



**iMDX 557  
RESIDENT PROCESSOR CARD  
INSTALLATION MANUAL**

---

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department  
Intel Corporation  
3065 Bowers Avenue  
Santa Clara, CA 95051

The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel's software license, or as defined in ASPR 7-104.9(a)(9).

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Intel Corporation.

The following are trademarks of Intel Corporation and its affiliates and may be used to identify Intel products:

BXP	int <sub>e</sub> l	iSBC	MULTICHANNEL
CREDIT	Intelelevision	iSBX	MULTIMODULE
i	int <sub>e</sub> l <sub>i</sub> gent Identifier	iSXM	Plug-A-Bubble
iICE	int <sub>e</sub> l <sub>i</sub> gent Programming	Library Manager	PROMPT
ICE	Intellec	MCS	RMX/80
iCS	Intellink	Megachassis	RUPI
i <sub>m</sub>	iOSP	MICROMAINFRAME	System 2000
iMMX	iPDS	MULTIBUS	UPI
Insite	iRMX		

REV.	REVISION HISTORY	DATE
-001	Original issue.	10/82





The syntax conventions used throughout this manual are described below.

## Notational Conventions

UPPERCASE	Characters shown in uppercase must be entered in the order shown. You may enter the characters in uppercase or lowercase.
<i>italic</i>	Italic indicates a meta symbol that may be replaced with an item that fulfills the rules for that symbol. The actual symbol may be any of the following:
Vx.y	Is a generic label placed on sample listings where the version number of the product that produced the listing would actually be printed.
[ ]	Brackets indicate optional arguments or parameters. When two or more elements are enclosed in brackets, <i>all</i> elements are optional, but only <i>one</i> element may be entered.
{ }	One and only one of the enclosed entries must be selected unless the field is also surrounded by brackets, in which case it is optional.
...	Ellipses indicate that the preceding argument or parameter may be repeated.
<code>input lines</code>	In interactive examples, user input lines are printed in white on black to differentiate them from system output.
<cr>	Indicates a carriage return.





# CONTENTS

INSTALLATION		PAGE			PAGE
Introduction .....		1	Test 6 – CPU Test .....		11
Description .....		1	Test 7 – Timer Count Test .....		12
Resident Processor Card .....		1	Test 8 – Timer Interrupt Test .....		12
System RAM .....		1	Test 9 – Software Interrupt Test .....		12
Upgrade Kit Current Requirements .....		1	Test A – PROM Checksum Test .....		12
Incoming Inspection .....		3	Test B – Failsafe Timer Test .....		12
Service and Repair Assistance .....		3	Test C – Short Moving Inversion Memory Test ...		12
Installation Considerations .....		4	Test D – Long Moving Inversion Memory Test ....		12
Series II Installation Procedure .....		4	V Variables .....		13
Model 800 Installation Procedure .....		4	Test Commands .....		14
Diagnostic Tests .....		6	TEST .....		15
IPC-85 Diagnostic .....		6	ERRONLY .....		17
IOC Diagnostic .....		7	DEBUG .....		18
Integral Disk Drive Test .....		8	LIST .....		19
General Test .....		9	SUMMARY .....		20
Keyboard Test .....		9	CLEAR .....		21
ECON86 Confidence Tests .....		10	DESCRIBE .....		22
Test Description .....		10	IGNORE .....		23
Test 0 – Data Bus Ripple Memory Test .....		10	RECOGNIZE .....		24
Test 1 – Address Bus Ripple Memory Test .....		11	REPEAT .....		25
Test 2 – Address Memory Test .....		11	EXIT .....		26
Test 3 – Refresh Memory Test .....		11	Multiple Commands per Line .....		27
Test 4 – March Memory Test .....		11	Running the ECON86 Test .....		27
Test 5 – Byte/Word Transfer Test .....		11	ECON86 Error Messages .....		28

## TABLES

TABLE	TITLE	PAGE	TABLE	TITLE	PAGE
1	RPC-86 Jumpers .....	6	3	ECON86 Test Procedure .....	27
2	IPC-85 Firmware Diagnostic Error Messages .....	7	4	ECON86 Test Error Messages .....	28

## FIGURES

FIGURE	TITLE	PAGE	FIGURE	TITLE	PAGE
1	Upgraded System Memory Allocation .....	2	3	Mainframe Cardcage Upgrade Configuration .....	5
2	Mainframe Cardcage Upgrade Configuration (Series II/80) .....	5	4	ECON Tests 0 Through C Example .....	29
				(Series II/85) .....	







## Introduction

This publication provides general information, installation procedures, and confidence test information for the iMDX 557 Resident Processor Card Kit. This kit upgrades an Intellec® Series II/85 Microcomputer Development System an Intellec® Series III Microcomputer Development System. In addition, the Model 800 Microcomputer Development System can also be upgraded to the functional equivalent of a Series III system.

## Description

The iMDX 557 Resident Processor Card Kit includes the following hardware and software:

- a. Resident Processor Card
- b. Resident 8086/8087/8088 Macro Assembler Diskette (Single Density)
- c. Resident 8086/8088 Utilities and Numerics Libraries Diskette (Single Density)
- d. Series III System Diskette (same contents as items b and c) (Double Density)
- e. Series III Diagnostic Confidence Test Diskette (one Single and one Double Density)
- f. ALTER 8086-Based Editor Diskette (one Single and one Double Density)
- g. Blank Flexible Diskette

## Resident Processor Card

The Resident Processor Card (RPC-86), which occupies two slots in the system cardcage, includes an 8086 16-bit HMOS microprocessor running at 8.0 MHz, 256K of onboard RAM, and 16K bytes of ROM. The 16K bytes of ROM contains a firmware program (DEBUG-86) for debugging 8086-based programs. A real-time counter is used to wake the 8085-based Integrated Processor Card (IPC-85) ISIS-II processor. Communication between the RPC-86 and IPC-85 processors is accomplished through a 2K byte window in the IPC-85 RAM. A total of 244K bytes of RAM is available to the user.

## System Ram

Figure 1 shows the upgraded Series III system memory allocation. The one-megabyte address space of the 8086 microprocessor is divided into 16 pages of 64K bytes each. Pages 0 through 3 are located on the RPC-86 processor board. Pages 4 through F are reserved for system expansion (the top 16K bytes of Page F contains the DEBUG-86 firmware).

