description
none	16 I/O MODULE TEST VERSION #- MM/YY *X	Header message.
none	16 I/O MODULE TEST 1 STARTING	Part 1 start message
none	16 I/O MODULE TEST 1 COMPLETE ERROR COUNT = XXXX ERROR BITS = YYYY	Part 1 complete message. XXXX is number of errors, YYYY is hexadecimal representation of data lines in error.
none	16 I/O MODULE TEST 2 STARTING	Part 2 start message.
none	16 I/O MODULE TEST 2 COMPLETE	Part 2 complete message.
1	COMPARE ERROR WIDTH = XX OUTPUT = YYYY INPUT = ZZZZ	During part 1 the transmitted and received patterns were not equal. In hex, XX is the number of bits transmitted, YYYY is the pattern transmitted, and ZZZZ is the pattern received.
2	INTERRUPT MASK ERROR	In part 2 the module interrupt mask is set to disable interrupts, but the interrupt line is set.
3	EXPECTED INTERRUPT DID NOT OCCUR	In part 2 an interrupt from the module did not occur when expected.
4	INTERRUPT NOT CLEARED	In part 2 the interrupt signal did not clear.
6	UNEXPECTED INTERRUPT	In part 2 an interrupt occurred when not expected.

SECTION XVII

FULL DUPLEX TTY/EIA INTERFACE MODULE TEST (TTYEIA)

17.1 INTRODUCTION

The TTYEIA tests the full duplex TTY/EIA interface module. The TTYEIA consists of four subtests as described in paragraphs 17.1.1 through 17.1.4. All of the tests are executed by either the E1 or the EA verbs.

17.1.1 TEST 1 – BAUD RATE TEST. The baud rate is checked to determine whether or not it is within the allowable range.

17.1.2 TEST 2 – PATTERN TEST. All possible bit patterns within a specified range are transmitted, received, and compared for correctness. The successive setting of the Enable Interrupts, Write Request, Interrupt, and Read Request lines is verified.

17.1.3 TEST 3 – TIMING ERROR TEST. This test sets up a five cycle time limited test clearing only the Write Request line. Checks are made to determine whether or not the Interrupt line is properly set and the Transmit in Progress line is properly cleared. If timing errors occur the Timing Error line is tested to verify that it is on.

17.1.4 TEST 4 – INTERRUPT TEST. This test causes five successive interrupts and checks to determine if expected or unexpected interrupt errors occur.

17.2 OPERATOR INTERFACE

TTYEIA runs under the DOCS (see Section III for a detailed explanation of the DOCS).

17.2.1 AVAILABLE VERBS. TTYEIA supplies the verbs listed in table 17-1.

When the LP verb is specified, the operator selects the test pattern to be used by responding to the following message:

```
LOOP PATTERN DEFAULT = 00 -
```

The desired test pattern is entered as hexadecimal characters.

17.2.2 STANDARD INITIALIZATION. In addition to the initialization questions required by the DOCS, TTYEIA also asks the questions listed in table 17-2.

17.2.3 PROGRAMMER-PANEL INITIALIZATION. In addition to the question IDLE ON ERRORS? the programmer-panel version of TTYEIA asks the questions listed in table 17-2.

17.3 SYSTEM REQUIREMENTS

TTYEIA requires the unit under test, a full duplex TTYEIA interface module, in addition to the hardware required by the DOCS.

17.4 ERROR MESSAGES AND NUMBERS

Table 17-3 lists the messages output by TTYEIA.

Table 17-1. TTYEIA Verbs

Verb	Function
IT	Initialize test
E1 or EA	Execute the test
L1 or LA	Loop on the test
LP	Loop on transmitting and receiving a test pattern

Table 17-2. TTYEIA Initialization Questions

Number	Message	Explanation
1	TTYEIA CRU BASE DEFAULT = 0000 -	The CRU base is determined by the chassis slot in which the module is placed.
2	TTYEIA INTERRUPT LEVEL DEFAULT = 06 -	The interrupt level is determined by pack panel wiring.
3	CPU CLOCK RATE (0 = 120, 1 = 100) DEFAULT = 00-	The clock rate is twice the ac power frequency.

Table 17-3. TTYEIA Messages and Numbers

Number	Message	Explanation
none	TTY/EIA MODULE TEST VERSION # - MM/YY *X	Header message
none	TTYEIA TEST STARTING	Test-start message
none	BAUD RATE = XXXX	XXXX is the measured baud rate in decimal
none	TTYEIA TEST COMPLETE	Test-complete message

Table 17-3. TTYEIA Messages and Numbers (Continued)

Number	Message	Explanation
1	STATUS ERROR EXPECTED = XX RECEIVED = XX	Error between expected and received status, where XX represents the inclusive or sum of the status lines as follows: 01 XMTING 02 TIMERR 04 RCR 08 WRQ 10 RRQ 20 DCD 40 DSR 80 INT
2	INTERRUPT LINE NOT SET	Module interrupt line was not set when expected.
3	RRQ OR WRQ NOT SET IN 250 MS	During transmit/receive, RRQ or WRQ were not set within 250 milliseconds.
4	TIMING ERROR LINE NOT SET	During timing error test, the timing error line was not set.
5	COMPARE ERROR OUTPUT = XX INPUT = XX	During transmit/receive cycle, the transmitted and received patterns were not equal. XX is the hexadecimal pattern.
6	UNEXPECTED TTYEIA INTERRUPT	An unexpected interrupt was caused by the module.
7	EXPECTED INTERRUPT DID NOT OCCUR	During interrupt test, an interrupt from the module did not occur when expected.
8	BAUD RATE = XXXXXX NOT IN SPECIFIED RANGE	Baud rate test was not in range for 75, 110, 300, 1200, 2400, 4800 or 9600 baud.
9	UNEXPECTED INTERRUPT LEVEL = XX	An interrupt not caused by the module occurred on the test level. The test is aborted.
A	XMTING NOT CLEARED IN 250 MS	During transmit/receive cycle in timing error test, the XMTING line was not reset in 250 milliseconds.

SECTION XVIII

PROM PROGRAMMER INTERFACE TEST (PROMPG)

18.1 INTRODUCTION

The PROMPG tests the hardware located on the PROM programmer interface card. The diagnostic will operate with the PROM programmer interface card in any CRU slot in a 990 computer. The PROMPG consists of four parts as described in paragraphs 18.1.1 through 18.1.4.

18.1.1 **PART 1 – REGISTER TEST.** This test checks the Data, Address, and Pulse Width registers.

18.1.2 **PART 2 – LOGIC TEST.** This test checks the logic on the PROM programmer interface card.

18.1.3 **PART 3 – SCOPE LOOP TEST.** This test checks for the presence of the programming pulse on the PROM programmer interface card.

18.1.4 **PART 4 – HOT MOCK-UP TEST.** This test compares the output to the input by looping back the address that is sent to the personality card.

18.2 OPERATOR INTERFACE

PROMPG runs under the DOCS (see Section III for a detailed explanation of DOCS).

18.2.1 **AVAILABLE VERBS.** PROMPG supplies the verbs listed in table 18-1.

18.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by the DOCS, PROMPG also asks the questions listed in table 18-2.

18.2.3 **PROGRAMMER-PANEL INITIALIZATION.** In addition to the question IDLE ON ERRORS? the programmer-panel asks the questions listed in table 18-2.

NOTE

If the entered CRU base is incorrect, the CRU base default will be displayed after the interrupt question.

18.3 SYSTEM REQUIREMENTS

PROMPG requires a PROM programmer interface module in addition to the hardware required by the DOCS. Although not required for diagnostic execution, a Tektronix 475 dual trace oscilloscope is recommended for use with test part 3. Part 3 does not produce any error messages and is only intended to drive an oscilloscope test of the circuits. Part 4 is a special factory test and requires Special Adapter (part number 944925-0001 or-0004) be inserted in the Personality Module.

18.4 ERROR MESSAGES AND NUMBERS

Table 18-3 lists the messages output by PROMPG.

Table 18-1. PROMPG Verbs

Verb	Function
IT	Initialize test
E1	Execute part 1, register test
E2	Execute part 2, logic test
E3	Execute part 3, scope loop
E4	Execute part 4, special factory test
EA	Execute parts 1 and 2 in order
L1	Loop on part 1
L2	Loop on part 2
L4	Loop on part 4
LA	Loop on parts 1 and 2 in order

Table 18-2. PROMPG Initialization Questions

Question	Comment
PROM PROGRAMMER CRU BASE ADDRESS DEFAULT = 0020 -	The CRU base address is determined by the chassis location.
PROM PROGRAMMER INT. LEVEL DEFAULT = 0015 -	The interrupt level is determined by the chassis location and backplane wiring.

Table 18-3. PROMPG Error Messages and Numbers

Number	Message
1	*ERROR* ADDRESS REGISTER
2	*ERROR* DATA REGISTER
3	*ERROR* PULSE WIDTH REGISTER
4	*ERROR* BUSY SET BEFORE GO IS GIVEN
5	*ERROR* BUSY DID NOT SET AFTER GO IS GIVEN
6	*ERROR* BUSY DID NOT RESET AT END OF PROGRAMMING CYCLE
7	*ERROR* EXPECTED INTERRUPT DID NOT OCCUR
8	*ERROR* TIMING ERROR BIT SET
9	*ERROR* NO TIMING ERROR WHEN GO RESETS DURING PROGRAMMING CYCLE
A	*ERROR* RESET DID NOT CLEAR UPPER BYTE OF ADDRESS REGISTER
B	*ERROR* RESET DID NOT CLEAR LOWER BYTE OF ADDRESS REGISTER
C	*ERROR* RESET DID NOT CLEAR DATA REGISTER
D	*ERROR* RESET DID NOT CLEAR PULSE WIDTH REGISTER
E	LOOP BACK ERROR
F	PROM PROGRAMMER UNEXPECTED INTERRUPT

SECTION XIX

MODEL 979/979A TAPE TEST (TAPTST)

19.1 INTRODUCTION

The TAPTST tests the TILINE Magnetic Tape Controller (TMTC) and the associated tape units.

The test portion of the TMTC and tape test is composed of 10 parts:

Part Number	Name
0	Controller test
1	Rewind and erase test
2	Basic write/read test
3	Write/backspace/read test
4	Creeping write/backspace/read test
5	Even/odd write/read test with skip
6	Special movement test
7	Creep and read/write ramp test
8	Random data check with varying length records
9	Power fail test

19.1.1 GENERAL NOTES. Whenever any part of the test finds an error, it checks the error print flag and, if it is set, prints an error message. In all cases, the appropriate error count is incremented.

Whenever any part starts executing, it checks the print header flag and, if it is set, prints the part number. Whenever a part completes and the print flag is set, a message saying that part is done is printed.

Whenever a command is issued to the TMTC, the register 7 status and the tape status bits are checked, unless the status check flag is set to zero.

Note that most tapes are heavily used (and therefore worn) near the front of the reel. Many data errors can be eliminated by using new tape or by placing a new BOT marker approximately 30 to 50 feet from the start of the reel and cutting off the worn tape. This is a precaution to eliminate media errors.

As a precaution to eliminate false errors, it is advisable to clean the heads on the tape handler before starting the test.

19.1.2 CONTROLLER TEST. Part 0 is a quick test of the TMTC. A tape unit does not have to be present because no commands are issued to a tape unit. If errors occur during the running of this part, there is a problem with the controller and the rest of the test probably will not run correctly.

19.1.3 REWIND AND ERASE TEST. Part 1 tests the ability of the controller and tape unit to perform the rewind and erase commands.

19.1.4 BASIC WRITE/READ TEST. Part 2 tests the ability of the controller and tape unit to perform the basic writing and reading of data.

19.1.5 WRITE/BACKSPACE/READ. Part 3 incorporates the backspace command in the basic write/read test.

19.1.6 CREEPING WRITE/BACKSPACE/READ TEST. Part 4 issues the same commands to the tape controller as Part 3 does, however, two consecutive write commands are issued.

19.1.7 EVEN/ODD/WRITE/READ TEST WITH SKIP. Part 5 performs the most extensive write/read operation in all ten parts of the test. This part is executed in two subsections. Section A writes and reads records of various lengths and verifies the data. Section B of Part 5 verifies that partial records can be read correctly.

19.1.8 SPECIAL MOVEMENT TEST. Part 6 of the tape test is broken into 5 sections. Before any of the sections of Part 6 are executed, the tape is moved forward by executing 4 erase commands with a maximum character count. This is done to get to a less worn position on the tape.

Section A writes and reads data of the following character lengths: 16, 17, 32, 33, 64, 65, 128, 129, 256, 257, 512, 513, 1024, 1025, 2048, 2049, 3095, and 3097.

Section B of Part 6 checks the special flag bit (8) of register 6.

Section C of Part 6 verifies that no excess data is read from tape when the data length specified in the issued command is greater than the length of the record. This is done for even and odd length records.

Section D of Part 6 verifies the approximate amount of data that an erase of zero length would destroy. Before this section is run, a flag is examined to see if this is a tape unit that is operating at 1600 bpi. If it is, this section is skipped.

Section E of Part 6 verifies that a write of 0 length does not transfer data to tape.

19.1.9 CREEP AND READ/WRITE RAMP TEST. Part 7 of the tape test is broken into 3 sections. Again the tape is moved forward by executing 4 erase commands with a maximum character count before any of Part 7 is executed.

Section A verifies that write commands can be issued at varying time intervals between issuances. This gives the tape varying time to rest or slow down.

Section B in Part 7 verifies that read commands can be issued at varying time intervals between issuances. This gives the tape varying time intervals to rest or slow down.

Section C of Part 7 verifies that the tape handler does creep in the forward direction if a number of WRITE/BACKSPACE/READ operations are performed. This capability of the tape unit is used if there is a bad spot on the tape during a write operation. The record being written is actually moved down the tape. Section C is run 5 times to verify constant creep on different sections of tape.

19.1.10 RANDOM DATA CHECK WITH VARYING LENGTH RECORDS. Part 8 verifies the capability of the TMTC and tape unit to write and read random data of varying lengths. The last part of part 8 verifies that an EOF can be written on tape.

19.1.11 POWER FAIL TEST. Part 9 verifies that when the system has battery support, a failure of the power supply will not cause data to be written on the tape. Part 9 requires operator intervention, so there are no verbs to loop on Part 9.

NOTE

Do not use this part if the computer does not have battery support. If this part of the test is used without standby power supply, memory will be cleared and the program lost.

When Part 9 starts executing, it asks the operator if the system has battery support. If the operator should respond with a 0 (indicating no battery support), a message is printed telling the operator not to run this part of the test. However, if the operator enters a 1 (indicating battery support), the operator is asked the following question:

HAS SYSTEM POWER BEEN CYCLED?-

The operator should respond with a 0 the first time he encounters this question, indicating that power has not been cycled. After seeing the zero, two records will be written and the head will be positioned between the two records. The following message is printed next, telling the operator to cycle system power and control returns to the verb decoder.

CYCLE SYSTEM POWER

At this time the operator should cycle system power (i.e., turn the 990/10 power off and then on). Of course, when the system is powered up, the series of system initialization questions must be answered. To finish executing Part 9, the verb E9 should be used. The operator should answer yes (=1) to the battery support and system power questions. At this time the test will verify by reading and comparing data that the previously written records on both sides of the head did not get destroyed when system power was cycled. If record 1 was destroyed, the following error message is printed.

PART 9 RECORD 1 ERROR

If record 2 was destroyed, the following error message is printed:

PART 9 RECORD 2 ERROR

The following shows the dialogue of Part 9.

Example 1.

```
VERB?-E9

PART 9
DOES THIS SYSTEM HAVE BATTERY SUPPORT?
-0 (indicates no battery)

DO NOT RUN PART 9 – NO BATTERY SUPPORT –

VERB?-
```

Example 2.

```
VERB? -E9

PART 9
DOES THIS SYSTEM HAVE BATTERY SUPPORT? -1
HAS SYSTEM POWER BEEN CYCLED? -0

CYCLE SYSTEM POWER-
VERB?-

(Cycle System Power)

(DOCS Initialization Questions)
ENTER TILINE BASE ADDRESS-F880
TAPE UNIT TO BE TESTED–(1, 2, 3, 4) -1
OTHER AVAILABLE UNITS–TERMINAL WITH -FF -FF
ENTER INTERRUPT LEVEL IN HEX-C
ENTER ZERO FOR INTERRUPTS AND 1 FOR IDLE SCAN -0
EXECUTE EA VERB? (DEF = 1) -0
VERB? -E9
PART 9
DOES THIS SYSTEM HAVE BATTERY SUPPORT? -1
HAS SYSTEM POWER BEEN CYCLED -1
PART 9 DONE
VERB?-
```

19.2 OPERATOR INTERFACE

TAPTST runs under the DOCS (see Section III for a detailed explanation of the DOCS).

19.2.1 AVAILABLE VERBS. TAPTST supplies the verbs listed in table 19-1.

19.2.1.1 IT Verb. The IT verb is used to initialize the tape parameters. These parameters are (1) tape TILINE address, (2) tape interrupt level, (3) unit to test, (4) other available units, and (5) interrupt or noninterruptible commands.

Table 19-1. TAPTST Verbs

Verb	Function
IT	Initialize test
E0	Execute Part 0
.	.
.	.
.	.
E9	Execute Part 9
EA	Execute all 10 parts
L1	Loop on Part 1
.	.
.	.
.	.
L8	Loop on Part 8

NOTE

There is no verb to loop on Part 9.

LA	Loop on Parts 0-8
CM	Execute Parts 0-8 for all available units and loop
IC	Issue command
IM	Issue multiple
LO	Loop on multiple commands
DC	Display the TMTC status
CD	Compare two blocks of data

To set up the list of available units, enter the number of the unit, 1 through 4. The default for the second unit is FF. This allows the operator to test one unit by entering the unit to be tested and depressing the space bar for other available units. If other units (up to 3) are available, they may be shown. The list is terminated with FF. This list is used and modified by the CM verb.

The following is an example of the IT verb:

```
VERB? -IT

ENTER TILINE BASE ADDRESS-
TAPE UNIT TO BE TESTED -(1, 2, 3, 4)-
OTHER AVAILABLE UNITS–TERMINATE WITH FF-
ENTER INTERRUPT LEVEL IN HEX-
ENTER 0 FOR INTERRUPTS AND 1 FOR IDLE SCAN-
```

19.2.1.2 E0 Through E9 Verbs. Use a verb E0 through E9 to execute a specific part of the test. If the part completes and if the operator typed a 1 to the initialization question:

```
PRINT ON ERRORS?-
```

the number of errors that occurred in part 1 will be printed. The following is an example of the E0 verb:

```
VERB? -E0

PART 0
PART 0 DONE
ERROR COUNTS
TIMING=0000 DATA=0000 STATUS=0000
```

19.2.1.3 L0 Through L8 Verbs. Use the L0 through L8 verbs to loop on Parts 0 through 8 of the tape test. L0 loops on Part 0, L1 loops on Part 1, and so forth. After each time the loop has been completed, the loop count will be printed. The following example is of the L1 verb:

```
VERB?-L1
PART 1
PART 1 DONE
LOOP COUNT=0001
PART 1
PART 1 DONE
LOOP COUNT=0002
PART 1
PART 1 DONE
LOOP COUNT=0003
```

To stop these verbs, cause an interrupt on the interactive device (i.e., enter any character). After each repetition of the loop, the loop count is output to the programmer panel.

19.2.1.4 EA Verb. Use the EA verb to execute all ten parts of the tape test in numerical order.

19.2.1.5 LA Verb. Use the LA verb to loop on Parts 0 through 8 in numerical order. After each time the loop has been completed, the loop count will be printed. To stop this verb, cause an interrupt on the interactive device (i.e., enter any character).

19.2.1.6 CM Verb. The CM verb can be used to execute Parts 0 through 8 with each tape unit as the unit to be tested. It does this by executing Parts 0 through 8, then it takes the next available unit as selected by the IT verb and makes it the unit to test.

The CM verb will run until interrupted and each time it finishes Parts 0 through 8, it will shift the next available unit to be the unit under test and the present unit under test replaces the unit to be tested in the list.

Before each unit is tested, the unit number is printed out in the following format:

```
UNIT TO BE TESTED=02
```

If the operator answered the initialization

```
PRINT ON ERROR?
```

with a 1, the error counts will be printed after each unit is tested.

The following is an example of the CM verb:

```
VERB; -CM
UNIT TO BE TESTED=01
PART 0
PART 0 DONE
PART 1
PART 1 DONE
  .
  .
  .
PART 8
PART 8 DONE
ERROR COUNTS
TIMING=0000 DATA=0000 STATUS=0000
UNIT TO BE TESTED=02
PART 0
PART 0 DONE
PART 1
PART 1 DONE
  .
  .
  .
PART 8
PART 8 DONE
ERROR COUNTS
TIMING-0000 DATA-0000 STATUS-0000
UNIT TO BE TESTED=03
```

After all units have been tested, the test will start over again with the first unit.

19.2.1.7 IC Verb. The IC verb can be used to issue a command to the TMTC from the 990 memory address input by the operator. The command is 8 words long.

Format	VERB? IC ADDR–XXXX
Action	Issue command at ADDR with the print flag ON so any errors are printed.

19.2.1.8 IM Verb. The IM verb can be used to issue multiple commands to the TMTC from the 990 memory.

Format VERB?–IM ADDR–XXXX #COMMANDS–XX

Action Issues number of commands where the command list starts at ADDR. The print flag is set to 1 so any errors will be printed.

19.2.1.9 LO Verb. The LO verb can be used to loop on a sequence of TMTC commands. The commands must each be 8 words long and must be in sequential memory.

Format VERB? –LO ADDR–XXXX #COMMANDS–XX CHECK ST?–X

Action Loop on a series of commands starting at ADDR. The number of commands is #COMMANDS. The status checker flag is set = 'CHECK ST?'.

19.2.1.10 DC Verb. The DC verb can be used to print out or display the status of the TMTC. It reads the TMTC slave registers, then formats and outputs them to the VDT or 733 ASR.

Format

VERB? –DC

TMTC REGS

TAPE STAT	COM	STAT/CT	OFFSET	NO CHAR	MEM AD	UNIT	R7 STAT
0010	06	000000	0000	0000	001EDC	02	A880

19.2.1.11 CD Verb. The CD verb can be used to compare two blocks of data. This verb compares the two blocks of data starting at the two addresses input by the operator for the number of words input. If any miscompares are found, the two addresses and their respective data are printed. The verb then goes on until it is done or another miscompare is found.

19.2.2 STANDARD INITIALIZATION. In addition to the initialization questions required by the DOCS, TAPTST also asks the questions listed in table 19-2.

Table 19-2. TAPTST Initialization Questions

Question	Comment
ENTER TILINE BASE ADDRESS –	
TAPE UNIT TO BE TESTED–(1,2,3,4) –	
OTHER AVAILABLE UNITS – TERMINATE WITH -FF –	
ENTER INTERRUPT LEVEL IN HEX –	
ENTER 0 FOR INTERRUPTS AND 1 FOR IDLE SCAN –	

NOTE

These questions are asked by the IT verb.

19.2.3 **PROGRAMMER-PANEL INITIALIZATION.** In addition to IDLE ON ERRORS? the programmer-panel version of TAPTST asks the questions listed in table 19-2.

After initialization, the test will run approximately 1 minute and 45 seconds until the next operator intervention is required. This time will be longer if errors occur. The error numbers will appear on the programmer panel.

When operator interaction is required, the test is starting Part 9. The 990/10 will go into the idle mode with 0000 hexadecimal on the programmer panel. At this point, the question being asked is:

DOES THIS SYSTEM HAVE BATTERY SUPPORT?

If battery support is present, the 990/10 will go into the idle mode. The question now being asked is:

HAS SYSTEM POWER BEEN CYCLED?

After responding with a NO (0), the operator should wait until the tape handler quits moving and then cycle system power.

The test must be reinitialized after the power has been cycled. After the test is reinitialized, the previous two questions of Part 9 will be asked. The operator should respond with a YES (1) to the second question. The program will finish Part 9, display a loop count on the front panel, and start executing the test over again.

19.3 SYSTEM REQUIREMENTS

The equipment recommended for TAPTST includes (1) a 990 minicomputer with 12K or more of memory, (2) an interactive device, and (3) a Model 979 Tape Unit and Controller (part number 947578-0001).

19.4 ERROR MESSAGES AND NUMBERS

When an error occurs in the execution of any part of the test, an error message will be printed out if the software error print flag is on.

The error message indicates which error has occurred. The number of the error will also appear on the programmer panel. The error messages are listed in table 19-3 in numerical order. It should be noted that the same message text may appear more than once with different error numbers. The reason for this is that some data checking routines are used by different parts of the test.

When an error is found, a message is output if the print error flag was set during initialization. In all cases the appropriate error count is incremented. There are three error counts: one for status errors, one for data errors, and one for timing errors.

Table 19-3. TAPTST Error Messages and Numbers

Number	Message
1-3	CONTROLLER FAILURE
4-7	ERROR X REWIND INTERRUPT TEST
8-A	ATTEMPT TO WRITE XXXX IN XXXX RESULTED IN XXXX BEING RETURNED

Table 19-3. TAPTST Error Messages and Numbers (Continued)

Number	Message
B	ADDR1 = XXXX COUNT = XXXX ADDR2 = XXXX CONT = XXXX
C	COMPARE ERROR-CHAR WRITTEN = XXXX CHAR READ = XXXX
D	DID NOT FILL LAST CHAR WITH FF
E	TOO MANY CHARACTERS WERE TRANSFERRED TO MEMORY
F	OFFSET DID NOT DECREMENT TO ZERO
10	OVERFLOW COUNT WAS 000000 BUT SHOULD HAVE BEEN 00 00 00
11	ERROR PART 6A XXXX MISCOMPARES WITH RECORD LENGTH = XXXX
12	ERROR PART 6B RECEIVED VRC ERROR IN WRITE OPERATION
13	ERROR PART 6B EXPECTED R7 ERROR PATTERN = XXXX RECEIVED = XXXX
14	ERROR PART 6C WROTE RECORD OF XXXX CHARACTERS ATTEMPTED TO READ 160 CHARACTERS DATA TRANSFER DID NOT STOP AFTER XXXX CHARACTERS
15	ERROR PART 6D AN ERASE OF 0 LENGTH DESTROYED LESS THAN 300 CHAR OF A 1600 CHAR RECORD
16	ERROR PART 6D AN ERASE OF 0 LENGTH DESTROYED MORE THAN 700 CHAR OF A 1600 CHAR RECORD
17	ERROR PART 6E A WRITE OF 0 LENGTH TRANSFERRED DATA TO TAPE
18	ERROR PART 7A FAILED TO READ RECORD XXXX
19	ERROR PART 7B FAILED TO READ FOLLOWING RECORDS WITHOUT ERROR RECORD NO XX
1A	PART 7C NO CREEP IN 20 BACKSPACE/WRITE OPERATIONS
1B	PART 7C NEGATIVE CREEP RECORD 2 CREPT BACKWARD AND DESTROYED RECORD 1 IN 12 HEX BACKSPACE/WRITE OPERATIONS

Table 19-3. TAPTST Error Messages and Numbers (Continued)

Number	Message
1C	E8 DATA ERROR XXXX MISCOMPARES WITH DATA LENGTH = XXXX
1D	PART 9 RECORD 1 ERROR DATA WRITTEN = XX DATA READ = XX
1E	PART 9 RECORD 2 ERROR DATA WRITTEN = XX DATA READ = XX
26	DID NOT GET EXPECTED INT WHEN TESTING R7
27	GOT FALSE INT WHEN TESTING R7
28	DID NOT RECEIVE INT IN 17 SEC
29	DID NOT GO IDLE IN 02 MIN
2A	DID NOT REACH BOT ON REWIND IN 4 MIN
2B	OFF LINE BIT DID NOT COME ON DURING REWIND AND UNLOAD IN 4 MIN
2C	** STATUS ERROR TMTC REGS TAPE STAT COM STAT/CT OFFSET NO CHAR MEM AD UNIT R7 STAT 0010 06 000000 0000 0000 001EDC 02 A880 COMMAND ISSUED TAPE STAT COM STAT/CT OFFSET NO CHAR MEM AD UNIT R7 STAT 0000 06 000000 0000 0100 003CB8 02 1000
2D	***END OF TAPE*** OPERATOR MUST START TEST OVER
2E	FAILED TO GET TILINE TIMEOUT WHEN WRITING TO A NON-EXISTENT MEMORY ADDRESS (1FFBFE)

SECTION XX

MEMORY MAPPING TEST (MAPTST)

20.1 INTRODUCTION

The MAPTST tests all of the map logic on the AU1 and AU2 boards. It also checks the microcode ROM bits that control the map logic.

There are two versions of the 990/10 MAP test:

1. Part number 945441-1202 or 1402 is used for Multiwire AU boards.
2. Part number 945441-1203 or 1403 is used for Printed Circuit AU boards.

This test is composed of 6 parts, with a main part of the test that calls each of the 6 parts in order.

Part Number	Name
1	Map file bit test
2	Map interrupt test
3	Address development test
4	Limit logic test
5	Latched map increment test
6	Instruction test

When each part starts executing, MAPTST outputs a starting message. When it completes a loop, it outputs the loop count. When each part completes, a completion message is printed.

When running the test, the operator should verify that each subtest beginning and completion message is output. If a subtest outputs a beginning message but no completion message, there is a problem in the computer, in which case the AU10 diagnostic test should be run.

20.1.1 MAP FILE BIT TEST. Part 1 verifies all of the map file registers by writing various patterns to the map files, then reading the patterns back and checking them.

20.1.2 MAP INTERRUPT TEST. Part 2 checks several special conditions associated with the map status bit and the map interrupt.

20.1.3 ADDRESS DEVELOPMENT TEST. Part 3 verifies that the map adder circuitry is working correctly.

20.1.4 LIMIT LOGIC TEST. Part 4 verifies that the limit adder circuits and the base selection circuitry are working correctly.

20.1.5 LATCHED MAP INCREMENT TEST. There is some special hardware that is used to increment the LMPA register when doing LMF instructions. This part verifies that the increment logic works properly.

NOTE

Part 5 uses all of the available memory past the end of the test and only does a complete test of the increment logic if 1-million words of memory are available.

After the test has determined available memory, it outputs a message telling the operator how much memory is available on the machine. The operator should verify that the correct memory configuration is printed out.

20.1.6 INSTRUCTION TEST. Part 6 should only be run on printed circuit AU boards. This is not a valid test for wire-wrap AU circuit boards. Part 6 verifies that all instructions with a source or destination operand will work correctly when they are preceded by a long distance instruction. This verifies the map logic on the AU1 board and the microcode that controls the map logic.

20.2 OPERATOR INTERFACE

MAPTST is a standalone test. For a detailed explanation of standalone tests, refer to section II in this manual.

Only the standard standalone test options are offered: idle on errors, print headers, or print errors. Consult section II on standalone tests for instructions in selecting these options.

MAPTST must be loaded into low memory (address >100 or less) since Part 5 uses memory to test the LMF instruction. Part 6 also uses memory at location >1000 beyond the middle of Part 6 (i.e., middle of Part 6 plus >1000).

20.2.1 EXECUTION. This test runs without operator intervention. When this test starts executing, it will output the following message:

```
MAPTST 990/10 MAP LOGIC TEST VERSION MM/YY *X
```

where MM/YY *X is the date and revision letter the test was released. This number is used to correlate the listing and the object code.

If the test runs without any errors, the messages will be as follows:

```
START MAP-FILE BIT TEST
BIT TEST COMPLETE

START MAP INTERRUPT TEST
INT TEST COMPLETE

BASE REG & ADDER VERIFICATION TEST
BASE REG TEST COMPLETE
LMF LMPA INCREMENT TEST
THE CURRENT MEMORY CONFIGURATION IS:
ADDRESSES (BYTE)
FROM      TO
000000    010000
LMF TEST COMPLETE
```

INSTRUCTION TEST
INSTRUCTION TEST COMPLETE

TEST COMPLETE LOOP COUNT = 0001

NOTE

Refer to the description of each part of the test for the meaning of the above messages.

After the loop count is printed, it will be output to the programmer panel, and the test will start over. The test will run until the operator halts it. All of the above messages can be suppressed by clearing the header print flag.

If the map test finds an error, it sets the fault light and outputs an error message. The standard error messages can be suppressed by clearing the error print flag.

If an unexpected interrupt occurs, one of the following messages will be output. (1) If a Level 2 interrupt occurs, one of the following messages will be printed:

(error type)
WORK POINTER PC & STATUS AT TIME OF ERROR WP=XXXX PC=XXXX
ST=XXXX

where the error type is one of the following:

ILLEGAL OP
PRIVILEGED ERROR INT

PARITY ERROR
TILINE TIMEOUT

and XXXX is the value at the time of the error.

If a map error occurred, the latched map address will also be output. (2) If an interrupt occurs from a level that is not being used, the following message will be output:

UNEXPECTED INT LEVEL=XX

where XX is the level of the interrupt 03-0F.

(3) If a level 1 interrupt (power fail) occurs, the machine will idle. (4) If a level 0 interrupt (power up) occurs, the following message will be output and the test restarted:

POWER RESTART INITIATED

20.2.2 **EXECUTION WITH A PROGRAMMER PANEL.** If an error is found, the error message number will be displayed on the programmer panel and the fault light will be lighted. If the halt-on-errors option has been selected, the computer will idle. If not, it will continue the test. If an unexpected interrupt occurs, the message:

FFFX

(where X is the interrupt level 2 through F) will be output to the programmer panel, and the computer will idle. If a Level 1 interrupt occurs, the computer will idle. If a Level 0 interrupt occurs, the test will be restarted.

20.3 EQUIPMENT REQUIREMENTS

The equipment required to run this includes (1) a 990/10 minicomputer with mapping, (2) a 911 VDT (913 VDT and 733 ASR/KSR can be selected) and a diagnostic load device.

20.4 ERROR MESSAGES AND NUMBERS

Whenever the test finds an error, it outputs the error message number to the programmer panel. Table 20-1 contains a list of the error message numbers.

NOTE

If an error message is printed in several parts, the subsequent parts of a message will have a message number of zero.

Table 20-1. MAPTST Error Messages and Numbers

Number	Message

Part 1

1 ERROR MAPFILE DATA:

	EXPECTED						READ					
MAP	L1	B1	L2	B2	L3	B3	L1	B1	L2	B2	L3	B3
0	FFE0	FFFF	FFE0	FFFF	FFE0	FFFF	FFE0	FFFF	FFE0	FFFF	FFE0	FFFF
1	FFE0	FFFF	FFE0	FFFF	FFE0	FFFF	FFE0	FFFF	FFE0	FFFF	FFE0	FFFF
2	FFE0	FFFF	FFE0	FFFF	FFE0	FFFF	FFE0	FFFF	FFE0	FFFF	FFE0	FFFF

Part 2 or 6

20 *ERROR*

ADDRESS (IN LISTING) OF ERROR 2080

WORK POINTER PC & STATUS AT TIME OF ERROR WP = 018C PC = 2122 ST = 420F WORKSPACE DATA

0000 FFFF FFFE 002C 0042 0042 218C 0000

21C4 0B1C 21B3 218C 1FA0 218C 2072 C20F

Table 20-1. MAPTST Error Messages and Numbers (Continued)

Number	Message
	Part 2
21	ERROR WHEN DOING LDS WITH MAP FILE IN ROM AREA MAP FILE 2 DATA ACTUAL EXPECTED XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
	Part 3
30	ERROR IN THE MAP FILE 2 BASE REGISTER OR THE MAP ADDER CIRCUITRY MAP FILE 2 L1 B1 L2 B2 L3 B3 AU ADDRESS 0000 0000 0000 0000 0000 0000 0020 LATCHED MAP ADDRESS EXPECTED ACTUAL 000020 000020
	Part 4
41	EXPECTED MAP INT DID NOT OCCUR MAP FILE 2 DATA = FFFF FF99 FFFF FF9A E000 FF9C MEMORY ADDRESS = 2020
42	ERROR * DID 'LDS' 'SETO' DATA STORED USING BASE 1 INSTEAD OF BASE 2 MAP FILE 2 DATA = 1000 009D 0000 009B 0000 009C MEMORY ADDRESS = 0000
43	ERROR UNEX MAP INTERRUPT

Table 20-1. MAPTST Error Messages and Numbers (Continued)

Number	Message
	Part 5
5	LMF ERROR MAP FILE ADDRESS = 003000 MAP FILE 1 DATA ACTUAL EXPECTED 1000 1002 1000 1006 1000 100A 1000 1002 1000 1006 1000 100A

SECTION XXI

DS25, DS50, DS200 DISK TEST (DSKTRI)

21.1 INTRODUCTION

The DSKTRI tests the TILINE disk controller (TDC), the disk pack and the disk drive of the DS25, DS50 and the DS200.

DSKTRI is composed of the following 6 parts:

Part Number	Name
1	Quick TDC test
2	Quick TDC and disk test
3	Addressing test
4	Special TDC test
5	Format cylinder 0
6	Interactive test

21.1.1 QUICK TDC TEST. Part 1 is a quick test of the TDC. A disk does not have to be present because no commands are issued to the disk. If this part of the test fails, there is a problem with the TDC and the rest of the test probably will not run correctly.

21.1.2 QUICK TDC AND DISK TEST. Part 2 is a quick test of the TDC and the disk. Part 2 performs an unformatted write and read operation to test these basic functions, issues a restore command with the status check flag set, and performs a seek test.

21.1.3 ADDRESSING TEST. Because Part 3 runs for several minutes, it has been broken into 4 parts, A, B, C, and D, and progress messages are printed after parts A and B complete, if headers are enabled.

Part A verifies that the TDC can address all of the sectors on one track correctly.

Part B verifies that the TDC can address every track on the disk.

Part C verifies the ability of the disk to perform read operations in very rapid sequence. Part C of the test runs approximately 11 minutes.

Part D verifies the three ID words and the CRC that make up the header of all the sectors on three different tracks.

21.1.4 SPECIAL TDC TESTS. Part 4 contains several tests that check for special conditions in the TDC. The status bits checked in this part are:

Status Bit	Definition
UE	UNIT ERROR
IE	HEADER ERROR
SE	SEARCH ERROR
DE	DATA ERROR
TO	TIME OUT ERROR
SI	SEEK INCOMPLETE
TT	TILINE TIMEOUT
RE	RATE ERROR

21.1.5 FORMAT CYLINDER 0. Part 5 formats cylinder 0 with the sectors per record and word count. This information is checked in Part 6 after a power down/power up sequence.

21.1.6 INTERACTIVE TEST. Part 6 requires operator intervention, so there are no verbs to loop on Part 6. This part verifies that the write protect status bit works and that the disk does not lose any information when the system power is cycled.

NOTE

Do not use this part if the computer does not have battery support.

When Part 6 starts executing, it asks the operator if the system has battery support. If the operator should enter a 0 (indicating no battery support), a message is printed telling the operator not to run this test.

This test checks the data previously formatted on the disk by parts, so it is necessary to run Part 5 immediately before running Part 6.

The operator is asked if Part 5 has been run. The operator should enter a 0 if it has not or a 1 if Part 5 has been run. If the operator enters a 0, a message will be output asking him to run Part 5.

If a 1 is input, a message will be output asking if the power has been cycled and if the write protect switch is in read-only position. If it has not, the operator should enter a 0. A message will then be output telling the operator to cycle the power and place the write protect switch in the read-only position.

After the power to the whole system including the disk units has been cycled, the operator should initialize the system. The E6 verb should then be used to execute Part 6.

If the computer has battery support and the operator answers yes to the next two questions, the write protect bit should be on. The diagnostic checks this bit and if the bit is not on, the following message is printed:

ERROR WRITE PROTECT TEST

DISK STATUS EXP=2000 REC=XXXX

After the write protect bit has been checked, the operator is asked to place the write protect switch to the read/write position. A reply is necessary from the operator to acknowledge the switch has been changed. Following this, the entire disc is checked to see if any data was lost on the power reset. This is verified by reading the data left from the second half of Part 5. If the data is not correct, the following message is printed:

ERROR DATA READ XXXX EXP XXXX FROM SURF XX CYL XXXX

The following shows two examples of the dialogue of Part 6.

Example 1:

```
VERB? -E6

PART 6
DOES THIS SYSTEM HAVE BATTERY SUPPORT? -0 (indicates no battery)

DO NOT RUN PART 6 –NO BATTERY SUPPORT–

VERB? -
```

Example 2:

```
VERB? -E6

PART 6
DOES THIS SYTEM HAVE BATTERY SUPPORT? -1 (system has battery support)

WAS PART 5 RUN LAST? -0
RUN PART 5
VERB? -E5
PART 5 SURFACE PACK ANALYSIS
PART 5A DONE
PART 5 DONE
VERB? -E6
PART 5
DOES THIS SYSTEM HAVE BATTERY SUPPORT? -1
WAS PART 5 RUN LAST? -1
HAS SYSTEM POWER BEEN CYCLED AND IS WRITE
PROTECT SWITCH IN READ ONLY POSITION? -1
CYCLE SYSTEM POWER AND PLACE WRITE PROTECT
SWITCH IN READ ONLY POSITION
VERB? -
```

(System power is cycled and write protect switch is placed in read-only position)

```
(DOCS Initialization Questions)
DISK TILINE ADDRESS D=FF80 -
DISK INTERRUPT LEVEL D=000 -
UNIT TO TEST 00 -
OTHER AVAILABLE UNITS (0-3 TERMINATE WITH FF)FF -
PRINT ON ERRORS? -1
PRINT HEADER MESSAGES? -1
EXECUTE EA VERB? (DEF=1) -0
VERB ? - E6
PART 6

DOES THIS SYSTEM HAVE BATTERY SUPPORT? -1
WAS PART 5 RUN LAST? -1
HAS SYSTEM BEEN CYCLED AND IS WRITE
PROTECT SWITCH IN READ ONLY POSITION? -1
ERROR WRITE PROTECT TEST
DISK STATUS EXP=2000 REC=XXX
SWITCH WRITE PROTECT SWITCH TO READ/WRITE -1
ERROR DATA READ 0000 EXP 5555 FROM SURF 00 CYCL 0000
ERROR DATA READ 555F EXP 5555 FROM SURF 03 CYCL 019A

PART 6 DONE
```

21.2 OPERATOR INTERFACE

DSKTRI runs under DOCS (see Section III for a detailed explanation of the DOCS versions).

21.2.1 AVAILABLE VERBS. DSKTRI supplies the verbs listed in table 21-1.

21.2.1.1 IT Verb. The IT verb is used to initialize the disk parameters. These parameters are:

1. Disk TILINE Address — The default value is F800.
2. Disk Interrupt Level — The default value is D.
3. Unit to Test — The default is unit 0.
4. Other Available Units — The default is FF.

Table 21-1. DSKTRI Verbs

Verb	Function
E1	Execute Part 1
.	.
.	.
.	.
E6	Execute Part 6
EA	Execute all 6 parts
L1	Loop on Part 1
.	.
.	.
.	.
L5	Loop on Part 5

NOTE

There are no verbs to loop on Part 6.

LA	Loop on Parts 1-4
CM	Execute Parts 1-4 for all available units
IC	Issue command
IM	Issue multiple commands
LO	Loop on multiple commands
DC	Display the TDC status
SR	Store registers
CD	Compare two blocks of data
FD	Format the whole disk
RH	Read headers
SA	Surface pack analysis

The following is an example of IT verb:

```
VERB ? -IT
DISK TILINE ADDRESS D=F800 -
DISK INTERRUPT LEVEL D=0D -
UNIT TO TEST 00 -
OTHER AVAILABLE UNITS (0-3 TERMINATE WITH FF)FF -
```

Whenever an error is found, a message is output and the appropriate error count is incremented. There are three error counts: one for status errors, one for data errors, and one for timing errors.

21.2.1.2 **E1 Through E6 Verbs.** These verbs are used to execute Part 1 through 6 of the TDC and disk diagnostic. After one part completes and if the operator has typed a 1 to the question:

PRINT ON ERRORS?-

during test initialization, the number of errors that occurred in that part will be printed. If errors occurred during the execution of that part, a message will be printed indicating the TDC has malfunctioned.

21.2.1.3 **L1 Through L5 Verbs.** The L1 through L5 verbs are used to loop on Parts 1 through 5 of the TDC and disk diagnostic. L1 loops on Part 1, L2 loops on Part 2, and so forth. After each time the loop has been completed, the loop count will be printed.

21.2.1.4 **EA Verb.** The EA verb is used to execute all 6 parts of the TDC and disk diagnostic in order, 1 through 6.

21.2.1.5 **LA Verb.** This verb is used to loop on Parts 1 through 4. After each loop has been completed, the loop count will be printed. To stop this verb, the operator should cause an interrupt on the 733 ASR/KSR or the VDT by entering an at sign (@).

21.2.1.6 **CM Verb.** The CM verb can be used to execute Parts 1 through 4 with each disk unit as the unit under test. It does this by executing Parts 1 through 4, then taking the next available unit (selected by the IT verb) as the unit under test.

This verb runs until interrupted and each time it finished Parts 1 through 5, it will shift to the next available unit. The present unit under test is put at the end of the available units list.

After each loop on Parts 1 through 4, the number of the unit to be tested is printed out. If the error flag is 0, the error counts are printed out after each loop.

The following is an example of the CM verb:

```
VERB ? -CM
PART 1
PART 1 DONE
PART 2
PART 2 DONE
PART 3
PART 3A DONE
PART 3B DONE
PART 3C DONE
PART 3 DONE
PART 4
PART 4 DONE
ERROR COUNTS
DATA = 0000 TIMING = 0000 STATUS = 0000
UNIT 00 UNDER TEST
PART 1
PART 1 DONE
PART 2
PART 2 DONE
PART 3
PART 3A DONE
PART 3B DONE
```

```
PART 3B DONE
PART 3 DONE
PART 4
PART 4 DONE
```

21.2.1.7 IC Verb. The IC verb can be used to issue a command to the TDC from the 990 memory address input by the operator. The command is 8 words long.

Format VERB ?-IC ADDR-XXXX

Action Issue command at ADDR with the print flag on so any errors are printed.

21.2.1.8 IM Verb. The IM verb can be used to issue multiple commands to the TDC from the 990 memory.

Format VERB ?-IM ADDR-XXXX #COMMANDS-XX

Action Issues #COMMANDS where the command list starts at ADDR. The print flag is set to 1 so any errors will be printed.

21.2.1.9 LO Verb. This verb can be used to loop on a sequence of TDC commands, each of which must be 8 words long and in sequential memory locations.

Format VERB ?-LO ADDR-XXXX #COMMANDS-XX CHECK ST?-X

Action Loop on a series of commands starting at ADDR.
The number of commands is #COMMANDS.
The status checker flag is set to CHECK ST?

21.2.1.10 DC Verb. The DC verb can be used to print out or display the status of the TDC. It reads the TDC slave registers, then formats and outputs them to the VDT or 733 ASR/KSR.

The following shows the format of the DC verb:

```
VERB? -DC

CONTROLLER STATUS

UTC REG.

DISK STAT  COMM SA  S/R  R A CYL A BYTE C  MEM AD SEL CONT STATUS
C0B0       00       00   00   00  0000  0000     0024CC  04   C800
```

21.2.1.11 CD Verb. This verb can be used to compare two blocks of data starting at the two addresses input by the operator for the number of words input. If any comparison errors are found, the words in error are printed. The verb then continues until it is done or another comparison error is found.

21.2.1.12 **SR Verb.** This verb performs a store register command, then prints out the disk descriptor words as shown in the following example:

```
VERB ? -SR
TOTAL WORD COUNT = 0F00
SECT/TRACK = 18 OVERHEAD/RECORD = 10
TRACKS/CYL = 02 CYL/DRIVE = 00CB
```

21.2.1.13 **RH Verb.** This verb reads the three header words from each sector on the track and verifies the data. On sector zero, the word count and sectors/record will never give an error. The word count and sectors/record on sector zero will be used as the standard for the remaining sectors. The known data (cylinder, head, and sector) is calculated by the program and compared on all sectors. The status checker flag is set to 1 so any disc status errors will be printed.

The following is an example of the RH verb:

```
VERB? -RH CYL-0XXX HEAD-0X

                DATA EXPECTED          DATA RECEIVED

CYL   HEAD SECT  S/R WC      CYL   HEAD SECT  S/R WC
XXXX  0X   XX    XX  XXXX    XXXX  0X   XX    XX  XXXX
```

21.2.1.14 **FD Verb.** This verb can be used to format the whole disk. Every track on the disk will be formatted with sectors/record equal to 1 and the word count at maximum. The status checker flag is set to 1 so any disk errors will be printed. It will take approximately 4 to 5 minutes to format a model DS50 disk. Larger disks will take longer.

NOTE

Be sure to use a scratch disk pack because data will be written on the disk.

The following shows the format of the FD verb:

```
VERB? -FD YOU SURE 0=NO -1
```

21.2.1.15 **SA Verb.** This verb executes a surface pack analysis of the entire disc pack. There are two options available for this verb: short test and long test. The test formats each track with two patterns ("CCCC" and "AAAA") for the short test and formats each track with four patterns ("CCCC", "AAAA", "3333" and "5555") for the long test. After each formatting operation, the data intergrity is checked two times with reverse offset (OS bit = 1, OSF bit = 0) and two times with forward offset (OS bit = 1, OSF bit = 1).

If an error occurs during the data intergrity check, the following algorithm is used to decide if a track is bad: for a given type of offset, three retrys with error should occur before two retrys without error. If a track is found to be bad, its cylinder and head no's are printed in decimal and the pattern is printed in hex. The test resumes with checking the next track. If more than 100_{10} tracks are found to be bad, the test is aborted and control returns to verb decoder.

Format:

VERB ? - SA

SURFACE PACK ANALYSIS

SELECT LENGTH: 0 = SHORT, 1 = LONG (DEFAULT = 0) –X*

If a track (or more) is found to be in error, the following error message is printed:

BAD TRACK (IN DECIMAL)

CYLINDER HEAD PATTERN

XXXX XX XXXX

If more than 100 bad tracks are found, the following message is printed:

MORE THAN 100 BAD TRACKS - TEST ABORT.

Upon the test completion (without bad tracks, with bad tracks or abortion) the following message is printed:

PACK ANALYSIS DONE

ERROR COUNTS **

DATA = XXXX TIMING = XXXX STATUS = XXXX

***NOTE 1**

Usually, the short test is adequate for most of the packs. The long test may be used where there is a doubt about a certain pack (example - bad correlation in between vendor map and short test map). It is noted that the running time for the long test is double in respect with the short test.

**** NOTE 2**

A bad track error is not counted toward any of specified error counts. Only Command Issuer errors or Status Checker errors are counted.

Run time of this test depends upon the type of disk under test (DS25, DS50, or DS200), and is given as follows:

	Run Time (hours)		
	DS25	**DS50**	**DS200**
Short Test	¼	½	2
Long Test	½	1	4

In order to be able to check that the test is operational, during this long test, the hex value of the cylinder under test is displayed on the computer front panel. The operator may check if the test is running by observing the value displayed on the front panel which should keep incrementing (at a rate function of disc type).

21.2.2 **STANDARD INITIALIZATION.** In addition to the questions required by the DOCS, DSKTRI also asks the following questions:

DISK TILINE ADDRESS D=F800 -
DISK INTERRUPT LEVEL D=0D -
UNIT TO TEST 00 -
OTHER AVAILABLE UNITS (0-3 TERMINATE WITH FF)FF -

NOTE

These questions are asked by the IT verb.

21.2.3 **PROGRAMMER-PANEL INITIALIZATION.** When running the disk diagnostic in the programmer-panel mode, the only question asked by the DOCS is IDLE ON ERRORS?. The questions following this will be those initialization questions asked by the IT verb. After all initialization questions are answered by the operator, the test will start executing.

The test will run approximately 16 minutes before the next operator interaction is required. Of course this time will be longer if errors occur. The error numbers will appear on the programmer panel. When operator interaction is required, the test is starting Part 6. The 990/10 will go into the idle mode with 0000 hexadecimal on the programmer panel. At this point, the question being asked is:

DOES THIS SYSTEM HAVE BATTERY SUPPORT?

If there is no battery support, the test is terminated.

If battery support is present, the 990/10 will go into the idle mode. Now the question being asked is:

WAS PART 5 RUN LAST?

After responding YES (1), the 990/10 will go into the idle mode with 0000 hexadecimal on the programmer panel. The question being asked is:

HAS DISK POWER BEEN CYCLED AND
IS WRITE PROTECT SWITCH IN READY ONLY POSITION?

At this point, turn the unit being tested off and on again leaving the write protect switch in the read-only position. After responding YES (1), the final operation for the operator to perform occurs when the 990/10 goes into the idle mode with 0001 hexadecimal on the programmer panel. At this time, the operator should clear the write protect by moving the write protect switch on the disk drive to the read/write position. To proceed with the test, the operator should respond with a YES (1). The program will finish Part 6, display a loop count on the programmer panel, and start executing the test over again.

21.3 SYSTEM REQUIREMENTS

The following list of equipment is recommended by DSKTRI:

(1) a 990/10 minicomputer with 12K or more of memory,

(2) an appropriate interactive device,

(3) a DS25 or DS50 disk kit,

(4) an appropriate loading device.

21.4 ERROR MESSAGES AND NUMBERS

When an error occurs in the execution of any part of the test, an error message will be printed if the print error flag is equal to 1. The error message indicates which error has occurred. The number of the error will also appear on the programmer panel. The error messages are listed in table 21-2.

Table 21-2. DSKTRI Error Messages and Numbers

Number	Message
	Part 1
1	STORE REG ERROR
2	ERROR R6 = 0 STATUS SHOULD BE C0X0 WAS XXXX
3	R7 SHOULD BE A100 WAS XXXX AFTER IO RESET
4	LOCK OUT NOT SET AFTER 2 READS
5	UTC REG DATA TEST WAS EXP REG. XXXX XXXX XX XXXX XXXX XX
	Part 2
6	ERROR ON WRITE/READ UNFORMATTED
7	ERROR ON SEEK TRACK ADDR EXP = XXXX REC = XXXX
8	ERROR ON SEEK SECTOR ADDR EXP 0 REC XX
9	ERROR ON SEEK CRC CHAR EXP = XXXX REC = XXXX
	Part 3
A	ERROR PART 3A DATA EXP XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
B	ERROR PART 3B TRACK ADDR XXXX DATA REC XXXX XXXX XXXX XXXX XXXX

Table 21-2. DSKTRI Error Messages and Numbers (Continued)

Number	Message
C	PART 3D CRC ERROR CRC EXP = XXXX REC = XXXX ID WORD1 EXP = XXXX REC = XXXX ID WORD2 EXP = XXXX REC = XXXX ID WORD3 EXP = XXXX REC = XXXX
D	PART 3D ID WORD ERROR ID WORD1 EXP = XXXX REC = XXXX ID WORD2 EXP = XXXX REC = XXXX ID WORD3 EXP = XXXX REC = XXXX
	Part 4
E	ERROR DID COMMAND WITH NO UNIT SEL R7 STATUS EXP A805 REC XXXX ERROR DID STORE REG WITH WC = XX WORDS TRANSFERRED XX
F	ERROR DID STORE REG WITH WC = XX WORDS TRANSFERRED XX
10	PART 4C ID ERROR EXP = XXXX REC = XXXX
11	PART 4E SEARCH ERROR EXP = A802 REC = XXXX
12	PART 4F DATA ERROR EXP = A850 REC = XXXX
13	PART 4G TIMEOUT ERROR NO TIMEOUT IN 200 MS
14	PART 4G STATUS ERROR R7 STATUS EXP = A804 REC = XXXX
15	ERROR PART 4 SI TEST R0 STATUS EXP = 08XX REC = XXXX R7 STATUS EXP = A801 REC = XXXX
16	ERROR PART 4I DID WRITE WITH WC = 0 BUT DATA NOT 0 ERROR PART 4 BUSY FLAG SET DATA (MSB SHOULD BE 0) XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
18	PART 4K TILINE TIMEOUT ERROR R7 STATUS EXP = A820 REC = XXXX
19	PART 4L RATE ERROR R7 STATUS EXP = A008 REC = XXXX

Table 21-2. DSKTRI Error Messages and Numbers (Continued)

Number	Message
	Part 4 (Continued)
1A	PART 4M UNIT ERROR EXP = AAAA REC = XXXX SE SL OS OSF XX XX XX XX
1C	PART 4D TRANSFER INHIBIT ERROR
1D	PART 4P HEAD SWITCH FAILURE FAILURE TO SWITCH HEADS 0 to 1, CYL 0
1E	PART 4P HEAD SWITCH FAILURE FAILURE TO SWITCH HEADS FROM MAX HEAD CYL 0, TO HEAD 0 CYL 1
	Part 6
20	ERROR WRITE PROTECT TEST DISK STATUS EXP = 2000 REC = XXXX
	Command Issuer
28	*ERROR* THE UTC DID NOT RESPOND IN 20 SEC
29	*ERROR* ALL ATTENTION LINES WERE NOT SET 3 SEC AFTER DOING A SEEK OR RESTORE COMMAND
2A	UNEX DISK INT AT XXXX
	Status Checker
2B	NR BIT FOR UNIT XX WAS NOT SET WHILE SEEKING
2C	ERROR ON SEEK OR RESTORE UNIT X REG 7 = XXXX DISK STATUS UNIT = X REG 0 = XXXX UTC REG. DISK STAT COMM SA S/R R A CYL A BYTE C MEM AD SEL CONT STATUS XXX 0X XX XX XX XXXX XXXX XXXXXX XX XXXX COMMAND ISSUED DISK STAT COMM SA S/R R A CYL A BYTE C MEM AD SEL CONT STATUS XXX 0X XX XX XX XXXX XXXX XXXXXX XX XXXX
2D	** STATUS ERROR CONTROLLER STATUS COMP = X ERR = X IDLE = X UTC REG. DISK STAT COMM SA S/R R A CYL A BYTE C MEM AD SEL CONT STATUS XXX 0X XX XX XX XXXX XXXX XXXXXX XX XXXX DISK STAT COMM SA S/R R A CYL A BYTE C MEM AD SEL CONT STATUS XXX 0X XX XX XX XXXX XXXX XXXXXX XX XXXX

SECTION XXII

COMMUNICATIONS INTERFACE MODULE TEST (CRCOMM)

22.1 INTRODUCTION

The CRCOMM operates under the DOCS. The test verifies full performance of and/or diagnoses problems associated with the communication interface module (part number 946104). The subgroup of tests to be performed is indirectly under the control of the operator in that the operator specifies the testing mode for the module. The module may be tested with or without a special test connector (part number 948550) attached, or with or without a modem (synchronous or asynchronous) attached. Those tests that cannot be meaningfully performed without either the test connector or modem attached are skipped.

It is assumed that the user has access to the installation and operation manual and the depot maintenance manual listed in the preface for the communications interface module.

CRCOMM consists of the following 22 tests:

Test 1 toggles RES MODEM LD OUT (output word 4, bit 3) and SYNC MODE (output word 3, bit 5) and tests the corresponding input bits for agreement.

Test 2 sets DTR and then toggles SRTS to determine if SDCD toggles in agreement. Then it checks to see if NSF (new status flag) and INTSUM are set and cleared properly by this condition.

Test 3 checks the 250-millisecond timer (TIMEXP) for proper setting and clearing, and checks the timing to within the design tolerance. It also checks INTSUM for proper setting and clearing.

Test 4 reads the ID switches on the module and prints no message (except TEST 4 COMPLETE) unless the switches are set to other than 7F, that is, switches 1-7 all closed (on). In the event there is a discrepancy, it prints the switch ID as read and indicates an error.

Test 5 checks to ensure that the interrupt logic on the module properly interrupts the processor when enabled, and not otherwise. If zero has been entered as the interrupt level this test will be skipped.

Test 6 writes an alternating pattern of bytes to output words 2 and 3, strobing each to the asynchronous/synchronous receiver/transmitter. It then reads the asynchronous/synchronous receiver/transmitter to verify correct operation. Then it swaps and repeats process.

Test 7 writes a WALKING 1 pattern to output words 2 and 3, strobing each to the asynchronous/synchronous receiver/transmitter. It then reads the asynchronous/synchronous receiver/transmitter registers, thus written back through the I/F to verify correct operation. Then it writes a WALKING 0 pattern and repeats until a 1 or a 0 have been walked through all 8 bit positions.

Test 8 checks the correct operation of the 1-millisecond pulse used as carrier detect reset on an asynchronous modem and newsync on a synchronous modem. The PULSED MODEM LD (output word 4, bit 4) bit is toggled various times to check proper setting of the RING INDICATOR input, and timing is checked to within the design tolerances.

Test 9 checks the CDR/NEWSYNC EIA driver and the RING EIA receiver by toggling PULSED MODEM LD and checking RING for agreement.

Test A toggles DTR and checks the toggling of DSR. It also checks to see if DSR setting causes NSF to set when enabled.

Test B toggles RTS and checks the toggling of CTS and DCD. It also checks to see if DCD setting causes NSF to set when enabled.

Test C toggles SRTS and checks the toggling of SDCD. It also checks to see if SDCD changes cause NSF to set when enabled.

Test D sets up the conditions that should cause WRQ to set and then checks to see if WRQ sets when enabled and resets when disabled. It also checks to see if INTSUM is set when WRQ is equal to 1.

Test E enables the half-duplex bit that masks the receive data to a space and verifies that no RRQ interrupts are generated.

Test F forces a line break condition (transmitted data a constant space) and tests proper operation of the BREAK bit. Since the breaking condition also forces a parity error and framing error, these bits, PE and FE, and RCVERRSUM are also tested for proper setting.

Test 10 performs a 1-byte data transfer in SELF-TEST, ASYNCHRONOUS mode to verify data transfer capability.

Test 11 sets up a data transfer, but with RRQ disabled, thus forcing an OVERRUN condition. It tests for the OVERRUN bit being set and RCVERRSUM being set. It then enables RRQ and checks to see if RRQ goes from a zero to a one. It also receives the character transmitted and verifies that the transfer was done correctly.

Test 12 writes the asynchronous/synchronous receiver/transmitter SYNC and DLE registers and then uses the transparent idle fill feature to transmit a DLE/SYNC sequence. This constitutes a transmitter UNDERRUN condition, so that the UNDERRUN bit can be tested for setting. Then the receiver is enabled and the DLE/SYNC sequence's proper transmission and reception is verified.

Test 13 performs a series of 200-byte data transfers in SELF-TEST, SYNCHRONOUS mode to verify data transfer capability. The tranfer is performed 4 times and must be executed successfully (no error conditions) all four times in order to pass.

Test 14 performs a series of 200-byte data transfers in ASYNCHRONOUS mode and at a different baud rate from Test 12. Other test parameters are essentially the same as in Test 12.

Test 15 performs a series of 200-byte data transfers in SYNCHRONOUS mode and at a different baud rate from Test 13. Other test parameters are essentially the same as in Test 13.

Test 16 performs a series of 200-byte data transfers as does test 13 but the communication module is not reset between transfers. This tests STRIP SYNC between blocks.

NOTE

Tests 13 and 15 use a clock-recovery method for synchronizing the data, unless a synchronous modem is used. Thus, although the data format is synchronous, the clocking scheme is essentially asynchronous.

All tests first master reset the module so that all output CRU interface bits are zero, all registers are zero, and all flip-flops are zero (inactive).

Some of the tests described above could either duplicate effort in prior tests, or be meaningless without the loopback connector installed, or without the correct type (asynchronous or synchronous) of modem connected. In this case, the test will be skipped, as illustrated by the following example:

```
VERB? - EA
TEST 1 COMPLETE
TEST 2 COMPLETE
TEST 3 COMPLETE
TEST 4 COMPLETE
TEST 5 COMPLETE
TEST 6 COMPLETE
TEST 7 COMPLETE
TEST 8 COMPLETE
TEST 9 SKIPPED
TEST A SKIPPED
TEST B SKIPPED
TEST C SKIPPED
TEST D COMPLETE
TEST E COMPLETE
TEST F COMPLETE
TEST 10 COMPLETE
TEST 11 COMPLETE
TEST 12 COMPLETE
TEST 13 COMPLETE
TEST 14 COMPLETE
TEST 15 SKIPPED
TEST 16 COMPLETE
```

22.2 OPERATOR INTERFACE

CRCOMM runs under the DOCS (refer to Section III for a detailed explanation of the DOCS versions).

22.2.1 AVAILABLE VERBS. CRCOMM supplies the verbs listed in table 22-1.

When a verb is executing, execution may be suspended by entering any single character on the keyboard, provided the interactive device is wired for interrupts.

22.2.2 STANDARD INITIALIZATION. In addition to the questions required by the DOCS, CRCOMM also asks the questions listed in table 22-2.

Table 22-1. CRCOMM Verbs

Verb	Function
IT	Initialize the program
ET	Execute test (This allows execution of tests 1-16)
EA	Execute all tests
LT	Loop on test
LA	Loop on all tests
DS	Display status
SP	Select patterns (Test 6)

Table 22-2. CRCOMM Initialization Questions

Question
1. ENTER COMM INTERFACE CRU BASE DEFAULT = 0140
2. LOOP BACK CONNECTOR INSTALLED? DEFAULT = 01
3. ENTER COMM INTERFACE INTERRUPT LEVEL DEFAULT = 09
4. IS COMM INTERFACE CONNECTED TO A MODEM? DEFAULT = 01
5. IS THE MODEM SYNCHRONOUS? DEFAULT = 00
6. IS THIS COMM INTERFACE A 2741? DEFAULT = 00
7. ID SWITCH VALUE? DEFAULT = 007F
8. IS THE COMPUTER LINE CURRENT 60 HZ? DEFAULT = 01

NOTE

Questions 1 through 3 are always asked. Questions 4 and 5 are printed only if the answer to question 3 is no (zero). Questions 6, 7, and 8 are printed regardless of the answer to question 3.

22.2.3 **PROGRAMMER-PANEL INITIALIZATION.** When operating under the programmer-panel version of CRCOMM, the first question asked will be:

IDLE ON ERRORS?

Subsequent questions will be those in table 22-2.

22.3 SYSTEM REQUIREMENTS

CRCOMM runs on a computer system with the following requirements: (1) 990 computer with 12K of memory, (2) an appropriate interactive device, (3) test connector, part number 948550-0001 (optional), and (4) communications interface module (either standard interface or the 2741 interface).

22.4 ERROR MESSAGES AND NUMBERS

The error messages in table 22-3 are considered self-explanatory in all cases, assuming some familiarity with the module. The notation IWm-n implies input word m, bit n, and OWm-n

implies output word m, bit n, as defined in either the installation and operation or depot maintenance manuals.

Table 22-3. CRCOMM Error Messages and Numbers

Number	Message
1	RSVOUT NOT SET (IW1-3) ERROR
2	RSVOUT NOT RESET (IV1-3) ERROR
3	MODE NOT SET (IW1-9) ERROR
4	MODE NOT RESET (IW0-9) ERROR
5	MODE NOT RESET (IW1-9) ERROR
6	MODE NOT RESET (IW0-9) ERROR
7	SDCD NOT SET (IW0-9) ERROR
8	NSF SET WHEN NOT ENABLED (IW1-12) ERROR
9	SDCD NOT RESET (IW1-2) ERROR
A	NSF NOT BEING SET (IW1-12) ERROR
B	INTR SUM NOT SET (IW1-14) ERROR
C	NSF NOT RESET (IW1-12) ERROR
D	INTR SUM NOT RESET (IW1-14) ERROR
F	TIMEXP NOT RESET (IW1-13) ERROR
10	TIMER TOO SHORT (IW1-13) ERROR
11	TIMER NOT BEING SET (IW1-13) ERROR
12	TIMER TOO LONG (IW1-13) ERROR
13	TIMER TOO LONG AFTER RETRIGGER (IW1-13) ERROR
14	TIMER SHORT AFTER RETRIGGER
15	INTERRUPT SUMMARY NOT SET BY TIMEXP (IW1-14) ERROR
16	ID SWITCH READ XXXX EXPECTED XXXX (IW4-0-6) ERROR
17	UNEXPECTED INTERRUPT AT LEVEL XXXX ERROR
18	UNEXPECTED INTERRUPT FROM COMM INTERFACE ERROR

Table 22-3. CRCOMM Error Messages and Numbers (Continued)

Number	Message
19	COMM INTERFACE DOES NOT INTERRUPT CPU ERROR
1A	RING NOT SET (IW1-4) ERROR
1B	RING NOT RESET (IW1-4) ERROR
1C	PMODLD TIMING LONG ERROR (OW4-4)
1D	PMODLD TIMING SHORT ERROR (OW4-4) WROTE READ XOR OW2 XXXX XXXX ZZZZ OW3 XXXX XXXX ZZZZ
1F	*** ERROR RW BUSY NOT RESET
20	DSR NOT BEING SET (IW1-0) ERROR
21	DSR NOT RESET (IW1-1) ERROR
22	CTS NOT BEING SET (IW1-1) ERROR
23	DCD NOT BEING SET (IW1-5) ERROR
24	CTS NOT RESET (IW1-1) ERROR
25	DCD NOT RESET (IW1-5) ERROR
26	SDCD NOT BEING SET (IW1-2) ERROR
27	WRQ SET WHEN NOT ENABLED (IW1-15) ERROR
28	WRQ NOT BEING SET (IW1-15) ERROR
29	RRQ SET WHEN NOT ENABLED (IW1-8) ERROR
2A	OVERRUN NOT SET (IW0-14) ERROR
2B	UNEXPECTED FRAMING ERROR (IW1-13)
2C	UNEXPECTED PARITY ERROR (IW0-12) PARITY DISABLED
2D	XMIT DATA = XX ADDRESS = AAAA RCV DATA = XX ADDRESS = BBBB ERROR COUNT = YY
2E	*** ERROR IN DATA TRANSFER *** IW0 = XXXX IW1 = XX IW2 = XX

Table 22-3. CRCOMM Error Messages and Numbers (Continued)

Number	Message
	IW3 = XX XMIT BUFF ADDRESS = XXXX BYTE CNT = XXXX RCV BUFF ADDRESS = XXXX BYTE CNT = XXXX
2F	RRQ GENERATED WHEN HALF DUPLEX SELECTED (OW4-6) ERROR
30	RECVERR NOT BEING SET (IW0-15) ERROR
31	PE NOT BEING SET (IW0-12) UPON BREAK (OW4-6) ERROR
32	FE NOT BEING SET (IW0-12) UPON BREAK (OW4-6) ERROR
33	TIMED OUT

SECTION XXIII

CRU EXPANSION CHASSIS INTERFACE TEST (CRUEXP)

23.1 INTRODUCTION

The CRUEXP tests the CRU expansion chassis interface board and the CRU buffer board and expansion chassis by using a 16 I/O TTL module (part number 945145-0001) located in the expansion chassis. The 16 I/O TTL module is operated in the loopback mode during this test. CRUEXP operates by looping data through the expansion chassis and verifying the data.

23.2 OPERATOR INTERFACE

CRUEXP supports the verbs listed in table 23-1 in addition to the verbs available in the DOCS.

23.2.1 IT VERB. The IT verb is used to set up the initial test parameters for the CRU expansion test. This verb prints the messages in table 23-2. The values for interrupts must be 0-F and the value of the CRU expansion chassis must be 1-7.

23.2.2 E1 THROUGH E2 VERBS. To execute either of the 2 parts of the test, enter the E1 or E2 verb. Each verb prints a start and test-complete message. Also, the E1 verb prints the number of errors that occurred and the bit positions of the errors.

23.2.3 EA VERB. The EA verb executes parts 1 and 2 of the test in order.

23.2.4 L1 THROUGH L2 VERBS. The L1 and L2 verbs loop on parts 1 or 2 of the test, respectively.

23.2.5 LA VERB. The LA verb loops on both parts 1 and 2 of the test.

Table 23-1. CRUEXP Verbs

Verb	Function
IT	Initialize test
E1	Execute Part 1, pattern test
E2	Execute Part 2, interrupt test
EA	Execute Parts 1 and 2
L1	Loop on Part 1
L2	Loop on Part 2
LA	Loop on Parts 1 and 2

Table 23-2. CRUEXP Initialization Questions

Question	Comment
16 I/O CRU BASE DEFAULT = 0400-	Enter the correct CRU base followed by a space.
16 I/O INTERRUPT LEVEL DEFAULT = 06-	Enter the correct interrupt followed by a space.
CRU EXPANSION CHASSIS NO. DEFAULT = 01-	Enter the correct expansion chassis number followed by a space.
CRU EXPANDER INTERRUPT LEVEL DEFAULT = 0A-	Enter the correct interrupt level followed by a space.

23.2.6 STANDARD INITIALIZATION. In addition to the initialization questions asked by the DOCS, CRUEXP asks the initialization questions listed in table 23-2. The IT verb simply reiterates the initialization questions.

23.2.7 PROGRAMMER-PANEL INITIALIZATION. When operating under the programmer-panel version of CRUEXP, the first question asked will be:

IDLE ON ERRORS?

Subsequent questions will be those in table 23-2.

23.3 SYSTEM REQUIREMENTS

In addition to the hardware listed in the section on the DOCS, CRUEXP also requires (1) 16 I/O data module (part number 945145-0002) with high to low interrupt, (2) 16 I/O data module test connector (part number 946760-0002), and (3) CRUX1 expansion kit (part number 944985-0001) consisting of CRU expansion PCB (part number 945005-0001), CRU buffer PCB (part number 944905-0001), and CRU expansion cable kit (part number 945001-0001).

23.4 ERROR MESSAGES AND NUMBERS

The error message descriptions and numbers are listed in table 23-3.

Table 23-3. CRUEXP Error Messages and Numbers

Number	Message
1	COMPARE ERROR WIDTH = XX OUTPUT = XXXX INPUT = XXXX
2	INTERRUPT MASK ERROR
3	EXPECTED INTERRUPT DID NOT OCCUR
4	INTERRUPT NOT CLEARED
5	UNEXPECTED CRU EXPANDER INTERRUPT
6	INTERRUPT BIT IN ID WORD NOT SET
7	CHASSIS ID WORD ERROR EXPECTED CHASSIS ID = XX RECEIVED CHASSIS ID = XX
8	INTERRUPT ID ERROR EXPECTED INTERRUPT LEVEL = XX RECEIVED = XX
9	INTERRUPT BIT SET IN ID WORD NO INT PRESENT

SECTION XXIV

TILINE COUPLER LOGIC TEST (TILCOU)

24.1 INTRODUCTION

The TILCOU is used to verify the interrupt and interlock logic on the TILINE coupler logic boards (part number 945091-0001) and the Absolute Value (ABS) instruction function in the 990 multiprocessor systems.

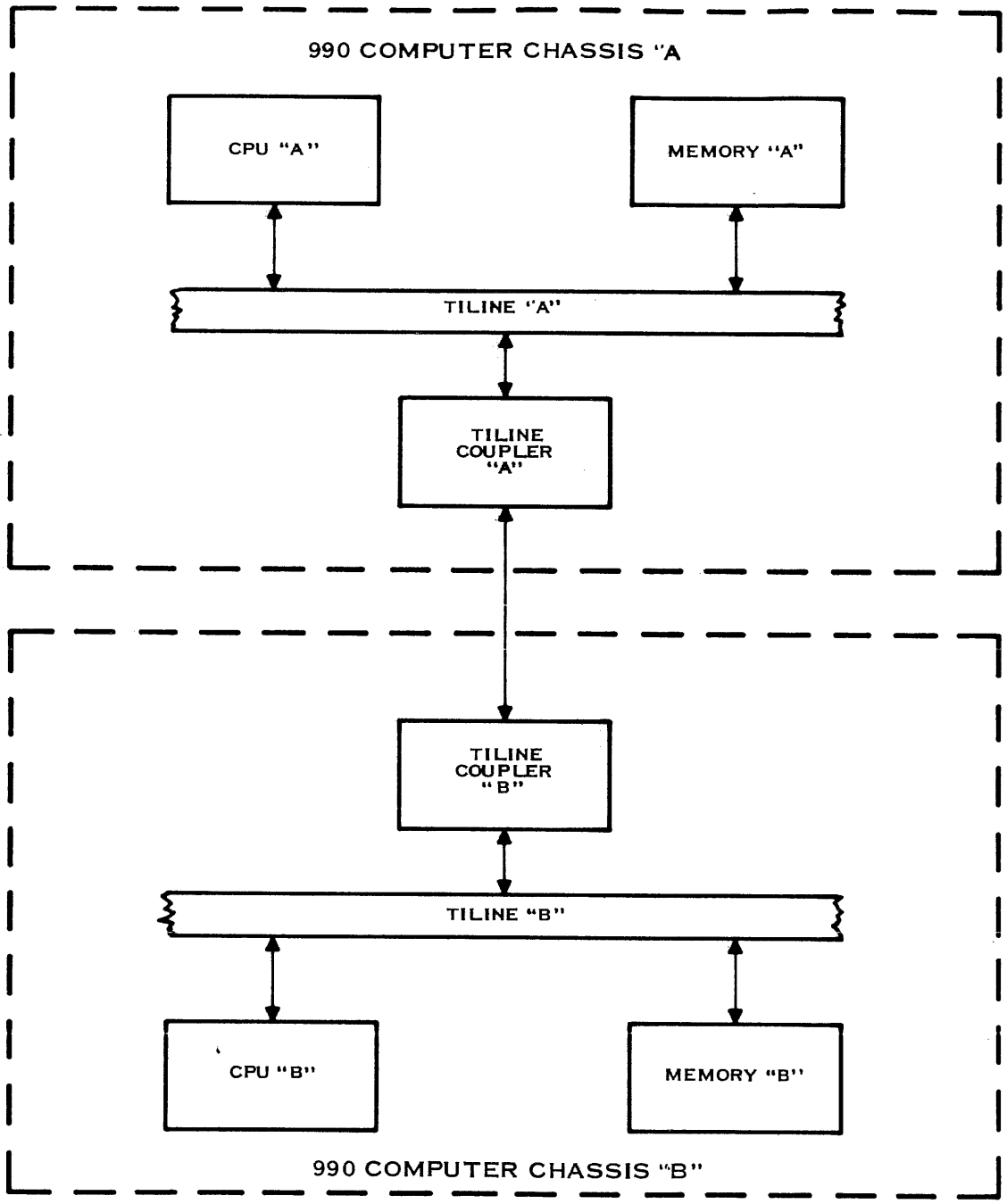
NOTE

In order to do a thorough test of the coupler logic, the RAM10 memory test should be run in each processor, with the addresses set up to test the memory across the TILINE couplers, in addition to the TILCOU test.

The coupler is designed to resolve timing conflicts resulting from simultaneous data transfer requests in multiprocessor systems. Two 990 computers each executing across the coupler are required to test the resolving logic. The operator should first test memory using RAM10 in each processor. If self-test ROMs are used in either chassis, do not connect the vectored data cable until both computers have loaded RAM10. Be sure to protect both program areas by use of the coupler switch settings.

TILCOU runs under control of the DOCS. Interrupts are checked by using two 990 computers, each of which sends the other an interrupt (via the TILINE coupler), and verifies that the interrupts occur.

In 990 multiprocessor systems two or more CPUs share common memory, and it is possible for more than one CPU to access the same memory simultaneously causing loss of data. To prevent this conflict software "memory busy" flags are used. However, if two CPUs check the status of the memory busy flag in successive memory cycles, the second CPU will see that the memory is not busy because the first CPU has had insufficient time to set the busy flag. Therefore it is possible for one CPU to modify a memory location between the read and write cycles of another CPU. All 990 instructions, except the ABS instruction, allow this problem to occur. The problem is avoided with the ABS instruction because the CPU issues the TILINE HOLD (TLHOLD–) signal which is implemented and propogated by the TILINE couplers and denies access to the TILINE (and to memory) by any other CPU during execution of the ABS instruction. Therefore, it is necessary to use the ABS instruction to examine and set memory busy flags in all memory sharing applications so that the integrity of the memory busy flags are maintained. This function of the ABS instruction is tested by having CPU "A", located on TILINE "A" (see figure 24-1), attempt execution of an ABS instruction across the TILINE couplers to an address in memory "B" on TILINE "B". Simultaneously CPU "B", located on TILINE "B", has access to TILINE "B" and is executing the ABS instruction to the same memory address in memory "B". Under these conditions, if the ABS software interlock is working properly, CPU "A" will be denied access to TILINE "B" (and to memory "B") until CPU "B" has had time to set the memory busy flag for memory "B". Once this flag is set and TILINE HOLD is released by CPU "B", CPU "A" may access TILINE "B" and attempt to access memory "B" but it will find the memory busy flag and be denied access until CPU "A" has completed its memory operations and cleared the memory busy flag.



(A) 138981

Figure 24-1. Simplified 990 Multiprocessor System

This function of the ABS instruction may be tested in the other direction by CPU "B" attempting execution of an ABS instruction to memory"A" while CPU "A" is executing an ABS instruction to the same memory "A" address.

24.1.1 INTERRUPT TEST. Part 1 of the TILINE coupler logic diagnostic tests the ability of the TILINE coupler CRU interface to cause maskable interrupts.

24.1.2 ABS INSTRUCTION. Part 2 of the TILINE coupler logic test tests the effectiveness of the ABS instruction function in maintaining the integrity of the memory busy flags.

24.2 OPERATOR INTERFACE

TILCOU runs under the DOCS (see Section III for more details on the DOCS versions).

24.2.1 MULTIPROCESSOR LOADING. This test should only be loaded to one processor, the local. The diagnostic will be downloaded to the other processor, called the remote, at initialization time. The remote processor must be in IDLE to load the TILINE coupler logic test if the local processor does contain self-test ROMs. Execute the IDLE instruction by entering 0340 to the programmer panel and pressing WPE, PCE, MAE, and MDE. Press the RUN switch and the IDLE LED should be illuminated.

24.2.2 STANDARD OPERATION. After the DOCS initialization questions are answered, the following questions are asked:

```
LOCAL COUPLER INTERRUPT LEVEL DEFAULT = 000D -
FIRST REMOTE ADDRESS RECOGNIZED BY LOCAL
PROCESSOR ADDRESSING REMOTES MEMORY DEFAULT=4000 -
```

(This address is the address that the local processor uses to load to the remote processor's memory.)

```
DOWN LOAD TO REMOTE PROCESSOR VERIFIED
ENTER PC = C0 WP = A0 AND PRESS RUN
```

(Now the operator should set the program counter of the remote processor to 00C0, the workspace pointer to 00A0 and press the run switch. The display of the remote processor should then display a 0001. The remote processor is then ready.)

```
THE REMOTE PROCESSOR SHOULD DISPLAY A
  1 ON THE PROG PANEL
IS REMOTE PROCESSOR READY ? -
```

Table 24-1 lists the verbs available in TILCOU.

The IT verb is used to set up the local processors interrupt vector and download the diagnostic to the remote processor. If an error occurs during download, the following message is printed:

```
DOWNLOAD ERROR
```

Table 24-1. TILCOU Verbs

Verb	Function
EA	Execute Part 1 and Part 2
LA	Loop on Parts 1 and 2
IT	Initialize test
E1	Execute Part 1, interrupt test
E2	Execute Part 2, ABS instruction
L1	Loop on Part 1
L2	Loop on Part 2
PE	Print Errors

The operator should correct the faulty condition, such as a loose memory board, then use the IT verb to download the test again.

24.2.3 PROGRAMMER-PANEL OPERATION. The TILINE coupler logic test can be run on the local processor that has a programmer panel as the only I/O device. All operator communication is through the programmer panel. In addition to the question IDLE ON ERRORS? the questions shown in paragraph 24.2.2 are asked.

After the operator has answered all of the initialization questions, the test will execute Parts 1 and 2.

24.3 SYSTEM REQUIREMENTS

TILCOU requires at least the following equipment: (1) two 990 computers, each with at least 8K words of memory, (2) a TILINE coupler expansion kit (part number 945091-0002) which includes two couplers, (3) an appropriate interactive device, and (4) an appropriate loading device.

NOTE

This test cannot be run on single processor systems.

24.4 ERROR MESSAGES AND NUMBERS

When an error occurs, an error message number from the list in table 24-2 is displayed on the programmer panel.

Table 24-2. TILCOU Error Messages and Numbers

Number	Message
1	INTERRUPT FROM REMOTE DID NOT OCCUR
2	STATUS REPORTS POWER NOT ON IN REMOTE
3	MASK DID NOT HOLD OFF INTERRUPT
4	UNEXPECTED TILINE COUPLER INTERRUPT
5	*ERROR* INTERRUPT FROM REMOTE PRESENT BEFORE MASK AT INITIALIZATION TIME
6	*ERROR* INTERRUPT FROM REMOTE PRESENT BEFORE READ BEFORE THE MASK
7	*ERROR* INTERRUPT FROM REMOTE DID NOT CLEAR AFTER RESET INSTRUCTION
8	*ERROR* REMOTE DID NOT DETECT INTERRUPT BEFORE THE MASK
9	*ERROR* INTERRUPT FROM LOCAL DID NOT OCCUR
A	*ERROR* REMOTE DID NOT SEE INT RESET WHEN READ BEFORE MASK
B	*ERROR* REMOTE PROCESSOR POWER STATUS BIT NOT RESET
C	*ERROR* REMOTE MASK DID NOT HOLD OFF INTERRUPT
D	*ERROR* TILINE HOLD PERMITTED ACCESS DURING ABS INSTRUCTION
E	TILINE TIMEOUT DURING DOWN LOAD

SECTION XXV

EPROM MEMORY BOARD TEST (EROMBT)

25.1 INTRODUCTION

The EROMBT tests the Texas Instruments EPROM (part number 945170-0001) memory board for the 990 family of computers. Tests 1, 2, and 3 are designed so that knowledge of the contents of the EPROMs is not required by the test. These tests do not attempt to test the EPROMs, but rather the TTL logic on the EPROM board that supports the addressing of the EPROMs and the data buffering out of the EPROMs. Test 4 tests the contents of the EPROM chips during burn-in testing.

NOTE

The EPROMs must contain a program and must not be programmed to all zero or all one data.

Test 1 of the EROMBT checks that the EPROM board causes an interrupt when an attempt is made to write to a memory address occupied by the board.

Test 2 checks the data output buffers on the EPROM board. The test uses the data contained in the EPROMs. Data in all the EPROMs is compared against data in all other EPROMs on a bit-for-bit basis. If a bit ever changes, it is known that the TTL logic for that data bit is functioning.

After all data bits have been checked, any bit that has failed to change state is reported as a suspected error. It is possible that a particular bit in a set of EPROMs actually does have the same state throughout the entire address space; however, it is unlikely.

A failure of the addressing logic may cause indications of data failure. Only in extreme cases should data failure cause indications of the addressing failure.

Test 3 checks the TTL addressing logic on the EPROM board. The test holds an address bit at differing states in 2 different address buffers. Buffer 1 holds the bit at a 0, while buffer 2 holds the bit at a 1. The data is compared by indirect reference through the address buffers, while all other address bits are varied through the EPROM address space. If data never varies through all comparisons, the address bit is reported to be in error. It is possible, but unlikely, that data in the EPROMs does not vary with respect to a given address bit.

A failure of the address bits will probably cause indications of data bit failure. Therefore, if both data and address bit failures are reported, it is best to suspect addressing failures as the real error.

The user should have prior knowledge of the EPROM contents. Otherwise, it may be difficult to distinguish indicated errors from real ones. Error indications, in this case, are based on the expectation that a given bit in a data word will vary from word to word.

Test 4 checks the EPROM contents for the following pattern:

ADDRESS	DATA
XXXX XX00 0000 0000_2 thru XXXX XX11 1111 1100_2	0000_{16}
XXXX XX11 1111 1110_2	$FFFF_{16}$

25.2 OPERATOR INTERFACE

EROMBT runs under the DOCS (refer to Section III for a detailed explanation of the DOCS versions).

25.2.1 AVAILABLE VERBS. EROMBT supports the verbs listed in table 25-1.

25.2.2 STANDARD INITIALIZATION. In addition to the initialization questions required by the DOCS, EROMBT also asks the following initialization questions:

ENTER BEGINNING ADDRESS DEFAULT = 0000 -
ENTER ENDING ADDRESS DEFAULT = 0000 -
AREA OF EROM TO BE TESTED XXXX TO XXXX - (X's user supplied)
ENTER EROM BOARD INTERRUPT LEVEL DEFAULT = 0003 -

25.2.3 PROGRAMMER-PANEL INITIALIZATION. In addition to the question IDLE ON ERRORS? the programmer-panel version asks the questions shown in the paragraph 25.2.2.

Table 25-1. EROMBT Verbs

Verb	Function
IT	Initialize the test.
EA	Execute Tests 1, 2, and 3 in order and return to the verb decoder.
LA	Execute Tests 1, 2, and 3 and loop through this test sequence.
E1-E4	Execute a single test once (Tests 1 through 4).
L1-L4	Execute a single test and loop (Tests 1 through 4).

25.3 SYSTEM REQUIREMENTS

In addition to the hardware required by the DOCS, EROMBT also requires that the unit under test, an EPROM memory board, be present.

25.4 ERROR MESSAGES AND NUMBERS

Table 25-2 lists the error messages output by EROMBT.

Table 25-2. EROMBT Error Messages and Numbers

Number	Message	Explanation
1	***ERROR*** - BEGINNING ADDRESS IS NOT LESS THAN ENDING ADDRESS	This is an initialization error. Reinitialize the test.
2	***ERROR*** - BEGINNING ADDRESS EQUAL ENDING ADDRESS	This is an initialization error. Reinitialize the test.
3	***ERROR*** - BEGINNING AND ENDING ADDRESS 8K APART	This is an initialization error. The maximum amount of memory on an EPROM board is 8K. Reinitialize the test.
4	***ERROR*** - SUBTEST L EXPECTED INTERRUPT ON WRITING TO ROM DID NOT OCCUR; POSSIBLE CAUSES: 1) INCORRECT INTERRUPT LEVEL SET IN INITIALIZATION 2) TEST AREA IS NOT IN ROM 3) ROM BOARD INTERRUPT MALFUNCTION	Before deducing an EPROM board failure, the user should take care to eliminate the other possibilities.
5	***ERROR*** - SUBTEST 2 SUSPECTED BITS IN ERROR (MSB TO LSB) BITS IN ERROR - 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,	This is the data error message. If an address message (message 6) also occurs, it could cause the malfunction indicated by error 5.
6	***ERROR*** - SUBTEST 3 SUSPECTED ADDRESS BITS IN ERROR (MSB TO LSB) BIT IN ERROR - 4,5,6,7,8,9,A,B,C,D,E,F	This indicates a probable addressing error.
7	***ERROR***-SUBTEST 4 EPROM DATA COMPARISON ERROR AT ADDRESS XXXX EXPECTED DATA: YYYY ACTUAL DATA: ZZZZ	Address XXXX contains ZZZZ. It should contain YYYY. (YYYY = 0000 or FFFF)

SECTION XXVI

911 VIDEO DISPLAY TERMINAL TEST (CRT911)

26.1 INTRODUCTION

The CRT911 tests a Model 911 Video Display Terminal (consisting of keyboard, a VDT controller, a video display unit and interconnecting cables). If a VDT controller contains logic for two controllers, the board responds to two CRU base addresses and the diagnostic must be executed separately for each controller using the appropriate CRU base address for each execution. The test consists of four parts. The first three parts test the CRT911 CRU board and require no operator intervention. The fourth part is interactive and requires extensive operator intervention.