## Upgrade Kit Current Requirements

Power requirements for the RPC-86 processor are as follows:

	+5V	+12V	-12V
RPC-86	5.6A	25mA	23mA

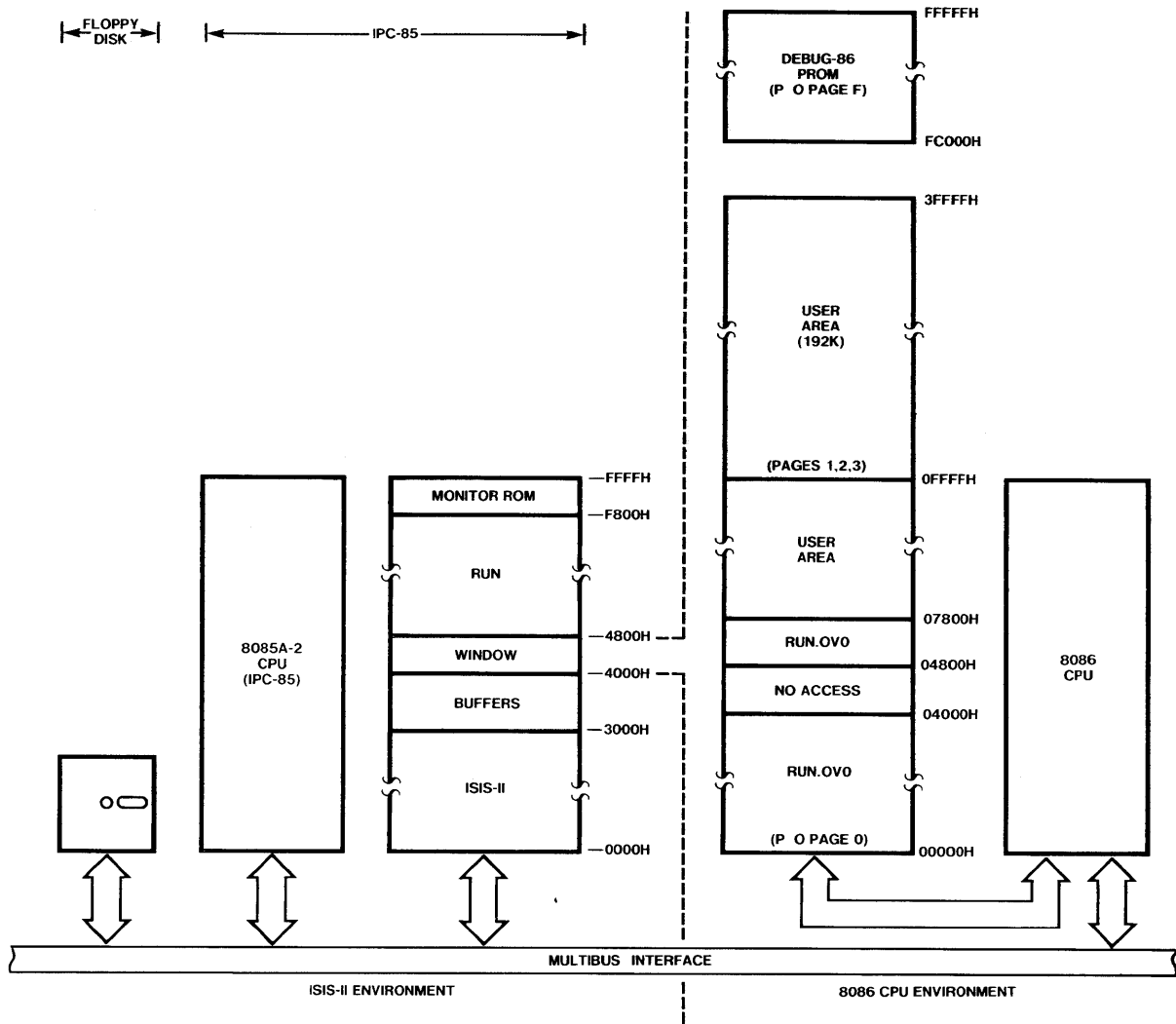


Figure 1. Upgraded System Memory Allocation

## Incoming Inspection

Inspect the exterior of the shipping carton immediately upon receipt for evidence of mishandling during transit. If the shipping carton is severely damaged or water-stained, request that the carrier's agent be present when the carton is opened. If the carrier's agent is not present, and the contents of the carton are damaged, keep the carton packing material for the agent's inspection.

For repairs to a product damaged in shipment, contact the Intel MCS D Technical Service Center to obtain a Return Authorization Number and further instructions. A purchase order will be required to complete the repair. A copy of the purchase order should be submitted to the carrier with your claim.

It is suggested that salvageable shipping cartons and packing material be saved for future use in the event the product must be reshipped.

## Service and Repair Assistance

The best service for your Intel product will be provided by an Intel Customer Engineer. These trained professionals will provide prompt, efficient on-site installation, preventive maintenance, or corrective maintenance services that will keep your equipment in the best possible operating condition.

Your Intel Customer Engineering can provide the service you need through a prepaid service contract or on an hourly charge basis. For further information, contact your local Intel office.

When it is impossible for you to use the services of an Intel Customer Engineer or when Intel service is not available in your local area, you may contact the Intel Service Center directly at one of the following numbers:

Telephone:

From Alaska, Arizona, or Hawaii call—(602) 869-4600

From all other U.S. locations call toll free—(800) 528-0595

TWX: 910-951-1330

Never return equipment to Intel for service or repair before you contact an Intel Customer Engineer or the Intel Service Center.

If return of your equipment is necessary, you will be given a Repair Authorization Number, shipping instructions, and other important information that will help Intel provide you with fast, efficient service. If the product is being returned because of damage sustained during shipment, or if the product is out of warranty, a purchase order is necessary in order for the Intel Service Center to make the repair.

When preparing the product for shipment to the Service Center, use the original factory packaging material is available. If the original packaging is not available, wrap the product in a cushioning material such as Air Cap SD-240, manufactured by the Sealed Air Corporation, Hawthorne, N.J. (or equivalent) and enclose in a heavy-duty corrugated shipping carton. Seal the carton securely, mark it "FRAGILE" to ensure careful handling, and ship it to the address specified by the Intel Service Center.

### NOTE

Customers outside of the United States should contact their sales source (Intel Sales Office or Authorized Intel Distributor) for directions on obtaining service or repair assistance.

## Installation Considerations

A bus master is any module (board) that has the ability to acquire and control the system bus through the bus exchange logic. In the Intellec Series II system cardcage, the board installed in the bottom slot has the highest bus priority and the board installed in the top slot (the IPC-85 processor) has the lowest bus priority. Bus masters for the Intellec Series II system include the IPC-85 processor, the RPC-86 processor, hard disk controller, dual double density diskette controller, and in-circuit emulator (ICE™) modules.

For optimum throughput efficiency, the RPC-86 processor must have the second lowest bus priority and, due to interrupt handling considerations, must always be installed directly below the IPC-85 processor. The controller for the hard disk drive or dual density diskette drive should have the highest bus priority. However, due to cabling requirements, the hard disk controller *must* reside in the mainframe cardcage even when the system includes an expansion chassis. The double density diskette controller may reside in the mainframe or expansion chassis cardcage.

For Intellec Series II/80 Development Systems with an IPB-80 processor, an IPC-85 Integrated Processor Card must be purchased and installed in place of the IPB-80 (see figure 2).

Figure 3 shows the recommended cardcage configuration for Series II/85 Development Systems including a drive controller.

### Series II Installation Procedure

Figures 2 and 3 show the recommended cardcage configurations for the various upgraded Series II systems. Install the RPC-86 processor as follows:

1. Turn off system power and remove mainframe front panel.
2. Remove the RPC-86 board from the shipping carton. Press firmly inward on all socket mounted integrated circuits to ensure proper seating. Verify that the plug-on jumpers are correctly installed (see table 1).
3. Install the RPC-86 processor as shown in figures 2 and 3. Press firmly inward on the extractor handles to seat boards in their mating connectors in the cardcage.
4. Replace the front panel and turn on system power. Check out the system by performing the disk-based ECON-86 confidence tests in this manual.

### Model 800 Installation Procedure

The installation of the RPC-86 in the Model 800 is similar to the Series II systems. The following jumpers must be set:

Remove the jumper from E213 to E214 and reinstall from E212 to E213.

1. Rotate key-operated switch to OFF and remove mainframe top cover.
2. Install the RPC-86 processor in cardcage slot 5. Press firmly downward on extractor handles to seat the board in its mating connectors in the cardcage.
3. Replace the mainframe top cover and rotate the key-operated switch to ON. Check out the RPC-86 processor by executing the ECON86 confidence tests described later in this manual.

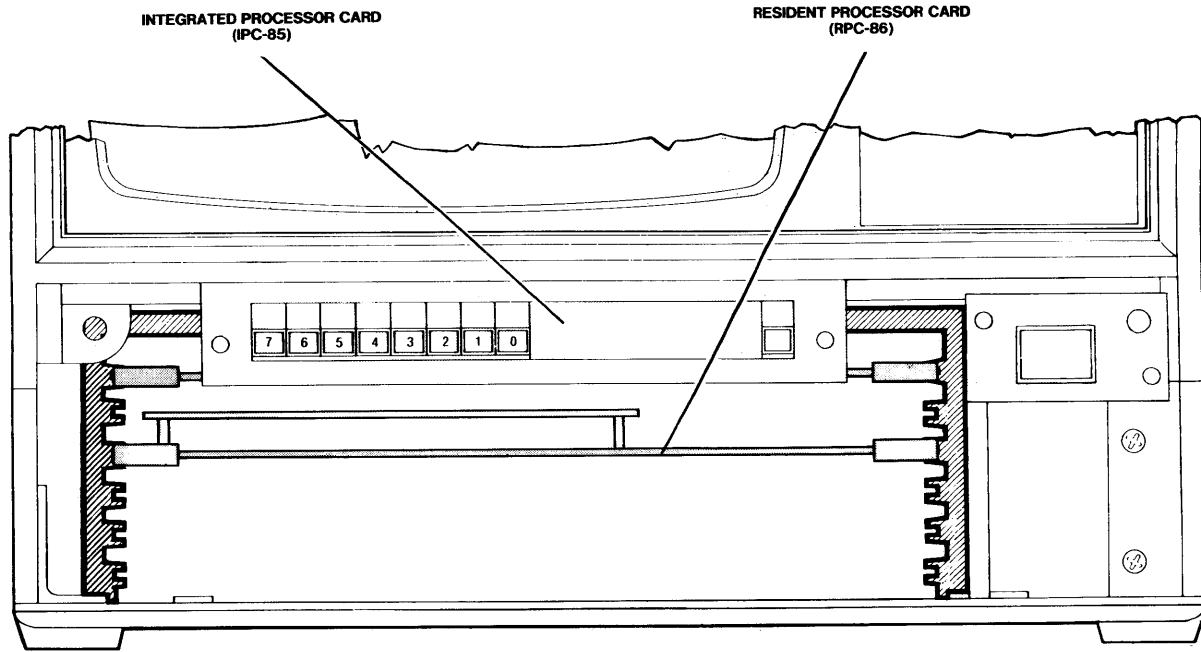


Figure 2. Mainframe Cardcage Upgrade Configuration (Series II/80)