26.2 OPERATOR INTERFACE

CRT911 runs under the DOCS. Refer to Section III for a detailed description of the DOCS characteristics.

26.2.1 AVAILABLE VERBS. CRT911 supplies the verbs listed in table 26-1. The CU and CD verbs are special purpose verbs included in CRT911. The following paragraphs discuss CU and CD.

Table 26-1. CRT911 Verbs

Verb	Action
IT	Initialize the CRT Test
E1	Execute Part 1 (Self-Test)
E2	Execute Part 2 (Memory)
E3	Execute Part 3 (Cursor Address)
E4	Execute Part 4 (Manual Intervention)
EA	Execute all four parts in order
L1	Loop on Part 1
L2	Loop on Part 2
L3	Loop on Part 3
LA	Loop on Parts 1-3
LB	Loop on Parts 1-3, first on one half of controller and then on the other half. Terminate with "@".
PE	Print the number of errors
CU	Cursor Move
CD	Modify Memory/Scope Loop

26.2.1.1 CU (Cursor Move). The Cursor Move verb moves the cursor from any CRT memory location to any other memory location (displayed or not displayed). Initially, the verb performs the following:

1. Resets Self-Test and clears Keyboard Interrupt Enable.
2. Enable video and nonblinking cursor.

Cursor commands are used with the CU verb to specify the move. The cursor commands are:

- U - Up one row
- D - Down one row
- L - Left one character position
- R - Right one charcter position
- C - To column zero
- Z - To row zero
- A - Direct cursor address
- I - Iterate direct cursor address

The CU verb is entered in response to the diagnostic message, VERB?-, by typing CU-x, where x is any of the cursor commands except A and I. The cursor commands, A and I, are entered as follows:

1. Wait for message, VERB?-.
2. Enter CU-A or CU-I.
3. Wait for message -nnnn (hexadecimal value of current cursor location).
4. Enter -mmmm (hexadecimal value of desired cursor location).

The following are examples of the CU verb:

- VERB? -CU -U-
- VERB? -CU -A- -0377 -0000-
- VERB? -CU -A- -0377-0000-U- (Chained Commands).

Enter @ to terminate the CU verb.

All cursor commands except A and I affect only the cursor position and provide no additional information. Cursor commands A and I cause the current cursor address to be read and displayed and allow the operator to specify any CRT memory location as the new cursor location. Cursor command A permits a new entry after each cursor move operation. The cursor command I continually writes the operator selected cursor address into the controller register.

26.2.1.2 **CD (Cursor Data).** The CD verb modifies the data at the current cursor address (which may be any address in VDT memory). The CD verb can also activate scope loops. The CD verb has the following five commands:

- S- Read (or read and modify) memory character at current cursor address.
- R- Read (or read and modify) memory character at current cursor address and move cursor right one position.
- F- Free form input mode from the keyboard to the screen.
- SL- Modify and read data at cursor address. Then scope loop on memory write operation using the operator input data.
- C- Exercise CRU interface in a scope loop.

26.2.2 **STANDARD INITIALIZATION.** Table 26-2 lists the initialization questions for CRT911.

26.2.3 **PROGRAMMER-PANEL INITIALIZATION.** In addition to the IDLE ON ERRORS? question, the programmer-panel version asks the questions listed in table 26-2.

26.3 SYSTEM REQUIREMENTS

In addition to the hardware required by the DOCS, a 911 video display terminal kit (part number 936477) is also required.

Table 26-2. CRT911 Initialization Questions

Message	Comment
CRT UNDER TEST CRU BASE ADDRESS DEFAULT = 0100	Enter the CRU base address if it differs from default.
CRT UNDER TEST INTERRUPT DEFAULT = 000A	Enter the interrupt level of CRT under test if it differs from default.
AC LINE FREQ 50 OR 60 HZ DEFAULT = 60HZ	If unit is 50 Hz model, enter 50 Hz.

26.4 MESSAGES

Table 26-3 lists the normal messages produced by CRT911 and table 26-4 lists CRT911 error messages.

Table 26-3. CRT911 Normal Messages

PART 1 CRT CONTROLLER SELF TEST
PART 1 DONE

PART 2 CRT MEMORY TEST
PART 2 DONE

PART 3 CURSOR ADDRESS TEST
PART 3 DONE

Table 26-3. CRT911 Normal Messages (Continued)

PART 4 MANUAL INTERVENTION TEST

** TERMINAL READY **

UNPLUG THE CRT CABLE TO TEST TERMINAL READY–

TERMINAL NOT READY

VIDEO ENABLE TEST VERIFY THE SCREEN IS BLANK–

CURSOR ENABLE TEST VERIFY THE CURSOR IS OFF–

BLINKING CURSOR TEST VERIFY THE CURSOR IS BLINKING–

RIPPLE DUMP TEST VERIFY THE RIPPLE DUMP PATTERN–

VIDEO INTENSITY TEST VERIFY THE INTENSITY SHIFT–

CHARACTER DISPLAY TEST VERIFY THE CORRECT CHARACTER IS DISPLAYED

```
THE CHARACTER SHOULD BE:  -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -

  -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -
  -   - ! - " - # - $ - % - & - ' - ( - ) - * - + - , - - - . -
/ - 0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - : - ; - < - = - > -
? - @ - A - B - C - D - E - F - G - H - I - J - K - L - M - N -
O - P - Q - R - S - T - U - V - W - X - Y - Z - [ - \ - ] - ^ -
_ - ` - a - b - c - d - e - f - g - h - i - j - k - l - m - n -
o - p - q - r - s - t - u - v - w - x - y - z - { - | - } - ~ -
 _
ENTER KEYS IN ANY ORDER ON  -CRT UNDER TEST-
```

TYPE ESC OR SPACE BAR TO END TEST

ENTER THE KEYBOARD KEYS IN THE ORDER DISPLAYED ON THE CRT SCREEN

INTERRUPT TEST DEPRESS ANY KEYBOARD KEY

KEYBOARD RATE TEST DEPRESS THE REPEAT KEY AND ANY OTHER KEY FOR 4 SEC.

RATE = 000A * CHAR/SEC (SHOULD BE 6-E)

BEEPER TEST VERIFY THE BEEPER IS WORKING

DO YOU WANT TO REPEAT THE BEEPER TEST? DEFAULT = 01

*typical value

Table 26-4. CRT911 Error Messages

Number	Text
1	**ERROR **VIDEO(FULL) VALUE NON-ZERO VIDEO(FULL) EXPECTED XXXX ACTUAL XXXX
2	**ERROR ** VIDEO(BLANK) VALUE NON-ZERO VIDEO(BLANK) EXPECTED XXXX ACTUAL XXXX
3	**ERROR **VIDEO(H-SYNC) COUNT NOT IN CORRECT RANGE VIDEO(H-SYNC) EXPECTED XXXX–XXXX ACTUAL XXX
4	**ERROR ** H-SYNC COUNT NOT IN CORRECT RANGE H-SYNC EXPECTED XXXX–XXXX ACTUAL XXXX
5	**ERROR **V-SYNC NOT CORRECT VALUE V-SYNC EXPECTED XXXX–XXXX ACTUAL XXXX
6	**ERROR ** AUDIO PULSE COUNT NOT IN CORRECT RANGE AUDIO EXPECTED XXXX–XXXX ACTUAL XXXX
7	CHAR. TIMING ERROR
8	CHAR. PARITY ERROR
9	ERROR, Z* IS NOT THE PREVIOUS WORD SELECT
A	MEMORY ERROR PART XX DATA SENT XX DATA REC. XX ACTUAL ADDR XXXX EXP ADDR XXXX
B	CURSOR ERROR PART XX EXP ADDR XXXX ACT ADDR XXXX
C	TERMINAL NOT READY
D	TERMINAL READY
E	PARITY ERROR ON KEY # XX
F	PARITY ERROR ON CHAR. XX
10	ERROR NO INTERRUPT WAS GENERATED WITHIN 30 SECONDS
11	MEMORY ERROR IN SCROLL TEST
12	ERROR KB TEST CHAR REC. XX CHAR EXP. XX

*NOTE: Z = 0 OR 1

SECTION XXVII

FLOPPY DISK TEST (FLPDSK)

27.1 INTRODUCTION

FLPDSK tests the 990 computer floppy disk drive and controller. FLPDSK consists of 11 subtests:

Subtest Number	Name
1	Controller Self Test
2	Microcode and Basic Command Test
3	Basic Head Movement Test
4	Read ID Test
5	Write/Read Test
6	Zero Fill and Sequential Sectoring Test
7	Revolution Time Test
8	Arm Slot Test
9	Media Test
10	Power Fail Test
11	Drive Not Ready and Write Protect Test

For Parts 1 and 2, a floppy disk drive is not required since only the controller is tested. If errors occur during these two parts, there is a high probability the remaining parts will not run correctly. It takes approximately 3 minutes to run Parts 1-9 of the test. If errors occur it takes longer.

27.2 OPERATOR INTERFACE

This test operates under the DOCS. Refer to Section III for an explanation of the DOCS versions.

27.2.1 AVAILABLE VERBS. In addition to the verbs supplied in the DOCS, FLPDSK also supplies the verbs listed in table 27-1. The first six verbs are self explanatory. The next nine verbs are utility verbs and are discussed in paragraphs 27.2.1.1 through 27.2.1.9.

Table 27-1. FLPDSK Verbs

Verb	Function
IT	Initialize test
EA	Execute test 1-10 once
LA	Execute test 1-9 and repeat
ET	Execute an individual test once (1-11)
LT	Execute an individual test and repeat (1-11)
CM	Execute tests 1-9 on all available units
IC	Issue command
IM	Issue multiple commands

Table 27-1. FLPDSK Verbs (Continued)

Verb	Function
LO	Loop or command(s)
CD	Compare two blocks of data
FD	Format disk
OT	Off track test
AL	Head alignment

27.2.1.1 **IC Verb.** The IC verb is used to issue a command to the floppy disk controller from a memory address input by the operator. The operator also specifies whether or not status will be checked after the command is completed. A status error occurs on write commands because the floppy disk controller has been interrupted while transferring data. An example of the use of the IC verb is shown below with the operator's entries underscored.

VERB? – IC ADDRESS – 5000 CHECK STATUS? – 1

In this example the command is located at memory address 5000_{16} and status will be checked.

27.2.1.2 **IM Verb.** The IM verb is used to issue multiple commands to the floppy disk controller starting at the memory address input by the operator. The operator is asked the number of commands, the address of the first command, and if status should be checked. An example of the IM verb is shown below. The operator's entries are underscored.

VERB? – IM # OF CMDS – 4 ADDRESS – 5000 CHECK STATUS? – 0

Memory Location (Hexadecimal)	Data	
5000	0000	– Command 1 = Reset
5002	2000	– Command 2 = Restore
5004	7001	– Command 3 = Write
5006	5010	
5008	0010	
500A	B000	– Command 4 – Stop

The instruction shown above will do the following: reset the controller, move the head to track 0, write 8 words or 16 bytes from memory location 5010_{16}. A STOP command should be issued following write commands. Status will not be checked on any of the instructions because of the negative answer given.

27.2.1.3 **LO Verb.** The LO verb is used to issue one or more commands to the floppy disk controller and repeat the sequence. The operator is asked the number of commands, the address of the

first command, and whether or not the status should be checked. Shown below is an example of the LO verb. The operator's entries are underscored.

VERB? – LO # OF CMDS – 4 ADDRESS – 5000 CHECK STATUS? – 1

Memory Location (Hexadecimal)	Data	
5000	D000	– Command 1 = Reset
5002	2000	– Command 2 = Restore
5004	C020	– Command 3 = Physical Seek to track 20
5006	C040	– Command 4 = Physical Seek to track 40

The instruction set shown above will do the following: reset the controller, move the head to track 0, move the head to physical track 20_{16}, move the head to physical track 40_{16} and repeat the sequence. The status will be checked after each command is completed. The loop count will be output on the interactive device and will appear on the computer programmer panel.

27.2.1.4 **CD Verb.** The CD verb is used to compare two blocks of data that are located in memory. The data is compared on a byte basis. The operator is required to make three entries: the memory address of the first block of data, the memory address of the second block of data, and the number of bytes of data. If some of the data does not compare, an error message will be printed indicating the number of bytes that did not compare. An example of the CD verb when some data does not compare is shown below. Operator entries are underscored.

VERB? – CD ADDR 1 = 5000 ADDR 2 = 6000 LENGTH = 2C

ERROR PART CD

001A BYTES OUT OF 002C DID NOT COMPARE

27.2.1.5 **FD Verb.** The FD verb is used to format the diskette. The test parts of the floppy disk test expect the diskette to be formatted before testing starts. The FD verb attempts to format all 77 tracks on the diskette, but will allow up to three bad tracks which cannot be formatted. An example of the FD verb is shown below.

VERB? – FD ARE YOUR SURE? – 1

ENTER # OF HEADS PER DISK – 1

The question ARE YOU SURE? is asked in order to remind the operator to not specify a good diskette which contains valuable data in the drive. If the diskette is so bad that there are over three bad tracks the following message is printed.

UNABLE TO FORMAT 4 TRACKS ON ONE SIDE OF FLOPPY

27.2.1.6 **OT Verb.** The OT verb is issued to check the drive, to make sure the drive will read data at the limits of the specification. The test requires that an alignment diskette (Shugart SA

120-1) be run. The test reads either track 47 or 49 (operator choice) and will print an error message if the data is not what was expected.

An example of the OT verb is shown below.

```
VERB? – OT
ENTER TEST TRACK # (EITHER 47 OR 49 HEX)–47
ENTER DESIRED NUMBER OF LOOPS–1
OFF TRACK DATA TEST START
```

27.2.1.7 **AL Verb.** The AL verb is used for alignment. It positions the head to the track specified by the operator and does an unformatted read until interrupted by the operator from the keyboard. The alignment diskette should be used (Shugart SA 120-1) to complete head alignment. For alignment information, consult the *Shugart Maintenance Manual*.

27.2.2 **STANDARD INITIALIZATION.** In addition to the DOCS questions, the default test values of the floppy disk CRU address, floppy disk interrupt level, unit under test and other available units are displayed by the program. All of the defaults may be changed by the operator and the changed value becomes the new default. Table 27-2 lists the initialization questions.

27.2.3 **PROGRAMMER-PANEL INITIALIZATION.** In addition to the question IDLE ON ERRORS? the programmer-panel version asks the same questions in table 27-2. For additional information on the operation in the front panel mode refer to Section III on the DOCS tests.

27.3 SYSTEM REQUIREMENTS

In addition to the system requirements of DOCS, FLPDSK requires the following:

- At least 8K words of memory
- Floppy disk controller
- Floppy disk drive
- A write protected diskette
- A formatted diskette
- An alignment diskette

27.4 MESSAGES

Table 27-3 lists the messages output by FLPDSK.

Table 27-2. FLPDSK Initialization Questions

Question	Comment
ENTER FLOPPY DISK CRU BASE, DEFAULT = 0080	
ENTER FLOPPY DISK INT LEVEL, DEFAULT = 07	
UNIT TO BE TESTED (1, 2, 3, 4), DEFAULT = 01	
ENTER OTHER AVAILABLE UNITS (TERMINATE WITH FF)	When unit numbers of other available units (1, 2, 3, 4); enter value of "FF" to end string of numbers. Used by CM verb only.
DO YOU WANT TO RUN TEST WITH INTERRUPTS? DEFAULT = 00	Enter 01 to run test with interrupts.
DO YOU WANT TO USE RETRYS IN MICROCODE? DEFAULT = 01	Enter 00 if you do not want to use retrys.
IS THE COMPUTER RUNNING ON 60 CYCLE CURRENT? DEFAULT = 01	Enter 00 if computer uses 50 Hz.

Table 27-3. FLPDSK Messages

Message Number	Text	Description
None	FLOPPY DISK TEST VERSION MM/YY *X	Header Message.
None	PART XX	Header Message.
None	PART XX DONE	Header Message.
None	# of CMDS–	Initialization question for LO and IM verbs. Action: Enter number of commands in hexadecimal.
None	ADDRESS–	Initialization question for IC, IM, and LO verbs. Action: Enter memory address of first command.
None	CHECK STATUS?–	Initialization question for IC, IM and LO verbs. Action: Enter 0 for no, 1 for yes.

Table 27-3. FLPDSK Messages (Continued)

Message Number	Text	Description
None	LOOP COUNT = XXXX	Header message to display loop count for LA and LT verbs.
None	ADDR1 =	Initialization message for CD verb. Action: Enter memory address of first data block.
None	ADDR2 =	Initialization message for CD verb. Action: Enter memory address for second data block.
None	LENGTH	Initialization message for CD verb. Action: Enter the number of bytes in hexadecimal to be compared.
None	ARE YOU SURE?	Initialization message for FD verb. Action: Enter 1 for yes, 0 for no.
None	ENTER # OF HEADS PER DISK	Initialization message for FD verb. Action: Enter 1 or 2 depending on diskette.
None	TEST #	Initialization message for LT and ET verb. Action: Enter a decimal digit 1-11 to execute a test.
None	**TEST NOT AVAILABLE**	Error message when trying to execute LT or ET verb.
None	UNIT UNDER TEST = XX	Information message used by CM verb when starting on a new unit.
None	DOES THIS SYSTEM HAVE BATTERY SUPPORT?	Question asked during power fail test. Action: Enter 1 if system has battery support, 0 for no.
None	DO NOT RUN PART 10- - NO BATTERY SUPPORT- -	Warning because of zero answer to previous question.

Table 27-3. FLPDSK Messages (Continued)

Message Number	Text	Description
None	WAS PART 9 RUN LAST?	Question asked during power fail test. This is the next question after answering 1 to previous question. Action: Enter 1 if part 9 was run last, 0 if no.
None	RUN PART 9 !!	Warning because of zero answer to previous question.
None	HAS SYSTEM POWER BEEN CYCLED?	Last question asked during power fail test. This is the next question after the previous question. Action: Enter 1 if power has been cycled, 0 if no.
None	CYCLE SYSTEM POWER!	Lead through message. Action: Cycle system power.
None	REMOVE DISKETTE FROM DRIVE AND CLOSE DOOR	Lead through message. Action: Remove diskette from drive being tested and close the door and respond.
None	INSERT WRITE PROTECTED DISKETTE BUT DO NOT CLOSE THE DOOR	Lead through message. Action: Insert a write protected diskette and respond.
None	CLOSE THE DOOR ON DRIVE	Lead through message. Action: Close the door and respond.
1 RAM XXXX	FAILURE COUNTS I/O CRC CRC WT MARK XXXXXXXX XXXX XXXX	Summary of error counts from doing 1000 resets.
WRITE XXXX	MICRO XXXX	
2	LED NUMBER XX FAILED TO COME ON	One of 3 red LEDs failed to come on.
3	ALL 4 LED'S DID NOT COME ON	One of the 4 LEDs (1 green and 3 red) failed to come on at same time.
4	RECEIVED EQUIPMENT CHECK ERROR WITH STATUS = XXXX	Fatal error received during attempt to do 1000 resets.

Table 27-3. FLPDSK Messages (Continued)

Message Number	Text	Description
5	CHECK SUM ERROR EXP=0000 REC = XXXX	The microcode did not sum to zero. This indicates that some microcode is incorrect.
6	XXXX WORDS DID NOT COMPARE ON RAM WRITE/READ WITH DATA = XXXX	Data (AAAA or 5555) was written into the controller RAM but not read out correctly.
7	RAM WRITE/READ ERROR REC = XXXX EXP = XXXX	Data (random) was written into controller RAM but not read out correctly.
8	EXPECTED 0000 AFTER CLEAR STATUS BUT RECEIVED XXXX	After issuing a clear, status was not cleared.
9	DID NOT REACH TRACK 0 AFTER RESTORE COMMAND	After issuing a restore command the head was not at track 0.
A	EXPECTED TRACK XX BUT READ XX AFTER STEP COMMAND	The head was not at the correct track after a step head command.
B	ERROR WHEN WRITING TO TRK XX SECT XX STATUS = XXXX NO XFER RDY BIT	Transfer ready bit did not come on after issuing write command.
C	ERROR WHEN READING FROM TRK XX SECT XX STATUS = XXXX NO XFER RDY BIT	Transfer ready bit did not come on after issuing read command.
D	DATA EXPECTED XXXX RECEIVED XXX WHILE READING TRACK XX SECTOR XX	Data did not compare after reading it off disc.
E	CONTROLLER BUFFER WAS NOT ZERO FILLED WITH SECTOR LENGTH = XX	Data read back shows zero fill works incorrectly.
F	A SEQUENTIAL WRITE OF 128 BYTES WROTE DATA INTO NEXT SECTOR	Data was written into the next sector when only one sector was written.
10	A SEQUENTIAL WRITE TO LAST SECTOR ON TRACK DID NOT INCREMENT HEAD TO NEXT TRACK	Controller failed to auto increment head during read.
11	A SEQUENTIAL READ FROM LAST SECTOR ON TRACK DID NOT INCREMENT HEAD TO NEXT RRACK	Controller failed to auto increment head during write.

Table 27-3. FLPDSK Messages (Continued)

Message Number	Text	Description
12	DATA DID NOT COMPARE ON SEQUENTIAL SECTORING/TRACK INCREMENT TEST	Controller did increment head but data read was not same as that written.
13	A COMPLETE REVOLUTION TOOK LONGER THAN 174 MS	The drive is turning the diskette too slow.
14	A COMPLETE REVOLUTION TOOK LESS THAN 158 MS	The drive is turning the diskette too fast.
15	DATA DID NOT COMPARE ON TRACK XX POSITION BEFORE WRITE XX POSITION BEFORE READ XX	Data could not be read after seeking to track from opposite direction that data was written on from.
16	TIMED OUT WHILE WRITING TO TRACK = XX STATUS XXXX	Transfer ready bit did not come on after issuing a write command.
17	TIMED OUT WHILE READING FROM TRACK = XX STATUS XXXX	Transfer ready bit did not come on after issuing a read command.
18	DATA DID NOT COMPARE WHILE READING TRACK XX EXP = XXXX REC = XXXX	Data did not compare after reading data that was written from other direction.
19	POWER WAS NOT CYCLED IN 1 MINUTE	Operator did not cycle power in power fail test.
1A	BAD DRIVE STATUS WITH DISKETTE OUT EXP = 8004 REC = XXXX	Incorrect status was received with diskette out of drive.
1B	BAD DRIVE STATUS WITH DOOR OPEN EXP = 8004 REC = XXXX	Incorrect status was received with door open on drive.
1C	BAD STATUS ON WRITE PROTECT EXP = 800 REC = XXXX	Incorrect status was received with write protect diskette in drive.
28	XXXX BYTES OUT OF XXXX DID NOT COMPARE	Error in CD verb.
29	UNABLE TO FORMAT 4 TRACKS ON ONE SIDE OF FLOPPY	Error in attempts to format diskette.
2A	TIMED OUT WHILE X-FERRING DATA STATUS = XXXX CMD = XXXX	Data was not transferred in allotted time by command issuer subroutine.

Table 27-3. FLPDSK Messages (Continued)

Message Number	Text	Description
2B	TIMED OUT AFTER ISSUING COMMAND XXXX STATUS XXXX WITH INTERRUPT	Timed out error in command issuer using interrupts.
2C	TIMED OUT AFTER ISSUING COMMAND XXXX STATUS XXXX WITHOUT INTERRUPT	Timed out error in command issuer with interrupts disabled.
2D	** STATUS ERROR ** COMMAND ISSUED = XXXX STATUS = XXXX	Status error reported by status checker routine.
2E	UNEXPECTED FLOPPY DISK INTERRUPT AT LOC XXXX	Unexpected interrupt reported by interrupt service routine
2F	DATA COMPARE ERROR TRACK SECT HEAD WCNT EXP RCED XXXX XX XX XXXX XXXX XXXX	Data miscompare in OT test
none	ENTER TEST TRACK # (EITHER 47 OR 49 HEX)	
none	ENTER THE DESIRED # OF LOOPS	
none	ILLEGAL LOOP COUNT	
none	THE RADIAL ALIGNMENT ROUTINE IS NOW RUNNING TO STOP DEPRESS @ KEY ON INTERACTIVE DEVICE	Reading track 26
none	THE AZIMUTH ALIGNMENT ROUTINE IS NOW RUNNING TO STOP DEPRESS @ KEY ON INTERACTIVE DEVICE	Reading track 4C
none	SELECT A TRACK****00 OR 01 OR 02	
none	NON ALLOWED TRACK***TRY AGAIN	
none	THE FLAG ALIGNMENT ROUTINE IS NOW RUNNING TO STOP DEPRESS @ KEY ON INTERACTIVE DEVICE	
none	AN ALIGNMENT DISKETTE IS NEEDED TO USE THIS VERB	
none	THIS TEST MUST BE RUN WITHOUT INTERRUPTS TO RUN THE OTHER TESTS USING INTERRUPTS RUN THE IT VERB	

SECTION XXVIII

DIGITAL TO ANALOG CONVERTER TEST (DACHK)

28.1 INTRODUCTION

The DACHK tests the 990 half-card digital/analog converter (part number 949029-000X) in a 990 computer system. DACHK consists of three parts.

- Part 1 tests the channel decode and reset functions.
- Part 2 determines if each bit (independently) can be set to a -1- without setting the other bits to a -1-.
- Part 3 determines if each bit (independently) can be set to a -0- without setting the other bits to a -0-.

28.2 OPERATOR INTERFACE

DACHK runs under the DOCS. However, it will not run with the front panel version of the DOCS. Refer to Section III for a detailed explanation of the DOCS characteristics.

28.2.1 AVAILABLE VERBS. In addition to the verbs supplied in the DOCS, DACHK also supplies the verbs listed in table 28-1.

28.2.2 STANDARD INITIALIZATION. In addition to the initialization questions required by the DOCS, DACHK asks the initialization questions listed in table 28-2.

Table 28-1. DACHK Verbs

Verb	Function
IT	Initialize test
E1	Execute part 1-channel decode test
E2	Execute part 2-ripple -1- bit test
E3	Execute part 3-ripple -0- bit test
EA	Execute parts 1-3 in order
LA	Loop on parts 1-3 in order
CB	Calibrate D/A
DA	Operator voltage/current request

28.3 SYSTEM REQUIREMENTS

In addition to the hardware described in the DOCS description on D/A converter, a D/A Test Fixture (part number 938155-0001) and a 5-digit DVM (digital voltmeter) are required by this test. DACHK requires 16K bytes of memory.

28.4 ERROR MESSAGES

Only the operator can detect improper conditions during the test. Therefore, there are no error messages in DACHK.

Table 28-2. DACHK Initialization Questions

Messages	Comments
ENTER D/A CRU ADR (DEF=0020)–	Enter the CRU address if it differs from the default; otherwise, enter a space.
ENTER DISPLAY MODE (0=VOLTS, 1=MA) DEF=00 (NOTE: IF READING VOLTS IN CURRENT MODE; ENTER 0)–	Even if the D/A is being used in the current mode, if measurements are made by the voltage drop across a 500 ohm resistor, then the voltage mode(0) should be chosen.
ENTER RANGE (0=5V, 1=10V) DEF=01–	Enter 0 if 0-5V or -5V through +5V. Enter 1 if 0-10V or -10V through +10V.
ENTER # OF CHAN; DEF=4(D)-	The (D) indicates the program is looking for a decimal response.
ENTER CODE (NB, OB or TC) DEF=TC–	NB=Natural Binary; OB=Offset Binary; TC=Two's Complement

SECTION XXIX

ANALOG TO DIGITAL CONVERTER TEST (ADCHK)

29.1 INTRODUCTION

The ADCHK tests the 990 half-card analog/digital converter (part number 949024-000X) in a 990 computer system. ADCHK consists of seven parts.

- Part 1 is an interface test that tests the control functions.
- Part 2 tests the channel address decode.
- Part 3 checks that adjacent channels are not shorted together.
- Part 4, 5, and 6 check the accuracy of the A/D converter at the Max, Mid, and Min points within the range of the A/D.
- Part 7 checks the response time of the A/D by taking 20,000 samples in approximately 1 second.

29.2 OPERATOR INTERFACE

ADACHK runs under the DOCS. However, it will not run with the front panel version of the DOCS. Refer to Section III for a detailed explanation of the DOCS characteristics.

29.2.1 AVAILABLE VERBS. In addition to the verbs supplied in the DOCS, ADCHK also supplies the verbs listed in table 29-1. The first ten verbs are self explanatory. The next eight verbs are discussed in paragraph 29.2.1.1 through 29.2.1.8.

29.2.1.1 CB Verb – Calibrate the A/D. This verb is used to calibrate a new A/D board or an A/D board that has been configured for a different mode. The verb is used to set offset at a near-zero voltage and to set range at a near-maximum voltage. After each calibration step the verb prints out:

'OFFSET OR GAIN: "OFFSET (0) OR GAIN (1), DEFAULT = 0"

The operator selects the next test step and testing will continue until the operator inputs an @; then the program prints out

'VERB?'

29.2.1.2 DA Verb – Sample a Channel Every 6*N Seconds. The verb will print the voltage read on an operator specified channel every 0, 6, 12, 18, . . . or 594 seconds continuously until the operator inputs an @. The multiple (n) of 6 seconds is determined by the operator. The verb asks the questions:

"ENTER CHAN (D)"
ENTER # OF 6 SEC PERIODS BETW. SAMPLES (D)"

29.2.1.3 CL Verb – Calibrate Loop. This verb takes samples at high speed with a scope display for very close tolerance calibrations. The verb is terminated by an @.

Table 29-1. ADCHK Verbs

Verb	Function
IT	Initialize test
E1	Execute part 1- interface test
E2	Execute part 2- channel address test
E3	Execute part 3- channel short test
E4	Execute part 4- max. accuracy test
E5	Execute part 5- mid. accuracy test
E6	Execute part 6- min. accuracy test
E7	Execute part 7- response time test
EA	Execute parts 1-7 in order
LA	Loop on parts 1-7 in order
CB	Calibrate A/D
DA	Sample 1 channel every 6*N seconds
CL	Take samples as fast as possible (for scope)
RC	Sample every channel
SO	Set 1 bit to a one (1)
SZ	Set 1 bit to zero (0)
TB	Test 1 bit continuously
BL	Loop on SBO, SBZ for 1 bit

29.2.1.4 **RC Verb – Read All Channels.** This verb takes one sample of each channel and prints the results.

29.2.1.5 **SO Verb – Set a Bit to One.** This verb sets a bit to 1 and then returns to VERB?.

29.2.1.6 **SZ Verb – Set a Bit to Zero.** This verb sets a bit to 0 and then returns to VERB?.

29.2.1.7 **TB Verb – Test One Bit.** This verb tests one bit continuously and prints out its value until terminated by an @.

29.2.1.8 **BL Verb – SBO, SBZ Loop.** This verb alternately sets a particular bit to 0 and to 1. The particular bit is specified by the program. The verb is terminated by an @.

29.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by the DOCS, ADCHK asks the initialization questions listed in table 29-2.

Table 29-2. ADCHK Initialization Questions

Messages	Comments
ENTER A/D TOLERANCE (DEF= 0001) –	This tolerance is used in tests 4-6.
ENTER PWR. FREQ. (0=60HZ, 1=50HZ) (DEF- 0) –	Select the line frequence which corresponds to the local ac power line. (used in test 7)
ENTER A/D CRU ADR (DEF= 0020) –	Enter the CRU base address if it differs from the default; otherwise, enter a space.
ENTER A/D INT. LVL (DEF= 0F) –	Enter the A/D interrupt if it differs from the default; otherwise, enter a space.
ENTER RANGE (0=5V, 1=10V) DEF= 01 –	Enter 0 if 0-5V or -5V through +5V. Enter 1 if 0-10V or -10V through +10V.
ENTER # OF CHAN; DEF= 64(D) –	The (D) indicates the program is looking for a decimal response.
ENTER CODE (NB, OB OR TC) DEF= TC –	NB = Natural Binary; OB = Offset Binary; TC = Two's Complement (See A/D specifications for explanation of codes, (part number 949024).

29.3 SYSTEM REQUIREMENTS

In addition to the hardware described in the DOCS description, an A/D Test Fixture (part number 938153-0001) and an Analogic DC Voltage Standard Model AN3100 (or equivalent) are required to test the A/D board. ADCHK requires 16K bytes of memory.

29.4 ERROR MESSAGES AND NUMBERS

Table 29-3 lists the error messages output by ADCHK.

Table 29-3. ADCHK Error Messages

Number	Message
11	'ERROR 1.1 CHAN XX; EXP DATA =XXXX' 'RECEIVED = YYYY'
12	'CAN NOT SET EXTERNAL TRIGGER'
13	'CAN NOT RESET EXT TRIG'
14	'CAN NOT SET INTERRUPT MASK'
15	'CAN NOT RESET INTERRUPT MASK'
16	'CAN NOT RESET RDY F.F. BY ADDRESSING BIT 15'

Table 29-3. ADCHK Error Messages (Continued)

Number	Message
17	'CAN NOT RESET RDY F.F. BY ADDRESSING BIT 6'
18	'INTER MASK DID NOT PREV. INTER.'
19	'DID NOT GET EXPECTED INTER. FROM CPU STROBE'
1A	'EXTER. TRIG DID NOT PREV. CPU STROBE INT'
1B	'DID NOT GET EXT. TRIGED INTER'
1C	'DID NOT GET RDY (BIT 15)'
1D	'-RESET-INSTR. FAILED TO CLR BOARD'
	'DATA AFTER RESET = XXXX'
	'" " LDCR (ZERO) = YYYY'
1E	'LDCR (ZEROS) INSTR. FAILED TO CLR BOARD'
	'DATA AFTER-LDCR- = XXXX'
	'DATA AFTER RESET = YYYY'
1F	'DID NOT GET EXT. TRIG'
1G	'BIT 13 = 0 DID NOT PREVENT EXT STORBE'
21	'ERR 2.1; CHAN XX FAILS TO RESPOND'
22	'ERR 2.2; CHAN XX RESPONDS WHEN CHAN YY IS ADR'
31	'ERR 3.1; EXPECTED VOLTAGE NOT PRESENT ON CHAN XX'
32	'ERR 3.2; UNEXPECTED VOLTAGE PRESENT ON CHAN XX'
41	'CHAN X EXP DATA = YYYY'
	'HEX DATA DEC DATA NUMB.
	ZZZZ TWW.WWWW 1 ZZZZ TWW.WWWW 1 ZZZZ TWW.WWWW 1
70	'ERROR 7.0 RESP TIME XXX PER CENT SLOW - -'
	'YYYY SAMPLES TAKEN IN ZZZZ CLK CYCLES'
91	'A/D INT OCCURS WHEN DISABLED'
	'WP = XXXX PC - YYYY'

SECTION XXX

LOCAL MULTI-DROP MODULE TEST (LOCLIN)

30.1 INTRODUCTION

The LOCLIN runs on a 990 computer with a minimum of 16K bytes of memory. The LOCLIN verifies the correct operation of the Local Multi-Drop Module (part number 949087) and verifies that the module can communicate with another similar module.

The LOCLIN consists of 15 subtests:

Subtest Number	Name
1	10 Ms Timer Test
2	250 Ms Timer Test
3	ID Switch Test
4	Interrupt Logic Test
5	Loopback Test
6	Walking 1 Test
7	DTR/DSR Toggle Test
8	RTS/CTS Toggle Test
9	RTS/DCD Toggle Test
A	WRQ Test
B	Overrun Condition Test
C	Half Duplex/RRQ Interrupt Test
D	Self Test - 200 Baud
E	Self Test - 1200 Baud
F	Communications Test - 9600 Baud

For the execution of subtests 1-E, only one Local Multi-Drop Module is necessary. To execute subtest F (communcations test) another Local Multi-Drop Module must be connected via a twisted pair cable and located in another computer. If errors occur during the running of subtests 1-E, there is a high probability part F will not run correctly. It takes approximately 25 seconds to run subtests 1-E. If an error occurs, the test takes longer to execute.

When running subtest F, the standard message table will execute in approximately 50 seconds per slave. If a built message table is used, it takes approximately 1.5 to 3 seconds per message.

30.2 OPERATOR INTERFACE

LOCLIN runs under any of the DOCS versions. Refer to Section III for a detailed explanation of the DOCS versions.

30.2.1 AVAILABLE VERBS. In addition to the verbs supplied in the DOCS, LOCLIN supplies the verbs listed in table 30-1.

30.2.2 STANDARD INITIALIZATION. In addition to the initialization questions asked by the DOCS, the default test values of the Local Multi-Drop Module CRU address, the interrupt level, and the power line frequency are presented to the operator at the beginning of the test. These defaults may be changed by the operator, and the changed value becomes the new default. Table 30-2 lists the initialization questions.

30.2.3 PROGRAMMER-PANEL INITIALIZATION. In addition to the IDLE ON ERRORS? question, the programmer panel version of LOCLIN asks the questions listed in table 30-2.

For additional information on operating in the programmer panel mode, refer to the Section III on the DOCS.

Table 30-1. LOCLIN Verbs

Verb	Function
ET	Run an individual test once (1–F)
LT	Run an individual test and loop (1–F)
EA	Run tests 1-E once
LA	Run tests 1-E and loop
DS	Display CRU status of Local Line Module

Table 30-2. LOCLIN Initialization Questions

Question	Comment
CRU BASE FOR LOCAL LINE MODULE DEFAULT=0160 –	Enter CRU base address.
INT LEVEL FOR LOCAL LINE MODULE DEFAULT=000C –	Enter interrupt level.
IS POWER LINE FREQUENCY 60HZ –	Enter 01 if line frequency is 60 Hz; enter 00 if line frequency is 50 Hz.

30.3 SYSTEM REQUIREMENTS

In addition to the DOCS requirements, LOCLIN requires 1) that the CPU have 16K bytes of memory, and 2) that a Local Multi-Drop Communications Module be installed.

30.4 MESSAGES

Table 30-3 lists the messages output by LOCLIN.

Table 30-3. LOCLIN Messages

Message Number	Text	Description
None	LOCAL LINE MODULE TEST VERSION MM/DD/YY	Header Message
None	TEST 1	Header Message
None	TEST 2	Header Message
None	TEST 3	Header Message
None	TEST 4	Header Message
None	TEST 5	Header Message
None	TEST 6	Header Message
None	TEST 7	Header Message
None	TEST 8	Header Message
None	TEST 9	Header Message
None	TEST A	Header Message
None	TEST B	Header Message
None	TEST C	Header Message
None	TEST D	Header Message
None	TEST E	Header Message
None	TEST F	Header Message
None	COMPLETE	Header Message
None	IS THERE ANOTHER COMPUTER IN THE SYSTEM?	Question in test F

Table 30-3 LOCLIN Messages (Continued)

Message Number	Text	Description
None	***HOOK UP ANOTHER COMPUTER***	Program command. Action: Hook up another computer with Local Line Module.
None	ENTER COMPUTER ID (0-1F HEX)	Program command. Action: Enter ID.
None	SHOULD CRC ERROR CHECKING BE USED?	Question in test F. Action: Enter 1 for yes, 0 for no.
None	SHOULD THE NUMBER OF ERRORS BE CHECKED?	Question in test F. Action: Enter 1 for yes, 0 for no.
None	ENTER MAX ALLOWABLE ERRORS BEFORE HALTING	Program command, resulting from answering yes to preceding question. Action: Enter a number in hex.
None	IS THIS COMPUTER THE MASTER?	Question in test F. Action: Enter 1 for master, 0 for slave.
None	DO YOU WANT TO USE STANDARD MESSAGE TABLE?	Question is master section of test F. Action: Enter 1 for standard table, 0 to build own table.
None	ENTER MESSAGE LENGTH (00 TO TERMINATE)	Program command in master section of test F, resulting from answering 0 to preceding question.
None	ENTER NO OF TRIES–	Program command in master section of test F following previous command. Action: Enter a hex number.
None	ENTER SLAVES (0-1F) TERMINATE WITH FF	Program command in master section of test F. Action: Enter ID of slave computers.
None	SLAVE UNIT UNDER TEST HAS ID=XX	Status message in master section of test F.
None	ENTER MASTER ID–	Program command in slave section of test F. Action: Enter ID of master computer.

Table 30-3. LOCLIN Messages (Continued)

Message	Text	Description
None	DO YOU WANT EVERY MESSAGE NO. PRINTED?	Question in slave section of test F. Action: Enter 1 to print message number, 0 = nothing.
None	MSG=XXXX	Status message in slave section of test F, resulting from a positive answer to preceding question.

Part 1 Error Messages

1	NSF SET WHEN NOT ENABLED (IW1-12) ERROR.
2	NSF WAS NOT SET IN 15 SECONDS.
3	INTR SUM NOT BEING SET (IW1-14) ERROR.
4	NSF NOT RESET (IW1-12) ERROR.
5	INTR SUM NOT RESET (IW1-14) ERROR.
6	NSF WAS SET IN XX SECONDS.
7	NSF SET WHEN ENABLED BUT NOT TOGGLED.
8	CTS NOT SET IN 16.6 MS.

Part 2 Error Messages

9	TIMEXP NOT RESET (IW1-13) ERROR.
A	TIMER TO SHORT (IW1-13) ERROR.
B	TIMEXP NOT BEING SET (IW1-13) ERROR.

Part 2 Error Messages (Continued)

C	TIMER TO LONG (IW1-13) ERROR.
D	TIMER TO LONG AFTER RETRIGGER (IW1-13) ERROR.
E	TIMER SHORT AFTER RETRIGGER.
F	INTRUPT SUMERY NOT SET BY TIMEXP (IW1-14) ERROR.

Part 3 Error Messages

10	ID SWITCH READ XXXX EXPECTED XXXX (IW4-0-6) ERROR.

Table 30-3. LOCLIN Messages (Continued)

Message Number	Text	Description
11	UNEXPECTED INTERRUPT AT LEVEL XXXX ERROR.	
12	UNEXPECTED INTERRUPT FROM MODULE ERROR.	
13	MODULE DOES NOT INTERRUPT CPU ERROR.	
Part 5 and 6 Error Messages		
14	WROTE READ XOR OW2 XXXX XXXX ZZZZ OW3 XXXX XXXX ZZZZ	
Test Read Write Busy Subroutine Error Messages		
15	**ERROR RWBUSY NOT SET.	
Part 7 Error Messages		
16	DSR NOT BEING SET (IW1-0) ERROR.	
17	DSR NOT RESET (IW1-0) ERROR.	
Part 8 Error Messages		
18	CTS NOT BEING SET (IW1-1) ERROR.	
19	DCD NOT BEING SET (IW1-5) ERROR.	
Part A Error Messages		
1A	WRQ SET WHEN NOT ENABLED (IW1-15) ERROR.	
1B	WRQ NOT BEING SET (IW1-15) ERROR.	
Part B Error Messages		
1C	OVERRUN NOT SET (IW0-4) ERROR.	
1D	UNEXPECTED FRAMING ERROR (IW0-3) ERROR.	
1E	UNEXPECTED PARITY ERROR (IW0-12) PARITY DISABLED.	
Compare Data Subroutine Error Messages		
1F	XMIT DATA = XX RCV DATA = XX	ADDRESS = AAAA ADDRESS = BBBB

Table 30-3. LOCLIN Messages (Continued)

Message Number	Text	Description
DS Verb Error Message		
20	***ERROR IN DATA TRANSFER*** IW0=XXXX IW1=XX IW2=XX IW3=XX XMIT BUFF ADDRESS=XXXX BYTE CNT=XXXX RCV BUFF ADDRESS=XXXX BYTE CNT=XXXX	
Part C Error Messages		
21	RRQ GENERATED WHEN HALF DUPLEX SELECTED (OW4-6) ERROR	
Part F Error Messages		
22	CRC WAS NOT CORRECT ON MESSAGE XXXX WITH LENGTH=XXXX.	
23	TIME OUT ERROR EXPECTING XXXX CHAR BUT RECEIVED XXXX CHAR MSG NO XXXX	
24	ISR ERROR EXPECTING XXXX CHAR BUT RECEIVED XXXX CHAR MSG NO XXXX	

SECTION XXXI

AUTOMATIC CALL UNIT TEST (ACUTST)

31.1 INTRODUCTION

The ACUTST is used to test the automatic call unit for the 990 family of computers. ACUTST consists of five parts as follows:

Part 1 – Checks all state transitions of the ACU Module.

Part 2 – Checks all digits and control codes.

Part 3 – Checks the timing of the Abandon Call and Retry Status.

Part 4 – Checks frequencies of ACU tones.

Part 5 – Checks ACU dialing capability using communications interface and modem.