122015-2

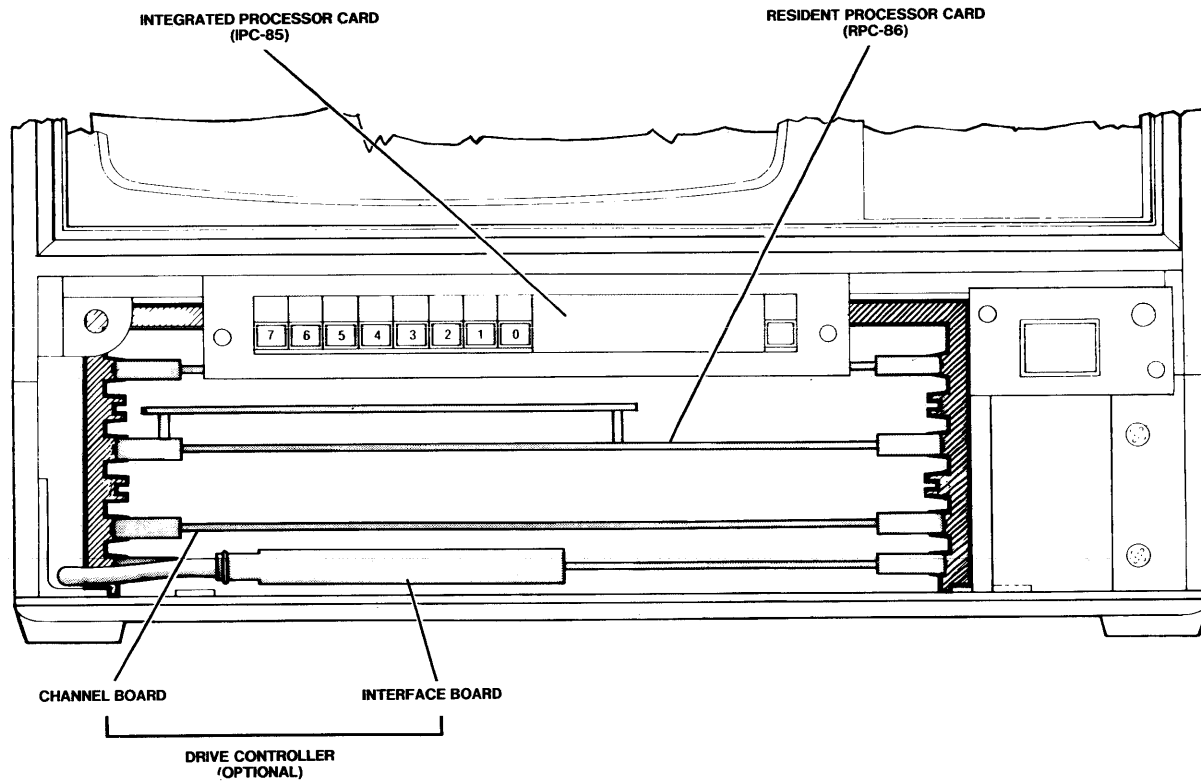


Figure 3. Mainframe Cardcage Upgrade Configuration (Series II/85)

122015-3

Table 1. RPC-86 Jumpers

To	From	To	From
E2	E3	E112	E113
E5	E9	E114	E115
E13	E14	E120	E128
E15	E16	E124	E125
E17	E18	E141	E165
E19	E53	E144	E145
E19	E118	E147	E158
E22	E23	E149	E164
E24	E129	E151	E152
E26	E27	E175	E176
E28	E32	E178	E179
E30	E31	E184	E185
E33	E34	E189	E193
E40	E41	E190	E194
E42	E43	E191	E195
E45	E54	E199	E200
E46	E55	E201	E202
E47	E56	E213	E214
E48	E57	E219	E225
E49	E58	E244	E248
E50	E59	E254	E255
E51	E60	E258	E259
E52	E61	E261	E262
E84	E85	E263	E264
E88	E89	E266	E268
E90	E91	E267	E269
E92	E93	E270	E272
E94	E95	E275	E276
E108	E109	E277	E278

## Diagnostic Tests

The power-up/reset diagnostic runs automatically when power is first applied to the system and whenever the front panel RESET switch is pressed. This diagnostic verifies operation of the basic IPC-85 and input/output controller (IOC) circuits, but is not comprehensive. If this is the initial system installation, or if there is any question about the system's operation, perform the IPC-85 and IOC diagnostics, and the ECON86 confidence tests.

The system gives not indication that it has passed the power-up/reset diagnostic. If failures are detected, an error message (or messages) will be displayed on the CRT.

## IPC-85 Diagnostic

The IPC-85 diagnostic tests circuit on both the IPC-85 and the IOC. It is more comprehensive than the power-up/reset diagnostic and is easily called, but does not test all system functions. Specifically, this diagnostic checks (1) ROM and parallel input-output (PIO) checksums, (2) IOC interrupts and RAM, (3) PIO interrupts and RAM, and (4) the 62K RAM on the IPC-85 processor board.

To run the IPC-85 diagnostic, proceed as follows:

1. Ensure that DIAGNOSTIC/LINE/LOCAL switch on rear panel is set to LINE (middle) position.

- Turn power on and press RESET. The system responds with the monitor sign-on message

```
SERIES II MONITOR, Vx.y
_
```

where *x.y* indicates the version and release number of the monitor.

- Call IPC-85 diagnostic by typing

```
[ESC] < : >
```

The IPC-85 diagnostic executes automatically and, if no errors are detected, the CRT displays

```
INTELLEC SERIES II DIAGNOSTIC Vx.y
TESTING CHECKSUMS-PASSED
TESTING IOC-PASSED
TESTING PIO-PASSED
TESTING RAM-PASSED
END DIAGNOSTIC
_
```

If the diagnostic detects errors, error messages will be displayed under the TESTING CHECKSUMS, TESTING IOC, TESTING PIO, or TESTING RAM messages as appropriate. The error messages and probable failures are shown in table 2.

**Table 2. IPC-85 Firmware Diagnostic Error Messages**

Error Message	Probable Failure*
FAILURE—BOOT CHECKSUM	IPC-85
FAILURE—IOC CHECKSUM	IOC
FAILURE—IOC INTERRUPTS	IPC-85 or IOC**
FAILURE—IOC NOT RESPONDING	IOC
FAILURE—IOC RAM	IOC
FAILURE—MONITOR CHECKSUM	IPC-85
FAILURE—PIO CHECKSUM	IOC
FAILURE—PIO INTERRUPTS	IPC-85 or IOC**
FAILURE—PIO NOT RESPONDING	IOC
FAILURE—PIO RAM	IOC
FAILURE—RAM BANK <i>mmK-nnK</i>	IPC-85

\*Before replacing any board, run IOC diagnostic. Make decision based on results of all tests.

\*\*Most probable cause is IPC-85, but may be IOC. Replace IPC-85 first.

### IOC Diagnostic

The IOC diagnostic test circuits that (except for the reset) are exclusively located on the IOC. It is therefore a good test to use to isolate troubles to either the IOC or the IPC-85. The IOC diagnostic also provides an audible indication of the test and can therefore be run as a starting point when the CRT is not providing correct indications.

In addition to testing the IOC, the IOC diagnostic also tests the keyboard, the CRT, and the integral disk drive. Probable causes of failures encountered during execution of the IOC firmware diagnostic are listed in table 2.

To run the IOC diagnostic, proceed as follows:

1. On the rear panel, set DIAGNOSTIC/LINE/LOCAL switch to DIAGNOSTIC (up) position. Press RESET.
2. The system runs a "five-beep" test automatically when RESET is pressed. If the test executes correctly, you will hear five beeps spaced as follows: two beeps, slight pause, then three beeps. After the fifth beep, the system displays the following sign-on message:

```
SERIES II IOC DIAGNOSTIC Vx.y
TYPE CNTL-@, FUNC-RUBOUT, "U" AND FUNC- "*"
          REQUESTED          RECEIVED
```

3. Type

	(CNTL and @ keys simultaneously)
	(FUNC and RUBOUT keys simultaneously)
	(FUNC, SHIFT, and *)

4. The system checks and displays each input and then displays the test menu as follows:

```
          REQUESTED          RECEIVED
↑@ 00000000 ↑@ 00000000
RD 11111111 RD 11111111
U 01010101 U 01010101
* 10101010 * 10101010
```

```
D-DISK
G-GENERAL
K-KEYBOARD/CRT
```