31.2 OPERATOR INTERFACE

ACUTST operates under the DOCS. Refer to Section III for information about the DOCS characteristics.

31.2.1 AVAILABLE VERBS. In addition to the verbs supplied in the DOCS, ACUTST supplies the verbs listed in table 31-1.

Table 31-1. ACUTST Verbs

Verb	Function
IT	Initialize Test
E1	Execute Part 1
E2	Execute Part 2
E3	Execute Part 3
E4	Execute Part 4
E5	Execute Part 5
EA	Execute Parts 1, 2, and 3
L1	Loop on Part 1
L2	Loop on Part 2
L3	Loop on Part 3
L4	Loop on Part 4

Table 31-1. ACUTST Verbs (Continued)

Verb	Function
L5	Loop on Part 5
LA	Loop on Parts 1, 2, and 3
LD	Loop on digits

When the LD option is specified, the operator selects the test digits or control codes to be used by responding to the questions listed in table 31-2.

Table 31-2. ACUTST LD Verb Questions

Question	Operator Response
RUN WITH INTERRUPTS: DEFAULT = 00	Enter 0 to run without interrupts or 1 to run with interrupts enabled.
DIALING TECHNIQUE (0 = PULSE 1 = TONE) DEFAULT = 00	Enter 0 for pulse dialing or 1 for tone dialing.
INPUT DIGITS (ENTER FF TO TERMINATE)	Enter the digit or the control code sequence, one hexadecimal digit at a time. Enter FF to terminate a sequence specification.

31.2.2 **TEST INITIALIZATION.** In addition to the initialization questions asked by the DOCS, the questions in table 31-3 are asked.

Table 31-3. ACUTST Initialization Questions

Question	Comment
ACU CRU BASE DEFAULT = 0000 –	The CRU base is determined by the chassis slot in which the module is placed.
DO YOU WANT TO USE INTERRUPTS DEFAULT = 00 –	For interrupts enter 1.
ACU INTERRUPT LEVEL DEFAULT = 00 –	The interrupt level is determined by backpanel wiring.
CPU CLOCK RATE (0=120 1 = 100) DEFAULT = 00 –	Enter a "0" for 120 Hz clock rate or a "1" for 100 Hz. The clock rate is two times the frequency of the ac power input to the computer.

31.2.3 **PROGRAMMER-PANEL INITIALIZATION.** In addition to the IDLE ON ERRORS?– question, the programmer panel version asks the same questions as listed in the previous section. For additional information, refer to the DOCS description.

31.3 SYSTEM REQUIREMENTS

In addition to the equipment required by the DOCS, an Auto Call Unit (part number 946110) is also required. ACUTST requires 8K words of memory.

31.4 ERROR MESSAGES

Table 31-4 lists the error messages output by ACUTST.

Table 31-4. ACUTST Error Messages and Numbers

Number	Message	Description
	AUTOMATIC CALL UNIT TEST	Header message.
	VERSION MM/YY *X	MM/YY *X is the release date and revision letter of the test.
	PART 1 – STATE TRANSITIONS TEST	Part 1 test start message.
	PART 1 – COMPLETE	Part 1 complete message.
	PART 2 – DIGIT TEST	Part 2 test start message.
	PART 2 – COMPLETE	Part 2 complete message.
	PART 3 – ACR TIMING TEST	Part 3 test start message.
	PART 3 – COMPLETE	Part 3 complete message.
	PART 4 – TONE FREQUENCY TEST	Part 4 test start message.
	PART 4 – COMPLETE	Part 4 complete message.
	PART 5 – NUMBER DIALING TEST	Part 5 test start message.
	PART 5 – COMPLETE	Part 5 complete message.
1	UNEXPECTED ACU INTERRUPT	An unexpected interrupt was caused by the module.
2	UNKNOWN INTERRUPT	An unknown interrupt occurred. The test is aborted.
3	EXPECTED INTERRUPT DID NOT OCCUR	An interrupt from the module did not occur when expected.
4	STATUS ERROR EXPECTED = XXXX RECEIVED = XXXX	Error between expected and received status.
5	PULSE ENABLE NOT SET	During pulse dialing pulse enable status bit did not get set.
6	PULSE TIMING ERROR	Error in number of pulses output by pulse dialer.
7	PULSE ENABLE DID NOT RESET	Pulse enable did not reset after digit was dialed.
8	ACR NOT REPORTED	Abandon Call and Retry status not reported when expected.
9	DDC NOT REPORTED	Digit Dial Complete did not set.
A	PND NOT SET	Present Next Digit did not set.
B	DSS NOT SET	Data Set Status did not set.
C	PND NOT RESET	Present Next Digit did not reset.
D	DSR NOT SET	Data Set Ready did not set.

SECTION XXXII

FAST LINE PRINTER TEST (FLPTST)

32.1 INTRODUCTION

The FLPTST runs on the TI 990/4 or 990/10 with 8K words of memory. The program verifies the operation of the Model 2230 and Model 2260 Line Printers when used with a 16 I/O printed circuit board.

The test consists of three parts:

1. Cable and basic data transfer test
2. Print pattern and slew/print rate test
3. VFU/slew print and power fail test.

It takes approximately 1 minute to run parts 1 and 2 of the test. If errors occur more time is required.

32.2 OPERATOR INTERFACE

FLPTST runs under the DOCS. For further information, consult Section III for more information about the DOCS.

32.2.1 AVAILABLE VERBS. Table 32-1 lists the verbs available under FLPTST in addition to those supplied by the DOCS.

Table 32-1. FLPTST Verbs

Verbs	Function
IT	Initialize Test
ST	Sliding Test
HA	Hammer Alignment
PB	Print Buffer
BB	Build Buffer
PD	Print Drum
RD	Ripple Dump
EA	Execute tests 1-3 once
LA	Execute tests 1-2 and repeat
ET	Execute an individual test once (1–3)
LT	Execute an individual test and repeat (1–3)
CM	Execute tests 1-2 on all available units

32.2.2 **STANDARD INITIALIZATION.** In addition to the questions asked by the DOCS, the default test values of the Fast Line Printer CRU address, Fast Line Printer Interrupt Level, and other available units are shown by the program. All of the default values may be changed by the operator. Table 32-2 lists the initialization questions.

32.2.3 **PROGRAMMER PANEL INITIALIZATION.** In addition to the IDLE ON ERRORS? question, the programmer-panel version asks the same questions as in table 32-2.

For additional information on the operation in the programmer panel mode, refer to Section III on the DOCS tests.

32.3 SYSTEM REQUIREMENTS

In addition to the hardware required by the DOCS, FLPTST requires the following hardware:

1. 16 I/O Board
2. Model 2230 or Model 2260 Line Printer.
3. 8K words of memory.

32.4 ERROR MESSAGES

Table 32-3 lists the messages output by FLPTST.

Table 32-2. FLPTST Initialization Questions

Question	Comment
ENTER PRINTER CRU BASE, DEFAULT = 0060 –	Enter CRU base.
ENTER PRINTER INTERRUPT LEVEL, DEFAULT = 0E –	Enter interrupt level.
DOES THIS SYSTEM HAVE OTHER LINE PRINTERS ? DEFAULT = 0 –	Enter 1 if system has other line printers.
DO YOU WANT TO RUN TEST WITH INTERRUPTS ? DEFAULT = 00 –	Enter 1 to run without interrupts.
IS COMPUTER LINE CURRENT 60 HZ ? DEFAULT = 01 –	Enter 00 if 50 Hz.
IS THIS A MODEL 2230 PRINTER ? DEFAULT = 1 –	Enter 00 if not a 2230 printer.

Table 32-3. FLPTST Error Messages and Numbers

Message Number	Text	Description
None	FLPTST-FAST LINE PRINTER	Test Header Message.
None	PART XX	Header Message.
None	PART XX DONE	Header Message.
None	LOOP COUNT = XXXX	Information message in LA and LT verb.
None	ERROR PART XX	Banner Message. This precedes each error message indicating which verb or part the error occurred in.
None	ENTER # OF LINES, DEFAULT = 0042 –	Initialization message for the RD (ripple dump) verb. Action: Enter the number of lines to be printed in hex.
None	ENTER BUFFER ADDR (WORD BOUNDARY), DEFAULT = XXXX –	Initialization message for the BB (build buffer) verb. Action: Enter starting column.
None	NO OF CHARACTERS IN HEX (EVEN #) –	Initialization message for the BB (build buffer) verb. Action: Enter number of characters.
1	INTERFACE VERIFICATION ERROR, OUTPUT = 1 INPUT = 0	I/O interface problem
2	INTERFACE VERIFICATION ERROR, OUTPUT = 0 INPUT = 1	I/O interface problem
3	PRINTER IS NOT READY	Status bit indicates printer not ready
4	PRINTER IS NOT ON LINE	Status bit indicates printer is not on line
5	DEMAND LINE IS NOT LOW	Demand is at wrong level for data transfer.
6	INTERRUPT DID NOT OCCUR AFTER PASSING DATA	Interrupt did not occur after sending data to the printer during the interrupt test.
7	INTERRUPT DID NOT OCCUR AFTER PASSING A PAPER FEED	Interrupt did not occur after sending a FORM FEED to the printer during the interrupt test.

Table 32-3. FLPTST Error Messages and Numbers (Continued)

Message Number	Text	Description
8	2230 LINE PRINTER WILL NOT PRINT 300 LPM	A 2230 type line printer will not print at least 300 lines per minute.
9	2260 LINE PRINTER WILL NOT PRINT 600 LPM	A 2260 type line printer will not print at least 600 lines per minute.
A	THIS IS A 2260, NOT a 2230 AS INITIALIZED	The printer is printing at twice the speed of a 2230 line printer; therefore, it must be a 2260.
B	PAPER SLEW IS TOO SLOW	The time to slew paper from top of form to top of form on 11 inch paper is too slow.
C	PRINTER SHOULD BE OFF BUT IS NOT	The printer should be off-line but the status bit does not reflect this.
D	DEMAND LINE SHOULD BE 1 BUT IS 0	The demand line does not reflect the proper status with printer off-line.
14	PRINTER IS NOT ONLINE BEFORE DATA IS OUTPUT	The printer is off-line when the Command Issuer Subroutine is entered.
15	PRINTER IS NOT READY BEFORE DATA IS OUTPUT	The printer is on line but not ready when the Command Issuer Subroutine is entered.
16	INTERRUPT OR DEMAND LINE FAILED TO COME ON AFTER TRANSFERRING DATA	Did not get interrupt after sending character with interrupts enabled or demand line failed to come on.
17	UNEXPECTED PRINTER INTERRUPT AT LOC = XXXX	Received unexpected interrupt from printer.

SECTION XXXIII

810 LINE PRINTER TEST (LP810)

33.1 INTRODUCTION

The LP810 tests the Texas Instruments Model 810 printer when the printer is connected to a 990 computer using the TTY/EIA interface module (part number 945075).

The LP810 diagnostic runs under control of the DOCS.

LP810 consists of the following 15 tests:

Test 01 tests the functions of the TTY/EIA interface card; TMS 9902, or TMS 9903 as follows:

1. Transmit bit
2. DSR bit
3. Interrupt capability
4. Baud rate of 4800

Test 02 is the first test that prints on the 810 printer. It uses the line feed character to force printing. This test prints each character in the character set 20 times on one line.

Test 03 prints a ripple dump. The ripple dump contains 80 or 132 characters per line depending on the option selected by the operator and consists of the full selected character set. Line feeds are used to force printing. Each character appears once in each column. The full character set ripple dump contains 95 lines and the standard character set ripple dump contains 63 lines.

Test 04 is the print buffer test. This test prints a diamond pattern of characters.

Test 05 is the carriage return test. This feature is tested by printing an English sentence with every other character omitted, giving a carriage return, and printing the characters that were omitted before. If the carriage return operates correctly, the following message is printed on one line in a readable form:

THIS LINE IS BEING PRINTED IN TWO PARTS. VERIFY THAT ONLY ONE LINE IS PRINTED.

Test 06 is the programmable forms length test. First the forms length is set to 33 lines. Then a form feed command is given and the following message is printed on the 810 under test:

THIS IS THE CENTER OF FORM PAGE ONE OF TEST 06

The forms length is then reset to 66 and the following message is printed on the top of the next page:

FORM FEED THIS IS TOP OF FORM PAGE TWO OF TEST 06

Test 07 tests the tab to line command. This test prints the following message ten times on the 810 under test:

THIS IS LINE XX AS NUMBERED ON THIS FORM YY LINES FROM TOP OF FORM

Where XX takes the values 1, 2, 4, 8, 16, 32, 40, 44, 46, and 47; and YY takes the values 4, 5, 7, 11, 19, 35, 43, 47, 49, and 50. On TI form 12461 the XX value corresponds to line number printed in the left margin of the form. The YY value is the count of lines from the top of form.

Test 08 tests the set vertical tab command and the vertical tab command. This test prints the following message ten times on the 810 under test:

THIS IS LINE XX AS NUMBERED ON THIS FORM YY LINES FROM TOP OF FORM

Where XX takes the values 1, 2, 4, 8, 16, 32, 40, 44, 46, and 47; and YY takes the values 4, 5, 7, 11, 19, 35, 43, 47, 49, and 50.

Test 09 tests horizontal tab set command and horizontal tab command. After numbering the columns, up arrows are printed in the following columns: 10, 15, 20, 40, 80, 90, 115, 117, 119, and 128.

Test 0A tests tab to column address command. After numbering the columns, up arrows are printed in the following columns: 10, 15, 20, 40, 80, 90, 115, 117, 119, and 128.

Test 0B tests line width set command. The columns are numbered with the line width set to a value selected by the operator, then the line width is set to 80. With this line width 240 'A' characters are sent to the printer. These characters are printed on three lines with 80 characters per line. If a line width of 132 was selected by the operator, the line width is then reset to 132 characters per line, and 132 'B' characters are printed.

Test 0C tests the printer lines per inch option. The printer is set to eight lines per inch, then eight lines of ten characters each are printed. The printer is then reset to six lines per inch, and six lines of ten characters each are printed.

Test 0D tests the printer characters per inch option (VCO and FCO printers only). The printer is set to 16.5 characters per inch, and six lines of seventeen characters each are printed. The printer is then reset to ten characters per inch, and six lines of ten characters each are printed.

Test 0E tests the printer vertical format control option. Store and recall vertical format information commands are tested, VFC channel 7 is programmed and stored, and VFC channel 8 is programmed and stored. Then VFC channel 7 is recalled and six lines of the following message are printed on the MC810 under test:

THIS IS LINE XX AS NUMBERED ON THIS FORM YY LINES FORM TOP OF FORM

Where XX takes the values 1, 2, 4, 8, 16, and 30; and YY takes the values 4, 5, 7, 11, 19, and 33. VFC channel 7 vertical tabs are set on the lines corresponding to the YY values except 33 which is top of form. Channel 8 is recalled and the above message is repeated on the second page of the test. The XX takes the values 3, 5, 17, 37, and 57; and YY takes the values 6, 8, 20, 40, and 60.

Test 0F is the manual intervention tests. When TEST 0F is executed, the operator is instructed as follows:

BEGIN MANUAL INTERVENTION TESTING POWER DOWN THE PRINTER
DEPRESS RETURN KEY

POWER UP THE PRINTER BUT DO NOT PUT THE UNIT ON LINE
DEPRESS RETURN KEY

VERIFY MESSAGE ON PRINTER
DEPRESS RETURN KEY

VERIFY MESSAGE ON PRINTER

The following 810 functions are tested:

1. DSR bit low when printer is off
2. DSR bit low when printer is off line
3. Select and deselect commands
4. Bell
5. Parity error detection

33.2 OPERATOR INTERFACE

LP810 runs under the DOCS. Refer to Section III for a detailed explanation of the DOCS versions.

33.2.1 AVAILABLE VERBS. In addition to the verbs available in the verb package, LP810 also supplies the verbs listed in table 33-1.

33.2.2 STANDARD INITIALIZATION. In addition to the initialization question required by the DOCS, LP810 asks the initialization questions listed in table 33-2.

33.2.3 PROGRAMMER-PANEL INITIALIZATION. In addition to the IDLE ON ERRORS? question, the questions in table 33-2 are asked.

Table 33-1. LP810 Verbs

Verb	Meaning
ET	Execute one of the tests (specified by test number)
LT	Loop on one of the tests (specified by test number)
EA	Execute all tests in order
LT	Loop on all tests in order
IT	Initialize test

Table 33-2. LP810 Initialization Questions

Question	Comment
ENTER PRINTER CRU BASE, DEFAULT = 0060 –	Enter CRU base address.
USE THE TTY/EIA BOARD (0) OR TMS 9902/9903 (1) DEFAULT = 00 –	Enter 1 for 9902/9903
*ENTER CLOCK FREQUENCY (0=3MHz, 1=4MHz) DEFAULT = 00 –	
ENTER PRINTER INTERRUPT LEVEL, DEFAULT = 0E –	Enter interrupt level.
DO YOU WANT TO RUN WITH INTERRUPTS? DEFAULT = 01 –	Enter the value 00 to run without interrupts.
DOES THIS PRINTER HAVE FULL ASCII CHARACTER SET? DEFAULT = 01 –	
DOES THIS PRINTER HAVE THE VCO OPTION? DEFAULT = 01 –	Vertical forms control/compressed print option.
IS THE COMPUTER LINE CURRENT 60 HZ? DEFAULT = 01 –	Enter the value 00 for 50 Hz.
DO YOU WANT TO RUN THE MANUAL INTERVENTION TEST? DEFAULT = 01 –	Enter the value 00 to disable that test.
ENTER COLUMN WIDTH (0 = 132, 1 = 80), DEFAULT = 01	Enter 01 to set column width of paper to 80; or the value 00 for 132 columns.

33.3 SYSTEM REQUIREMENTS

In addition to hardware required by the DOCS, the following equipment is required to run the test:

Model 810 Printer Interface Kit (part number 938120)

33.4 ERROR MESSAGES

The error messages are listed in table 33-3.

*This question is asked only when the TMS 9902/9903 is selected.

Table 33-3. LP810 Errors

Number	Message
	*****ERROR TEST 01 MUST RUN WITH OUT ERROR
2	ERROR BAUD RATE LESS THAN 4800 BAUD
3	ERROR BAUD RATE GREATER THAN 4800 BAUD
4	ERROR TRANSMIT STILL IN PROGRESS AFTER TIMED OUT
5	ERROR PRINTER IS OFF LINE DATA SET READY IS LOW AFTER TIME OUT
6	ERROR EXPECTED NEW STATUS FLAG INTERRUPT DID NOT OCCUR (DSR HIGH-TO-LOW)
7	ERROR EXPECTED NEW STATUS FLAG INTERRUPT DID NOT OCCUR (DSR LOW-TO-HIGH)
8	ERROR EXPECTED WRITE REQUEST INTERRUPT DID NOT OCCUR
9	ERROR UNEXPECTED INTERRUPT AT PRINTER INTERRUPT LEVEL
A	ERROR DSR TRUE WHILE PRINTER IS POWERED DOWN
B	ERROR DSR TRUE WHILE PRINTER IS OFF LINE
C	ERROR DSR LOW-TO-HIGH NEVER SET, LINE BUFFER NEVER RECEIVED CHARACTERS

SECTION XXXIV

DS10 DISK TEST (DS10PD)

34.1 INTRODUCTION

The DS10PD runs on a 990 computer with a minimum of 24K bytes of memory. This test checks the DS10 TILINE Disk Controller (TDC) and associated units. The test consists of six parts:

Number	Name	Subtests
1	Quick Controller Test	1-3
2	Disk and Controller Test	4-19
3	Addressing Test	20-22
4	Memory Addressing Test	25
5	Surface Pack Analysis	23
6	Interactive Part	24

For execution of Part 1, a disk unit is not needed since Part 1 tests only the TDC. If errors occur in Part 1, there is a high probability that the other parts will not run correctly. It takes approximately 16 minutes to run Parts 1-4. If errors occur, more time is required.

34.2 OPERATOR INTERFACE

This diagnostic runs under the DOCS.

34.2.1 AVAILABLE VERBS. This test operates using the 990 DOCS and supplies the verbs listed in table 34-1 in addition to the verbs supplied in the DOCS.

34.2.2 STANDARD INITIALIZATION. In addition to the initialization questions asked by the DOCS, the default test values of disk TILINE address, disk interrupt level, unit under test, and other available units are presented to the operator at the beginning of the test. These defaults may be changed by the operator, and the changed value will become the new default. Also, the operator is asked for a maximum error count value. Table 34-2 lists the initialization questions.

34.2.3 PROGRAMMER-PANEL INITIALIZATION. When operating the programmer-panel version, IDLE ON ERRORS? is asked followed by the questions in table 34-2.

Table 34-1. DS10PD Verbs

Verb	Function
IT	Initialize Test
EA	Run parts 1-6 once
LA	Run parts 1-5 and loop
E1-E6	Run an individual part once
L1-L5	Run an individual part and loop
CM	Loop on parts 1-5 for all available units
IC	Issue command
IM	Issue multiple commands
LO	Loop on multiple commands
DC	Display the U. T. C. status
SR	Store registers
CD	Compare two blocks of data
FD	Format the whole disk
DT	Display trace table

Table 34-2. DS10PD Initialization Questions

Question	Comment
DISK TILINE ADDRESS D=F800 –	Enter address.
DISK INTERRUPT LEVEL D=0D –	Enter interrupt level.
UNIT TO TEST 00 –	Enter unit number (0-3) of unit under test.*
OTHER AVAILABLE UNITS (0-3 TERMINATE WITH FF)F –	Enter unit number(s) of any other available* disk units. Last entry must be the value FF.
MAX ERROR COUNT EQUAL – FFFF –	Enter error count limit.

*Each disk is treated as a separate unit. The fixed disk is unit 0 and the removable disk is unit 1 on the first drive.

34.3 SYSTEM REQUIREMENTS

In addition to the hardware required by the DOCS, DS10PD also requires the following:

1. 990 computer with 24K bytes of memory and TILINE
2. A disk kit: TILINE disk, DS10 disk unit.

34.4 ERROR MESSAGES

Table 34-3 lists the error messages output by DS10PD.

Table 34-3. DS10PD Error Messages

Subtest Number	Error Number	Message
	1	*ERROR* THE UTC DID NOT RESPOND IN 20 SEC
	2	UNEX DISK INT AT XXXX
	4	MAX ERROR COUNT EXCEEDED, STOP
	5	****THAT TEST NUMBER IS NOT CURRENTLY AVAILABLE
1	6	UTC REG DATA TEST 4 WAS EXP REG
1	7	XXXX XXXX XX
2	8	ERROR R6=0 STATUS SHOULD BE COXO WAS XXXX
2	9	R7 SHOULD BE A100 WAS XXXX AFTER 10 RESET
3	10	STORE REG ERROR
3	11	SELF TEST ERROR REG TWO SHOULD READ 0000 REG TWO WAS--XXXX PLEASE CONSULT MAINTENANCE MANUAL BEFORE CONTINUING THE DIAGNOSTIC
3	12	LOCK OUT NOT SET AFTER 2 READS
5	13	DATA ERROR DATA READ WAS XXXX
5	14	DATA SHOULD HAVE READ -XXXX
6	15	WRITE FORMAT ERROR FORMAT READ WAS

Table 34-3. DS10PD Error Messages (Continued)

Subtest Number	Error Number	Message
5	16	XXXX XXXX XXXX XXXX
6	17	FORMAT SHOULD HAVE READ
7	18	RETRY ERROR - RETRIED WHEN NOT REQUIRED
8	19	TRANSFER INHIBIT ERROR
9	20	ID TEST ERROR HAVE READ REG 8 SHOULD HAVE A810 BUT WAS -XXXX
10	21	SE TEST ERROR REG 8 SHOULD READ A802 BUT WAS -XXXX
11	22	DE TEST ERROR REG 8 SHOULD READ A840 BUT WAS -XXXX
12	23	ERROR PART 2 XX TEST STATUS EXP=XXXX STATUS REC=XXXX
13	24	ERROR PART 4 SI TEST RO STATUS EXP=08XX REC=XXXX R7 STATUS EXP=A801 REC=XXXX
14	25	ERROR PART 21 DID WRITE WITH WC=0 BUT DATA NOT 0
15	26	ERROR PART 2 BUSY FLAG SET DATA (MSB SHOULD BE 0)
15	27	XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
16	28	TILINE TIMEOUT ERROR R7 STATUS EXP=A820 REC=XXXX
17	29	EXPECTED RATE ERROR, DID NOT RECEIVE IT R7 STATUS EXP=A808 REC=XXXX
18	30	WRITE AMP FAILURE R7 STATUS EXP=C800 REC=
19	31	ERROR PART 2N CONTROLLER DID NOT SWITCH FROM MAX HEAD CYL FF TO HEAD 0 CYL 100
20	32	DATA ERROR IN SECTOR XXXX DATA READ WAS

Table 34-3. DS10PD Error Messages (Continued)

Subtest Number	Error Number	Message
20	33	SECTOR COUNT ERROR ACCESSED SECTOR GREATER THAN 20
21	34	ERROR IN CYL ADDRESSING TEST AT CYL-XXXX RECEIVED -XXXX FOR CYL ADDR
22	35	ERROR IN VARIABLE FORMAT TEST AT -XXXX SECTORS PER RECORD RECEIVED-XXXX AS DATA
25	36	PART 4 DATA NOT EQUAL TO ADDRESS ADDRESS = XXXX DATA = XXXX
25	37	COMPARISON ERROR IN MAP MEMORY MAP ADDRESS = XXXX MAP BIAS = XXXX TILINE ADDRESS = XXXXXX EXPECTED DATA = XXXX DATA READ = XXXX
23	38	XX ERRORS IN A TRIES DID FORMAT WITH WE=XXXX S/R=XX DATA=XXXX DATA READ=XXXX FROM CYL=XXXX SURF=XX
23	39	ERROR PART 5 DID FORMAT WITH WE=XXXX S/R=XX DATA=XXXX DATA READ=XXXX FROM CYL=XXXX SURF=XX
23	40	PART 5 LOOP ERROR
24	41	ERROR DATA READ XXXX EXP XXXX FROM SURF XX CYL XX
	0003	*STATUS ERROR CONTROLLER STATUS COMP=X ERR=X IDLE=X U.T.C. REG DISK STAT COM SA S/R RA CYLA BYTEC MEMAD SEL CONT STAT XXXX XXXX XX XX XX XXXX XXXX XXXXXX XX XXXX COMMAND ISSUED DISK STAT COM SA S/R RA CYLA BYTEC MEMAD SEL CONT STAT XXXX XXXX XX XX XX XXXX XXXX XXXXXX XX XXXX

NOTE

Preceding all error messages, the following message will appear:

ERROR IN TEXT XXXX

NOTE

Units are selected as follows:

Unit	Sel.
0	08
1	04
2	02
3	01

NOTE

The disk command codes and the disk surface designators are located in the same words. The command codes are located in the left bytes and the surface designators are located in the right bytes. The commands and their identifying codes are shown in the following list.

Comm Left Byte	Command
00	STORE REGISTERS
01	WRITE FORMAT
02	READ DATA
03	WRITE DATA
04	UNFORMATTED READ
05	UNFORMATTED WRITE
06	INDEPENDENT SEEK (DUMMY COMMAND)
07	RESTORE

SECTION XXXV

TRACE MODULE TEST (TRACE)

35.1 INTRODUCTION

The TRACE tests the trace module for the 990 computer. The TRACE consists of five parts as described in the following paragraphs.

35.1.1 **PART 1.** Part 1 does a reset instruction and then does the following:

1. Verifies the input bits reset properly.
2. Verifies the traffic control bits can be loaded.
3. Verifies that the registers reset properly.
4. Tests trace bit to see if it sets and resets.
5. Tests internal clock bit to see if it sets and resets.
6. Tests glitch latch enable bit to see if it sets and resets.

35.1.2 **PART 2.** Part 2 does the following:

1. Loops on resetting module and checking CRU input bits and registers.
2. Loops on writing and reading the trace module registers.
3. Writes into one register and checks to see if it modifies another register.

35.1.3 **PART 3.** Part 3 does the following:

1. Verifies address pointer decrements properly.
2. Writes a pattern to all memory locations and then verifies the data.
3. Galloping RAM test.

This part runs for approximately 10 seconds on the 990/10 and about 16 seconds on the 990/4.

35.1.4 **PART 4.** Part 4 verifies the tracing capability of the module. It does the following:

1. Verifies event-delay status bit sets and resets.
2. Verifies buffer full status sets and resets.
3. Verifies that the event counter, delay counter, and address pointer decrement at the internal clock speed.
4. Verifies that event-delay status can generate an interrupt.
5. Verifies buffer full status can generate an interrupt.

6. Verifies the internal clock frequency by checking it against the RTC.

7. Checks the carry propagation paths in the event counter.

35.1.5 **PART 5.** Part 5 uses the diagnostic mode to simulate trace data. The diagnostic mode allows register 7 data to be traced into the RAM with computer control. This part does the following:

1. If either the data probe or the emulator data cables are attached, an error message is output. The trace module does not go into the diagnostic module in this case.

2. Checks to see if one can get any diagnostic mode clocks.

3. Checks the data path from the diagnostic register to the RAM at internal clock speed.

4. Checks the above with external clocks.

5. Checks the compare data and mask logic.

6. Checks the qualifier and mask compare logic.

7. Checks the glitch latch data path at internal clock speed.

8. Checks the functionability of the glitch latch.

9. Verifies that the "inverse of event" bit in the event control register functions properly.

10. Verifies that the "every event" bit and "event transition" bit in the event control register work properly.

11. Verifies that one can get out of the diagnostic mode.

35.2 OPERATOR INTERFACE

TRACE runs under the DOCS. Refer to Section III for a detailed explanation of the DOCS.

35.2.1 **AVAILABLE VERBS.** In addition to the verbs supplied in the DOCS, TRACE also supplies the verbs listed in table 35-1.

35.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by the DOCS, TRACE also asks the CRU address, Interrupt Level and Line Frequency. Table 35-2 lists the initialization questions.

35.2.3 **PROGRAMMER-PANEL INITIALIZATION.** The programmer panel version asks IDLE ON ERRORS? followed by the questions in table 35-2.

35.3 SYSTEM REQUIREMENTS

TRACE requires the unit under test, a trace module, in addition to the hardware required by the DOCS. The trace module diagnostic must be run on a 990 with at least 8K of memory. The exact amount will depend on the DOCS version used. The 990 must have real time clock interrupts available for part 4 of the test to run properly. An error message will be printed if no real time clock interrupts were received. *In order to run the test, the trace module data cable should not be connected.* An error message will flag this condition. The trace module control cable should be disconnected for the entire diagnostic.

Table 35-1. TRACE Verbs

Verb	Function
E1	Execute Part 1 - Interface Test
E2	Execute Part 2 - Register Test
E3	Execute Part 3 - RAM Test
E4	Execute Part 4 - Trace Mode Test
E5	Execute Part 5 - Diagnostic Mode Test
EA	Execute Part 1-5
L1	Loop on Part 1
L2	Loop on Part 2
L3	Loop on Part 3
L4	Loop on Part 4
L5	Loop on Part 5
LA	Loop on Part 1-5
IT	Initialize Test
PE	Prints the number of times errors occurred.
RG	Allows modification of any of the trace module's registers.
LR	Loops on loading and reading a particular register. Scoping aid.
DT	Dumps trace module RAM.
LT	Continuously traces data into the RAM.

35.4 ERROR MESSAGES

Table 35-3 lists the messages output by TRACE.

Table 35-2. TRACE Initialization Questions

Question	Comment
ENTER CRU BASE OF THE TRACE MODULE DEFAULT = 0200 –	Enter CRU base address.
ENTER INT LEVEL OF THE TRACE MODULE DEFAULT = 07 –	Enter interrupt level.
ENTER LINE FREQ (DEFAULT = 3C) –	Enter the value 32_{16} if line frequency is 50 Hz.

Table 35-3. TRACE Error Messages

Error Number	Message	Comment
1	TRAFFIC CONTROL BITS WON'T LOAD OR READ IS MODULE AT CRU BASE XXXX	The traffic control bits are 3 bits which determine to what register data is routed. When read, these bits were not the same as when they were written. The bits that are different between the received and expected data are the bits that are not working.
2	REG XXXX DID NOT RESET PROPERLY	After a reset, registers 1-6 should be all ones and register 0 and 7 should be all zeros. The bits that are different in the received and expected data are the bits that did not reset properly.
3	TRACE BIT CANNOT BE SET TO XXXX	
4	INTERNAL CLOCK BIT CANNOT BE SET TO XXXX	
5	LATCH ENABLE CANNOT BE SET TO XXXX	
6	WRITING INTO REG XXXX AFFECTED REG XXXX AFFECTED REG CONTAINS XXXX BUT SHOULD BE ZERO	Only the least significant 16 bits of the affected register are shown. Do an RG verb to find the rest after interrupting the error message by pressing the space bar.
7	REG XXXX CANNOT BE LOADED AND VERIFIED	The register cannot be either written correctly or read correctly. The LR verb is useful for diagnosting the problem.

Table 35-3. TRACE Error Messages (Continued)

Error Number	Message	Comment
8	ADDRESS POINTER DID NOT LOAD PROPERLY	Try the RG verb with the expected data going to register 6 which is the address pointer. Apparently, a certain pattern cannot be loaded right.
9	ADDR POINTER DID NOT AUTOMATICALLY DECREMENT	When the RAM is written to or read from, the address pointer is automatically decrement. Try the LT verb and watch the address pointer decrement on a scope.
A	RAM LOCATION XXXX CANNOT BE LOADED	If only one bit is different between received and expected data, then most likely that particular RAM chip is bad.
B	NO RTC INTERRUPTS RECEIVED	Part 4 of the test requires real time clocks in order to check the internal clock on the trace module. Apparently the computer you are on is not getting any real time clocks. They might not be wired up on your system. Do an E5 verb to finish the test.
C	WRITING INTO LOC XXXX AFFECTED LOC XXXX AFFECTED LOC CONTAINS XXXX	Address lines may be shortened together either externally to the RAM chips or internally. Only the LS part of the affected RAM location is printed. To find out the rest, do a DT verb after you interrupt the error message by pressing the space bar.
D	ADDRESS POINTER DID NOT DECREMENT 255 TIMES	Try the LT verb to see why. Look at the address pointer on the scope.
E	RAM LOC XXX CANNOT BE LOADED	Same as Error A except it occurred in different part of the test.
F	THERE IS MORE THAN XXXX PERCENT DIFF BETWEEN AN INTERVAL MEASURED WITH THE RTC AND THAT MEASURED WITH THE TRACE CLOCK EVENT COUNTER IS EXPECTED	Either the internal clock frequency is wrong or the event counter is getting extra clocks from somewhere.

Table 35-3. TRACE Error Messages (Continued)

Error Number	Message	Comment
10	THERE IS MORE THAN XXXX PERCENT DIFF BETWEEN AN INTERVAL MEASURED WITH THE RTC AND THAT MEASURED WITH THE TRACE CLOCK EVENT COUNTER IS LESS THAN EXPECTED	Either the internal clock frequency is lower than it should be or the event counter is missing some clocks.
11	BUFFER FULL STATUS DID NOT SET	If Error 14 also occurred, most likely, clocks are not getting to the counters. Using the RG verb, read the address pointer (REG 6) after interrupting the error message by pressing the space bar. If the zero, the address pointer did decrement correctly, but the logic to set buffer full status when the RAM is full probably did not work.
12	EVENT DID NOT DEC 255 TIMES	The event counter may have missed a clock or two or it could have gotten an extra one. The counter may have a bad chip. Use the LT verb to watch the event counter decrement on a scope.
13	DELAY COUNTER DID NOT DECREMENT 255 TIMES	Same as above except for delay counter instead of event counter.
14	EVENT-DELAY STATUS DID NOT SET INITIAL EVENT COUNT = XXXX PRESENT EVENT COUNT = XXXX	This is the first error that can be received in Part 4. If initial and present event count are both equal then the event counter never got any event clocks. The LT verb might help to find out why. If the present event count is zero, then the logic used to set the event-delay status may be bad.
15	AN INTERRUPT WAS EXPECTED AND RECEIVED BUT CRU BIT NOT SET ON TRACE MOD	Bit 31 on the CRU input was never set although the computer got a trace interrupt.
16	SETTING TRACE BIT TO 0 DOES NOT RESET BUFFER FULL STATUS	Setting the trace bit to zero should clear all input status.
17	SETTING TRACE BIT TO 0 DOES NOT RESET EVENT-DELAY STATUS	Same as Error # 16.
18	EVENT-DELAY STATUS DID NOT GENERATE AN INTERRUPT	With the interrupt host bit set on the trace module, the computer should have received an interrupt when event delay status set but it did not.

Table 35-3. TRACE Error Messages (Continued)

Error Number	Message	Comment
19	BUFFER FULL STATUS DID NOT GENERATE AN INTERRUPT	Same as Error #18 except with buffer full status instead of event-delay status.
1A	TRACE BIT DID NOT RESET WHEN AUTO HOLD AND BUFFER FULL BOTH PRESENT	The module should stop tracing when the RAM fills up and you have selected the auto hold mode, but it did not. Use the LT verb to put the module in trace mode. Then check with scope.
1B	BIT XXXX HEX OF THE COMPARE DATA AND MASK LOGIC MALFUNCTIONED CODE = XXXX	Bit 13_{16} is the least significant bit. If the code is 1, 3, 5, or 7 then the bit number that errored is a logic one in the compare register. If the code is 2, 3, 4, or 7 then the bit number that errored is a logic one in the compare mask register. If the code is 4, 5, 6, or 7 then the bit number that errored is a logic one in the diagnostic register. With a code of 1 or 4 the event counter should have decremented. Interrupt the computer in the middle of that error message and scope around the bit that errored in the compare and mask logic.
1C	THE EMULATOR CABLE IS CONNECTED. THIS PART OF THE TEST WON'T WORK WITH THE EM CABLE CONNECTED.	The trace module will not go into the diagnostic mode with any data cable attached. Detach the cable and rerun.
1D	TRACE DATA PROBE IS CONNECTED. THIS PART OF THE TEST WON'T WORK WITH THE DATA PROBE CONNECTED.	Same as error number 1C.
1E	THE DIAG MODE CLOCK DID NOT DECREMENT THE ADDRESS POINTER OR COUNT EVENTS	This is the first non-trivial error message for Part 5. Either the module will not go into the diagnostic mode at all or the diagnostic clocks are not getting to the counters.
1F	DIAG REG CANNOT BE TRACED INTO RAM LOC XXXX WITH INT CLK	
20	SETTING TRACE BIT TO ZERO DID NOT STOP MODULE FROM CONTINUOUSLY TRACING	

Table 35-3. TRACE Error Messages (Continued)

Error Number	Message	Comment
21	EVENT COUNTER DID NOT DECREMENT RIGHT	Use the LT verb to watch it decrement on the scope.
22	QUALIFIER AND MASK COMPARE LOGIC MALFUNCTIONED INPUT DATA = XXXX QUALIFIER MASK AND REG = XXXX	Interrupt computer in the middle of the error message and then use a scope to trace through the qualifier and mask compare logic. Note that the data should still be in the register.
23	DIAG REG CANNOT BE TRACED INTO RAM LOC XXXX WITH DIAG CLOCK	
24	UNLATCHED INPUT DATA FROM DIAG REG WAS NOT TRACED INTO LOC XXXX WITH INT CLOCK	
25	GLITCH LATCH NOT WORKING	
26	INVERSE OF EVENT BIT IN EVENT CONTROL REG NOT WORKING	A compare not equal was set and the event counter decremented when it should not have.
27	DIAG CLK DOES NOT DEC EVENT COUNTER RIGHT WITH EVENT CONTROL REG (BIT 1) = XXXX EVENT CONTROL REG (BIT 2) = XXXX	Bit 1 is the ext/int event bit. Bit 2 is the event transition bit. When Bit 1 is one, external events decrement the event counter. When Bit 2 is a one, event transitions decrement the event counter.
28	UNEXPECTED TRACE INT AT XXX	
29	MODULE DID NOT RESET PROPERLY	When a reset is executed, all CRU input bits with the exception of 3 bits are set to zero. Two of these bits tell what cable is attached to the trace module and the other is an unused bit. Check to make sure you have the trace module plugged into the correct spot. You can modify the CRU base by an IS verb.
2A	CANNOT LEAVE DIAGNOSTIC MODE	

SECTION XXXVI

EMULATOR TEST (EMUTST)

36.1 INTRODUCTION

The EMUTST tests an emulator and 9900 buffer combination. It also tests the interface between the emulator and a trace module. The emulator and trace modules may occupy any CRU slots and the emulator may be connected to any unshared interrupt level. This diagnostic will operate in a 990 with 16K bytes of memory.

This test consists of five parts as described in the following paragraphs.

36.1.1 PART 1. Part 1 performs emulator memory tests in the following manner. First, each memory location address is written to itself as the write data. The hardware read after write data is used as the compare data. Next, the test waits 2 seconds to allow dynamic RAM data to decay if refresh is not working properly. The test then reads each memory location and compares the read data to the address. This section of the test verifies refresh and checks to see that all memory locations are available. The test then writes and uses read after write data for comparison with data patterns of 0000, FFFF, AAAA and 5555 in that order.

36.1.2 PART 2. Part 2 loads and reads data from the trace counter register and the breakpoint register. Each register starts with data = 0s and increments data to all 1s in each register.

36.1.3 PART 3. Part 3 tests load trap, reset trap and idle status. First the Reset WP (0004) and PC (0024) are put in memory locations 0000 and 0002. JMP$ instructions fill the rest of the first 256 words of memory. Then the Load (FE00) and PC (FE20) are put in memory locations FFFC and FFFE. An Idle instruction is put at location FE20 and JMP$ instructions fill the rest of control memory. A Load Trap is then executed. If IDLE status is not detected either the trap failed or the IDLE instruction or status bit failed.

The Control Memory Idle instruction is then replaced with a JMP$. An Idle instruction is placed at memory location 0024. A Reset Trap is then performed checking Reset Trap, Idle instruction and Idle status.

36.1.4 PART 4. Part 4 tests Trace Data, Trace Count, and Interrupt and Auto Hold as controlled by the trace counter and breakpoint register. Part 4 consists of four parts. A header message is printed before each part is executed showing the condition of the trace, breakpoint and interrupt for that part of the test. E4 should not be executed with target system control memory selected or data compare errors will occur.

The first part of E4 generates an interrupt when the trace counter reaches zero.

The second part of E4 generates an Auto Hold on a PC breakpoint.

The third part of E4 generates an Auto Hold on an MA breakpoint.

The fourth part of E4 generates an Auto Hold on an MA write breakpoint after the trace counter passes through zero in the noncontinuous trace mode.

36.1.5 **PART 5.** Part 5 tests the interface between the trace module and the emulator. To do this, code is executed by the emulator in its 4K memory. The code is traced by the emulator and trace module during execution. The trace module traces Memory Read Data and the emulator traces Memory Read Addresses qualified by TSE from the trace module. The emulator breakpoint is set to breakpoint at address 1E in the MA mode. When address 1E is accessed, compare by the breakpoint logic sends Event to the trace module. Event causes the trace module to send Halt to the emulator, the emulator holds and sends Hold to the trace module, the trace module then halts. The trace data in both the trace module and the emulator is then verified. The trace module clock is the end-of-memory cycle clock from the emulator.

The test is run again with the trace module tracing all memory addresses and the emulator not tracing. The emulator breakpoint register is set to 10, at which address the previous halt sequence is repeated.

36.2 OPERATOR INTERFACE

EMUTST runs under the DOCS. Refer to Section III for a detailed explanation of the DOCS versions.

36.2.1 AVAILABLE VERBS. In addition to the verbs supplied in the DOCS, EMUTST also supplies the verbs listed in table 36-1.

The nonstandard verbs are described in the following paragraphs. For the SM, RM, WM, TC, BP, HD, and TP verbs, all messages are printed when the verb is first called. Each loop then uses any values it needs from those entered on the first pass and no more print occurs unless errors are detected. To exit, type @. The verbs loop whenever called until LP is executed and the no loop option is selected.

Table 36-1. EMUTST Verbs

Verb	Function
E1	Execute Part 1 - Memory Tests
E2	Execute Part 2 - Register Tests
E3	Execute Part 3 - Trap and Idle Tests
E4	Execute Part 4 - Trace and Breakpoint Tests
E5	Execute Part 5 - Emulator Trace Module Interface Tests
EA	Execute Part 1, 2, 3, and 4
L1	Loop on Part 1
L2	Loop on Part 2
L3	Loop on Part 3
L4	Loop on Part 4
L5	Loop on Part 5
LA	Loop on Parts 1, 2, 3, and 4
IT	Initialize Test
DS	Display Status
PE	Print Error Count

Table 36-1. EMUTST Verbs (Continued)

Verb	Function
SM	Select Memory
RM	Read Memory
WM	Write to Memory
LD	Load Memory
TC	Trace Counter Inspect and Change
BP	Breakpoint Register Inspect and Change
HD	Assert Hold
RH	Release Hold
TP	Execute Trap
TM	Test Memory at Speed
XT	External Trigger
LP	Loop on Verbs
CS	Clear Status

36.2.1.1 **DS Verb.** The DS verb displays the emulator status in the following format:

DS

DATA	HR	HA	CC	CE	CM	EM	EI	IS	MP	TF	TB	TT	BP
0000	1	1	0	0	1	1	0	0	1	00	0	0	00

where:

DATA	= EMULATOR	CRU BITS 0-15
HR	= HOLD REQUESTED	CRU BIT 16
HA	= HOLD ACKNOWLEDGED	CRUBIT 17
CC	= COMMAND COMPLETE	CRU BIT 18
CE	= CLOCK ERROR STATUS	CRU BIT 20
CM	= CONTROL MEMORY STATUS	CRU BIT 21
EM	= EMULATOR MEMORY STATUS	CRU BIT 22
EI	= EMULATOR IDLE STATUS	CRU BIT 23
IS	= INTERRUPT STATUS	CRU BIT 24
MP	= MICROPROCESSOR TYPE	CRU BIT 25
TF	= TRACE FUNCTION STATUS	CRU BITS 26, 27
TB	= TRACE BUFFER STATUS	CRU BIT 28
TT	= TRACE TRIGGERED STATUS	CRU BIT 29
BP	= BREAKPOINT FUNCTION STATUS	CRU BITS 30, 31

36.2.1.2 **PE Verb.** The PE verb prints the error count, which is the total number of all errors, of the most recently executed verb. This verb is normally used after Loop verbs with headers and error print suppressed.