If REQUESTED and RECEIVED data do not match, the system will display ERROR and indicate faulty bits. For example:

```
          REQUESTED          RECEIVED
↑@ 00000000 ↑@ 00000000
RD 11111111 W 11010111 ERROR 00101000
U 01010101 U 01010101
* 10101010 # 10100011 ERROR 00001001
```

### Integral Disk Drive Test

This test may be selected only when IOC diagnostic test menu is displayed. Proceed as follows:

1. Type



The system prompts

```
DISK TEST
Insert SCRATCH disk and type "#".
```

2. Insert the scratch diskette; ensure that the diskette is write enabled. Type

```
#
```

3. The system runs the test in approximately 40 seconds. The indicator on the drive lights during the test, and you can hear the drive operating. If the system passes the test, the test menu is displayed. If there is a failure, the system displays

```
READ ERROR
```

or

```
ERROR nnnnnnnn
```

### General Test

To select and execute the general test from the IOC diagnostic test menu,

1. Type

```
G
```

2. The system displays

```
TEST PASSED
```

or

```
TEST FAILED
```

and the menu.

### Keyboard Test

To select and execute the keyboard test from the IOC diagnostic test menu,

1. Type

```
K
```

2. The system displays a full screen (25 lines, 80 characters per line) of characters.
3. Type each keyboard character. The system displays a full screen of each typed character. Note that pressing CNTL and any other key displays ↑ x where x is other key. Letters are displayed as capitals unless TPWR key is also pressed and latched. Letters follow the SHIFT key when the TPWR key is released. Note that pressing FUNC and any other key displays FUNC x and fills the screen with xs where x is the other key.

#### NOTE

On systems upgraded with the iMDX 511 Firmware Enhancement Kit, the RPT key becomes the FUNC key.

4. On the rear panel, set the DIAGNOSTIC/LINE/LOCAL switch to LINE (middle) position. Press RESET. System displays:

```
SERIES II MONITOR, Vx.y
```

```
· _
```

## ECON86 Confidence Tests

The ECON86 confidence test, which runs under Version 4.1 (or later) of the ISIS-II Disk Operating System, provides a complete check of the RPC-86, including its CPU, RAM, ROM, timers, and interrupts. The test can also check the system address and data lines to any additional offboard RAM memory (above 256K).

### Test Description

The ECON86 confidence test comprises the 14 individual tests listed below and described in the following paragraphs. Once invoked, these tests run automatically without operator intervention.

Test	Function
0	Data Bus Ripple Memory Test
1	Address Bus Memory Ripple Test
2	Address Memory Test
3	Refresh Memory Test
4	March Memory Test
5	Byte/Word Transfer Test
6	CPU Test
7	Timer Count Test
8	Timer Interrupt Test
9	Software Interrupt Test
A	PROM Checksum Test
B	Failsafe Timer Test
C	Short Moving Inversion Memory Test
D	Long Moving Inversion Memory Test

The entire 256K of RAM located on the RPC-86 processor board is checked during the memory tests. The test range for any offboard memory tests may be specified using the V variables described later in the paragraph headed "V VARIABLES."

### Test 0 — Data Bus Ripple Memory Test

The V variables are examined to determine testable memory. For all memory segments except the first 64K bytes, the first memory location is written, read, and verified for the following hexadecimal values: 0, 1, 2, 4, 8, 10, 20, 40, 80, 100, 200, 400, 800, 1000, 2000, 4000, and 8000. This test locates catastrophic failures (dead chips). The first 64K bytes are not tested because such testing requires moving the test code to the second 64K segment. Since the first segment has not properly tested, there is the possibility of a catastrophic failure.

### **Test 1 — Address Bus Ripple Memory Test**

The V Variables are examined to determine the testable memory. For all testable memory segments, the segment is filled with zeros. Then an incrementing byte data value is written to the following hexadecimal address within the segment: 0, 1, 2, 4, 8, 10, 20, 40, 80, 100, 200, 400, 800, 1000, 2000, 4000, and 8000. The same addresses are examined for the expected data values. The Address Ripple test is only performed on full 64K byte segments, thus the first 64K bytes are not tested. This test locates shorted and open address lines.

### **Test 2 — Address Memory Test**

The V variables are examined to determine testable memory. For all testable memory except the first 64K bytes, memory is filled with an incrementing value every 17 words, ensuring that no value is used more than once. Then, all testable memory is verified to contain the incrementing pattern. Next, the test code is move to the second 64K bytes, and the first 64K bytes are tested. The Address Memory test ensures uniqueness across segment boundaries. Uniqueness within a segment is verified by the March and Inversion Memory tests.

### **Test 3 — Refresh Memory Test**

The V variables are examined to determine testable memory. For all testable memory except the first 64K bytes, memory is filled with a constant pattern (AA55H). The test executes a software delay loop of about ten seconds. All memory is then verified to contain the original pattern. This test ensures that refresh occurs automatically whenever the CPU does not access a particular memory segment. The first 64K bytes are not tested because they are refreshed by the same circuitry as the second 64K bytes.

### **Test 4 — March Memory Test**

The V variables are examined to determine testable memory. For all memory segments, memory is filled with an initial pattern. The first location in the segment is read, verified, complemented, read and verified. The process is repeated until all words in the segment have been tested. The segment now contains the complement of the initial pattern. Next, the test begins with the last word in the segment, again reading, complementing, reading, and so on towards the beginning of the segment. The entire sequence (forward and reverse march) is repeated for four different initial patterns (AA55H, 55AAH, 0000H, and FFFFH).

### **Test 5 — Byte/Word Transfer Test**

The V variables are examined to determine testable memory. For all memory segments the first two words are written with the values 1234 and 5678. These words are then verified. The odd word between these two words is verified to be 7812. The odd word is then written with 9ABCH and verified. The first two words are then verified to be BC34H and 569AH. This test verifies the ability of the memory to correctly handle 8 and 16 bit data transfers.

### **Test 6 — CPU Test**

This test executes the 8086 instruction set and verifies the results. If a failure occurs, the test terminates immediately; execution of the remaining tests should not be attempted.

**Test 7 — Timer Count Test**

This test, which verifies the count accuracy of the 8253 Programmable Interval Timer chip, programs counter 1 to interrupt on terminal count (mode 0). At a specified time, the count is read and compared with the expected results. If an error occurs, the test immediately displays the counter number together with the expected versus received count.

**Test 8 — Timer Interrupt Test**

This test, which checks the interrupt capability of the 8253 Programmable Interval Timer, programs counter 0 to interrupt on terminal count (mode 0). The test verifies that interrupt level 2 is generated when counter 0 reaches the terminal count. If the test fails, it outputs a FAILED message.