36.2.1.3 **SM Verb.** The SM verb determines if the emulator memory (0-1FFE) or control memory (FE00-FFFE) will be accessed in the emulator or the target system. A sample of SM use follows:

```
SM
SELECT CONTROL MEMORY (EMULATOR=1 TARGET=0) -1
SELECT EMULATOR MEMORY (EMULATOR=1 TARGET=0) -1
VERB? -
```

36.2.1.4 **RM Verb.** The RM verb reads eight word groups of emulator accessed memory. After a group is printed the next 8 words of memory may be printed by pressing the RETURN key. If some other address is desired it can be printed out by entering the address and then RETURN. To exit the RM verb, type @. A sample of RM use follows:

```
ADDR    0     2     4     6     8     A     C     E
0800    0000  0000  0000  0000  0000  0000  0000  0000  -
0810    0000  0000  0000  0000  0000  0000  0000  0000  - 100
0100    0000  0000  0000  0000  0000  0000  0000  0000  - @
VERB? -
```

36.2.1.5 **WM Verb.** The WM verb writes a word of data to emulator accessed memory. Enter the address to be written to. The contents of that address will be printed out followed by a dash. Enter new data or leave old data intact. The next address and contents will be printed. It can be modified, or exit WM by typing @. A sample of WM use follows:

```
WM
MEMORY ADDRESS = -50
0050    DATA = 0000    -FFFF
0052    DATA = 0000    -@
VERB?-
```

36.2.1.6 **LD Verb.** The LD verb will load 256 consecutive words of 990 memory to 256 consecutive words of emulator accessed memory. This verb can be used to load user generated code to the emulator for execution. A sample of LD use follows:

```
LD
ENTER STARTING ADDRESS OF PROGRAM TO BE LOADED
LOADING PROGRAM TO EMULATOR MEMORY
LOAD COMPLETE
VERB? -
```

36.2.1.7 **TC Verb.** The TC verb allows inspection and change of trace counter contents. After TC verb prints the contents of the trace counter and a dash, new data may be entered. A sample of TC use follows:

```
TC
TRACE COUNTER DATA = A9 -90
VERB ? - TC
TRACE COUNTER DATA = 90 -
VERB? -
```

36.2.1.8 **BP Verb.** The BP verb allows inspection and change of the breakpoint register in a manner identical to the TC verb. A sample of BP use follows:

```
BP
BREAKPOINT REGISTER DATA = 001E -FFEE
VERB? -BP
BREAKPOINT REGISTER DATA = FFEE -
VERB? -
```

36.2.1.9 **HD Verb.** The HD verb puts the emulator in the hold mode.

36.2.1.10 **RH Verb.** The RH verb takes the emulator out of the hold mode.

36.2.1.11 **TP Verb.** The TP verb allows a reset trap or a load trap to be executed. The user enters the trap type, workspace pointer and program counter. After the PC value is entered, the desired trap is executed by the emulator. A sample of TP use follows:

```
TP
ENTER TRAP TYPE (1=RESET 0=LOAD)
THEN ENTER WP AND PC. -0 -10 -100
VERB?
```

36.2.1.12 **TM Verb.** The TM verb executes an At Speed Memory Test. E1 memory tests are CRU accesses to emulator accessed memory. TM executes a memory test using the 9900 on the buffer board to achieve a full speed test. This test loops on changing data patterns until @ is entered at the keyboard. Upper bound must be greater than 50_{16} to allow room for the test in memory. The test is loaded at location 0000. A front panel loop count shows the number of times the TM verb has tested the memory specified. A sample of TM use follows:

```
TM
ENTER LOWER AND THEN UPPER BOUND OF
MEMORY TO BE TESTED. -400 -1FFE
EXECUTING AT SPEED MEMORY TEST
```

36.2.1.13 **XT Verb.** The XT verb allows manual testing of the emulator's external trigger. After the XT message is printed, ground the external trigger pin on the emulator. If the logic is functioning 'External Trigger Received' will be printed out. A sample of XT use follows:

```
XT
TOGGLE EXTERNAL TRIGGER INPUT, THIS TEST WILL END ONLY IF TRIGGER IS
SUCCESSFUL OR SPACE IS ENTERED AT KEYBOARD.
EXTERNAL TRIGGER RECEIVED.
VERB? -
```

36.2.1.14 **LP Verb.** The LP verb asks if the user wants to loop on verbs. If answered yes, the following verbs will loop when executed.

36.2.1.15 **CS Verb.** The CS verb puts the emulator in known status conditions which follow:

1. Puts emulator in hold
2. Selects emulator control memory
3. Selects emulator 4K memory
4. Turns trace function off (clear CRU bits 26 and 27)
5. Masks interrupt
6. Disables auto hold
7. Selects emulator trace clock
8. Selects internal breakpoint
9. Clears interrupt status
10. Turns breakpoint function off (clears CRU bits 30 and 31)

CS also turns off Trace in the trace module if one is present. The trace module CRU base must have been specified during IT.

36.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by the DOCS, EMUTST also asks the questions listed in table 36-2.

36.2.3 **PROGRAMMER-PANEL INITIALIZATION.** The programmer-panel version of EMUTST asks the IDLE ON ERRORS? - question followed by the questions shown in table 36-2.

36.3 SYSTEM REQUIREMENTS

In addition to a 990 with 16K bytes of memory and an interactive device, the following equipment is needed:

9900/9980 Emulator Module	945925
9900 Buffer Module	949995
Trace Module (Connected to emulator)	949910

36.4 ERROR MESSAGES

Table 36-3 lists the messages output by EMUTST.

Table 36-2. EMUTST Initialization Questions

Question	Comment
ENTER TRACE MODULE CRU BASE ADDRESS DEFAULT = 0100–	Enter CRU address.
ENTER THE EMULATOR CRU BASE ADDRESS DEFAULT 0140 –	Enter CRU address.
ENTER THE EMULATOR INT. LEVEL DEFAULT = 0004 –	Enter interrupt level.

Table 36-3. EMUTST Error Messages

Number	Meaning
#1	Emulator unexpected interrupt.
#2	Hold requested was not returned. Verify CRU base address is correct.
#3	Hold acknowledged not returned.
#4	Control memory status (bit 21) failed. It should have been a one.
#5	Emulator memory status (bit 22) failed. It should have been a one.
#6	Memory Read failed to return command Complete address = XXXX.
#7	Memory Write failed to return command Complete address = XXXX.
#8	Memory data error address = XXXX. Data = XXXX expected = XXXX.
#9	Trace counter data error, expected = XX. Actual = XX.
#10	Compare register data error. Expected = XXXX. Actual = XXXX.
#11	Command complete failed to set after a write to trace counter.

Table 36-3. EMUTST Error Messages (Continued)

Number	Meaning
#12	Command complete failed to set after a read from trace counter.
#13	Command complete failed to set after a write to compare register.
#14	Command complete failed to set after a read from compare register.
#15	Trace function status (bit 26) failed. It should have been a zero.
#16	Control memory status (bit 21) failed. It should have been a zero.
#17	Emulator memory status (bit 22) failed. It should have been a zero.
#18	Load Trap or emulator idle failed.
#19	Reset trap or emulator idle failed.
#20	Hold requested failed to become a one after hold request was set to a one.
#21	Hold acknowledged failed to become a one after hold request was set to a one.
#22	Breakpoint function (bit 30) failed. It should have been a zero.
#23	Breakpoint function (bit 31) failed. It should have been a zero.
#24	Trace function (bit 26) failed. It should have been a one.
#25	Trace function (bit 27) failed. It should have been a one.
#26	Emulator failed to interrupt CPU.
#27	Interrupt status (bit 24) failed. It should have been a one. Indicates auto hold failed.

Table 36-3. EMUTST Error Messages (Continued)

Number	Meaning
#28	Trace buffer status (bit 28) failed. It should have been a zero.
#29	Auto hold failed to set hold requested to a zero.
#30	Auto hold failed to set hold acknowledged to a zero.
#31	Trace buffer status (bit 28) failed. It should have been a one.
#32	Trace function status (bit 27) failed. It should have been a zero.
#33	Breakpoint status (bit 20) failed. It should have been a one.
#34	Breakpoint status (bit 31) failed. It should have been a one.
#35	Trace trigger status (bit 29) failed. It should have been a one.
#36	Clock error status received.
#37	Upper bound less than lower bound.
#38	At Speed Memory test is failing to execute.
#39	T.M. Cable status (bit 21) indicates data cable not present.
#40	T.M. data source status (bit 23) fails. Indicates emulator is not data source.
#41	T.M. Halt pending status (bit 31) failed. It should have been a one.
#42	T.M. Tracing status (bit 20) failed. It should have been a zero.
#43	Trace module data error address = XX. Expected = XXXXX. Actual = XXXXX
#44	Memory read timed out.

SECTION XXXVII

REMOTE DISKETTE SYSTEM LOOPBACK TEST (RMTFLP)

37.1 INTRODUCTION

The RMTFLP tests the components in the remote diskette system of a 990 computer. The RMTFLP consists of local and remote tests which are discussed in paragraphs 37.1.1 and 37.1.2 respectively.

37.1.1 **LOCAL TEST.** The local test exercises the CRU-based remote diskette buffer board with bit patterns sent through on-board loopback jumpers. The resulting patterns are compared to a list of expected results. A non-compare causes the generation of an error message.

37.1.2 **REMOTE TEST.** The remote test checks the buffer board, the interface cable and the remote interface board. Remote drives are de-selected to turn on loopback logic. Bit patterns are sent to the specified drive. The resulting patterns are compared to a list of expected results. Non-compares cause generation of error messages.

37.2 OPERATOR INTERFACE

RMTFLP runs under the DOCS. For a detailed explanation of the DOCS versions, refer to Section III of this manual.

37.2.1 **AVAILABLE VERBS.** In addition to the verbs supplied in the DOCS, RMTFLP supplies the verbs listed in table 37-1.

Table 37-1. RMTFLP Verbs

Verb	Function
IT	Initialize test
EL	Execute local loopback test
ER	Execute remote loopback test
LL	Loop on local loopback test
LR	Loop on remote loopback test

37.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by the DOCS, RMTFLP asks the question listed in table 37-2.

Table 37-2. RMTFLP Initialization Questions

Question	Comment
"BUFFER BOARD CRU BASE ADDRESS DEFAULT = 0040"	Enter correct CRU base address.

37.2.3 **PROGRAMMER-PANEL INITIALIZATION.** The programmer-panel version of RMTFLP asks:

IDLE ON ERRORS?

followed by the question in table 37-2.

37.3 SYSTEM REQUIREMENTS

The RMTFLP will run on a 990 computer with at least 16K bytes of memory, an appropriate loading device and an interactive device.

The system must also include a remote diskette system which is tested by this diagnostic.

37.4 ERROR MESSAGES

When a non-compare occurs during the local or remote test, an error message is output to the interactive device as follows:

VALUE SENT	VALUE RECEIVED	VALUE RECEIVED
XXXX	XX	XX

where the Xs represent hexadecimal digits.

SECTION XXXVIII

32-INPUT/TRANSITION DETECTION MODULE TEST (INPMOD)

38.1 INTRODUCTION

The INPMOD tests the 32-input/transition detection module with the outputs of the 32-bit output data module connected to the corresponding inputs of the 32-bit input module. INPMOD consists of seven tests as described in paragraphs 38.1.1 through 38.1.7.

38.1.1 **TEST 1.** Test 1 checks the reset and clear functions of the 32-input module. All bits on the output module are cycled off and on to check that the input module reads each condition correctly.

38.1.2 **TEST 2.** All possible patterns are sent to the output module and a check is made to ensure that the input module reads each pattern correctly.

38.1.3 **TEST 3.** Each bit on the output module is cycled off and on 100 times. The value read by the input module is checked after each transition to determine if the input module works properly.

38.1.4 **TEST 4.** This test checks the transition operation of the input module. Bits 0 through 31 are unmasked to allow interrupt generation. Each bit on the output module is individually cycled from low to high to check that the transition generates an interrupt. If an interrupt is generated the bit address and current value of the bit are checked for correctness. Similar checks are made by cycling each bit from high to low.

38.1.5 **TEST 5.** The transition interrupt scanner is tested by causing a transition on each line and checking that the input module recognizes each transition in order.

38.1.6 **TEST 6.** Each bit is individually masked and checked to determine that transitions on the selected line do not generate an interrupt, and that transitions on all other lines do generate interrupts.

38.1.7 **TEST 7.** With the board level mask set to one, checks are made to determine that interrupts generated on the input module also generate CRU interrupts.

38.2 OPERATOR INTERFACE

INPMOD runs under the DOCS. For a detailed explanation of the DOCS versions, refer to Section III of this manual.

38.2.1 **AVAILABLE VERBS.** In addition to the verbs supplied in the DOCS, INPMOD supplies the verbs listed in table 38-1.

Table 38-1. INPMOD Verbs

Verb	Function
IT	Initialize Test
EA	Execute all Tests
E1	Execute Reset and Clear Test (Test 1)
E2	Execute Pattern Test (Test 2)

Table 38-1. INPMOD Verbs (Continued)

Verb	Function
E3	Execute Bit Cycle Test (Test 3)
E4	Execute Transition Test (Test 4)
E5	Execute Scanner Test (Test 5)
E6	Execute Bit Mask Test (Test 6)
E7	Execute Interrupt Test (Test 7)
LA	Loop on all Tests
L1	Loop on Test 1
L2	Loop on Test 2
L3	Loop on Test 3
L4	Loop on Test 4
L5	Loop on Test 5
L6	Loop on Test 6
L7	Loop on Test 7
LB	Load CRU Base
LC	Load CRU
ST	Store CRU
SO	Set CRU Bit to One
SZ	Set CRU Bit to Zero
TB	Test CRU Bit
BP	Bit Pole
BL	Loop on Set Bit

38.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by the DOCS, INPMOD asks the questions listed in table 38-2.

Table 38-2. INPMOD Initialization Questions

Question	Comment
ENTER THE 32-INPUT MODULE CRU ADDRESS DEFAULT = 0040–	Enter correct CRU address.
ENTER INPUT MODULE INTERRUPT LEVEL DEFAULT = 04–	Enter correct interrupt level.
ENTER THE 32-BIT OUTPUT MODULE ADDRESS DEFAULT = 0080–	Enter correct module address.

38.2.3 **PROGRAMMER-PANEL INITIALIZATION.** The programmer-panel version of INPMOD asks;

IDLE ON ERRORS?

followed by the questions in table 38-2.

38.3 SYSTEM REQUIREMENTS

The INPMOD will run on a 990 computer with at least 8K words of memory, an appropriate loading device, the 32-input/transition detection module under test, a 32-bit output data module, 32 input/output functional test cables and an appropriate interactive device.

38.4 TEST AND ERROR MESSAGES

INPMOD test and error messages are listed in table 38-3.

Table 38-3. INPMOD Test and Error Messages

Hexadecimal Number	Message	Description
None	32-BIT INPUT MODULE TEST VERSION MM/YY *X	Header Message, MM/YY *X is the release date and revision letter.
None	32-BIT INPUT MODULE TEST STARTING	Test Start Message
None	TEST X YY . . . Y	Test Start Message, X is the test number (1-7), YY . . . Y is the test name.
None	TEST COMPLETE	Test completed with no errors.
None	TEST ABORTED	Test aborted because of error.
1	ERROR 01	Reset did not mask Board Interrupt.
2	ERROR 02 ILLEGAL INTERRUPT INTERRUPT ADDRESS = XX	Reset did not clear interrupt at input line XX
3	ERROR 03 PATTERN EXPECTED 00000000 PATTERN RECEIVED XXXXXXXX ERROR PATTERN XXXXXXXX	Invalid bit pattern (All 32 bits should be low after reset) (See Note)
4	ERROR 04 PATTERN EXPECTED FFFFFFFF PATTERN RECEIVED XXXXXXXX ERROR PATTERN XXXXXXXX	Invalid bit pattern (All 32 bits should be high) (See Note)
5	ERROR 05	No interrupt after turning all 32 bits on
6	ERROR 06 ILLEGAL INTERRUPT INTERRUPT ADDRESS = XX	Board clear did not clear interrupt at input line XX
7	ERROR 07 PATTERN EXPECTED 00000000 PATTERN RECEIVED XXXXXXXX ERROR PATTERN XXXXXXXX	Invalid bit pattern (All bits should be low) (See Note)
8	ERROR 08	No interrupt after turning all bits off.

Table 38-3. INPMOD Test and Error Messages (Continued)

Hexadecimal Number	Message	Description
9	ERROR 09 ILLEGAL INTERRUPT INTERRUPT ADDRESS = XX	Reset did not clear interrupt at input line XX
11	ERROR 11 PATTERN EXPECTED XXXXFFFF PATTERN RECEIVED XXXXXXXX ERROR PATTERN XXXXXXXX	Invalid pattern (Bits 0-15) (See Note)
12	ERROR 12 PATTERN EXPECTED FFFFXXXX PATTERN RECEIVED XXXXXXXX ERROR PATTERN XXXXXXXX	Invalid Pattern (Bits 16-31) (See Note)
15	ERROR 15 BIT XX FAILED CYCLE TEST	Bit XX did not get read in as zero.
16	ERROR 16 BIT XX FAILED CYCLE TEST	Bit XX did not get read in as one.
20	ERROR 20 BIT XX TRANSITION DID NOT GENERATE INTERRUPT	High-to-low transition (on input line) of bit XX did not generate an interrupt.
21	ERROR 21 INVALID ADDRESS INTERRUPT ADDRESS = XX	Invalid interrupt address
22	ERROR 22 BIT - XX - BAD LEVEL	Bad bit level (interrupting bit should be read in as high).
23	ERROR 23 ILLEGAL INTERRUPT INTERRUPT ADDRESS = XX	Illegal interrupt
24	ERROR 24 BIT XX TRANSITION DID NOT GENERATE INTERRUPT	Low-to-high transition (on input line) of bit XX did not generate an interrupt.
25	ERROR 25 INVALID ADDRESS INTERRUPT ADDRESS = XX	Invalid Address
26	ERROR 26 BIT - XX - BAD LEVEL	Bad bit level (interrupting bit should be read in as low)
27	ERROR 27 ILLEGAL INTERRUPT INTERRUPT ADDRESS = XX	Illegal Interrupt
30	ERROR 30	Transition on bit zero did not generate interrupt.

Table 38-3. INPMOD Test and Error Messages (Continued)

Hexadecimal Number	Message	Description
31	ERROR 31 INVALID ADDRESS INTERRUPT ADDRESS = XX	Scanner error - Invalid interrupt address
32	ERROR 32	Scanner error - No interrupt
33	ERROR 33 ILLEGAL INTERRUPT INTERRUPT ADDRESS = XX	Scanner error - Illegal interrupt
36	ERROR 36 BIT XX TRANSITION DID NOT GENERATE INTERRUPT	Bit mask error - No interrupt on unmasked line.
37	ERROR 37 ILLEGAL INTERRUPT INTERRUPT ADDRESS = XX	Bit mask error, Illegal interrupt on masked line.
40	ERROR 40	No CRU interrupt
50	ERROR 50 UNEXPECTED INTERRUPT RELATIVE PC = XXXX	Unexpected CRU Interrupt
51	ERROR 51 UNKNOWN INTERRUPT RELATIVE PC = XXXX	Unknown CRU Interrupt

NOTE: Data sent to the output module is inverted at its output lines. This data is again inverted when read in by the input module. The PATTERN RECEIVED represents the data read in by the input module and is the complement of the actual pattern on the input lines.

SECTION XXIX

32-BIT OUTPUT DATA MODULE TEST (OUTMOD)

39.1 INTRODUCTION

The OUTMOD tests the 32-bit output data module with its outputs connected to the corresponding inputs of the 32-bit input/transition detection module. OUTMOD consists of three tests as described in paragraphs 39.1.1 through 39.1.3.

39.1.1 TEST 1. This test checks the response to a RSET instruction. All bits on the output module are reset and checked for the correct response.

39.1.2 TEST 2. All possible patterns are output by the output module and checked for correctness by the input module.

39.1.3 TEST 3. Each bit on the output module is cycled on and off 100 times and each transition verified to determine if the output module works properly.

39.2 OPERATOR INTERFACE

OUTMOD runs under the DOCS. For a detailed explanation of the DOCS versions, refer to Section III of this manual.

39.2.1 AVAILABLE VERBS. In addition to the verbs supplied in DOCS, OUTMOD supplies the verbs listed in table 39-1.

Table 39-1. OUTMOD Verbs

Verb	Function
IT	Initialize Test
EA	Execute all Tests
E1	Execute Reset Test (Test 1)
E2	Execute Pattern Test (Test 2)
E3	Execute Bit Cycle Test (Test 3)
LA	Loop on all Tests
L1	Loop on Test 1
L2	Loop on Test 2
L3	Loop on Test 3
LB	Load CRU Base
LC	Load CRU
ST	Store CRU
SO	Set CRU Bit to One
SZ	Set CRU Bit to Zero
TB	Test CRU Bit
BP	Bit Pole
BL	Loop on Set Bit

39.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by DOCS, OUTMOD asks the questions listed in table 39-2.

Table 39-2. OUTMOD Initialization Questions

Question	Comment
ENTER THE 32-OUTPUT MODULE CRU ADDRESS DEFAULT = 0080–	Enter correct CRU address.
ENTER THE 32-BIT INPUT MODULE ADDRESS DEFAULT = 0040–	Enter correct address.

39.2.3 **PROGRAMMER-PANEL INITIALIZATION.** The programmer-panel version of OUTMOD asks:

IDLE ON ERRORS?

followed by the questions in table 39-2.

39.3 SYSTEM REQUIREMENTS

The OUTMOD will run on a 990 computer with at least 16K bytes of memory, an appropriate loading device, the 32-bit output data module under test, a 32-input/transition detection module, 32 input/output functional test cables and an appropriate interactive device.

39.4 TEST AND ERROR MESSAGES

OUTMOD test and error messages are listed in table 39-3.

Table 39-3. OUTMOD Test and Error Messages

Hexadecimal Number	Message	Description
None	32 OUTPUT MODULE TEST VERSION MM/YY *X	Header Message, MM/YY *X is the release date and revision letter of the test.
None	32-BIT OUTPUT MODULE TEST STARTING	Test Start Message
None	TEST X YY . . . Y	Test Start Message, X is the Test Number (1-3), Y. . . . . Y is the Test Name.

Table 39-3. OUTMOD Test and Error Messages (Continued)

Hexadecimal Number	Message	Description
None	TEST COMPLETE	Test completed with no errors.
None	TEST ABORTED	Test aborted because of error.
1	ERROR 01 PATTERN EXPECTED 00000000 PATTERN RECEIVED XXXXXXXX ERROR PATTERN XXXXXXXX	Bit(s) did not get read in as zeroes by the input module. (See Note)
2	ERROR 02 PATTERN EXPECTED FFFFFFFF PATTERN RECEIVED XXXXXXXX ERROR PATTERN XXXXXXXX	Bit(s) did not get read in as ones by the input module. (See Note).
3	ERROR 03 PATTERN EXPECTED XXXXXXXX PATTERN RECEIVED XXXXXXXX ERROR PATTERN XXXXXXXX	Pattern Error (See Note).
4	ERROR 04 BIT XX (DECIMAL) FAILED CYCLE TEST	Bit cycle error - Bit XX did not go low at output line. (See Note).
5	ERROR 05 BIT XX (DECIMAL) FAILED CYCLE TEST	Bit cycle error - Bit XX did not go high at output line. (See Note).

NOTE: Data sent to the output module is inverted at its output lines. This data is again inverted when read in by the input module. The PATTERN RECEIVED represents the data read in by the input module and is the complement of the actual pattern on the output lines.

SECTION XL

5MT/6MT SERIAL INTERFACE MODULE TEST (FIVMOD)

40.1 INTRODUCTION

The FIVMOD is used to test the 5MT/6MT Interface Module in a 990 computer. FIVMOD consists of four subtests as described in paragraphs 40.1.1 through 40.1.4.

40.1.1 TEST 1 – RESET/CRU INTERFACE TEST. This test executes a RESET instruction and checks for correct setting of the status bits. Test 1 also sets the CRU control bits and checks to verify correct status levels.

40.1.2 TEST 2 – BIT ADDRESSING TEST. This test checks the random write/read operation using the self-test feature of the 5MT/6MT module.

40.1.3 TEST 3 – SEQUENTIAL LOAD TEST. This test checks the sequential write/read operation using the self-test feature of the 5MT/6MT module.

40.1.4 TEST 4 – INTERRUPT/BUSY TEST. This test checks the interrupt bit, the clear-interrupt bit, the busy bit, the auto interrupt, and the CRU interrupt.

40.2 OPERATOR INTERFACE

FIVMOD runs under the DOCS on a 990 computer. For a detailed discussion of the DOCS see Section III of this manual.

40.2.1 AVAILABLE VERBS. In addition to the verbs supplied in the DOCS, FIVMOD supplies the verbs listed in table 40-1.

Table 40-1. FIVMOD Verbs

Verb	Function
IT	Initialize test
E1	Execute test 1
E2	Execute test 2
E3	Execute test 3
E4	Execute test 4
EA	Execute tests 1, 2, 3, and 4
L1	Loop on test 1
L2	Loop on test 2
L3	Loop on test 3
L4	Loop on test 4
LA	Loop on tests 1, 2, 3, and 4

Table 40-1. FIVMOD Verbs (Continued)

Verb	Function
LB	Load CRU base
LC	Load CRU
ST	Store CRU
SO	Set CRU bit to one
SZ	Set CRU bit to zero
TB	Test CRU bit
BP	Bit pole
BL	Loop on set bit
RW	Read write cycle
DR	Dump input RAM
LR	Load output RAM

40.2.2 STANDARD INITIALIZATION. In addition to the initialization questions required by DOCS, FIVMOD asks the questions listed in table 40-2.

Table 40-2. FIVMOD Initialization Questions

Question	Comment
ENTER THE 5MT/6MT I/F MODULE CRU ADDRESS DEFAULT = 0080–	Enter correct CRU address
ENTER 5MT/6MT INTERRUPT LEVEL DEFAULT = 07–	Enter correct interrupt level

40.2.3 PROGRAMMER-PANEL INITIALIZATION. The programmer-panel version of FIVMOD asks:

IDLE ON ERRORS? –

followed by the questions in table 40-2.

40.3 SYSTEM REQUIREMENTS

The FIVMOD will run on a 990 computer with at least 8K words of memory, an appropriate loading device, an appropriate interactive device and the 5MT/6MT serial interface module under test.

40.4 ERROR MESSAGES

FIVMOD error messages are listed in table 40-3.

Table 40-3. FIVMOD Error Messages

Error Number	Message
1	ERROR 01 EXPECTED PATTERN = 0000 RECEIVED PATTERN = XXXX
2	ERROR 02 EXPECTED PATTERN = 6EFF RECEIVED PATTERN = XXXX
3	ERROR 03 EXPECTED PATTERN = 0000 RECEIVED PATTERN = XXXX
4	ERROR 04 BIT XX DID NOT GO HIGH
5	ERROR 05 BIT XX DID NOT GO LOW
6	ERROR 06 BIT XX NOT LOW
7	ERROR 07 EXPECTED PATTERN = CCCC RECEIVED PATTERN = XXXX
8	ERROR 08 EXPECTED PATTERN = DDDD RECEIVED PATTERN = XXXX
9	ERROR 09 EXPECTED PATTERN = 0000 RECEIVED PATTERN = XXXX
10	ERROR 10 EXPECTED PATTERN = XXXX RECEIVED PATTERN = XXXX
11	ERROR 11 NO BUSY BIT
12	ERROR 12 NO INTERRUPT
13	ERROR 13 NO AUTO-INTERRUPT CLEAR
14	ERROR 14 INTERRUPT NOT CLEARED
15	ERROR 15 NO CRU INTERRUPT
16	ERROR 16 UNEXPECTED CRU INTERRUPT
17	ERROR 17 UNKNOWN INTERRUPT
18	ERROR 18 BUSY BIT DOES NOT GO LOW

SECTION XLI

MODEL 804 CARD READER TEST (CRDRDR)

41.1 INTRODUCTION

The CRDRDR is used to test the 804 card reader in a 990 computer system. CRDRDR consists of three subtests as described in paragraphs 41.1.1 through 41.1.3.

41.1.1 TEST 1 – STATUS CHECKS TEST. This test checks the 804 interface board to verify that all status bits can be reset.

41.1.2 TEST 2 – ERROR FLIP FLOP TEST. This test checks the ability of the interface board to detect and report software timing violation and end of record set errors.

41.1.3 TEST 3 – READ CARD CHECK. This test checks to verify the test deck by comparing it with the test deck pattern in memory.

41.2 OPERATOR INTERFACE

CRDRDR runs under the DOCS on a 990 computer. For a detailed discussion of the DOCS see Section III of this manual.

41.2.1 AVAILABLE VERBS. In addition to the verbs supplied in the DOCS, CRDRDR supplies the verbs listed in table 41-1.

Table 41-1. CRDRDR Verbs

Verb	Function
EA	Run test 1, 2, and 3 once
E1	Run test 1 once
E2	Run test 2 once
E3	Run test 3 once
L1	Loop on test 1
L2	Loop on test 2

41.2.2 STANDARD INITIALIZATION. In addition to the questions required by the DOCS, CRDRDR asks the questions listed in table 41-2.

41.2.3 PROGRAMMER-PANEL INITIALIZATION. The programmer panel version of CRDRDR asks:

IDLE ON ERRORS?–

followed by the questions in table 41-2.

Table 41-2. CRDRDR Initialization Questions

Question	Comment
ENTER CARD READER CRU BASE, DEFAULT = 0040	Enter correct address
DO YOU WANT TO RUN TEST WITH INTERRUPTS? DEFAULT = 1	For yes press space bar For no enter 0
ENTER CARD READER INTERRUPT LEVEL, DEFAULT = 04	Enter correct interrupt level

41.3 SYSTEM REQUIREMENTS

CRDRDR will run on a 990 computer with at least 12K words of memory, an appropriate interactive device, a Model 804 Card Reader Kit, and a card reader test deck.

41.4 ERROR MESSAGES

CRDRDR error messages are listed in table 41-3.

Table 41-3. CRDRDR Error Messages

Error Number	Message
1	CARD PRESENCE HIGH***SHOULD BE LOW
2	ERROR FLIP-FLOP NOT RESET
3	COLUMN INDEX NOT RESET
4	CARD PRESENCE WAIT LIMIT EXCEEDED
5	READ, FEED, OR TIMING ERROR
6	CARD READER NOT READY
7	UNEXPECTED INTERRUPT AT LOCATION XXXX
8	CARD READER DATA COMPARE ERROR COLUMN XX PUNCH EXPECTED XX XX XX XX XX XX XX XX XX XX XX XX PUNCH READ XX XX XX XX XX XX XX XX XX XX XX XX
9	TIMING ERROR FLIP-FLOP NOT SET WHEN C/I INTERRUPT NOT RESET.
A	INTERFACE GOT COLUMN INDEX WHEN END OF RECORD FLIP-FLOP SET.
B	CARD READER TIMED OUT ON TB 15 HIGH IN READER SUBROUTINE

Table 41-3. CRDRDR Error Messages (Continued)

Error Number	Message
C	CARD READER TIMED OUT ON TB 15 LOW IN READER SUBROUTINE
D	CARD READER TIMED OUT ON TB 15 HIGH IN TEST 2
E	CARD READER TIMED OUT ON TB 15 LOW IN TEST 2
F	CARD READER TIMED OUT ON TB 7 LOW IN TEST 3

SECTION XLII

*TILINE FLOPPY DISK TEST (DDFLOP)

42.1 INTRODUCTION

The DDFLOP tests the TILINE Floppy Controller (TFC) and the associated drive units. DDFLOP is composed of eight parts as described in paragraphs 42.1.1 through 42.1.8.

42.1.1 PART 1 – TFC TEST. Part 1 runs a performance test on the TFC. A disk drive need not be present because none is selected. A failure in this part of the DDFLOP indicates a faulty TFC which means other parts of the test probably will not run correctly.

42.1.2 PART 2 – RANDOM READ AND ARM SLOP TEST. Part 2 performs 150 random reads and then checks the arm slop by reading each track from each direction.

42.1.3 PART 3 – ADDRESSING TEST. Part 3 checks the ability of the TFC and disk drive to select each track and each sector. In addition, 16 words of data are transferred to each track, read back, and checked for correctness.

42.1.4 PART 4 – ZEROS, ONES, AND WORST CASE DATA TEST. Part 4 performs a transfer of a sector of zeros, a sector of ones, and a sector of a Worst case data pattern (F6F6 for dual density or AAAA for single density diskette). The data is all transferred to and read back from cylinder 76 for greatest bit concentration.

42.1.5 PART 5 – CT, SI, ZERO FILL, AND BUSY FLAG TEST. Part 5 is divided into four parts, a, b, c, and d.

a. Part 5a, verifies that the controller can generate a command timeout (CT).

b. Part 5b, verifies that the controller can generate a seek incomplete (SI).

c. Part 5c, insures that data is not destroyed by a zero fill if a write data is executed with a word count of zero.

d. Part 5d, checks for normal operation of the busy flag (IDLE bit).

42.1.6 PART 6 – HEAD SWITCH TEST. Part 6 checks the ability of the unit to switch from head zero to head one and from head one to head zero for a double density media installed. If a single density media is installed, the test becomes a track switch test checking the ability of the unit to switch from one track to the next.

42.1.7 PART 7 – INTERACTIVE PART 1. Part 7 checks the operation of the door lock, the activity LED, write protect, unsafe, not ready, off line, and the dual and single density media sensors. Operator intervention is required so there is no loop on capability (L7 or C7, figure 42-1) for Part 7. The use of both a single and a double density diskette is required. The double density diskette must be formatted for Part 7 to run properly.

* Trademark of Texas Instruments Incorporated

42.1.8 **PART 8 – INTERACTIVE PART 2.** Part 8 is a power fail data integrity test. Operator intervention is required so there is no loop on capability (L8 or C8) for Part 8. A formatted diskette must be installed for part 8 to run properly.

42.2 OPERATOR INTERFACE

DDFLOP runs under the control of DOCS on a 990 Computer with TILINE. Refer to Section III of this manual for a detailed description of DOCS.

42.2.1 **AVAILABLE VERBS.** In addition to the verbs supplied by the DOCS package, DDFLOP provides the verbs listed in Table 42-1.

Table 42-1. DDFLOP Verbs

Verb	Function
IT	Initialize Test
E1	Execute Part 1
E2	Execute Part 2
E3	Execute Part 3
E4	Execute Part 4
E5	Execute Part 5
E6	Execute Part 6
E7	Execute Part 7
E8	Execute Part 8
E9	Execute Part 9
L1	Loop On Part 1
L2	Loop On Part 2
L3	Loop On Part 3
L4	Loop On Part 4
L5	Loop On Part 5
L6	Loop On Part 6
L9	Loop On Part 9

NOTE

There are no verbs to loop on Parts 7 or 8.

C2	Loop On Part 2 for all available floppy disk units
C3	Loop On Part 3 for all available floppy disk units
C4	Loop On Part 4 for all available floppy disk units
C5	Loop On Part 5 for all available floppy disk units
C6	Loop On Part 6 for all available floppy disk units
C9	Loop On Part 9 for all available floppy disk units
EA	Execute Parts 1-9 on the floppy disk unit under test
LA	Loop on Parts 1-6 and 9 on floppy disk unit under test
CA	Loop on Parts 1-6 for all available floppy disk units

NOTE

No floppy disk unit is required for the execution of part 1 but part 1 is included in CA to insure the exercising of the TFC.

Table 42-1. DDFLOP Verbs (Continued)

Verb	Function
PE	Print error count
IC	Issue command
IM	Issue multiple commands
LO	Loop on multiple commands
DC	Display controller status
ST	Execute controller self test Part (n)
FD	Format diskette
OT	Execute off track data test
RM	Read controller memory
WM	Write controller RAM
AL	Disk alignment verb

42.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by DOCS, DDFLOP asks the questions listed in table 42-2.

Table 42-2. DDFLOP Initialization Questions

Question	Comment
FLOPPY TILINE ADDR. DEFAULT = F800 –	Enter correct TILINE address
FLOPPY INTERRUPT LEVEL DEFAULT = 0D –	Enter correct interrupt level
UNIT TO BE TESTED DEFAULT = 00 –	Enter correct unit designator
OTHER AVAILABLE UNITS (0-3 TERM WITH FF) DEFAULT = FF –	Enter FF or correct unit designator

42.2.3 **PROGRAMMER – PANEL INITIALIZATION.** The programmer-panel version of DDFLOP asks:

IDLE ON ERRORS?

followed by the questions in table 42-2.

42.3 SYSTEM REQUIREMENTS

The DDFLOP will run on a 990 Computer with TILINE and at least 12K words of memory, an appropriate loading device, a TILINE Floppy Kit part number 2261685-0001, and an appropriate interactive device.

42.4 ERROR MESSAGES

DDFLOP error messages are listed in table 42-3.

Table 42-3. DDFLOP Error Messages

Error Number	Message
	Part 1
11	1-1 NO COMMUNICATIONS WITH CONT REG XX CHANGE THE CONTROLLER
12	1-2 NO SET/RESET BIT XX OF CONT REG X MAY BE THE CONTROLLER OR THE TILINE
13	1-3 NO SET/RESET CONT REG 5 BIT XX MAY BE THE CONTROLLER OR THE TILINE
14	1-4 NO SET/RESET CONT REG 0 BIT XX MAY BE THE CONTROLLER OR THE TILINE
15	1-5 STORE REG ERROR WORD1 WORD2 WORD3 XXXX XXXX XXXX
16	1-6 NO UNIT SELECTED, REG 0 SHOULD=C0X0 REG 7 SHOULD = A001 REG 0 = XXXX REG 7 = XXXX
17	1-7 FAIL SELF TEST CONT REG 2 = XXXX EXPECTED REG 7 = A100 AFTER IO RESET REG 0 = XXXX REG 7 = XXXX
18	1-8 TFC REG DATA TEST WAS EXPT REG XXXX XXXX XX
19	1-9 FAILURE IN REG 7 INTERRUPT LOGIC
1A	1-A FAILURE REG 0 INTERRUPT LOGIC FLOPPY INTERRUPT DID NOT OCCUR IN 8 MS
1B	1-B TILINE TIMEOUT ERROR REG 7 STATUS EXP = A020 REG 0 = XXXX REG 7 = XXXX NO TILINE TIMEOUT IN 3 SECONDS
1C	1-C ERROR DID STORE REG WITH WC = XX BUT WORDS TRANSFERRED = XX CONTROLLER MALFUNCTION

NOTE

Parts 2 and 3 have no error messages of their own. However, they may output error messages listed in other parts.

	Part 4
41	4-1 ERROR DID RESTORE WITH NO UNIT SELECTED EXPECTED UNIT ERROR (REG 7 = A0001) REG 0 = XXXX REG 7 = XXXX

Table 42-3. DDFLOP Error Messages (Continued)

Error Number	Message
	Part 5
51	5-1 ERROR PART 5 SI TEST REGO STATUS EXP=04XX REG7 STATUS EXP= A0001) REG 0 = XXXX REG 7 = XXXX
52	5-2 ERROR PART 5C DID WRITE WITH WC=0 BUT DATA CHANGED EXPECTED DATA = E5E5 BUT GOT DATA = XXXX
53	5-3 ERROR PART 5 BUSY FLAG SET DATA (MSB SHOULD BE 0) XXXX
54	5-4 PART 5A TIMEOUT ERROR NO COMMAND TIMEOUT IN 500 MS
55	5-5 COMMAND TIMEOUT ERROR REG 7 STATUS EXP=A004 REG 0 = XXXX REG 7 = XXXX
56	5-6 DID A READ WITH WORD COUNT = 0 EXPECTED DATA = 0 BUT GOT DATA = XXXX
	Part 6
61	6-1 FAILURE TO SWITCH HEADS FROM 0 TO 1 CYL 0
62	6-2 FAILURE TO SWITCH HEADS FROM 1 TO 0 CYL 0
63	6-3 PART 6 HEAD SWITCH FAIL ON DATA COMPARE
	Part 7
71	7-1 ERROR WRITE PROTECT TEST EXPT REG 0 = 20X0 REG 0 = XXXX REG 7 = XXXX
72	7-2 ERROR OFF LINE, NOT READY UNSAFE TEST EXPT REG 0 = D0X0 REG 0 = XXXX REG 7 = XXXX
73	7-3 ERROR IN STORE REGISTERS VALUES EXPT XXXX XXXX XXXX RECD XXXX XXXX XXXX
	Part 8
81	8-1 DATA INTEGRITY TEST FAILED. MAY BE DUE TO A BADLY FORMATTED DISKETTE

Table 42-3. DDFLOP Error Messages (Continued)

Error Number	Message
	COMMAND ISSUER, INTERRUPT, AND COMPARE DATA ROUTINES
91	9-1 UNIT=X REG.7 = XXXX FLOPPY STATUS UNIT=X REG.0 = XXXX
92	9-2 NR BIT FOR UNIT XX RESET TOO SOON
93	9-3 UNEXPECTED FLOPPY INTERRUPT AT XXXX REG 0 = XXXX REG 7 = XXXX
94	9-4 THE TFC DID NOT RESPOND IN 10 SECONDS
95	9-5 THE ATTN LINES DID NOT SET IN 10 SECONDS
96	9-6 IDLE BIT SET WITHOUT A COMPLETE OR ERROR REG 0 = XXXX REG 7 = XXXX
97	9-7 IDLE DID NOT OCCUR WITHIN 2.5 SECONDS AFTER AN IO RESET REPLACE THE CONTROLLER
98	9-8 DATA COMPARE ERROR CYL SECT HEAD WONT EXPT RECD XXXX 00XX 00XX XXXX XXXX XXXX
	Part 9
9A	DATA NOT EQUAL TO ADDRESS ADDRESS = XXXX DATA = XXXX
9B	COMPARISON ERROR IN MAPPED MEMORY MAP ADDRESS = XXXX MAP BIAS = XXXX TILINE ADDRESS = XXXXXX EXPECTED DATA = XXXX DATA READ = XXXX

SECTION XLIII

TMS 9900/9980 EMULATOR AND BUFFER TEST (EMU900)

43.1 INTRODUCTION

The EMU900 tests the emulator, part number 941555-0001, combined with one of the following buffers:

1. TMS9900-1 Buffer Board, Part Number 949973-0001
2. Buffer Module Assembly, Part Number 941560-0001
3. SBP9900A Buffer Board, Part Number 949972-0001
4. Buffer Module Assembly, Part Number 941550-0001
5. TMS9980A/81 Buffer Board, Part Number 949974-0001
6. Buffer Module Assembly, Part Number 941565-0001

EMU900 exercises all functions of the 9900/9980 Emulator and its associated Buffer Board. EMU900 also includes a verb (EM) which completely tests the Expansion Memory Board when used.

The EMU900 consists of 13 parts which are described in paragraphs 43.1.1 through 43.1.13.

43.1.1 PART 1 – HOST/EMULATOR INTERFACE TEST (E1 VERB). Part 1 tests the 16 bit data field on the host/emulator CRU interface; interrupt capability; entry into and exit from run mode; and all control lines associated with each of the above operations. This test does not check out the 8 bit command field.

The test begins by executing the RESTART verb. If the control program is operating properly the pattern 4869_{16} (ASCII representation for HI) will appear on the host CRU-IN Field. This is the first and most basic check of the control program and the CRU interface. Next an invalid opcode is output to the control program to see whether it will properly respond by raising COMCMP (Command Complete) high which indicates an error condition. Then the RESTART verb is executed to force the emulator to the beginning of the control program, the communications protocol is then partially executed by raising HSTRED (Host Ready) high and waiting for EMREDY (EMU Ready) to go low. For a detailed explanation of the control program and the communications protocol refer to the 9900/9980 emulator hardware specification, part number 941552-0001. In this partially executed mode the control program will echo any pattern that it sees on the Host CRU-OUT, back to the Host CRU-IN. The test uses this feature and writes patterns of FFFF, 8000_{16}, 4000_{16}, 2000_{16}, 1000_{16}, 800_{16}, . . . 0000_{16} to the CRU-OUT and checks for them on the CRU-IN.

The next portion of the test checks out the interrupt processing; the IDLE bit present at CRU-IN; and whether or not the emulator can enter into and exit the RUN mode successfully. INTPEN (Interrupt Pending) is cleared so that the emulator will recognize a new interrupt. If the INTPEN bit will not clear, an error message is printed and the entire test will be aborted. Once INTPEN is

cleared, HSTRED is set high which causes EMREDY to go low. When EMREDY goes low this causes an interrupt to be generated. The interrupt should be present at INTPEN although in this case this will not create a CRU interrupt since CRU interrupts have been previously masked off with INTENA (Interrupt Enable). If INTPEN does not indicate an interrupt present an error message will be printed out. Next, an interrupt is again generated using EMREDY but this time the CRU interrupts are not masked off. This causes the test to process the interrupt through its interrupt processor and if for any reason the interrupt is not processed, an error message will be printed. Next, prototype memory is filled with JMP $ instructions and an IDLE instruction is placed at address 500_{16}. The PC is then set to 500_{16} and the program is run. During this operation, HLTACK (Halt Acknowledge), and IDLE (IDLE Bit) are checked for proper operation. Finally, the interrupt feature is checked once more to see that HLTACK creates an interrupt when the Emulator switches from RUN mode to CONTROL mode. This concludes the part 1 Interface Test.

43.1.2 **PART 2 – MEMORY TESTS (E2 VERB).** Part 2 tests emulator or target memory (depending on which has been previously selected with the "SM" – Select Memory verb), and Control Memory. Both sets of memory are tested with a pattern test, an address test and a speed test. The verb requests beginning and ending test addresses and uses these bounds in the pattern and address tests. This feature of the Memory test can be used to check out the expansion memory board. The initial beginning and ending bounds at startup of the test are from 0000_{16} to 1FFE which would include only 8K bytes of Prototype memory, either emulator or target. Once the bounds are modified they remain as modified throughout all loop and execute verbs using these memory tests. The speed tests do not use the same bounds as the pattern and address tests but instead check memory in blocks of either 0-8K bytes, 0-16K bytes or 0-64K bytes depending on whether the buffer connected is a TMS9980A/81 and depending on whether the beginning address is above a certain 8K byte boundary.

The pattern test is the same for both prototype (EMU 8K byte or target) and control memory. It consists of writing and reading specific patterns (8000_{16}, 4000_{16}, 2000_{16}, . . . , 0000_{16}, FFFF) to each separate memory location. Each pattern will be written to all memory in one operation and then read back in one operation until all patterns have been used. If any bit errors occur while writing or reading memory if there are any data bus problems, this test should find them.

The address test will be the same in both control and prototype memory. In this test each memory location is loaded with values that are equal to the address being accessed. In this way the address lines are tested to see if a value is written into the wrong location. If an error occurs, the address of the error, expected data and actual data are returned.

The type of procedure used for the speed test is the same for both control and prototype memory. The test consists of downloading a test procedure into lower memory which successively writes and reads alternating 1's and 0's to each memory location, testing all memory locations one at a time. The procedure starts out by filling memory with AAAA's, then, beginning with the first available memory location it writes 5555_{16} to that location and immediately reads it back, shifts it left by one and writes it again to the same location, reads it back again, shifts it to the right this time and writes it once again to the same location. The final result should be that memory will be filled with 5555_{16}'s. If any speed errors occur during this test they will propagate through and the final value will be different from 5555_{16}. After the test is through running the diagnostic reads all memory and reports any errors along with the address of the error and the actual and expected values.

43.1.3 PART 3 – BITS REGISTER TEST (E3 VERB). Part 3 tests the hardware bits register by using two test subroutines (RIPTST and WRTPAT) and by exercising all the verb commands which set and reset each of the bit locations on the register. RIPTST (Ripple Pattern Test) is a subroutine used in this and subsequent execute verbs which write and read various patterns (8000_{16}, 4000_{16}, 2000_{16}, 1000_{16}, . . . , 0000_{16}) to a selected hardware register, reporting any errors and displaying both expected and actual data patterns. This test locates bad bit locations in a register. WRTPAT (Pattern Test) is another subroutine which writes and reads a set of patterns to a selected hardware register. The patterns used in this test are FFFF, AAAA, 5555_{16}, and 0000_{16}. This subroutine reports errors in a manner similar to that used in RIPTST.

Following these tests each individual bit position on the register is set and checked using the particular control program verb associated with it. For instance, the "DT" (Define Trace) verb is used to set bits 12-15 on the register, and the "DB" (Define Breakpoint) verb is used to set bits 5-7 depending upon the conditions specified by the verb. Using both these hardware and software methods of testing, the complete operation of the bits register can be checked out.

43.1.4 PART 4 – LOWER AND UPPER BOUNDS HARDWARE TEST (E4 VERB). Part 4 tests the lower and upper trace-bounds-registers. Both registers are tested identically. The registers are first tested with "RIPTST" (Ripple Test) and "WRTPAT" (Pattern Test) as described in paragraph 43.1.3 above. Finally the patterns 55AA and AA55 are written and read successively to each of the registers using the DT (Define Trace) verb. This is accomplished by first setting up the DT verb and then executing a RE (Run Emulation) verb. As soon as the RE verb is executed the control program sets up all necessary hardware registers including both upper and lower bounds registers with the previously specified values. In this way both the hardware and the software functions of the registers are tested.

43.1.5 PART 5 – BREAKPOINT HARDWARE REGISTER TEST (E5 VERB). Part 5 tests both breakpoint hardware registers A and B. Both registers are tested identically. The registers are first tested with "RIPTST" (Ripple Test) and "WRTPAT" (Pattern Test) as described in paragraph 43.1.3 above. Finally the patterns 55AA and AA55 are written and read successively to each of the registers using the DB (Define Breakpoint) verb. This is accomplished by first setting up the DB verb and then executing the RE (Run Emulation) verb. As soon as the RE verb is executed the control program sets up all necessary hardware registers including both upper and lower bounds registers with the previously specified values. In this way both the hardware and the software functions of the registers are tested.

43.1.6 PART 6 – TRACE FUNCTION AND HARDWARE TESTS (E6 VERB). Part 6 tests the complete operation of the tracing functions of the emulator with the exception of the trace module/emulator interface. Part 6 consists of five subtests as described in paragraphs 43.1.6.1 through 43.1.6.5.

43.1.6.1 Trace Counter Subtest. This subtest checks to see that the emulator will make simple traces in prototype memory in the MA mode. Memory is filled with JMP $ instructions; an IDLE instruction is placed in memory and the RE verb is executed. The result is that the IDLE bit becomes active. When this occurs the test halts the emulator and reads the value in the trace counter which should be equal to the original value loaded minus one. This same test is repeated using trace counter values of 01, 02, 04, . . . , 80_{16}. When a value of 01_{16} is used the trace counter will decrement through 0 and will immediately halt. This, therefore, tests out the emulator's ability to halt when the trace memory full (TMF) condition is met.

43.1.6.2 **Bounds Comparison Circuitry Subtest.** This subtest consists of three subparts which together completely check out the bounds comparison circuitry. For the first subpart, a test procedure is downloaded into prototype memory after filling memory with JMP $ instructions. The procedure accesses one memory location starting with 8000_{16} - 0000_{16} dividing the memory value by 2 each time. In this way all bits of the compare registers are checked out. After a memory location is accessed, the emulator idles, is halted and the value in the trace counter is compared to its expected value. The trace counter is always initialized at FF, therefore, the expected value will always be FE. The downloaded procedure is executed in the MA mode with the upper-bound = lower-bound = 8000_{16}, 4000_{16}, 2000_{16}, . . . , 0000_{16}, and the workspace pointer is always set equal to the upper-bound = lower-bound. The trace counter is always reset to FF. The second subpart of the subtest checks out the other half of the output lines (A>B, address>bound) on the lower bound address comparators. It does so by setting the lower bound to 30_{16}, the upper bound to 0 and placing the test procedure in the excluded range ($0\text{-}30_{16}$), then a procedure is executed which accesses memory above the lower bound of 30_{16} thus causing the A>B lines to toggle. The fact that the code executes beneath the lower bound range and above the upper bound range automatically causes the A<B outputs on the lower bound comparators and the A>B outputs on the upper bounds comparators to toggle so there is no need for a separate test for those output lines. In a similar manner the third subpart of the subtest checks out the A<B lines on the upper bound comparators by accessing memory below the upper bound.

43.1.6.3 **Trace Memory Address and Data Lines Subtest.** This subtest checks out the address and data lines of the 512 byte trace RAM. In checking the address lines, memory is first filled with an FFFE pattern and then a 0 is placed at one location in memory. Next the entire trace RAM is read to see if the 0_{16} appears anywhere else in memory. This is done for each location in memory. In checking the data lines a type of pattern test using patterns of 8000_{16}, 4000_{16}, 2000_{16}, . . . , 0000_{16} is executed.

43.1.6.4 **Diagnostic Memory Address and Data Lines Subtest.** This subtest checks out the address and data lines of the diagnostic RAM along with four of the input lines of the 2-to-1 data selector, by outputting the patterns 0, 2, 4, 10_{16}, 14_{16}, A, and 1E. These patterns are used instead of 0, 5, A and F because this particular RAM and multiplexer traces data lines 11 through 14 instead of 12 through 15 which means that the test patterns must be shifted one place to the left in order to appear as 0, 5, A, and F. The test also checks to see that the emulator will function in the PC and EX modes. This is accomplished by downloading and executing a program which traces a known number of samples in each of these modes. After the procedure is executed the trace counter is checked for the correct final value.

43.1.6.5 **Diagnostic Memory Speed Subtest.** This subtest uses the two control lines IAQ and DBIN in checking the ability of the emulator to trace the processor's control lines IAQ and DBIN which feed into the trace memory through the 2-to-1 data selector. It does so by loading and executing a test procedure which toggles these control lines at a high rate of speed. The test procedures contain instructions selected because of their particular read and write machine cycles. For instance, in testing the DBIN line an instruction is needed that has a read cycle followed closely by a write cycle.

43.1.7 **PART 7 – BREAKPOINT FUNCTION TESTS (E7 VERB).** Part 7 tests out the breakpoint mode functions and breakpoint comparator circuitry. The breakpoint comparator circuitry is checked first by loading a test procedure into prototype memory which accesses workspace register 0 one time with a MOV R0,R1 instruction. The workspace pointer and either breakpoint A or B depending on which is being tested is set to one of a group of values (8000_{16}, 7800_{16}, 4000_{16}, 2000_{16}, 1000_{16}, 800_{16}, 7C0, 400_{16}, 200_{16}, 100_{16}, 80_{16}, 40_{16}, 3E, 20_{16}, 10_{16}, 8, 4, 2, 0) and the halt on breakpoint function is enabled so that when the downloaded procedure is executed it should cause the emulator to halt. If the emulator halts too soon or if it does not halt at all, an

error message is output. The above values will successively test each address bit of the compare registers. The reason for the values 7800_{16}, 7C0, and 3E is that they set all the bits high on each of the MSB-LSB comparator packages respectively thus guaranteeing that if any of the input lines to these packages are bad the user will be able to pinpoint more accurately which of the packages is at fault.

Next, all possible breakpoint mode combinations are checked by executing a procedure out of prototype memory which generates each of the possible cycles used by the emulator to qualify the halt condition (memory read, memory write, and instruction acquisition). The test checks not only to see that the emulator will halt on a breakpoint but that it will also not halt if the halt-on-breakpoint feature is disabled. Both sets of breakpoint compare circuitry A and B are tested identically with the same verb.

43.1.8 PART 8 – CPU TEST (E8 VERB). Part 8 loads and executes the standard AU04 Standalone CPU Test in emulator memory starting at address 0100_{16}. If the test encounters any CPU errors, it will place an error code in lower prototype memory depending on the type of error (error codes start at address 0000_{16}). After the test completes and IDLEs, the diagnostic checks for any error codes returned and reports all commands that failed. At the end of the error list the AU04 test inserts an end-of-list pointer (FFFF). If no errors occur this will be the first value in memory. If for any reason an incorrect error code is encountered the diagnostic reports an undefined error.

43.1.9 PART 9 – EMULATOR/TRACE MODULE INTERFACE TEST (E9 VERB). Part 9 tests the trace module/emulator interface. The trace module must be connected to the emulator via the data and control cables during the execution of this test.

The test consists of loading and executing a single test procedure from prototype memory. The test procedure is designed such that it checks out both the data and all control lines between the trace module and emulator. It is, however, not possible to test all the control lines at the same time, therefore, the test is executed twice, each time setting up different trace configurations. The first time the procedure is executed it tests all data lines, control lines and qualifiers with the exception of D1, the IAQ data line qualifier, P4-48 of the trace module/emulator data connector. The second execution of the procedure checks out D1. The data lines are tested by tracing a shifting pattern starting with 0001_{16} and continuing through 8000_{16} while the qualifiers and control lines are tested by causing the trace module to issue a HALT when it encounters certain conditions in the test procedure. The emulator must then acknowledge the HALT and issue a HOLD which causes the trace module to stop. When the trace module is halted this causes the emulator to halt also.

43.1.10 PART 10 – EXECUTE BUFFER/TARGET SYSTEM INTERFACE TEST (TS VERB). Part 10 tests the target system/buffer interface using any of the available buffers. The appropriate target system must be connected in order to execute this test.

With the specially modified target system the diagnostic is able to test the CRU data lines and all control lines. The CRU data lines are tested using the RC and WC (Read and Write CRU Field) control program verbs and reading and writing patterns of 2, 4, 10_{16}, 100_{16}, and 8000_{16}. Next, the interrupt capability is tested by setting up all interrupt traps in prototype memory and generating interrupts for levels 1, 2, 4, 8, and 15 (levels 1, 2, 3, and 4 for the TMS9980A/81). Each interrupt should return a code to workspace register 0 equal to its interrupt level. Thus, interrupt level 8 returns a code of 8 in register 0 (workspace pointer set to 100_{16}). Following this the interrupt request line (INTREQ) to the emulator is tested by tracing a procedure executing out of prototype memory which causes INTREQ to toggle. Next, RESET and LOAD are tested by setting up their interrupt vectors in prototype memory and checking for a proper return code in register

0 as in the previous interrupt test. Finally, target system HOLD is tested with a test procedure executing out of prototype memory. The test procedure sets and checks the HOLD line. If HOLD is operating properly, a code of FEED is returned to register 0. If an error occurs the procedure returns an error code of 0001 for HOLD stuck high or 0002 for HOLD stuck low. This concludes the target system/buffer interface test.

43.1.11 PART 11 – EXPANSION MEMORY BOARD TEST (EM VERB). Part 11 consists of three basic parts. The first part tests the 256 X 4 bit memory mapping/write protect RAM by writing and reading the patterns 0000, 0100_{16}, 0200_{16}, 0400_{16}, 0800_{16}, 0500_{16}, $0A00_{16}$, and 0F00. The patterns are written beginning at one of two different addresses depending on which buffer board is connected: addresses 1000_{16} - 11FE for the TMS9980A/81 Buffer Board and addresses 2000_{16} - 21FE for the SBP9900A and TMS9900-1 buffer boards. This part finishes leaving the RAM with F's written in all locations which disables the write protect feature for all of expansion memory and maps all addressing to expansion memory.

The next test of Part 11 checks all of the expansion memory by using the existing pattern, address and speed tests from the E2 verb. Any errors that occur during this portion of the test will be reported exactly as they would while executing the E2 verb. All the tests from the E2 verb are used in this portion of the EM verb with the exception of the emulator control memory pattern, address and speed tests.

The final test of Part 11 checks the write protect Violation Register and Flag. First, certain memory locations are cleared and then all memory is write protected and mapped to the expansion memory board. The write protect violation flag is then checked to make sure it is not active (high). Then FFFF is written to a selected memory location (8000_{16}, 4000_{16}, 2000_{16}, . . . , 0000) and the write protect violation flag is checked for active condition. Next, the write protect violation register is read at address 2200_{16} in control memory (address 1200_{16} for the TMS9980A/81) to see that the proper address was stored. The write protect violation flag is then checked to make sure that it was cleared after reading the violation register, and finally expansion memory is read at the selected location to make sure that the data remained unchanged.

43.1.12 PART 12 – EXPANSION MEMORY WRITE PROTECT REGISTER TEST (EW VERB). Part 12 tests the write protect violation register and flag. First, certain memory locations are cleared and then all memory is write protected and mapped to the expansion memory board. The write protect violation flag is then checked to make sure it is not active (high). Then, a selected memory location (8000_{16}, 4000_{16}, 2000_{16}, . . . , 0000) is written with a FFFF and the write protect violation flag is checked for active condition. Next, the write protect violation register is read at address 2200_{16} in Control Memory (address 1200_{16} for the TMS9980A/81) to see that the proper address was stored. The write protect violation flag is then checked to make sure that it was cleared after reading the violation register, and finally expansion memory is read at the selected location to make sure that the data remained unchanged.

43.1.13 PART 13 – LED TEST (IE VERB, RE VERB, IE VERB). Part 13 uses the following test procedure to check the operation of the emulator green LED lamp. To test the LED the diagnostic must not be in the GC (Generate Command) mode. The user simply enters an IE verb which will disable all tracing and breakpoints (note that the IE verb will not modify breakpoint operation while in the GC mode), followed by a RE (Run Emulation) verb which will cause the LED to light and remain lit. To turn off the LED the user again enters the IE verb. This concludes the LED test.

43.2 OPERATOR INTERFACE

EMU900 runs under DOCS on a 990 computer. Refer to Section III of this manual for a detailed description of DOCS.

43.2.1 AVAILABLE VERBS. In addition to the verbs supplied by the DOCS package, EMU900 provides the verbs listed in table 43-1.

Table 43-1. EMU900 Verbs

Verb	Function
IT	INITIALIZE TEST
L1	LOOP ON HOST/EMULATOR INTERFACE TEST
L2	LOOP ON MEMORY TESTS EXCEPT SPEED TESTS
L3	LOOP ON BITS REGISTER TEST
L4	LOOP ON LOWER AND UPPER BOUNDS HARDWARE TEST
L5	LOOP ON BREAKPOINT HARDWARE REGISTER TEST
L6	LOOP ON TRACE FUNCTION AND HARDWARE TESTS
L7	LOOP ON BREAKPOINT FUNCTION TESTS
L8	LOOP ON CPU TEST
L9	LOOP ON EMULATOR/TRACE MODULE INTERFACE TEST
L0	LOOP ON BUFFER/TARGET SYSTEM INTERFACE TEST
LE	LOOP ON EXPANSION MEMORY TEST
LW	LOOP ON EXPANSION MEMORY WRITE PROTECT REGISTER TEST
S1	LOOP ON PROTOTYPE MEMORY SPEED TEST
S2	LOOP ON CONTROL MEMORY SPEED TEST
LA	LOOP ON PARTS 1-8
E1	EXECUTE HOST/EMULATOR INTERFACE TEST
E2	EXECUTE MEMORY TESTS
E3	EXECUTE BITS REGISTER TEST
E4	EXECUTE LOWER AND UPPER BOUNDS HARDWARE TEST
E5	EXECUTE BREAKPOINT HARDWARE REGISTER TEST
E6	EXECUTE TRACE FUNCTION AND HARDWARE TESTS
E7	EXECUTE BREAKPOINT FUNCTION TESTS
E8	EXECUTE CPU TEST
E9	EXECUTE EMULATOR/TRACE MODULE INTERFACE TEST
TS	EXECUTE BUFFER/TARGET SYSTEM INTERFACE TEST
EM	EXPANSION MEMORY BOARD TEST
EW	EXPANSION MEMORY WRITE PROTECT REGISTER TEST

Table 43-1. EMU900 Verbs (Continued)

Verb	Function
EA	EXECUTE PARTS 1-8
HT	HALT EMULATOR
RS	RESET EMULATOR
SC	SELECT CLOCK
GC	GENERATE COMMAND
XT	EXECUTE COMMAND TABLE
LT	LOOP ON EXECUTE COMMAND TABLE
SL	SHOW COMMAND TABLE LIST
IE	INITIALIZE EMULATOR
SM	SELECT MEMORY (EMULATOR OR TARGET)
CK	READ CLOCK SELECTION
RR	READ PROTOTYPE CPU REGISTERS
WR	WRITE PROTOTYPE CPU REGISTERS
RM	READ MEMORY
WM	WRITE MEMORY
RC	READ TARGET SYSTEM CRU FIELD
WC	WRITE TARGET SYSTEM CRU FIELD
DT	DEFINE TRACE CONFIGURATION
DB	DEFINE BREAKPOINT CONFIGURATION
RT	READ TRACE SAMPLES
RE	RUN EMULATION
CP	RUN PROGRAM IN CONTROL RAM
PC	SELECT PROTOTYPE OR CONTROL MEMORY
PE	PRINT ERROR COUNT
DS	DUMP EMULATOR STATUS
LB	LOAD CRU BASE
LD	LOAD CRU
ST	STORE CRU
SO	SET CRU BIT TO ONE
SZ	SET CRU BIT TO ZERO
BT	TEST CRU BIT
BS	BIT SAMPLE

Table 43-1. EMU900 Verbs (Continued)

Verb	Function
HD	CONVERT HEX TO DECIMAL
DH	CONVERT DECIMAL TO HEX
BP	SET BREAKPOINT
CB	CLEAR BREAKPOINT
BL	LOOP ON SET BIT
LO	LOOP ON ONE
LZ	LOOP ON ZERO

NOTE

The EMULATOR and a BUFFER BOARD only should be connected to execute verbs E1-E8, EM, L1-L8, and LE. The TRACE MODULE should not be connected during these tests or error messages will result. The TRACE MODULE should be connected only during use of the E9 Verb (Emulator/Trace Module Interface Test) and its associated loop verb, L9. For the TS and associated loop verb LO, only an EMULATOR, BUFFER BOARD, and TARGET SYSTEM should be connected. For the EM, EW, and associated LE verb, only an EMULATOR, BUFFER BOARD and EXPANSION MEMORY BOARD should be connected.

43.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by DOCS, EMU900 asks the questions listed in table 43-2.

Table 43-2. EMU900 Initialization Questions

ENTER TRACE MODULE CRU BASE ADDRESS DEFAULT=0100 -

ENTER THE EMULATOR CRU BASE ADDRESS DEFAULT=0140 -

******WARNING******

Emulator and I/O device interrupt levels must not be the same. Enter the emulator int. level default=0006.

43.2.3 **PROGRAMMER-PANEL INITIALIZATION.** The programmer-panel version of EMU900 asks:

IDLE ON ERRORS?

followed by the questions of table 43-2.

43.3 SYSTEM REQUIREMENTS

The EMU900 will run on a 990 computer with at least 48K bytes of memory, emulator-buffer modules (refer to paragraph 43.1), trace module part number 949910-0001, an appropriate loading device and an appropriate interactive device.

43.4 ERROR MESSAGES

EMU900 error messages are listed in table 43-3.

Table 43-3. EMU900 Error Messages

Error Number	Message
01	UNEXPECTED EMULATOR INTERRUPT
02	HLTACK LINE FAILED TO RESPOND AFTER HALT WAS ISSUED
03	EMREDY LINE FAILED TO RESPOND AFTER HALT WAS ISSUED
04	COMMUNICA. ERR: HLTACK FAILED TO GO HIGH FOR "RE"
05	COMMUNICA. ERR: EMREDY LINE LOW, EXPECTED HIGH
06	COMMUNICA. ERR: EMREDY LINE FAILED TO GO LOW
07	COMMUNICA. ERR: EMREDY NOT HIGH AFTER HOST READY SET LOW
08	INVALID COMMAND
09	INDEX OUT OF BOUNDS
0A	SEMANTIC ERROR
0B	WARNING: COMMND BIT HIGH, ERROR MESSAGE MUST FOLLOW
0C	BAD OPCODE AT COMMAND #XXXX
0D	EMU FAILED TO RESTART
0E	BAD OPCODE=XXXX
0F	FIRMWARE FAILED TO START AT TOP AFTER HALT
10	CRU INTERFACE ERR: COMCMP BIT STUCK HIGH
11	CRU INTERFACE ERR: COMCMP BIT STUCK LOW
12	CRU INTERFACE ERR: IDLE BIT STUCK LOW
13	CRU INTERFACE ERR: EXPTD CRU DATA=XXXX, ACTL DATA=XXXX
14	INTPEN LINE STUCK HIGH, INTERRUPT TEST ABORTED

Table 43-3. EMU900 Error Messages (Continued)

Error Number	Message
15	INTERRUPT TEST ERR: EMREDY NEVER WENT LOW TO CAUSE INTERRUPT
16	INTPEN FAILED TO SHOW INTERRUPT W/EMREDY; CRU INTRPT MASKED
17	INTPEN FAILED TO CAUSE CRU INTERRUPT W/EMREDY
18	HLTACK WAS NOT LOW IN CONTROL MODE
19	HLTACK DOES NOT INDICATE EMULATION RUNNING
1A	EMULATOR FAILED TO RE-ENTER CONTROL MODE AFTER EMULATION
1B	HLTACK FAILED TO CREATE INTERRUPT
1C	PATRN ERR: @ADDR=XXXX, EXPTD DATA=XXXX, ACTL DATA=XXXX
1D	ADDRESS TEST ERR: ADDR & DATA NOT EQUAL, ADDR=XXXX, DATA=XXXX
1E	@SPEED EMULATION FAILED
1F	@SPEED MEM ERR: @ADDR=XXXX, EXPTD DATA= XXXX, ACTL DATA=XXXX
20	@SPEED CONTROL TEST FAILED TO COMPLETE
21	COMMND BIT FAILED, TEST ABORTED
22	BITS REG. ERR: IDLE BIT STUCK LOW
23	BITS REG. ERR: IDLE BIT STUCK HIGH
24	BITS REG. FAILED TO CLEAR, ACTL DATA=XXXX
25	BITS REG. ERR: BIT #XX FAILED "DB" COMMAND
26	BITS REG. ERR: MAP BIT FAILED
27	BITS REG. ERR: EMU 4K SELCT BIT FAILED
28	BITS REG. ERR: BIT #XX FAILED "DT" COMMAND, EXPTD DATA=XXXX, ACTL DATA=XXXX
29	PATRN ERR: @ADDR=XXXX, EXPTD PATRN=XXXX, ACTL PATRN=XXXX
2A	RIPPLE ERR: @ADDR=XXXX, TEST DATA=XXXX, ACTL DATA=XXXX
2B	LOWER BOUND REG. FAILED TO CLEAR, ACTL DATA= XXXX
2C	UPPER BOUND REG. FAILED TO CLEAR, ACTL DATA= XXXX

Table 43-3. EMU900 Error Messages (Continued)

Error Number	Message
2D	UP BND REG FAILED "DT" COMMAND, EXPTD DATA= XXXX ACTL DATA=XXXX
2E	LO BND REG FAILED "DT" COMMAND, EXPTD DATA= XXXX ACTL DATA=XXXX
2F	BREAKPT A REG FAILED TO CLEAR, ACTL DATA= XXXX
30	BREAKPT B REG FAILED TO CLEAR, ACTL DATA= XXXX
31	BP A FAILED "DB" COMMAND, EXPTD DATA=XXXX, ACTL DATA=XXXX
32	BP B FAILED "DB" COMMAND, EXPTD DATA=XXXX, ACTL DATA=XXXX
33	EMULATOR FAILED TO HALT ON TRACE MEMORY FULL
34	EMULATOR FAILED TO INDICATE IDLE EXECUTION
35	TRACE COUNTER ERR: EXPTD COUNT=XX, ACTL COUNT=XX
36	UB COMPAR FAILED ON XXXX, EXPTD TRC CNT=XX, ACTL CNT=XX, UB=XXXX, LB=XXXX
37	LB COMPAR FAILED ON XXXX, EXPTD TRC CNT=XX, ACTL CNT=XX, UB=XXXX, LB=XXXX
38	COMPAR ADDR ERR: FAILED COMPARING XXXX, EXPTD TRC CNT=XX, ACTL CNT=XX
39	TRC MEM ADDR ERR: @ADDR= XXXX, ACTL DATA= XXXX
3A	TRC MEM PATRN ERR: @ADDR=XX, EXP PATRN= XXXX, ACTL PATRN=XXXX
3B	DIAG RAM ERR: @ADDR=XX, EXPTD DATA=X, ACTL DATA=X
3C	PC MODE ERR: WRONG ADDR TRACED, EXPTD ADDR=XXXX, ACTL ADDR=XXXX
3D	EA MODE ERR: WRONG ADDR TRACED, EXPTD ADDR=XXXX, ACTL ADDR=XXXX
3E	PC MODE ERR: BAD TRACE CNT, EXPTD CNT=XX, ACTL CNT=XX
3F	EA MODE ERR: BAD TRACE CNT, EXPTD CNT=XX, ACTL CNT=XX
40	IAQ/DBIN TRACE SPEED ERR: @ADDR=XX, EXPTD PATRN=X, ACTL PATRN=X

Table 43-3. EMU900 Error Messages (Continued)

Error Number	Message
41	EMULATOR HALTED, SHOULD NOT HAVE
42	EMULATION FAILED COMPLETION
43	COMPAR REG FAILED COMPARING ADDR XXXX
44	EMULATOR FAILED TO HALT ON BREAKPT IN READ MODE
45	EMULATOR FAILED TO HALT ON BREAKPT IN WRITE MODE
46	EMULATOR FAILED TO HALT ON BREAKPT IN IAQ MODE
47	EMULATOR HALTED TOO SOON
48	EMULATOR HALTED AT WRONG PLACE IN IAQ MODE
49	EMULATOR HALTED AT WRONG PLACE IN WRITE MODE
4A	EMULATOR HALTED AT WRONG PLACE IN READ MODE
4B	CPU TEST FAILED TO COMPLETE
4C	FOLLOWING CPU TEST ERROR NOT DEFINED
4D	FAILURE EXECUTING XXXX
4E	EMULATOR & TRACE MODULE FAILED TO HALT ON BREAKPT
4F	EMULATOR FAILED TO HALT ON BREAKPT; TRACE MODULE HALTED
50	EMULATOR HALTED; TRACE MODULE FAILED TO HALT ON BREAKPT
51	BAD TM DATA: EXPTD D0-D3/DATA=XX/ XXXX, ACTL D0-D3/DATA=XX/ XXXX
52	BAD EMU ADDR TRACED: EXPTD ADDR=XXXX, ACTL ADDR=XXXX
53	TARGET CRU ERR: EXPTD DATA=XXXX, ACTL DATA= XXXX
54	INTERRUPT # XX FAILED, ACTL INTERRUPT=XX
55	9980 GENERATED INTERRUPT FOR NOP, EXPTD PC=0202, ACTL PC=XXXX
56	TARGET RESET FAILED
57	INTREQ FAILED TO TOGGLE: EXPTD TRC=X, ACTL TRC=X (TRC BITS=INTREQ/IAQ/DBIN/X)
58	TEST PROGRAM FAILED TO EXECUTE

Table 43-3. EMU900 Error Messages (Continued)

Error Number	Message
59	TARGET HOLD STUCK HIGH
5A	TARGET HOLD STUCK LOW
5B	DATA WRITTEN TO WRONG ADDRESS: ADDR=XXXX, EXPTD DATA=X, ACTL DATA=X
5C	PATRN ERR: ADDR=XXXX, EXPTD PAT=X, ACTL PAT=X
5D	WRITE PROTECT VIOLATION FLAG STUCK HIGH, SHOULD BE LOW
5E	WRITE PROTECT VIOLATION FLAG FAILED TO CLEAR @ADDR=XXXX
5F	WRITE PROTECT VIOLATION FLAG LOW, SHOULD BE HIGH
60	WRITE PROTECT VIOLATION REG. ERR: EXPTD ADDR=XXXX ACTL=XXXX
61	DATA WRITTEN TO PROTECTED AREA: @ADDR= XXXX, EXPTD DATA=0000, ACTL=XXXX

SECTION XLIV

TMS 9940 EMULATOR AND BUFFER TEST (EMU940)

44.1 INTRODUCTION

The EMU940 tests the emulator, buffer, and the interface between the emulator and trace module. The emulator and trace module may occupy any 990 CRU slot. The emulator may be connected to any unshared interrupt level. The EMU940 consists of nine parts plus a target system test as described in paragraphs 44.1.1 through 44.1.10.

44.1.1 PART 1 – ECHO AND INTERRUPT TESTS. Part 1 tests the emulator interrupt and the host computer-to-emulator interface by halting the emulator and setting the Host Ready line to high causing an emulator interrupt. When the Host Ready line is high the emulator echoes all data sent by the host on the CRU data lines (bits 0-15). The host first sends FFFF and checks the echoed data. Next, a zero bit is shifted through the ones and checked, completing part 1 of EMU940.

44.1.2 PART 2 – IE AND SM COMMAND TESTS. Part 2 tests the INITIALIZE EMULATOR and SELECT MEMORY Commands.

The IE command initializes the emulator by setting the memory bounds for 2K bytes of main memory and 128 bytes of RAM. It also clears all breakpoints and sets up the emulator to trace all of the memory. The SM command can read or modify the memory configuration.

In testing these commands, an IE is executed, then the SM command is executed, reading the memory selected by IE. If the returned data is incorrect one of the commands failed, but each must be checked by the user to determine which failed.

44.1.3 PART 3 – MEMORY TESTS. Part 3 tests the emulator memory. Since the emulator can contain as much as 8K bytes of memory, Part 3 recognizes the size of the emulator memory and prints a message informing the user how much memory is being tested. No control memory is tested. Part 3 consists of 5 tests as described in the following 5 paragraphs.

The first test checks the memory address lines by writing the address of each location to itself in the main memory and the workspace memory. In the main memory and workspace memory flag words, an increment-by-3 data pattern is written. These data patterns are then read and checked. Next, address data is shifted in main memory to toggle the most significant bits. This identifies bad address lines to the level of a single chip. The amount of shift used is determined by the size of memory.

The second test writes and reads 0000 to all memory locations.

The third test writes and reads 5555 to all memory locations.

The fourth test writes and reads AAAA to all memory locations.

The fifth test writes and reads FFFF to all memory locations.

44.1.4 PART 4 – CPU REGISTER TESTS. Part 4 tests the Emulator CPU Registers and the Read Register and Write Register commands. The test writes the register's select value to each register (WP=0 PC=2 ST=4). It then reads each register and checks the data.

44.1.5 PART 5 – CRU TESTS. Part 5 tests the 9940 Emulator CRU functions. It also tests the Write and Read CRU instructions. Part 5 consists of 8 tests.

The first test checks the CRU Flag Bits by shifting a 0 and a 1 thru the Flag Bits.

The second test checks Direction Bits 0-15. The test writes 0000 to P0-P15 and then, sets one Direction Bit at a time to point-out, (causing the data to loop back) and checks the data. Next, one Direction Bit at a time is set to point-in (no loop back).

The third test checks Direction Bits 16-31 in the same manner part 2 tests bits 0-15.

The fourth test checks P0-P15 by shifting a 0 and a 1 thru the bits. All Direction Bits point out.

The fifth test checks P16-P31 by shifting a 0 and a 1 thru the bits.

The sixth test checks the decrementer and P/C-bit by loading shifting 0 and 1 data patterns and reading them. Dynamic testing of the decrementer is done in a later test.

The seventh test checks the MPSI-bits and CB1. The test-bit and CB1 are set to cause loopback of MPSI data. Then 0 and 1 are shifted thru the bits.

The eighth test checks CB0, INT1, INT2, CB1, CB2, and CB3. The test-bit and CB0-CB3 are set high. P0-P10 are then read and the data checked. These lines are now an external CRU interface due to CB0 being set high. 1E7 is the expected read-data. Next, INT1 and INT2 are checked. The test-bit allows P30 and P31 to modify INT1 and INT2 so they can be checked. P11-P16 are next read-checked. The read-data should be 3F since CB1-CB3 are set high.

44.1.6 PART 6 – TRACE AND BREAKPOINT TESTS. Part 6 tests the run, trace and breakpoint function of the emulator. Part 6 consists of 13 tests.

The first test checks the Run Emulation command and the emulator idle detection logic. The emulator is setup to start execution on an Idle instruction. The Run Emulation command is then executed. If Idle is not detected, either Idle or the Run Emulation command failed.

The second test checks MA Trace with two trace ranges. The emulator traces addresses 0100-0102 and 0108-010A only. A breakpoint occurs after 255 addresses have been traced (trace buffer full).

The third test checks EX Trace and IAQ Breakpoint. Only the instruction at address 0102 is in the trace range, but all addresses accessed by the instruction should be traced. The BP should occur on the instruction at address 0104.

The fourth test checks MA Breakpoint and checks IAQ BP to make sure a BP does not occur on an address if it is not an IAQ.

The fifth test checks BP on MA Write by first addressing the BP address during a read and then during a write. No BP should have occurred on the read access.

The sixth test checks BP on MA Read by first addressing the BP address on a write cycle.

The seventh test checks for an Illegal Address BP at a nonexisting memory location.

The eighth test checks for an Illegal Address BP at a legal address that is not selected by the MS command.

The ninth test checks to see that MA BP stops even if the MA is an IAQ.

The tenth test allows the Emulator to trace more than 255 samples. It should not breakpoint after 255 trace samples.

The eleventh test does a Trace Memory test. This test traces all 0's by tracing writes to address 0000.

The twelfth test checks Trace Memory by tracing FE00FF. The test-bit is set to put address lines 8-15 into the flag bits of the trace memory and trace address 00FF only. Bit 7 is set to zero by the control program, causing FE to be read from the flag bits.

The thirteenth test traces high addresses only to put as many 1's as possible in the trace memory bits that are used to trace Address Bits 0-7.

44.1.7 PART 7 – DECREMENTER AND INTERRUPT TESTS. Part 7 tests the 9940 Interrupts and the Decrementer. Part 7 consists of four tests as described by the following four paragraphs.

The first test causes a level 3 interrupt and checks to see that the right interrupt vector was taken.

The second test checks the level 1 interrupt in the same manner as the first test.

The third test puts the decrementer in the timer mode and loads a count into the decrementer. The test then determines if a decrementer interrupt occurred. This test also checks, when the decrementer is in the Idle and CB3=1 modes, that the decrementer interrupt does not occur.

The fourth test puts the decrementer in the count mode. By outputting positive transitions on P17 the decrementer is made to count down. Enough transitions are issued to count the decrementer down to zero. Checks are then made to insure that a decrementer interrupt occurred and that the decrementer reloaded with the proper value.

44.1.8 PART 8 – 9940 CPU TESTS. Part 8 tests control loading and executing of the standard 990/4 diagnostic (AU04) to test the emulator CPU functions. The AU04 is divided into two modules so that it will fit into 2K bytes of memory. After the two AU04 Modules are executed, a third module which tests the additional instructions found in the 9940 is executed. This third module also executes a copy of the ROM Self Test. The third module executes for about 90 seconds. If an error is detected during the CPU tests an error message number will be printed. The following list of CPU error messages describes the error associated with each number.

Error Message Number	Instruction That Failed
0000	C
0001	CB
0002	MOV
0003	MOVB

Error Message Number	Instruction That Failed
0004	A
0005	AB
0006	S
0007	SB
0008	SOC
0009	SOCB
000A	SZC
000B	SZCB
000C	JMP
000D	JLT
000E	JLE
000F	JEQ
0010	JHE
0011	JGT
0012	JNE
0013	JL
0014	JH
0015	JOP
0016	JNC
0017	JOC
0018	JNO
0019	COC
001A	CZC
001B	XOR
1000	SELF TEST FAILED
1200	DCA
1400	DCS
2001	SRA
2002	SLA
2003	SRL
2004	SRC
2005	LI
2006	AI
2007	ANDI

Error Message Number	Instruction That Failed
2008	ORI
2009	CI
200A	STWP
200B	LWPI
200C	CLR
200D	NEG
200E	INV
200F	INC
2010	INCT
2011	DEC
2012	DECT
2013	SETO
2014	SWPB
2015	ABS
2016	B
2017	BL
2018	BLWP
2019	X
201A	RT
201B	DIV
201C	MPY
201D	LIIM

44.1.9 PART 9 – TRACE MODULE INTERFACE TESTS. Part 9 tests the Emulator-Trace Module interface. The Emulator and Trace Module interface cables must be connected before starting part 9. The control cable must be disconnected to run all other tests. Part 9 consists of three tests as described in the following three paragraphs.

Test 1 sets-up the emulator and trace module to trace a program to be executed in the emulator. The emulator's trace memory is controlled by the TSE signal from the trace module. A BP address is set in the emulator, but BP is not enabled. The trace module is set to count one external event. When the BP address is detected, the Event signal is sent to the trace module. The trace module then sends the Halt Request to the emulator. The emulator halts and sets control mode– to the trace module low, halting the trace module. The trace data in the emulator and trace module is then checked to verify proper operation of the interface.

Test 2 traces the same program as used in test 1 except the Trace Module mask register allows all qualifiers to condition tracing. The qualifiers allow only reads to be traced.

Test 3 is similar to tests 1 and 2 except only odd bytes are traced.

44.1.10 TARGET SYSTEM TEST. The target test is performed by the TS verb. This test checks the emulator-to-target system interface and consists of six subtests as described in the following six paragraphs.

Test 1 checks P0-P31 by loading a 32 bit shift register in the Target System through the P0-P31 I/O lines. Bit 0 is a zero, all other bits are ones. The zero is then shifted through all bits and read in each bit position through the P0-P31 I/O lines. If an error is detected the bit sequence from left to right of the displayed data is, P15-P0 then P31-P16.

Test 2 checks the CRU expansion function. All 256 CRU addresses are sent to the target system. Each address is latched up in the target system and checked. A flip-flop also latches the CRU write bit so it can be checked.

Test 3 checks the MPSI interface. The target system connects the CRUOUT line to the TD line and the CRUCLOCK to TC. To check the MPSI read function 0000, 5555, AAAA and FFFF are sent over the CRU lines to be read by the MPSI interface. The MPSI write function is tested by performing a write to the MPSI CRU address. The data is latched in the target system and checked by reading P16-P31.

Test 4 checks the clock output. The target system has a clock-counting latch. The counter is turned on and a 16 bit STCR instruction is executed. The counter is then turned off. The counter should have counted 36_{16} clocks.

Test 5 checks the Hold, Hold Acknowledged and Idle signals. The emulator is placed in Hold and then in Idle. These states are then read over the host-target system interface and checked.

Test 6 checks the interrupts in a manner similar to E7, except that the host-target system-interface controls the interrupts and the EC line.

44.2 OPERATOR INTERFACE

EMU940 runs under the control of DOCS on a 990 computer. Refer to Section III of this manual for a detailed description of DOCS.

44.2.1 AVAILABLE VERBS. In addition to the verbs supplied by the DOCS package, EMU940 provides the verbs listed in table 44-1.

Table 44-1. EMU940 Verbs

Verb	Function
E1	EXECUTE PART 1 – ECHO AND INTERRUPT TESTS
E2	EXECUTE PART 2 – IE AND SM COMMAND TESTS
E3	EXECUTE PART 3 – MEMORY TESTS
E4	EXECUTE PART 4 – CPU REGISTER TESTS
E5	EXECUTE PART 5 – CRU TESTS
E6	EXECUTE PART 6 – TRACE AND BREAKPOINT TESTS

Table 44-1. EMU940 Verbs (Continued)

Verb	Function
E7	EXECUTE PART 7 – DECREMENTER AND INTERRUPT TESTS
E8	EXECUTE PART 8 – 9940 CPU TESTS
E9	EXECUTE PART 9 – TRACE MODULE INTERFACE TESTS
EA	EXECUTE PARTS 1-7
L1	LOOP ON PART 1
L2	LOOP ON PART 2
L3	LOOP ON PART 3
L4	LOOP ON PART 4
L5	LOOP ON PART 5
L6	LOOP ON PART 6
L7	LOOP ON PART 7
L8	LOOP ON PART 8
L9	LOOP ON PART 9
LA	LOOP ON PARTS 1-7
TS	EXECUTE TARGET SYSTEM TEST
IT	INITIALIZE TESTS
GC	GENERATE COMMAND TABLE
XT	EXECUTE COMMAND TABLE
LT	LOOP ON COMMAND TABLE
SL	SHOW COMMAND TABLE
HT	HALT EMULATOR
SS	SINGLE STEP CLOCK
LP	LOOP CONTROL
DS	DISPLAY STATUS
SO	SET CRU BIT TO ONE
SZ	SET CRU BIT TO ZERO
SC	SELECT INTERNAL OR EXTERNAL CLOCK
SB	SET BREAKPOINT
RB	REMOVE BREAKPOINT

Table 44-1. EMU940 Verbs (Continued)

Verb	Function
PE	PRINT ERROR COUNT
IE	INITIALIZE EMULATOR
SM	SELECT MEMORY
CK	READ CLOCK STATUS
RR	READ CPU REGISTER
WR	WRITE TO CPU REGISTER
RM	READ EMULATOR MEMORY
WM	WRITE TO EMULATOR MEMORY
RC	READ EMULATOR CRU
WC	WRITE TO EMULATOR CRU
DT	DEFINE TRACE
DB	DEFINE BREAKPOINT
RT	READ EMULATOR TRACE DATA
RE	RUN EMULATION
CM	CONTROL MEMORY TEST COMMAND

44.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions asked by DOCS, EMU940 asks the questions listed in table 44-2.