**Test 9 — Software Interrupt Test**

This test checks all 256 software interrupts, the interrupt on overflow (INT0) condition, and the single-step capability. If an error occurs, the test immediately displays the type of failure.

**Test A — PROM Checksum Test**

This test reads the DEBUG86 PROM and computes and verifies its checksum. If the PROM cannot be read or if the checksum is wrong, the test outputs a FAILED message.

**Test B — Failsafe Timer Test**

This test attempts to read a non-existent absolute memory location (F0000H), and verifies that the failsafe timer on the IPC-85 board times out approximately 10 mS after the read command is issued. For this test to execute properly, location F0000H must not be existent in the system. If the test fails, it outputs a FAILED message.

**Test C — Short Moving Inversion Memory Test**

This test performs a moving inversion algorithm with only one's fill data pattern. If DEBUG is true, the Inversion test prints the Block being tested and the type of test (except when testing the first 64K segment, when only the block is printed) as shown in the sample run in Test D.

**Test D — Long Moving Inversion Memory Test**

This test performs the moving inversion algorithm with both zero fill and one's fill data patterns, and with both incrementing and decrementing address values. This test takes about four times as long as Test C (about 36 minutes). If DEBUG is true, the Inversion test prints the block being tested, and the type of test as follows:

```
000DH Long Moving Inversion Memory Test
Block 1000:0000, length 0080 words
  Zero fill, Forward
  One fill, Forward
  Zero fill, Backward
  One fill, Backward
```

```

Block 2000:0000, length 2000 words.
  Zero fill, Forward
  One fill, Forward
  Zero fill, Backward
  One fill, Backward
Block 0000:0000, length 2000 words.
  etc.

```

## V Variables

The 1-megabyte address space of the 8086 processor is divided into 16 pages of 64K bytes each. Pages 0 through 3 are located on the RPC-86 board. Pages 4 through F are reserved for system memory expansion. The top 16K bytes of Page F is the onboard RPC-86 PROM memory, and any attempt to test this area will generate an error message.

Variables V(0) through V(4) are used by the RAM memory tests to determine the test range. If the variables are not specified, testing defaults to pages 0 through 3 (absolute addresses 0 through 3FFFF hex).

Variable V(0) is the test descriptor

where

- V(0)=0 tests pages 0 through 3.
- V(0)=1 tests a range of memory; the starting and ending addresses are determined by V(1), V(2), V(3), and V(4). When V(0)=1, indicating a range of memory to test, the starting and ending addresses are compared. If the range is less than 256 bytes, the test aborts and outputs an error message.
- V(0)=2 tests all memory; the ending address is determined by V(3) and V(4).

Variable V(1) specifies the starting page number, which must be in the range 1 through F.

Variable V(2) defines the 16-bit offset address within the page number specified by V(1). To determine the absolute starting address, the test simply appends the offset to the page number specified. For example,

If V(1)=2 and V(2)=FC20H, the absolute starting address is 2FC20H. The offset defined by V(2) can be any number from 0 to FFFFH.

Variable V(3) specifies the ending page number, which must be in the range 1 through F.

Variable V(4) defines the 16-bit offset address within the page number specified by V(3). The absolute address is determined in the same manner as V(1) and V(2), above.

The V variables must be specified before any testing of offboard RAM memory; otherwise the testing defaults to the RPC-86 RAM. Each variable can be entered individually or as a string.

**Example 1.** Test all RAM (assuming one additional 64K RAM board) Each variable can be individually input as follows:

```
* V(0) = 2 <cr>
* V(3) = 4 <cr>
* V(4) = FFFF <cr>
```

In this example, the test range is 10000H through 4FFFFH.

Alternatively, the same V variables can be entered as a string:

```
* V(0) LENS = 2,0,0,4,FFFF <cr>
```

**Example 2.** Test offboard memory from 47000H to 62000H:

```
* V(0) LENS = 1,4,7000,6,2000 <cr>
```

Note that in this example the starting absolute address as well as the ending absolute address must be specified.

**Example 3.** Once the V variables have been specified, the same test range is used in all subsequent offboard RAM testing until a new set of variables is specified. The V variables are cleared automatically when exiting ECON86, or may be cleared during an ECON86 test session:

```
* V(0) = 0 <cr>
```

## Test Commands

Eleven commands: TEST, ERRORLY, DEBUG, LIST, SUMMARY, CLEAR, DESCRIBE, IGNORE, RECOGNIZE, REPEAT, and EXIT, allow the operator to specify the test sequence and report the result in an orderly manner. These test commands are described in following paragraphs. As shown on the following pages, any command consisting of four or more letters can be abbreviated to the first three letters of the command. Additionally, the letter T is accepted as an abbreviation for TEST.

The function and syntax of each command, including examples, are given on the following pages.

# TEST

## Syntax

$$\text{TEST} \left[ \begin{array}{l} \text{test}\#[, \text{test}\#] \dots \\ \text{test}\#[, \text{test}\#] \dots \end{array} \right] \left[ \text{REPEAT} \left\{ \begin{array}{l} \text{UNTIL ERROR} \\ \text{UNTIL NOERROR} \\ \text{nnnn} \\ \text{FOREVER} \end{array} \right\} \right]$$

where

*test#* is a test number as described earlier in this manual.

*nnnn* is the number of test iterations in hexadecimal (H), decimal (T), octal (Q), or binary (Y); the default number base is hexadecimal if the base suffix (H, T, Q, or Y) is omitted. The maximum number of iterations specifiable by *nnnn* is  $65,535_{10}$ .

## Abbreviation

TES (or T)

## Description

The TEST command uses the operator-supplied elements to load and execute one or more software procedures (i.e., tests). The tests are executed in numerical order regardless of the order in which they are specified. If a *test#* (i.e., test number) is specified for which no test exists, an error results. If no *test#* is specified, all 19 tests will be executed.

The REPEAT element, together with one of its four modifiers FOREVER, UNTIL ERROR, UNTIL NOERROR, and *nnnn*, allows the selected test(s) to be repeated.

If REPEAT FOREVER is specified, or if REPEAT is specified without a modifier, the tests will execute in numerical order, regardless of errors, and loop repeatedly until the operator hits the CNTL and C (CNTL-C) keys simultaneously.

When REPEAT UNTIL ERROR is specified, the test(s) will execute in numerical order and loop only until or more of the tests return an error condition.

When REPEAT UNTIL NOERROR is specified, the test(s) will execute in numerical order and loop until all tests run without error.