Table 44-2. EMU940 Initialization Questions

ENTER TRACE MODULE CRU BASE ADDRESS. DEFAULT = 0100-

ENTER THE EMULATOR CRU BASE ADDRESS. DEFAULT = 0140-

ENTER THE TARGET SYSTEM CRU BASE ADDRESS. DEFAULT = 0020

****WARNING****

Emulator and I/O device interrupt levels must not be the same.
Enter the emulator int. level. Default = 0006.

44.2.3 **PROGRAMMER-PANEL INITIALIZATION.** The programmer-panel version of EMU940 asks:

IDLE ON ERRORS?

followed by the questions in table 44-2.

44.3 SYSTEM REQUIREMENTS

The EMU940 will run on a 990 Computer with at least 48K bytes of memory, an appropriate loading device, an appropriate interactive device, a TMS9940 Emulator Board Part Number 941570-0001, a TMS9940 Buffer Module Assembly Part Number 949971-0001, and optionally a Trace Module Part Number 949910-0001.

44.4 ERROR MESSAGES

EMU940 error messages are listed in table 44-3.

Table 44-3. EMU940 Error Messages

Error Number	Message
1	HALT ACKNOWLEDGED FAILED TO RETURN AFTER HALT WAS SENT TO THE EMULATOR
2	EMULATOR READY FAILED TO RETURN AFTER HALT WAS SENT TO THE EMULATOR
3	UNEXPECTED EMULATOR INTERRUPT
4	EMULATOR RETURNED AN ILLEGAL ERROR CODE ILLEGAL CODE XXXX
5	EMULATOR READY FAILED TO RESET AFTER HOST READY WAS SET
6	EMULATOR READY FAILED TO SET AFTER HOST READY WAS RESET
7	EMULATOR RETURNED AN INDEX OUT OF BOUNDS ERROR MESSAGE
8	EMULATOR RETURNED AN ILLEGAL COMMAND ERROR MESSAGE
9	EMULATOR RETURNED A SEMANTICS ERROR MESSAGE
10	ECHO COMPARE ERROR. EXPECTED=XXXX ACTUAL=XXXX
11	IE OR SM COMMAND FAILED. EXPECTED SM DATA=XXXX ACTUAL SM DATA=XXXX
12	MEMORY DATA ERROR ADDRESS = XXXX DATA = XXXX EXPECTED = XXXX
13	REGISTER DATA ERROR DATA = XXXX EXPECTED = XXXX REG = XXXX (0=WP 2=PC 4=ST)
14	RUN EMULATION OR EMULATOR IDLE FAILED
15	BREAKPOINT FAILED TO OCCUR.
16	TRACE COUNT ERROR. EXPECTED = XX ACTUAL = XX
17	TRACE DATA ERROR. EXPECTED = XXXX ACTUAL = XXXX WORD # XX

Table 44-3. EMU940 Error Messages (Continued)

Error Number	Message
18	PC ERROR. EXPECTED VALUE = XXXX ACTUAL VALUE = XXXX
19	CRU DATA ERROR. EXPECTED = XXXX ACTUAL = XXXX BASE ADDRESS = XXXX
20	EXPECTED BUFFER TYPE = FFFF. ACTUAL BUFFER TYPE = XXXX
21	DECREMENTER INTERRUPT DID NOT CLEAR
22	DECREMENTER REGISTER AND T/-C DID NOT CLEAR
23	TEST BIT FAILED TO CLEAR
24	STATUS REGISTER ERROR. EXPECTED = XXXX ACTUAL = XXXX
25	DECREMENTER FAILED TO INTERRUPT AFTER 2000 COUNTS
26	BREAKPOINT OCCURRED. IT SHOULD NOT HAVE.
27	DECREMENTER CAUSED AN INTERRUPT ON THE WRONG COUNT.
	COUNTS REMAINING AT TIME OF INTERRUPT = XXXX
28	DECREMENTER FAILED TO RELOAD WITH PROPER VALUE. EXPECTED = 2000 ACTUAL = XXXX
29	CPU TEST ERROR. MESSAGE # XXXX
30	CPU TEST TIMED OUT. PASS OR FAIL NOT REPORTED.
31	INTERRUPT PENDING DID NOT CLEAR
32	INTERRUPT DID NOT OCCUR WHEN HOST READY WAS TOGGLED
33	INTERRUPT PENDING DID NOT SET WHEN HOST READY WAS TOGGLED
34	DATA ERROR. EXPECTED = XXXXXXXX ACTUAL = XXXXXXXX
35	EXPANSION ADDRESS ERROR. EXPECTED ADDRESS = XX ACTUAL ADDRESS = XX
36	CRU BIT WAS A ZERO. IT SHOULD HAVE BEEN A ONE.
37	CRU BIT WAS A ONE. IT SHOULD HAVE BEEN A ZERO.

Table 44-3. EMU940 Error Messages (Continued)

Error Number	Message
38	MPSI READ ERROR. EXPECTED = XXXX ACTUAL = XXXX
39	T.M. CABLE STATUS (BIT 21) INDICATES DATA CABLE NOT PRESENT.
40	T.M. DATA SOURCE STATUS (BIT 23) FAILS. INDICATES EMULATOR IS NOT DATA SOURCE.
41	T.M. HALT PENDING STATUS (BIT 31) FAILED. IT SHOULD HAVE BEEN A ONE.
42	T.M. TRACING STATUS (BIT 20) FAILED. IT SHOULD HAVE BEEN A ZERO.
43	TRACE MODULE DATA ERROR ADDRESS = XX EXPECTED = XXXXX ACTUAL = XXXXX
44	MPSI WRITE ERROR. EXPECTED = XXXX ACTUAL = XXXX
45	CLOCK COUNT ERROR. EXPECTED = XXXX ACTUAL = XXXX
46	TARGET SYSTEM HOLD ACKNOWLEDGED FAILED.
47	TARGET SYSTEM IDLE FAILED.
48	TARGET SYSTEM RESET FAILED.
49	TARGET SYSTEM INT1 FAILED.
50	TARGET SYSTEM INT2 FAILED.
51	TARGET SYSTEM EC INPUT FAILED.
52	CONTROL ROM CHECKSUM ERROR.
53	CONTROL RAM DATA ERROR. ADDRESS = XXXX

SECTION XLV

EXTERNAL AUTOCALL UNIT INTERFACE TEST (EXTACU)

45.1 INTRODUCTION

The EXTACU is used to perform fault isolation on the External Autocall Unit Interface Module (EACUIF), Part Number 2263480-0001. It also has a special verb to test the function of an external autocall unit (when connected to the EACUIF) by dialing a user entered number. Once a fault is identified it must be corrected before running other tests. EXTACU consists of six parts as described in paragraphs 45.1.1 through 45.1.6.

45.1.1 PART 1 – CRU BIT TEST. Part 1 is used to check the latch and reset capabilities of CRU output bits 1, 2, 3, 4, 5, 8, 9, A, & D and CRU input bits 1, 2, 3, 6, 7, 7, and E.

45.1.2 PART 2 – STATE SEQUENCER TEST. Part 2 tests the EACUIF state sequencer by taking control of the internal clock, putting the state sequencer in the test mode, stepping through the sequences, and checking for the proper outputs at each step.

45.1.3 PART 3 – PND, DSS, AND ACR INTERRUPT TEST. Part 3, the interrupt test, will be executed if the interrupts available flag was set during execution of the IT or IS verb. Part 3 checks for the proper occurrence of the PND, DDS, and ACR interrupts. Part 3 also checks to determine that interrupts do not occur without the proper enabling signals and that each interrupt can be reset by the proper reset signal.

45.1.4 PART 4 – INTERNAL CLOCK TEST. Part 4 exercises the state sequencer using the EACUIF internal clock.

45.1.5 PART 5 – EIA DRIVERS AND PWI INTERRUPT TEST. Part 5 tests the EACUIF line drivers and receivers and the PWI interrupt by using a loopback connector. The test will not run until the operator indicates that the loopback connector is installed. The PWI interrupt test will run only if the interrupt available flag was set during execution of the IT or IS verb.

45.1.6 PART 6 – DIAL NUMBER TEST. The dial number test uses the DN verb to exercise the EACUIF board connected to an autocall unit (an external autocall unit is required to run this test). The test will dial a telephone number input by the operator. This test is a functional test only and printouts (if any) are not intended to be used for fault isolation. Up to 22 characters may be entered but the last one must be an "X" to complete the entry. Example input:

258-3443X, dials the phone number 258-3443.

45.2 OPERATOR INTERFACE

EXTACU runs under the control of DOCS on a 990 computer. Refer to section III of this manual for a detailed description of DOCS.

45.2.1 AVAILABLE VERBS. In addition to the verbs supplied by the DOCS package, EXTACU provides the verbs listed in table 45-1.

Table 45-1. EXTACU Verbs

Verb	Function
IT	Initialize Test
E1	Execute Part 1
E2	Execute Part 2
E3	Execute Part 3
E4	Execute Part 4
E5	Execute Part 5
L1	Loop On Part 1
L2	Loop On Part 2
L3	Loop On Part 3
L4	Loop On Part 4
L5	Loop On Part 5
EA	Execute Parts 1-5
LA	Loop On Parts 1-4

NOTE

Part 5 is not included in the LA verb because it requires that a special connecter be installed that will cause the other parts to fail.

DN	Executes Part 6
PE	Prints the current error count

45.2.2 **STANDARD INITIALIZATION.** In addition to the questions required by DOCS, EXTACU asks the questions listed in table 45-2.

Table 45-2. EXTACU Initialization Questions

ENTER THE EXTACU CRU BASE ADDRESS D = 0080 -

ENTER THE EXTACU INTERRUPT LEVEL D = 0C -

ARE INTERRUPTS AVAILABLE? D = 1 -

45.2.3 **PROGRAMMER-PANEL INITIALIZATION.** The programmer-panel version of EXTACU asks:

IDLE ON ERRORS?

followed by the questions in table 45-2.

45.3 SYSTEM REQUIREMENTS

The EXTACU will run on a 990 computer with 16K bytes of memory, appropriate loading and interactive devices, an External Autocall Unit Interface Kit Part Number 2263907-0001, and an external autocall unit (optional).

45.4 ERROR MESSAGES

EXTACU error messages are listed in table 45-3.

Table 45-3. EXTACU Error Messages

Error Number	Message
	PART 1
11	1-1 FAILED TO GENERATE A CLEAR WHEN COMP-CLR WAS SET HIGH. CHECK THE CLEAR (COMPCLR) CKTRY. EXPECTED CRU = 0000 RECEIVED CRU = XXXX
12	1-2 FAILED TO GENERATE A CLOCK PULSE WHEN COMPCLK SET HIGH. CHECK THE EXTERNAL CLOCK (COMPCLK) CKTRY. EXPECTED CRU = 1000 RECEIVED CRU = XXXX
13	1-3 CLOCK FAILED TO RESET WHEN COMPCLK WAS CLEARED. CHECK THE EXTERNAL CLOCK (COMPCLK) CKTRY. EXPECTED CRU = 0000 RECEIVED CRU = XXXX
14	1-4 FAILED TO GO INTO THE TEST MODE WHEN THE TEST BIT WAS SET. CHECK THE TEST (TEST) CKTRY. EXPECTED CRU = 4000 RECEIVED CRU = XXXX
15	1-5 SYSTEM STAYED IN THE TEST MODE WHEN THE TEST BIT WAS RESET. CHECK THE TEST (TEST) CKTRY. EXPECTED CRU = 0000 RECEIVED CRU = XXXX

Table 45-3. EXTACU Error Messages (Continued)

Error Number	Message
16	1-6 FAILED TO GENERATE DIGIT PRESENT (INPUT BIT 6) WHEN DPR BIT WAS SET. CHECK THE DIGIT PRESENT (DPR) CKTRY. EXPECTED CRU = 0040 RECEIVED CRU = XXXX
17	1-7 FAILED TO CLEAR DIGIT PRESENT (INPUT BIT 6) WHEN DPR BIT WAS RESET. CHECK THE DIGIT PRESENT (DPR) CKTRY. EXPECTED CRU = 0000 RECEIVED CRU = XXXX
18	1-8 FAILED TO GENERATE CALL REQUEST (INPUT BIT 7) WHEN CRQ BIT SET. CHECK THE CALL REQUEST (CRQ) CKTRY. EXPECTED CRU = 0080 RECEIVED CRU = XXXX
19	1-9 FAILED TO CLEAR CALL REQUEST (INPUT BIT 7) WHEN CRQ BIT RESET. CHECK THE CALL REQUEST (CRQ) CKTRY. EXPECTED CRU = 0000 RECEIVED CRU = XXXX
1A	1-A FAILED TO SET CRU INPUT BIT X (IN THE TEST MODE) WHEN OUTPUT BIT X WAS SET. CHECK THE TEST CKTRY AND APPROPRIATE SIGNALS: ANB2 (BIT 1), ANB4 (BIT 2), ANB8 (BIT 3) EXPECTED CRU = XXXX RECEIVED CRU = XXXX
1B	1-B FAILED TO RESET CRU INPUT BIT X (IN THE TEST MODE) WHEN OUTPUT BIT X RESET. CHECK THE TEST CKTRY AND APPROPRIATE SIGNALS: ANB2 (BIT 1), ANB4 (BIT2), ANB8 (BIT 3) EXPECTED CRU = XXXX RECEIVED CRU = XXXX

Table 45-3. EXTACU Error Messages (Continued)

Error Number	Message
1C	1-C FAILURE IN CRU COMMUNICATIONS. CHECK CRU CKTRY. EXPECTED CRU = XXXX RECEIVED CRU = XXXX
	PART 2
21	2-1 COUNTER ENABLE WAS NOT SET AT THE PROPER TIME. FAILURE IN THE COUNTER ENABLE (CCTEN) CKTRY. SEQUENCE COUNTER IS IN STATE X EXPT STATE IS X.
22	2-2 COUNTER ENABLE WAS SET AT THE WRONG TIME. FAILURE IN THE COUNTER ENABLE (CCTEN) CKTRY. SEQUENCE COUNTER IS IN STATE X EXPT STATE IS X.
23	2-3 STATE SEQUENCER FAILED TO STEP TO EXPT COUNT. CHECK COUNTER FOR PROPER OPERATION. SEQUENCE COUNTER IS AT STATE X EXPT STATE IS X.
24	2-4 PNDSTAT WAS NOT SET AT THE PROPER TIME. CHECK THE PNDSTAT CKTRY. SEQUENCE COUNTER IS IN STATE X EXPT STATE IS X.
25	2-5 PNDSTAT WAS SET AT WRONG TIME. CHECK THE PNDSTAT CKTRY. SEQUENCE COUNTER IS AT STATE X EXPT STATE IS X.
26	2-6 FAILED TO RESET DIGIT PRESENT AT STATE 6 OF THE STATE SEQUENCER. CHECK DIGIT (DPR & BDPR) CKTRY.

Table 45-3. EXTACU Error Messages (Continued)

Error Number	Message
	PART 3
31	3-1 FAILED TO RECEIVE PND INT WHEN EXPECTED. CHECK PND INTERRUPT CKTRY.
32	3-2 FAILED TO RESET PND INT WITH SIGNAL X SIGNALS: 0 = DPR SET 1 = PNDINT RESET 2 = I/O RESET GENERATED 3 = COMPCLR SET 4 = CALL REQUEST LOW 5 = INT ON/OFF OFF
33	3-3 FAILED TO RECEIVE A DSS INT WHEN EXPECTED. CHECK DSS INTERRUPT CKTRY.
34	3-4 FAILED TO RECEIVE A DSS INT WHEN EXPECTED. CHECK ACR INTERRUPT CKTRY.
35	3-5 FAILED TO RESET DSS INT WITH SIGNAL X SIGNALS: 0 = DPR SET 1 = PNDINT RESET 2 = I/O RESET GENERATED 3 = COMPCLR SET 4 = CALL REQUEST LOW 5 = INT ON/OFF OFF
36	3-6 FAILED TO RESET ACR INT WITH SIGNAL X SIGNALS: 0 = DPR SET 1 = PNDINT RESET 2 = I/O RESET GENERATED 3 = COMPCLR SET 4 = CALL REQUEST LOW 5 = INT ON/OFF OFF

Table 45-3. EXTACU Error Messages (Continued)

Error Number	Message
37	3-7 GENERATED A PND INT WITH PND INT ENABLE LOW.
	PART 4
41	4-1 SEQUENCER FAILED TO COUNT ON INTERNAL CLOCK.
	CHECK THE EXTACU INTERNAL CLOCK CKTRY (CLOCKA).
	PART 5
51	5-1 FAILED LOOPBACK ON SIGNAL NBX.
	FAILURE IN LINE DRIVER/RECEIVER CKTRY.
52	5-2 FAILED TO LOOPBACK ON SIGNAL DPR.
	FAILURE IN LINE DRIVER/RECEIVER CKTRY.
53	5-3 FAILED TO LOOPBACK ON SIGNAL CRQ.
	FAILURE IN LINE DRIVER/RECEIVER CKTRY.
54	5-4 FAILED TO RECEIVE A PWI INTERRUPT WHEN EXPECTED. CHECK PWI INTERRUPT CKTRY.
55	5-5 FAILED TO RESET PWI INT WITH SIGNAL X SIGNALS:
	0 = DPR SET
	1 = PNDINT RESET
	2 = I/O RESET GENERATED
	3 = COMPCLR SET
	4 = CALL REQUEST LOW
	5 = INT ON/OFF OFF
56	5-6 GENERATED A PWI INT WITH SIGNAL PWI INT DISABLE SET.
	CHECK THE PWI INT CKTRY.

SECTION XLVI

TTY/EIA REMOTE TERMINAL TEST (RMTEIA)

46.1 INTRODUCTION

The RMTEIA tests the proper operation of the special cable (with the Ring indicator wired to RCR). RMTEIA also verifies the whole communications system (EIA card, cables, data sets, and remote terminal) by performing a keyboard printer interaction test. There are two special loopback tests to verify line and data set integrity. RMTEIA is composed of four parts which are described in paragraphs 46.1.1 through 46.1.6.

46.1.1 PART 1 – KEYBOARD/PRINTER INTERACTIVE TEST. Part 1 checks the integrity of the complete loop and is executed on the host computer. The test checks for RING in a timing loop printing a W every 30 seconds until RING is received. After receiving RING the test presents DTR and waits 15 seconds for the modem to respond with DSR and another 30 seconds after that for DCD. Once DCD is received, instructions are sent to the operator at the remote terminal to type any desired character pattern and verify that the character pattern is printed at the terminal. When the carriage return is depressed, the test is terminated and the total input is echoed back to the printer on the interactive terminal. The test monitors for DCD and DSR during execution and for errors caused by loss of DCD or DSR.

46.1.2 PART 2 – RIPPLE DUMP TEST. The ripple dump test is provided to allow adjustment of the print contrast control on the 743/745 KSR Data Terminal. Execution of this test on the host computer will cause the computer to go into a wait loop looking for DCD and printing a W every 30 seconds. Once DCD is received the ripple dump pattern is transmitted to the remote terminal. As in Part 1, the host computer monitors for DCD and DSD and prints an error if they are lost. The operator may allow the test to run to completion (about 5 minutes on a 743/745 KSR, 90 seconds on an 820 KSR at 1200 Baud), or may terminate the test by removing DCD (lifting the telephone receiver).

46.1.3 PART 3 – ANALOG LOOPBACK TEST. Part 3 is used to verify the transmission link to and from the HOST modem. Analog loopback test is executed with the AL switch on the HOST modem depressed. A ripple dump of characters 20_{16} through 7F are transmitted, looped back to the computer, and compared for correctness to verify the loop. The test will print an error if a timing error is received or if DCD is lost. The test will also print an error if the transmitted and received data do not match. A loop option will put this test into an infinite scope loop.

46.1.4 PART 4 – DIGITAL LOOPBACK TEST. Part 4 is identical to the Part 3 test except it is run on the remote modem. Instead of depressing the AL switch on the host modem the DL switch is depressed on the remote modem. This test will verify the integrity of the telephone lines with the rest of the loop. If the DL test passes and the selftest on the remote modem passes but the part 1 test fails, the problem is most likely in the remote terminal.

46.2 OPERATOR INTERFACE

RMTEIA runs under the control of DOCS on a 990 Computer. Refer to section III of this manual for a description of DOCS.

46.2.1 **AVAILABLE VERBS.** In addition to the verbs supplied by the DOCS package, RMTEIA provides the verbs listed in table 46-1.

Table 46-1. RMTEIA Verbs

Verb	Function
IT	INITIALIZE TEST
ET	EXECUTE TEST (PART 1 OR PART 2)
LT	LOOP ON TEST
EA	EXECUTE PARTS 1 AND 2
LA	LOOP ON PARTS 1 AND 2
AL	ANALOG LOOPBACK TEST (PART 3)
DL	DIGITAL LOOPBACK TEST (PART 4)

46.2.2 **STANDARD INITIALIZATION.** In addition to the initialization questions required by DOCS, RMTEIA asks the questions listed in table 46-2.

Table 46-2. RMTEIA Initialization Questions

Question	Comment
CRU ADDRESS DEFAULT=00	Enter correct CRU address
INTERRUPT LEVEL DEFAULT=06	Enter correct interrupt level

46.2.3 **PROGRAMMER-PANEL INITIALIZATION.** The programmer-panel version of RMTEIA asks:

IDLE ON ERRORS?

followed by the questions in table 46-2.

46.3 SYSTEM REQUIREMENTS

The RMTEIA will run on a 990 computer with 16K bytes of memory, appropriate loading and interactive devices, a Remote TTY/EIA Kit, Part Number 2265150-0001 (300 Baud), and a Remote TTY/EIA Kit, Part Number 2256150-0002 (1200 Baud).

46.4 ERROR MESSAGES

RMTEIA error messages are listed in table 46-3.

Table 46-3. RMTEIA Error Messages

Error Number	Message
01	DSR DIDN'T OCCUR WITHIN 15 SECONDS AFTER RING
02	DCD DIDN'T OCCUR WITHIN 30 SECONDS AFTER DCD

Table 46-3. RMTEIA Error Messages (Continued)

Error Number	Message
03	***ERROR*** LOST DATA CARRIER DETECT
04	***ERROR*** LOST DATA SET READY
05	A TIMING ERROR HAS BEEN DETECTED (TIMERR = 1)
06	***ERROR*** FAILED LOOPBACK DATA SENT DATA RECEIVED XX XX

APPENDIX A

MODIFYING 990 OBJECT CODE

APPENDIX A

MODIFYING 990 OBJECT CODE

A.1 INTRODUCTION

The assemblers produce object code that may be linked to other object code modules or programs and loaded into the 990 computer, or may be loaded into the computer directly. Object code consists of records containing up to 71 ASCII characters each. The format, described in the next paragraph, permits correction using a keyboard device. Reassembly to correct errors is unnecessary. An example of output code is shown in figure A-1.

A.1.1 OBJECT CODE FORMAT. The object record consists of a number of tag characters, each followed by one or two fields as defined in table A-1. The first character of a record is the first tag character, which tells the loader which field or pair of fields associated with the preceding tag character. When the assembler has no more data for the record, the assembler writes the tag character 7 followed by the check sum field, and the tag character F, which requires no fields. The assembler then fills the rest of the record with blanks and begins a new record with the appropriate tag character.

Tag character 0 is followed by two fields. The first field contains the number of bytes of relocatable code, and the second field contains the program identifier assigned to the program by an IDT directive. When no IDT directive is entered, the field contains blanks. The loader uses the program identifier to identify the program, and the number of bytes of relocatable code to determine the load bias for the next module or program. SDSMAC, an assembler, and the cross assembler place a single tag character 0 at the beginning of each program. PX9ASM, another assembler, is unable to determine the value for the first field until the entire module has been assembled, so PX9ASM places a tag character 0 followed by a zero field and the program identifier at the beginning of the object code file. At the end of the file, PX9ASM places another tag character zero followed by the number of bytes of relocatable code and eight blanks.

Tag characters 1 and 2 are used with entry addresses. Tag character 1 is used when the entry address is absolute. Tag character 2 is used when the entry address is relocatable. The hexadecimal field contains the entry address. One of these tags may appear at the end of the object code file. The associated field is used by the loader to determine the entry point at which execution starts when the loading is complete.

```
00000SAMPROG 90040C0000A0020BC06DB00029004 2C0020A0024BC81BC002A7F219F
A0028B0241B0000BCB41B0002B0380A00CAC0052C00A2B02E0C0032B0200B0F0F7F1DEF
A00D6BC0A0C00CAB04C3BC160C00CCBC1A0C00D0BC072B0281B3A00A00ECB02217F151F
A00EEB0900B06C1A00EAB1102A00F2B0543B11F8B2C20C0032BC101B0B44BE0447F18EF
A0100BDD66B0003B0282C00A2B11EDB03407F832F
200CE0010C          7FCABF
:
(A)132255
```

Figure A-1. Object Code Example

Table A-1. Object Output Tags Supplied by Assemblers

Tag Character	Hexadecimal Field (Four Characters)	Second Field	Meaning
0	Length of all relocatable code	8-character Program Identifier	Program Start
1	Entry address	None	Absolute Entry Address
2	Entry address	None	Relocatable Entry Address
3	Location of last appearance of symbol	6-character symbol	External Reference last used in relocatable code
4	Location of last appearance of symbol	6-character symbol	External Reference last used in absolute code
5	Location	6-character symbol	Relocatable External Definition
6	Location	6-character symbol	Absolute External Definition
7	Checksum for current record	None	Checksum
9	Load address	None	Absolute load address
A	Load address	None	Relocatable load address
B	Data	None	Absolute data
C	Data	None	Relocatable data
D†	Load Bias	None	Specify Memory Area
F	None	None	End-of-record
G	Location	6-character symbol	Relocatable symbol definition
H	Location	6-character symbol	Absolute symbol definition

† Not output by the assemblers

Tag characters 3 and 4 are used for external references. Tag character 3 is used when the last appearance of the symbol in the second field is in relocatable code. Tag character 4 is used when the last appearance of the symbol is absolute code. The hexadecimal field contains the location of the last appearance. The symbol in the second field is the external reference. Both fields are used by the linking loader to provide the desired linking to the external reference.

For each external reference in a program, there is a tag character in the object code, with a location, or an absolute zero, and the symbol that is referenced. When the object code field contains absolute zero, no location in the program requires the address that corresponds to the reference (an IDT character string, for example). Otherwise, the address corresponding to the reference will be placed in the location specified in the object code by the linking loader. The location specified in the object code similarly contains absolute zero or another location. When it contains absolute zero, no further linking is required. When it contains a location, the address corresponding to the reference will be placed in that address by the linking loader. The location of each appearance of a reference in a program contains either an absolute zero or another location into which the linking loader will place the referenced address.

Figure A-2 illustrates the chain of the external reference EXTR. The object code contains the following tag and fields:

4C00EEXTR

At location C00E, the address C00A points to the preceding appearance of the reference. The chain includes both absolute and relocatable addresses and consists of absolute addresses C00E, C00A, C006, and C002, relocatable addresses 029E, 029A, and 0298, absolute addresses B00E, B00A, B006, and B002, and relocatable addresses 0290 and 028E. Each location points to the preceding appearance, except for location 028E, which contains zero. The zero identifies location 028E as the first appearance of EXTR, the end of the chain.

Tag characters 5 and 6 are used for external definitions. Tag character 5 is used when the location is relocatable. Tag character 6 is used when the location is absolute. Both fields are used by the linking loader to provide the desired linking to the external definition. The second field contains the symbol of the external definition.

Tag character 7 precedes the checksum, which is an error detection word. The checksum is formed as the record is being written. It is the two's complement of the sum of the 8-bit ASCII values of the characters of the record from the first tag of the record through the checksum tag, 7.

Tag characters 9 and A are used with load addresses for data that follows. Tag character 9 is used when the load address is absolute. Tag character A is used when the load address is relocatable. The hexadecimal field contains the address at which the following data word is to be loaded. A load address is required for a data word that is to be placed in memory at some address other than the next address. The load address is used by the loader.

Tag characters B and C are used with data words. Tag character B is used when the data is absolute, such as an instruction word or a word that contains text characters or absolute constants. Tag character C is used for a word that contains a relocatable address. The hexadecimal field contains the data word. The loader places the data word in the memory location specified in the preceding load address field, or in the memory location that follows the preceding data word.

```
0229                        *
0230                        *       DEMONSTRATE EXTERNAL REFERENCE LINKING
0231                        *
0232                              REF  EXTR
0233  028C                        RORG
0234  028C  C820                 MOV  @EXTR,@EXTR
      028E  0000
      0290  028E'
0235  0292  28E0                 XOR  @EXTR,3
      0294  0290'
0236  B000                        AORG >B000
0237  B000  3220                 LDCR @EXTR,8
      B002  0294'
0238  B004  0420                 BLWP @EXTR
      B006  B002
0239  B008  0223                 AI   3,EXTR
      B00A  B006
0240  B00C  38A0                 MPY  @EXTR,2
      B00E  B00A
0241  0296                        RORG
0242  0296  C820                 MOV  @EXTR,@EXTR
      0298  B00E
      029A  0298'
0243  029C  28E0                 XOR  @EXTR,3
      029E  029A'
0244  C000                        AORG >C000
0245  C000  3220                 LDCR @EXTR,8
      C002  029E'
0246  C004  0420                 BLWP @EXTR
      C006  C002
0247  C008  0223                 AI   3,EXTR
      C00A  C006
0248  C00C  38A0                 MPY  @EXTR,2
      C00E  C00A
(A)132256
```

Figure A-2. External Reference Example

Tag character F indicates the end of record. It may be followed by blanks.

Tag characters G and H are used when the symbol table option is specified with SDSMAC or the cross assembler. Tag character G is used when the location or value of the symbol is relocatable, and tag character H is used when the location or value of the symbol is absolute. The first field contains the location or value of the symbol, and the second field contains the symbol to which the location is assigned.

The last record of an object code file has a colon (:) in the first character position of the record, followed by blanks.

A.1.2 MACHINE LANGUAGE FORMAT. Some of the data words preceded by tag character B represent machine instructions. Comparing the source listing with the object code fields identifies the data words that represent machine instructions. Figure A-3 shows the manner in which the bits of the machine instructions relate to the operands in the source statements for each format of machine instructions.

A.1.3 SYMBOL TABLE. When the SYMT option is specified (SDSMAC and cross assembler only), the symbol table is included in the object code file. One entry, using tag character G or H as appropriate, is supplied for each symbol defined in the assembly.

A.1.4 OBJECT CODE LISTING. When the OBJ option is specified (SDSMAC or cross assembler), the assembler prints the object code following the source code listing. When the cross reference listing is also specified, the object listing follows the cross reference listing. The object code shown in figure A-1 is shown in the object code listing format in figure A-4. Notice that blanks have been inserted for clarity, and a sequence number included at the right.

A.2 PROCEDURES FOR CHANGING OBJECT CODE

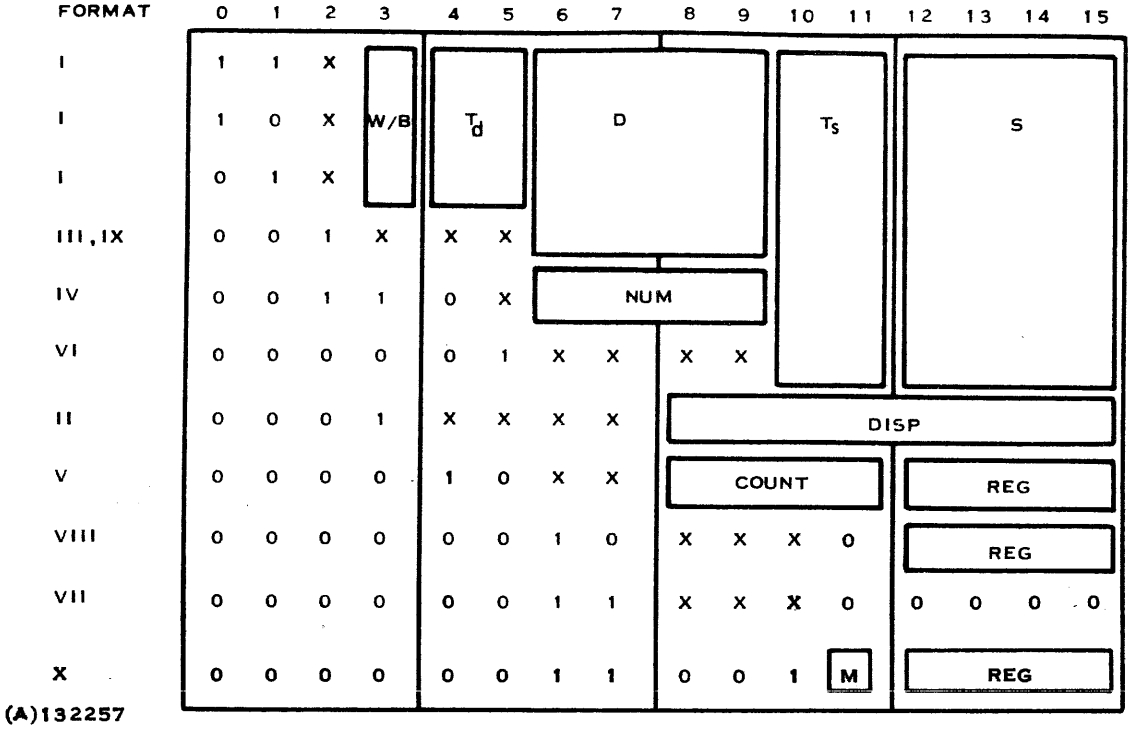
To correct object code without reassembling a program, change the object code by changing or adding one or more records. One additional tag character is recognized by the loader to permit specifying a load point. The additional tag character, D, may be used in object records changed or added manually.

Tag character D is followed by a load bias (offset) value. The loader uses this value instead of the load bias computed by the loader itself. The loader adds the load bias to all relocatable entry addresses, external references, external definitions, load addresses, and data. The effect of the D tag character is to specify the area of memory into which the loader loads the program. The tag character D and the associated field must be placed ahead of the object code generated by the assembler.

Correction of object code may require only changing a character or a word in an object code record. The user may duplicate the record up to the character or word in error, replace the incorrect data with the correct data, and duplicate the remainder of the record up to the 7 tag character. Because the changes the user has made will cause a checksum error when the checksum is verified as the record is loaded, the user must change the 7 tag character to F.

When more extensive changes are required, the user may write an additional object code record or records. Begin each record with a tag character 9 or A followed by an absolute load address or a relocatable load address, respectively. This may be an address into which an existing object code record places a different value. The new value on the new record will override the other value when the new record follows the other record in the loading sequence. Follow the load address with a tag character B or C and an absolute data word or a relocatable data word, respectively. Additional data words preceded by appropriate tag characters may follow. When additional data is to be placed at a nonsequential address, write another load address tag character followed by the load address and data words preceded by tag characters. When the record is full, or all changes have been written, write tag character F to end the record.

When additional memory locations are loaded as a result of changes, the user must (except in single-module standalone tests) change the hexadecimal field following the tag character 0 that contains the number of bytes of relocatable code. For example, when the object file written by the assembler contained 1000_{16} bytes of relocatable code, and the user has added 8 bytes in a new object record, additional memory locations will be loaded. The user must find the 0 tag character in the object code file and change the value following the tag character from 1000 to 1008; he must also change the 7 tag character to F in that record.



X — IS A BIT OF THE OPERATION CODE THAT IS EITHER 0 OR 1 ACCORDING TO THE SPECIFIC INSTRUCTION IN THE FORMAT

W/B — IS A BIT OF THE OPERATION CODE THAT IS 0 IN INSTRUCTIONS THAT OPERATE ON WORDS, AND 1 IN INSTRUCTIONS THAT OPERATE ON BYTES

T_D — IS A PAIR OF BITS THAT SPECIFY THE ADDRESSING MODE OF THE DESTINATION OPERAND, AS FOLLOWS

- 00 = WORKSPACE REGISTER ADDRESSING
- 01 = WORKSPACE REGISTER INDIRECT ADDRESSING
- 10 = SYMBOLIC MEMORY ADDRESSING WHEN D = 0
- 10 = INDEXED MEMORY ADDRESSING WHEN D ≠ 0
- 11 = WORKSPACE REGISTER INDIRECT AUTOINCREMENT ADDRESSING

D — IS THE WORKSPACE REGISTER FOR THE DESTINATION OPERAND

T_S — IS A PAIR OF BITS THAT SPECIFY THE ADDRESSING MODE OF THE SOURCE OPERAND AS SHOWN FOR T_D

S — IS THE WORKSPACE REGISTER FOR THE SOURCE OPERAND

NUM — IS THE NUMBER OF BITS TO BE TRANSFERRED

DISP — IS A TWO S COMPLEMENT NUMBER THAT REPRESENTS A DISPLACEMENT

REG — IS A WORKSPACE REGISTER ADDRESS

COUNT — IS A SHIFT COUNT

M — IS A MAP REGISTER FILE NUMBER (0 OR 1)

Figure A-3. Machine Instruction Formats

```
                                OBJECT FILE LISTING
0 0000SAMPROG   9 0040  C 0000  A 0020  B C06D  B 0002  9 0042  C 0020  A 0024  B C81B  C 002A  7 F219  F          0001
A 0028  B 0241  B 0000  B CB41  B 0002  B 0380  A 00CA  C 0052  C 00A2  B 02E0  C 0032  B 0200  B 0F0F  7 F1DE  F 0002
A 00D6  B C0A0  C 00CA  B 04C3  B C160  C 00CC  B C1A0  C 00D0  B C072  B 0281  B 3A00  A 00EC  B 0221  7 F151  F 0003
A 00EE  B 0900  B 06C1  A 00EA  B 1102  A 00F2  B 0453  B 11F8  B 2C20  C 0032  B C101  B 0B44  B E044  7 F18E  F 0004
A 0100  B DD66  B 0003  B 0282  C 00A2  B 11ED  B 0340  7 F832  F                                                0005
2 00CE  0 010C        7 FCAB  F                                                                                  0006

(A) 134412
```

Figure A-4. Object Code Listing Format

When added records place corrected data in locations previously loaded, the added records must follow the incorrect records. The loader processes the records as they are read from the object medium, and the last record that affects a given memory location determines the contents of that location at execution time.

The object code records that contain the external definition fields, the external reference fields, the entry address field, and the final program start field must follow all other object records. An additional field or record may be added to include reference to a program identifier. The tag character is 4, and the hexadecimal field contains zeros. The second field contains the first six characters of the IDT character string. External definitions may be added using tag character 5 or 6 followed by the relocatable or absolute address, respectively. The second field contains the defined symbol, filled to the right with blanks when the symbol contains less than six characters.

Example 1: To include a 500-word load bias in the example object code shown in figure A-1, perform the following steps.

1. Insert the D tag with a hexadecimal bias of 1F4 (=500 decimal) before the first record of the object code. The first few characters should now be

   D01F400000SAMPROG...

2. Change the checksum tag (7) to the end of record tag:

   Before modification, the last line of the example is

   200CE0010C 7FCABF

   After modification, this line becomes

   200CE0010C FFCABF

Example 2: To change relocatable address 20_{16} to contain 0, perform the following steps.

1. Locate relocatable address 20_{16} in the object code, that is, find a tag A followed by 0020. There are two ways to do this:

   A. Look at the object code listing, figure A-4. The tags are separated from the code by spaces. It is easy to pick out A 0020 in the first line.

B. If the object listing is unavailable, A 0020 can still be found by examining the object code, figure A-1, and accounting for the fields; the first tag is 0. Tag 0 takes two data fields: the length of all relocatable code 0000 and an 8-character program identifier SAMPEOF. The next tag is 9, which takes a single field 0040. C must be the next tag, which takes one field 0000. The next tag must be A, which takes the field 0020. Thus, the relocatable address 0020 is found.

2. Modify the characters ...A0020... to ...A0000...

3. Change the checksum tag to an end of record tag: before=7FCABF, after=FFCABF.

APPENDIX B

OPERATION OF THE 990 MAINTENANCE UNIT

APPENDIX B

OPERATION OF THE 990 MAINTENANCE UNIT

B.1 INTRODUCTION

This appendix describes the 990 maintenance unit, which is used to load diagnostics into a system when no other diagnostic loader device is available.

B.2 DESCRIPTION

The 990 maintenance unit contains a 990 programmer panel and a 733 ASR/KSR tape cassette transport housed in a portable aluminum carrying case (see figure B-1). This unit contains a power supply and the electronics necessary to control the tape transport and interface to the 990 computer. The 990 maintenance unit has storage space for manuals, CPU ROMs, and interface cables. A special CRU adapter board is available to allow the 990 maintenance unit to interface with 960 systems via the CRU interface.

B.2.1 TEST CONFIGURATIONS. The maintenance unit accommodates two different test configurations depending on the system under test. In the first configuration (figure B-2), the maintenance unit ties into the 990 system under test at the interface connector for the programmer panel or operator panel, if applicable (after removal of the existing interface cable from the chassis-mounted programmer panel or operator panel). In this configuration, diagnostic programs are loaded into program memory from the cassette tape transport on the maintenance unit and the test results are displayed on the DATA LEDs on the programmer panel in the maintenance unit. This configuration is used when the system is not equipped with an operational 733 ASR data terminal.

B.2.1.1 Standalone Programmer Panel. The programmer panel may also be detached from the maintenance unit and used as a standalone unit. In this case, the existing interface cable from the chassis-mounted operator or programmer panel is removed from its connector at the computer and the maintenance unit's programmer panel interface cable is connected in its place (see figure B-3). In this configuration, the key switch on the chassis-mounted panel controls the application of ac power to the computer but all other functions are controlled by the standalone programmer panel.

This configuration is used when the system under test is equipped with an operator panel or inoperative programmer panel but contains an operative 733 ASR data terminal. In this case, the diagnostic cassettes are loaded into the system from the 733 ASR and the programmer panel is used to display test data from selected registers in the processor board or from selected memory locations on any of the boards containing memory storage.