When REPEAT *nnnn* is specified, the test(s) will execute in numerical order *nnnn* times. If *nnnn* = 0, the tests will not be executed, although the first test will be loaded into memory.

## Examples

1. Run all tests (0-D hexadecimal) in numerical sequence; the setting of V variables is not required.

```
*TEST<cr>
```

2. Run a single test (e.g., Test B).

```
*TEST B<cr>
```

3. Run Tests A, B, C and D as long as one or more of these tests fail.

```
*TEST A TO D REPEAT UNTIL NOERROR<cr>
```

4. Run Tests 9, C, and D until one or more of these tests fail.

```
*TEST 9, C TO D REPEAT UNTIL ERROR<cr>
```

5. Run Test A 500 (decimal) times.

```
*TEST A REPEAT 500T<cr>
```

6. Run Tests B and C continuously regardless of error or no error.

```
*TEST B, C REPEAT FOREVER<cr>
```

or

```
*TEST B, C REPEAT<cr>
```



# ERRONLY

## Syntax

```
ERRONLY = 0  
ERRONLY = 1
```

## Abbreviation

ERR

## Description

The ERRONLY command is used to selectively suppress (ERRONLY = 1) or display (ERRONLY = 0) PASSED messages. The errone switch is cleared during CON86 initialization (i.e., the default condition is ERRONLY = 0).

## Examples

1. Run Test 9 ten (decimal) times and suppress PASSED messages.

```
*ERR = 1<cr>  
*TEST 9 REPEAT 10T<cr>
```

If no errors occur, the test displays only the prompt (\*) after the tenth test iteration.

2. Run Test 9 fifty (hexadecimal) times and print PASSED or FAILED message. (assumes that ERR = 1 has been entered previously).

```
*ERR = 0<cr>  
*TEST 9 REPEAT 50<cr>
```

## DEBUG

### Syntax

```
DEBUG = 0  
DEBUG = 1
```

### Abbreviation

```
DEB
```

### Description

The DEBUG command is used to selectively suppress or display error messages. The debug switch is cleared during ECON86 initialization (i.e., the default condition is DEBUG = 0).

### Examples

1. Run Test 9 through Test C and display error messages.

```
*DEBUG = 1 <cr>  
*TEST 9 TO C <cr>
```

2. Run Test C 100 (octal) times and suppress error messages. (Assumes that DEBUG = 1 has been entered previously.)

```
*DEBUG = 0 <cr>  
*TEST C REPEAT 100Q <cr>
```

# LIST

## Syntax

```
LIST 'pathname'
```

## Abbreviation

```
L I S
```

## Description

The LIST command causes a copy of all subsequent output, including prompts, input, line echo, and error messages, to be sent to the ISIS-II file *pathname*. If the ':CO:' (the console display) is specified, there is effectively no list file (the initial setting).

## Example

Run Test 5 and print (copy) all output, including error messages, on line printer.

```
*DEBUG = 1<cr>  
*LIST \:LP:'<cr>  
*TEST 5<cr>
```

## SUMMARY

### Syntax

```
SUMMARY [ test# [ , test# ] . . . ] [ EO ]
```

### Abbreviation

SUM

### Description

For each specified test, the following information is displayed by the SUMMARY command: the test number, the number of times executed, the number of times an error occurred, and whether the test was ignored or not. If no test(s) is specified, a summary of all tests will be included. If EO (Errors Only) is specified, only those tests with a non-zero error count will be displayed. The summary listing will be concluded with a statement as to whether any of the specified tests show a non-zero error count. Note that all error counts are given in hexadecimal.

### Examples

1. Display summary of Tests 7, 8, and 9.

```
*SUMMARY 7 TO 9 <cr>
00007H TIME COUNT TEST
      0001 FAILED IN 105A TRIALS
00008H TIMER INTERRUPT TEST
      0000 FAILED IN 01FF TRIALS
00009H SOFTWARE INTERRUPT TEST
      000C FAILED IN 1FFF TRIALS
```

2. Display summary of Tests 7, 8, and 9 only if a non-zero error count exists.

```
*SUMMARY 7 TO 9 EO <cr>
00008H TIMER INTERRUPT TEST
      0001 FAILED IN 01FF TRIALS
```

## CLEAR

### Syntax

```
CLEAR [ test# [ , test# ] . . . ]  
      test# TO test#
```

### Abbreviation

CLE

### Description

For each specified test, or for all tests if *test* range is missing, the execution count and the error count are set to zero. The CLEAR command does not affect the status (ignored or recognized) of a test, nor is the CLEAR command affected by the status of a test.

### Example

Clear execution count and error count on Tests A, B, and C.

```
*CLEAR A,B,C<cr>
```

## DESCRIBE

### Syntax

```
DESCRIBE [ test# [ , test# ] . . . ]  
         [ test# TO test# ]
```

### Abbreviation

DES

### Description

The DESCRIBE command displays the name, or description, of the specified test(s), and whether the test(s) would be ignored by the TEST command. If *test* range is missing, the descriptions of all tests will be displayed.

### Example

Describe Tests 6, 7, 8, and 9. (Assume that the IGNORE command has previously been specified for Test 9.)

```
* DESCRIBE 6 TO 9 <cr>  
00006H CPU TEST ****IGNORED****  
00007H TIMER COUNT TEST  
00008H TIMER INTERRUPT TEST  
00009H SOFTWARE INTERRUPT TEST
```

## IGNORE

### Syntax

```
IGNORE [ test# [ , test# ] . . . ]  
       [ test# TO test# ]
```

### Abbreviation

IGN

### Description

The IGNORE command allows the operator, at the beginning of the ECON86 session, to declare which test(s) is not to be run. The IGNORE command remains valid until negated, all or in part, by the RECOGNIZE command.

### Example

Run all tests except Tests 5, 6, and 8.

```
* IGNORE 5,6,8 <cr>  
* TEST <cr>
```

## RECOGNIZE

### Syntax

```
RECOGNIZE [ test# [ , test# ] . . . ]  
          test# TO test#
```

### Abbreviation

REC

### Description

The RECOGNIZE command allows the operator to negate all or part of a previously issued IGNORE command.

### Example

Assume that Tests 5, 6, and 8 are presently ignored and it is desired to run all tests except Test 6; i.e., Test 6 will remain ignored.

```
* RECOGNIZE 5,8 <cr>  
* TEST <cr>
```



## REPEAT

### Syntax

```
REPEAT [nnnn]
.
.
.
ENDR
```

where

*nnnn*

is the number of repeat iterations in hexadecimal (H), decimal (T), octal (Q), or binary (Y); the default number base is hexadecimal if the base suffix (H, T, Q, or Y) is omitted. The maximum number of repeat iterations specifiable by *nnnn* is 65,535<sub>10</sub>.