B.2.2 OPERATING CONTROLS AND INDICATORS. The maintenance unit's operating controls and indicators are shown in figure B-4 and listed and described in table B-1. Basically, the maintenance unit contains a POWER ON/OFF switch which controls power to the maintenance unit, a RESET switch which initializes the controller board in the maintenance unit, a REWIND switch which is used to rewind cassette tapes, a LOAD switch which is used to initiate diagnostic load operations plus the conventional controls and indicators found on the 990 programmer

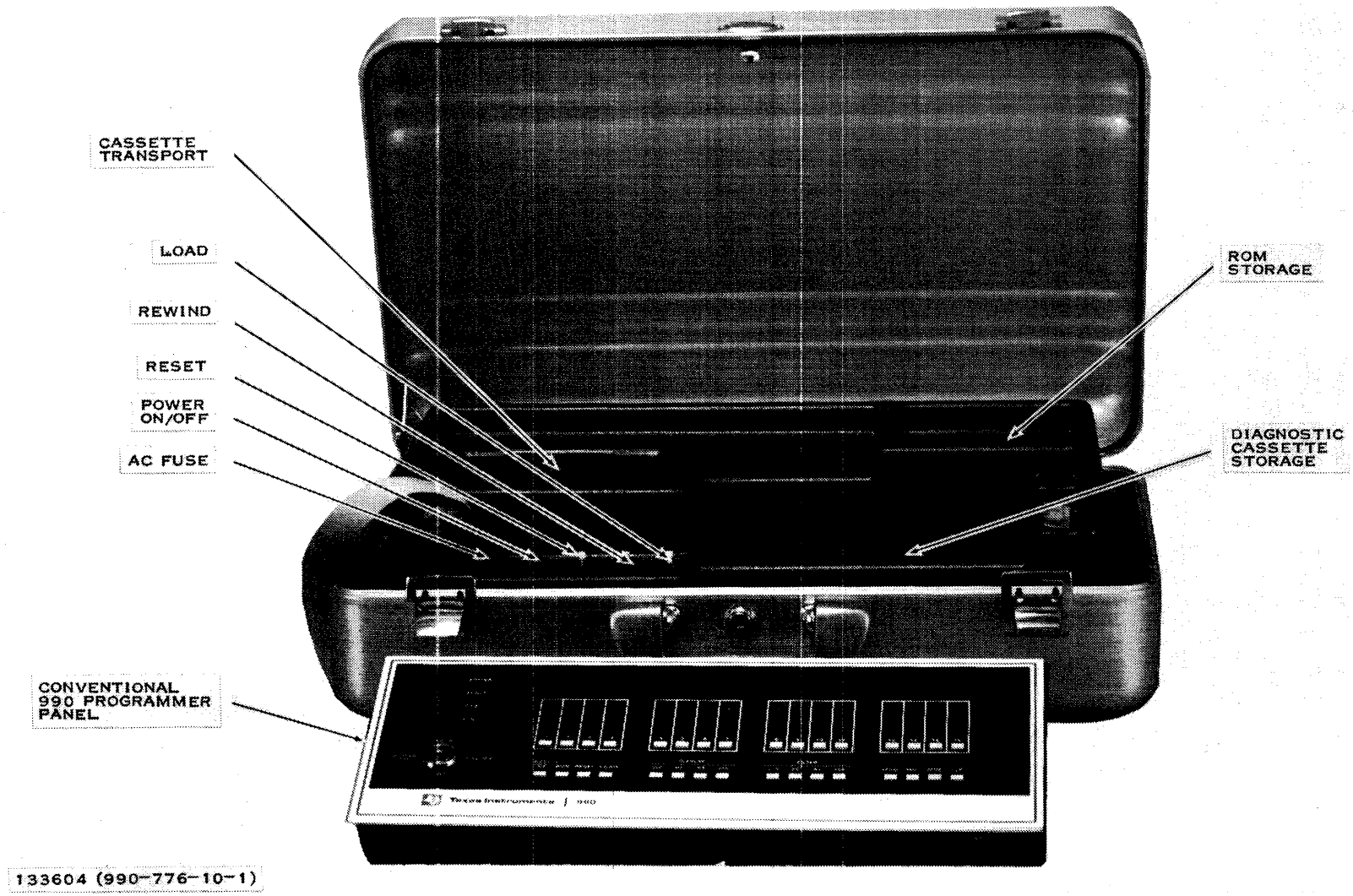


Figure B-1. 990 Maintenance Unit

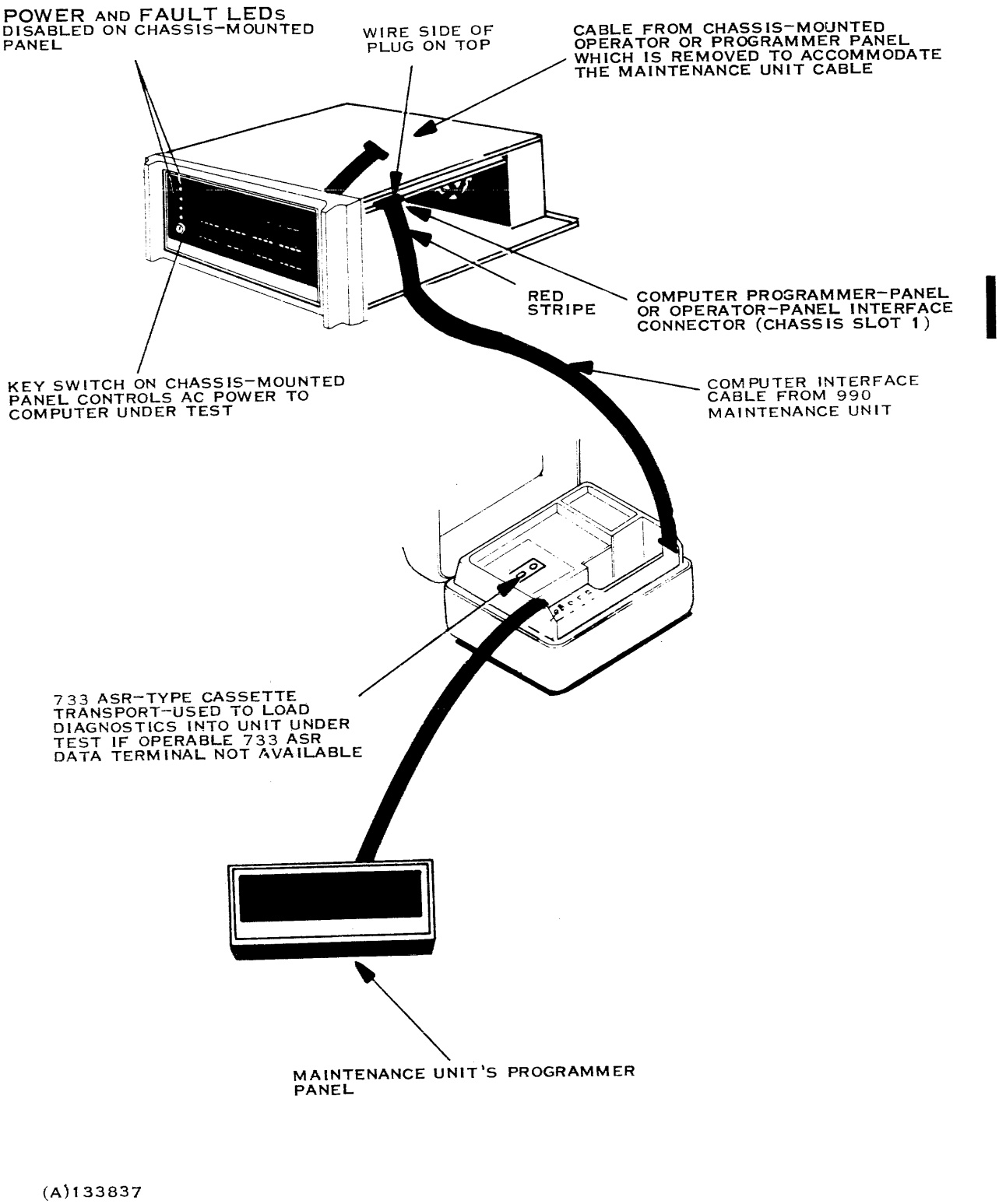


Figure B-2. Test Setup for System not Containing 733 ASR Data Terminal

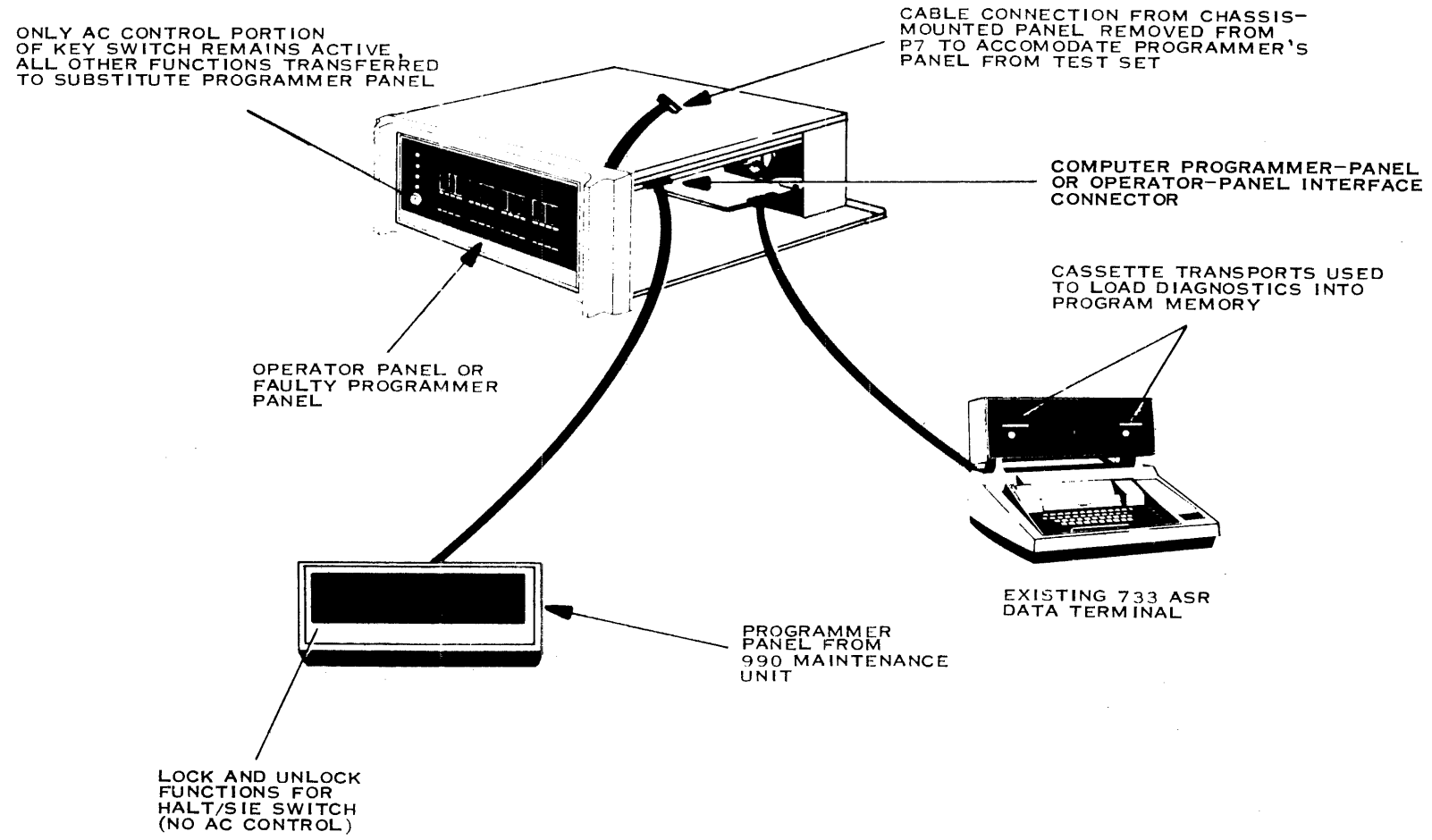
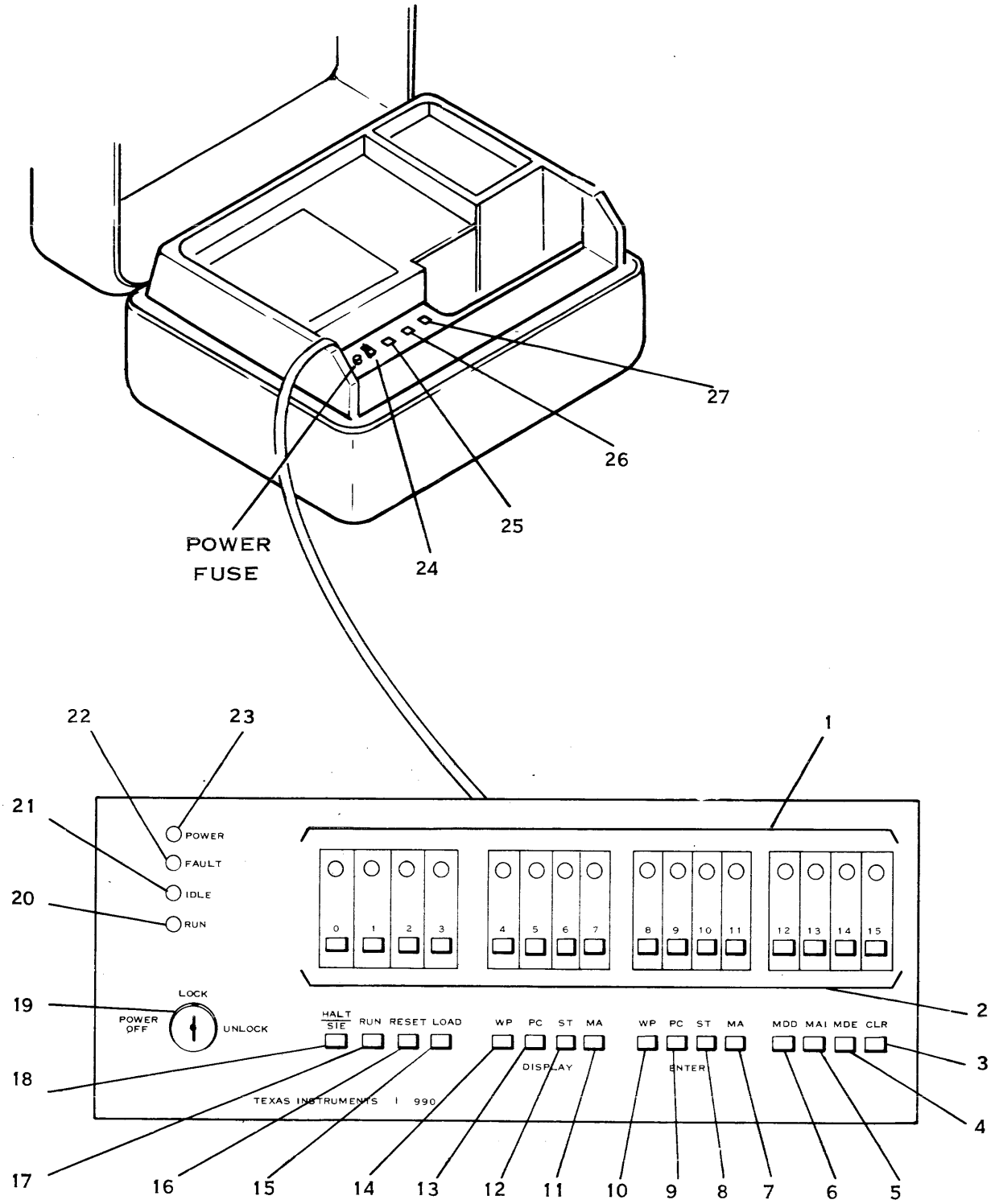


Figure B-3. Alternate Test Setup Using Programmer Panel Only



(A)133607

Figure B-4. Maintenance Unit Controls and Indicators

Table B-1. Maintenance Unit Controls and Indicators

Ref. No.	Control or Indicator	Function
1	DATA LEDs	The DATA LEDs are used to display data being entered into a CPU register or memory location or data presently stored in a register or memory location. During execution of the diagnostics, error numbers (in hex) are displayed on the right byte and the left byte is forced to all 1's (see error message number descriptions in 990 Computer Diagnostics Handbook). A lit LED denotes a logic 1, an extinguished indicator denotes logic 0. The LSB is displayed on the far right of the LEDs. In the HALT mode, the LEDs display a computer register contents, memory contents or value entered into computer memory via the data entry switches depending on which switches are pressed (see ref. 5, 7, 9, 11, 13, 15).
2	DATA entry switches	Used in conjunction with the ENTRY switches on the panel to enter data and addresses into selected computer registers and memory locations (active only when the panel is in the HALT mode of operation). In the HALT mode, the data LED located immediately above each data entry switch lights as each switch is pressed. The value indicated by the DATA LEDs is then stored in the register or memory address selected by the entry switches.
3	CLR switch	When pressed, this switch clears the DATA LED displays.
4	MDE switch	This switch is pressed to transfer a value displayed on the DATA LEDs to the memory location defined by the contents of the memory address (MA) register in the computer.
5	MAI switch	The memory address increment (MAI) switch is pressed to increment the value stored in the memory address register by a value of 2.
6	MDD switch	When pressed, this switch causes the contents of the memory location defined by the contents of the memory address register to be displayed on the DATA DISPLAY LEDs.
7	ENTER MA switch	When pressed, this switch causes the value displayed by the DATA LEDs to be entered into the computer's memory address register
8	ENTER ST switch	When pressed, the value displayed on the DATA LEDs is entered into the computer's status register.
9	ENTER PC	When pressed, the value displayed on the DATA LEDs is loaded into the computer's program counter.
10	ENTER WP	When pressed, the value displayed on the DATA LEDs is loaded into the computer's workspace pointer register.
11	DISPLAY MA	When pressed, the value stored in the computer's memory address register is displayed on the DATA LEDs.
12	DISPLAY ST	When pressed, the contents of the computer's status register is displayed on the DATA LEDs.
13	DISPLAY PC	When pressed, the contents of the computer's program counter is displayed on the DATA LEDs.

Table B-1. Maintenance Unit Controls and Indicators (Continued)

Ref. No.	Control or Indicator	Function
14	DISPLAY WP	When pressed, the contents of the computer's workspace pointer register is displayed on the DATA LEDs.
15	LOAD switch	For a programmer panel on the chassis (or PANEL mode of operation for the maintenance unit), pressing the switch causes the computer to trap to the ROM loader.
16	RESET switch	Pressing the RESET switch results in an IORESET – pulse being generated which resets all units in the system.
17	RUN switch	When the computer is halted (programmer panel is active), pressing the RUN switch returns the computer to the RUN mode of operation and deactivates the panel.
18	HALT/SIE switch	When the computer is in the RUN mode (RUN LED is lit), pressing the HALT/SIE switch causes the computer to halt and begin processing the front panel software *if* the key switch is set to the UNLOCK position. Pressing the switch when the computer is not in the RUN mode causes the computer to execute a single instruction at the present PC (program counter) address. The contents of the program counter are incremented by two and displayed on the DATA LEDs.
19	Key switch	The key switch (OFF/LOCK/UNLOCK) switch prevents unauthorized computer program intervention. The key must be inserted into the switch and the switch set to the UNLOCK position in order to enable the output of the HALT/SIE switch to the computer. The ac power control function for the 990 computer is controlled by the key switch on the chassis-mounted operator or programmer panel.
20	RUN LED	The RUN LED lights when a low-active RUN– signal is generated by the computer indicating the computer is in the RUN mode. When this LED is lit, all switches on the panel except the HALT/SIE switch are disabled and the DATA LEDs are driven under program control. When the RUN LED is extinguished, the panel controls are active.
21	IDLE LED	Lights when the computer is executing an idle instruction (indication of computer inactivity for most interrupt driven software).
22	FAULT LED	The FAULT LED lights when the computer has detected a self-test diagnostic failure or if the maintenance unit has detected a tape data fault.
23	POWER LED	Lights when the POWER ON/OFF switch on the maintenance unit is set to the ON position and the maintenance unit's internal power supply is functioning properly.
24	POWER ON/OFF switch	Controls ac power to the maintenance unit.
25	RESET switch	Initializes the logic on the maintenance unit's maintenance controller board. Has no affect on the system under test.
26	REWIND switch	Causes the cassette tape in the cassette transport to rewind back to the beginning of the tape (tape motion stops when clear leader is sensed).
27	LOAD switch	When the programmer panel is in the HALT mode (RUN LED extinguished), this switch causes the program on cassette tape to be loaded into memory. When the load is complete, program execution begins.

panel. However, the function of three of the programmer panel controls and indicators are slightly different when the programmer panel is used as a part of the maintenance unit. These differences include:

- Key switch – Does not affect ac power to the computer or maintenance unit, but otherwise exercises the same key control over program intervention in the computer.
- POWER LED – Indicates the status of the power supply in the maintenance unit rather than the power supply in the computer chassis.
- FAULT LED – May be lit by either the computer (self-test failure) or by the maintenance controller board in the maintenance unit in the event of a faulty tape read operation.

The functions of all other programmer panel controls and indicators are exactly the same as those of a conventional chassis-mounted programmer panel (see figure B-4 and table B-1).

B.2.2.1 Maintenance Unit Operating Procedures. Some of the more common operating procedures are briefly described in the following paragraphs. These procedures include:

- Changing panel mode of operation
- Mounting cassette tapes in maintenance unit
- Diagnostic load from cassette transport in maintenance unit
- Entering data into CPU registers or memory locations
- Displaying data from CPU registers or memory locations
- Single instruction execution

Changing Panel Mode of Operation. The programmer panel in the 990 maintenance unit may be operated in one of two modes including RUN and HALT. The computer initially comes up in the RUN mode when ac power is applied to the computer through the key switch on the chassis-mounted front panel. During this time, the RUN LED and all DATA LEDs on the programmer panel in the maintenance unit light and remain lit until the mode of operation changes. If the key switch on the programmer panel in the maintenance unit is set to the LOCK position, all controls on the panel are disabled. To change from the RUN mode of operation to the HALT mode, the key must be inserted in the switch and the key switch must be rotated to the UNLOCK position. At this point, only the HALT/SIE switch on the panel is enabled. When the HALT/SIE switch is pressed, the computer ceases normal program execution and traps to the panel software utility which is located in ROM. At this time, the RUN LED on the programmer panel extinguishes and the outputs of the switches on the programmer panel are constantly monitored by software through the programmer panel CRU type interface. At this point, the panel is operating in the HALT mode.

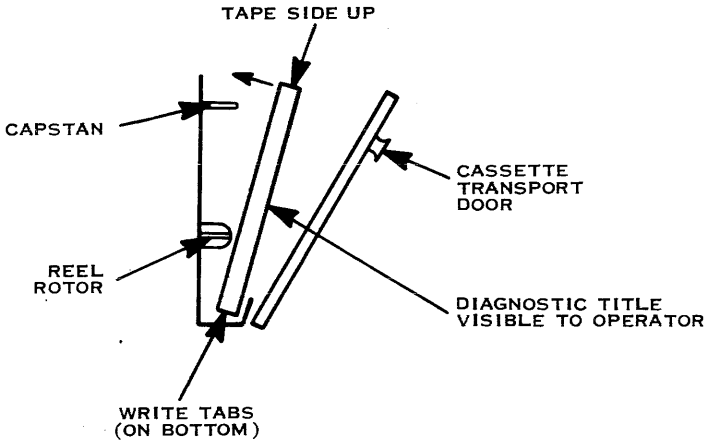
In the HALT mode, a diagnostic tape may be loaded from the maintenance unit or information may be entered into or displayed from selected CPU registers or program memory locations.

In order to switch from the HALT mode of operation to the RUN mode, the RUN switch must be pressed on the programmer panel. This causes the CPU on the processor board to begin program execution at the memory address indicated by its program counter.

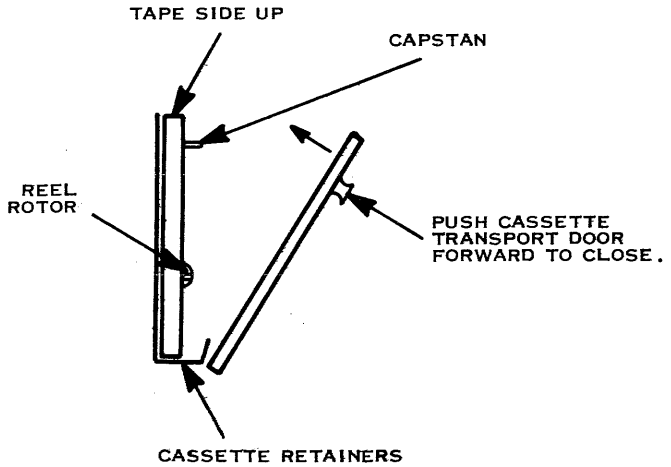
Mounting and Removing Tape Cassettes. In order to load a diagnostic tape cassette into the maintenance unit, the cassette transport door must be opened and the cassette inserted with the tape end up. The desired diagnostic title should be facing the operator as shown in figure B-5. The cassette should then be firmly pressed into the transport such that the capstan and reel motors properly engage the cassette tape and reels. The transport door must then be closed to complete installation of the cassette.

To remove a cassette from the transport, the door should be opened to the first stop and then opened the rest of the way using a quick downward motion. This causes the cassette to eject from the tape transport. When not in use, the transport door should be closed to prevent accumulation of dust or dirt in the tape drive mechanism and read head.

Loading a Diagnostic Into Program Memory. Before initiating a load from a cassette tape, the POWER ON/OFF switch should be set to the ON position and the tape fully rewound by pressing the REWIND switch on the maintenance unit (reference no. 26 in table B-1). The tape motion will automatically stop when the clear leader at the beginning of the tape is sensed by the maintenance controller board.



STEP 1. CASSETTE IS PROPERLY ORIENTED AND ALIGNED IN TRANSPORT



STEP 2. CASSETTE IS PRESSED INTO POSITION AND DOOR IS CLOSED.

(A)133608

Figure B-5. Cassette Tape Installation, Simplified Diagram

The Key switch on the 990 computer chassis-mounted operator or programmer panel must be set to one of the ON positions (ON, LOCK, or UNLOCK) to apply ac power to the computer. At this point, the RUN and POWER LEDs should light on the programmer panel in the maintenance unit. The panel's mode of operation must then be changed to the HALT mode by setting the Key switch on the maintenance unit's programmer panel to the UNLOCK position and pressing the HALT/SIE switch. The RUN LED on the panel should extinguish indicating the CPU is now processing the panel software utility. At this point, the panel software begins examining the switch outputs from the programmer panel. The cassette diagnostic load operation may now be initiated by pressing the LOAD switch (reference no. 27 in table B-1) on the maintenance unit. The panel software recognizes the combination of the Load signal with the Maintenance Unit Present signal as a "Load from maintenance unit cassette" command (refer to the flowchart in figure B-6). As a result, the panel software branches to the self-test program (if the loader ROM is equipped with self-test). If the self-test fails to execute correctly, the CPU lights the FAULT LED on the programmer panel in the maintenance unit and inhibits the diagnostic load operation.

However, if the self-test executes satisfactorily (or if self-test is not present), the CPU branches to the ASR loader program which is also stored in ROM. Since the load signal occurred with maintenance unit present, the software retains the CRU base address of the front panel.

At this point, the maintenance unit waits for RUN to turn on and then begins transmitting tape read data over the programmer panel CRU interface under control of the 733 loader program. Each time the maintenance controller board in the maintenance unit has a byte of data ready for the computer, it sets CRU bit 12 which is interpreted by software as a Read Request. When the 733 loader recognizes the Read Request, it serially transfers the next byte of data from the maintenance unit and resets CRU bit 12, freeing the maintenance unit to ready the next data byte. If the maintenance unit has another byte of data available before the loader has accepted the previous data byte (CRU bit 12 still set to the computer), the load operation is aborted and the FAULT LED lights on the maintenance unit's programmer panel. When the 733 loader decodes an End of File tag, it sets Clear Write Request via CRU output bit 13 which causes the maintenance unit to stop the cassette transport and return to an idle state.

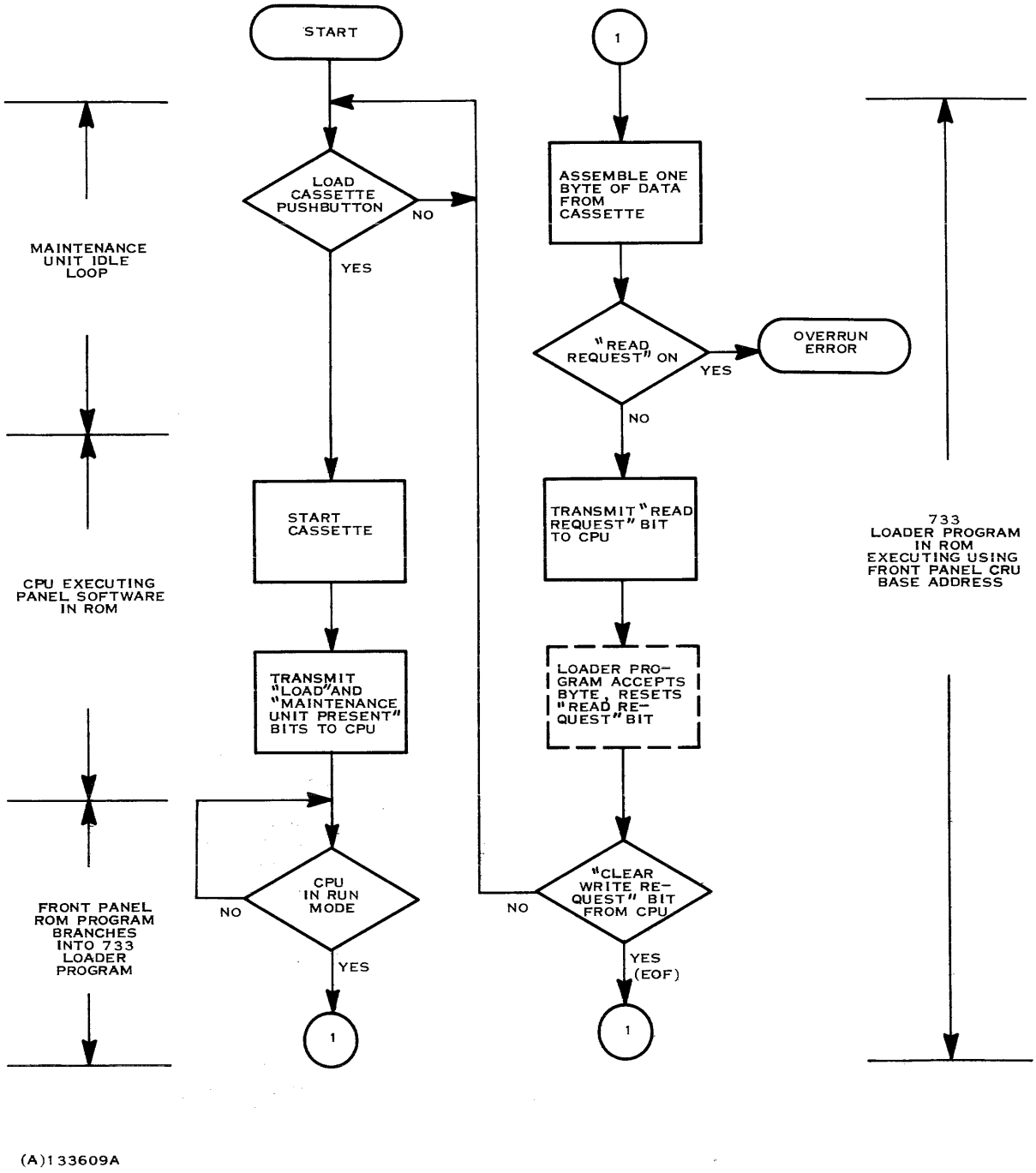


Figure B-6. Diagnostic Load from Maintenance Unit, Flowchart

APPENDIX C

OPERATION OF THE 990 PROGRAMMER PANEL

APPENDIX C

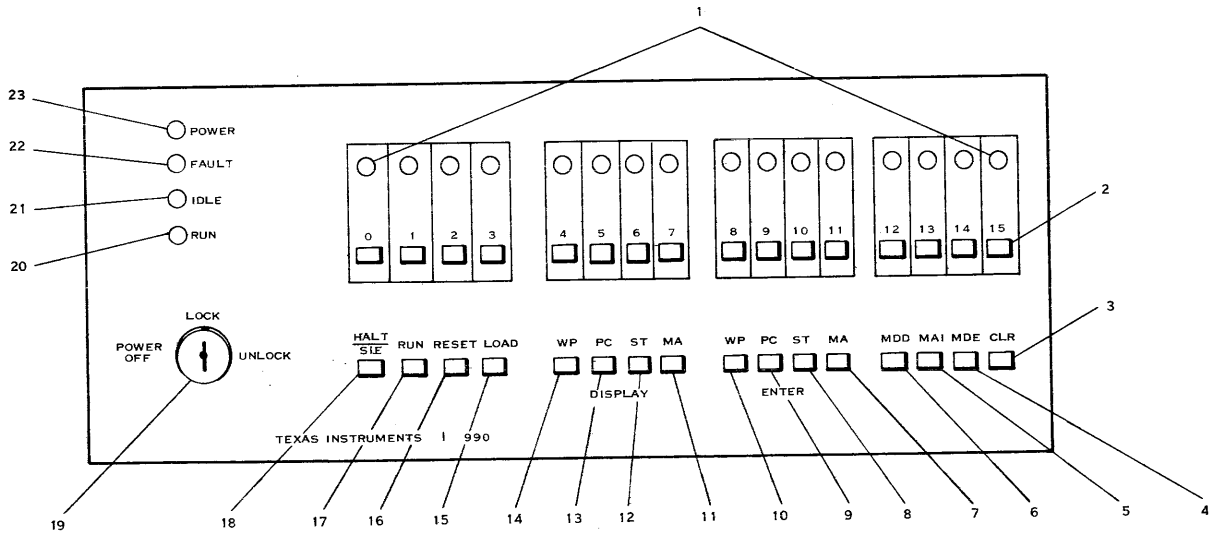
OPERATION OF THE 990 PROGRAMMER PANEL

C.1 INTRODUCTION

The programmer panel controls and indicators are shown in figure C-1 and listed and described in table C-1. Basically, the panel functions in one of two modes: run or halt.

When power is initially applied to the computer, the system comes up in the run mode, which locks out the programmer controls. In this mode of operation, the RUN LED and all DATA DISPLAY LEDs on the panel light and remain lit.

In order to set the programmer panel to the halt mode, the key must be inserted in the key switch and the switch rotated to the UNLOCK position. At this time, pressing the HALT/SIE switch halts the computer and activates the controls on the programmer panel. The panel may be returned to the run mode by pressing the RUN switch.



(B)133312

Figure C-1. 990 Programmer-Panel Controls and Indicators

Table C-1. Programmer-Panel Controls and Indicators

Reference Number	Control or Indicator	Function
1	DATA LEDs	In the run mode of operation, all DATA LEDs light except when the computer halts. At this point, the contents of the CPUs program counter is displayed. A lighted LED denotes logic 1, an extinguished indicator denotes logic 0. The LSB is displayed on the far right of the LEDs. In the halt mode, the LEDs display a computer register contents, memory contents, or a value entered into computer memory via the data entry switches, depending on which switches are pressed (see references 5, 7, 9, 11, 13, 15).
2	DATA entry switches	Used in conjunction with the ENTRY switches on the panel, these switches enter data and addresses into selected computer registers and memory locations (active only when the panel is in the halt mode of operation). In the halt mode, the data LED located immediately above each data entry switch lights as each switch is pressed. The value indicated by the DATA LEDs is then stored in the register or memory address selected by the entry switches.
3	CLR switch	When pressed, this switch clears the DATA LED displays
4	MDE switch	This switch is pressed to transfer a value displayed on the DATA LEDs to the memory location defined by the contents of the memory address (MA) register in the computer.
5	MAI switch	The memory address increment (MAI) switch is pressed to increment the value stored in the CPU's memory address register by a value of 2.
6	MDD switch	When pressed, this switch causes the contents of the memory location defined by the contents of the memory address register to be displayed on the DATA DISPLAY LEDs.
7	ENTER MA switch	When pressed, this switch causes the value displayed by the DATA LEDs to be entered into the computer's memory address register.
8	ENTER ST switch	When pressed, the value displayed on the DATA LEDs is entered into the computer's status register.
9	ENTER PC switch	When pressed, the value displayed on the DATA LEDs is loaded into the computer's program counter.
10	ENTER WP switch	When pressed, the value displayed on the DATA LEDs is loaded into the computer's workspace pointer register.
11	DISPLAY MA switch	When pressed, the value stored in the computer's memory address register is displayed on the DATA LEDs.
12	DISPLAY ST switch	When pressed, the contents of the computer's status register is displayed on the DATA LEDs.

Table C-1. Programmer-Panel Controls and Indicators (Continued)

Reference Number	Control or Indicator	Function
13	DISPLAY PC switch	When pressed, the contents of the computer's program counter is displayed on the DATA LEDs.
14	DISPLAY WP switch	When pressed, the contents of the computer's workspace pointer register is displayed on the DATA LEDs.
15	LOAD switch	When the panel is in the halt mode, pressing this switch causes the computer to trap to the ROM loader starting address.
16	RESET switch	Pressing the RST switch results in an IORESET – pulse being generated, which resets all units in the system.
17	RUN switch	When the computer is halted (programmer panel is active), pressing the RUN switch returns the computer to the run mode of operation and deactivates the panel.
18	HALT/SIE switch	When the computer is in the run mode (RUN LED is lighted), pressing the HALT/SIE switch causes the computer to halt and begin processing the programmer-panel software if the key switch is set to the UNLOCK position. Pressing the switch when the computer is not in the run mode causes the computer to execute a single instruction at the present PC (program counter) address. The contents of the program counter are incremented by two and displayed on the DATA LEDs.
19	Key switch	The key switch (OFF/LOCK/UNLOCK) prevents unauthorized computer turnon or program intervention. In order to apply ac power to the chassis, the key must be inserted into the switch and the switch set to the LOCK position. At this point, power is applied to the computer, but the programmer panel is locked out. In the UNLOCK position, the computer may be halted by pressing the HALT/SIE switch. The key may be removed from the switch in either the OFF or LOCK position.
20	RUN LED	The RUN LED lights when a low-active RUN– signal is generated by the computer indicating the computer is in the run mode. When this LED is lighted, all switches on the panel except the HALT/SIE switch are disabled and the DATA LEDs are driven under program control. When the RUN LED is extinguished, the panel controls are active.
21	IDLE LED	The IDLE LED lights when the computer is executing an idle instruction (indication of computer inactivity for most interrupt-driven software).
22	FAULT LED	The FAULT LED lights when the computer has detected a diagnostic test failure.
23	POWER LED	The POWER LED lights when power is applied to the unit (key switch on the panel set to the LOCK or UNLOCK position).

APPENDIX D

READING THE LOAD MAP LISTING

APPENDIX D

READING THE LOAD MAP LISTING

D.1 INTRODUCTION

Texas Instruments uses the 990/10 program development system link editor SDSLNK to produce the linked object modules for the diagnostic programs. Some of the diagnostic programs, particularly standalone programs, consist of single modules for which no linking is required. Other diagnostic programs, including all of the verb-package diagnostics, require linking. This linking is performed at the factory; however, users who wish to modify the standard object modules (for example to use a VDT in place of a 733 ASR/KSR) need to be able to read the load map listing (LMI) that accompanies the diagnostic programs. For further information about the link editor, SDSLNK, consult the *990/10 Program Development System Operation Guide.*

D.2 LOAD MAP LISTING

SDSLNK prints a load map as shown in the example in figure D-1. The heading identifies the version (part number and revision letter) of SDSLNK and shows the date and time of the run.

The load map consists of setup data (listing of link control file and input/output file assignments) and the maps for each procedure and phase.

These procedure/phase maps consist of a header line, a module list, a list of definitions, a list of global definitions, and a list of unresolved references.

The heading for each procedure/phase map identifies it (Procedure N or Phase N), gives the name, origin within the linked output, and length.

The module list gives the name, module number relative to entire link, origin, length, type, date of assembly/compilation, time of assembly/compilation, and processor (i.e., SDSMAC) for each module in the link.

The defintions section lists in alphabetical order the defintions in the modules of the output file. The definitions and the corresponding values are printed four per line. The definitions are symbols that appear as operands in DEF statements in the source code of the modules. When the listing shows an asterisk preceding the definition, the symbol is not referenced in another module. When the value is followed by an asterisk, the value is self-defining (absolute) and does not require modification when relocated.

The global definitions section lists in alphabetical order the global symbols in the module, if any symbols are defined as global symbols. The format is similar to the format of the definition section, and asterisks have the same significance.

The unresolved references section lists each unresolved reference followed by the number of the module in which it appears. An unresolved reference is a symbol that is used as an operand of a REF statement in the source code of the module, for which definition is not found in another module of the link edit. An unresolved reference that appears in more than one module will be listed for each module in which it appears.

After data for all phases has been printed and all processing is complete, SDSLNK prints the final message:

**** LINKING COMPLETED

An example listing is shown in figure D-1.

```
 TI 990/10 SDSLNK 936060 **    01/00/00   00:28:09                PAGE 1
COMMAND LIST


NOSYMT
PHASE 0,FLPDSK
INCLUDE DSC6.OBJECT.PARTS (FLOPSV)
INCLUDE DSC6.OBJECT.PARTS (FLOPT1)
INCLUDE DSC6.OBJECT.PARTS (FLOPT2)
END


 TI 990/10 SDSLNK 936060 **    01/00/00   00:28:09                PAGE 2
LINK MAP

CONTROL FILE = DSC6.LNKTMP.DIAG (FLPDSK)

LINKED OUTPUT FILE = DSC6.OBJECT.DIAG (FLPDSK)

LIST FILE = DSC6.LSTING.DIAG (FLPDSK)

OUTPUT FORMAT = ASCII


(A)136426A
```

Figure D-1. SDSLNK Load Map Example (Sheet 1 of 2)

```
TEXAS INSTRUMENTS DSD SOFTWARE FACTORY     PART NUMBER 02250105-9006 REV. *A
TI 990/10 SDSLNK 939187 *A   04/15/78  07:43:52                      PAGE   3


PHASE 0, FLPDSK   ORIGIN = 0000  LENGTH = 2B18

MODULE    NO   ORIGIN    LENGTH      TYPE        DATE       TIME       CREATOR

FLOPSV    1     0000       11F8     INCLUDE    04/13/78   15:14:29    SDSMAC
FLOPT1    2     11F8       07D8     INCLUDE    04/11/78   08:12:59    SDSMAC
FLOPT2    3     19D0       1148     INCLUDE    04/11/78   08:01:55    SDSMAC


                                D E F I N I T I O N S

 NAME     VALUE NO     NAME     VALUE NO     NAME     VALUE NO     NAME     VALUE NO

 ADDR1    05AE  1      ADDR2    05B0  1     *BEGMS1   1602  2     *BEGMS2   1608  2
*CD       0506  1      CISUER   0DD2  1     *CM       04A2  1      COMLEN   05B2  1
 COMPAR   0558  1      DEVCRU   0324  1      DEVINT   0326  1      EA       12FC  2
 EB       1346  2      EEMOUT   1362  2      ENDMS1   160C  2     *ENDMS2   1612  2
 ERCNT    05AC  1      ERMS2    1384  2      ERTRY    0336  1      ET       1228  2
*FD       05F8  1     *IC       0342  1     *IM       035C  1      INTFLG   0332  1
 INTOCC   11A6  1     *INTSAV   150A  2      ISRPC    115E  1     *IT       00B0  1
*ITVERB   00BC  1      LA       131C  2     *LO       036E  1     *LOOPFL   0438  1
 LOPCNT   045C  1      LOPCTR   048C  1      LT       12C6  2      NORMEX   11A4  1
 PART10   2642  3      PART11   290E  3      PART5    19D0  3      PART6    1D98  3
 PART7    20BA  3      PART8    21FA  3      PART9    235E  3      PGMEND   2B18  3
 PWRFLG   0338  1      SCHKDS   1138  1     *SCHKER   10B8  1      SETUP    1388  2
 SETUP2   13AE  2      STATUS   113A  1      TESTAB   11F8  2      TESTNO   1264  2
 TESTWP   12A6  2      UNITO    0328  1      UNITFL   0334  1

(A)136427A
```

Figure D-1. SDSLNK Load Map Example (Sheet 2 of 2)

APPENDIX E

DIAGNOSTICS PART NUMBERS

Although the diagnostics are not sold individually, there are part numbers assigned to each diagnostic, which allows reference to individual diagnostics and individual diagnostics are available at the depot level. Each part number associated with the diagnostics is shown in table 1-1.

Associated with each diagnostic is a collection of materials such as listings, program descriptions, submodules, and others. Each is assigned a part number which has the form XXXXXXX - YYYY. The XXXXXXX part corresponds to the individual diagnostic and the YYYY part corresponds to the specific material being referenced. For example, 2250119-0009 is the part number of the Microfiche Listings Kit for RAM10, the 990/10 random access memory test. 2250119 indicates RAM10 and 0009 indicates Microfiche Listings Kit.

The materials available for each diagnostic are listed below:

Program Description (PD)

Operating Procedure (OP)

Assembly Listing (LIST)

Source (SRC)

Object (OBJ)

Fully Linked Object (FLO)

Microfiche Listings Kit (SP), including PD, LML's and LIST's

Diagnostic kits for the 990 computers are shown in the following list:

Part Number	Description
0937782	Diagnostics Kit 990
0937782-1	Diagnostics Kit (CCO) - 990
0937782-2	Diagnostics Kit (CDO) - 990
0937782-3	Diagnostics Kit (FDO) - 990
0937782-4	Diagnostics Kit (DS31 OBJ) - 990
0937782-5	Diagnostics Kit (DS25 OBJ) - 990
0937782-6	Diagnostics Kit (DS50 OBJ) - 990
0937782-9	Diagnostics Kit (Fiche Kit) - 990
0937782-10	Diagnostics Kit (DS10 OBJ) - 990

Base part numbers for the DOCS load modules are listed below:

Part Number	Description
2250162	MXDOCS, Standard-DOCS
2250163	MNDOCS, Mini-DOCS
2250164	FPDOCS, Front Panel DOCS

All three versions of the DOCS are available on cassette, cards, and diskette, and all except FPDOCS are available on disk. The media must be specified when ordering any version of the DOCS. Microfiche Listings Kits are available for all three versions of the DOCS.

ALPHABETICAL INDEX

INTRODUCTION

The following index lists key words and concepts from the subject material of the manual together with the area(s) in the manual that supply major coverage of the listed concept. The numbers along the right side of the listing reference the following manual areas:

- Sections - References to Sections of the manual appear as "Section x" with the symbol x representing any numeric quantity.
- Appendixes - References to Appendixes of the manual appear as "Appendix y" with the symbol y representing any capital letter.
- Paragraphs - References to paragraphs of the manual appear as a series of alphanumeric or numeric characters punctuated with decimal points. Only the first character of the string may be a letter; all subsequent characters are numbers. The first character refers to the section or appendix of the manual in which the paragraph is found.
- Tables - References to tables in the manual are represented by the capital letter T followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the table). The second character is followed by a dash (-) and a number:

  Tx-yy

- Figures - References to figures in the manual are represented by the capital letter F followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the figure). The second character is followed by a dash (-) and a number:

  Fx-yy

- Other entries in the Index - References to other entries in the index are preceded by the word "See" followed by the referenced entry.

USER'S RESPONSE SHEET

Manual Title: Model 990 Computer Diagnostics Handbook (945400-9701)

Manual Date: 15 January 1979 **Date of This Letter:** _____

User's Name: _____ **Telephone:** _____

Company: _____ **Office/Department:** _____

Street Address: _____

City/State/Zip Code: _____

Please list any discrepancy found in this manual by page, paragraph, figure, or table number in the following space. If there are any other suggestions that you wish to make, feel free to include them. Thank you.

CUT ALONG LINE

Location in Manual	Comment/Suggestion
_____	_____
	_____
	_____
	_____
_____	_____
	_____
	_____
	_____
_____	_____
	_____
	_____
	_____

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
FOLD ON TWO LINES (LOCATED ON REVERSE SIDE), TAPE AND MAIL

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 7284 DALLAS, TX

POSTAGE WILL BE PAID BY ADDRESSEE

TEXAS INSTRUMENTS INCORPORATED
DIGITAL SYSTEMS GROUP

ATTN: TECHNICAL PUBLICATIONS
P.O. Box 2909 M/S 2146
Austin, Texas 78769

FOLD

TEXAS INSTRUMENTS
INCORPORATED
DIGITAL SYSTEMS DIVISION
POST OFFICE BOX 2909 AUSTIN, TEXAS 78769