### Abbreviation

REP

### Description

This compound command repeats the sequence of test commands between REPEAT and ENDR (end repeat). The default condition is “forever” if *nnnn* is not specified. Note in the following example that a nested prompt (.\* ) is issued while inputting the REPEAT command parameters.

### Example

Over an extended period of time, compile a cumulative list and a backup list of the test summary.

```
* REPEAT <cr>
.* LIST \ALIST' <cr>
.* TEST <cr>
.* SUM <cr>
.* TEST \ALIST' <cr>
.* TEST <cr>
.* SUM <cr>
.* ENDR <cr>
```

## **EXIT**

### **Syntax**

**E X I T**

### **Abbreviation**

**E X I**

### **Description**

When at the ECON86 prompt (\*) level, the EXIT command ends the test session and returns control to the ISIS-II Disk Operating System.

### **Example**

```
*EXIT <cr>
```

## Multiple Commands Per Line

Two or more of the ECON86 commands (except the compound command REPEAT...ENDR) may be entered on a single line as follows:

```
cmd1; cmd2; cmd3; ...; cmdn
```

Each command must be separated by a semicolon, and the execution of commands is from left to right. For example, the multiple command

```
* TEST 9; TEST 3; SUMMARY; EXIT<cr>
```

runs Test 9 and then Test 3, outputs a test summary, and returns control to the ISIS-II Disk Operating System. Alternatively, the same multiple command may be abbreviated as follows:

```
* T 9; T 3; S U M; E X I<cr>
```

## Running the ECON86 Test

Step-by-step procedures for initializing ECON86 and running the test routine are provided in table 3. To simplify the procedure, the software DEBUG switch is turned off by default. Figure 4 is an actual run of tests 0 through C.

Any test in progress may be aborted by simultaneously pressing the CNTL and C (CNTL-C) keys, which returns the program to the ECON86 prompt (\*) level. Note that it may take up to 20 seconds to effectively abort an onboard memory test because the test manager must restore the original onboard memory contents. If you abort any onboard memory test, the test will print out any errors that occurred before the abort command.

Table 3. ECON86 Test Procedure

ECON86 Test Initialization
<p>Turn on system power. Monitor signs on:</p> <p>SERIES II MONITOR Vx.y</p> <p>—</p> <p>Turn on system drive, load Series III Diagnostic Confidence Test Diskette, and press mainframe front panel RESET switch. After ISIS-II system signs on, initialize ECON86 in one of the following ways:</p> <p>If your system does <i>not</i> include a hard disk subsystem, initialize ECON86 as follows:</p> <p>ISIS-II, Vx.y</p> <pre>- RUN ECON86&lt;cr&gt;</pre> <p>Enhanced Series-III Diagnostic Confidence Test, Vx.y</p> <p>*</p> <p>If your system <i>includes</i> a hard disk subsystem, initialize ECON86 as follows:</p> <p>ISIS-II Vx.y</p> <pre>- F4; RUN; F4; ECON86&lt;cr&gt;</pre> <p>Enhanced Series-III Diagnostic Confidence Test, Vx.y</p> <p>*</p> <p>Comment: ECON86 is ready for execution. If you are unable to initialize ECON86, initialize and run CONFID test as described in the <i>Series II Installation Manual</i>, order number 9800559-05.</p>

## ECON86 Error Messages

Error messages that may occur when running the ECON86 test with the DEBUG switch turned on (DEBUG=1) are listed in table 4.

**Table 4. ECON86 Test Error Messages**

Test	Error Message	Probable Fault
0	ADDRESS EXPECTED RECEIVED XXXXXH XXH XXH	Offboard RAM, * RPC-86 board
1	Same as Test 0	Offboard RAM, * RPC-86 board
2	Same as Test 0	Offboard RAM*
2	V VARIABLE ERROR	Wrong V variable input
3	Same as Test 0	RPC-86 board
4	Same as Test 0	Offboard RAM*
4	V VARIABLE ERROR	Wrong V variable input
5	Same as Test 0	RPC-86 board
6	Same as Test 0	Offboard RAM*
6	V VARIABLE ERROR	Wrong V variable input
7	Same as Test 0	RPC-86 board
8	Same as Test 0	Offboard RAM, * RPC-86 board
9	Same as Test 0	RPC-86 board
A	COUNTER 1 IS NOT ACCURATE, EXPECTED E8H, RECEIVED XXH	RPC-86 board
B	NO INTERRUPT OCCURRED	RPC-86 board
B	MASK REGISTER FAILURE	RPC-86 board
B	WRONG VECTOR ADDRESS	RPC-86 board
C	SOFTWARE INTERRUPT LEVEL XXH FAILED	RPC-86 board
C	SOFTWARE INTERRUPT ON OVERFLOW OCCURRED	RPC-86 board
C	SINGLE STEP INTERRUPT FAILED	RPC-86 board
D	FAILED	RPC-86 board
E	FAILED	IPC-85 board
F	Same as Test 0	Offboard RAM*
F	V VARIABLE ERROR	Wrong V variable input
10	Same as Test 0	RPC-86 board
11	Same as Test 0	Offboard RAM*
11	V VARIABLE ERROR	Wrong V variable input
12	Same as Test 0	RPC-86 board

\*Page number of defective offboard RAM board is identified by most-significant hexadecimal digit in ADDRESS portion of error message.

```

=>RUN ECON86<cr>
Enhanced Series-III Diagnostic Confidence Test, X004
*
*describe<cr>
0000H Data Bus Ripple Memory Test
0001H Address Bus Ripple Memory Test
0002H Address Memory Test
0003H Refresh Memory Test
0004H March Memory Test
0005H Byte/Word Transfer Test
0006H CPU Test
0007H Timer Count Test
0008H Timer Interrupt Test
0009H Software Interrupt Test
000AH PROM Checksum Test
000BH Failsafe Timer Test
000CH Short Moving Inversion Memory Test
000DH Long Moving Inversion Memory Test
*
*
* **** V variables 0,1,2,3, & 4 determine the amount of memory tested.
* ****
* **** if V(0) = 0 then pages 1,2, & 3 are tested.
* ****           (DEFAULT)
* **** if V(0) = 1 then V(1) specifies starting page ( 1 to 0FH ),
* ****           V(2) specifies starting offset ( 0000H to 0FFFFH,,
* ****           V(3) specifies ending page ( 1 to 0FH)
* ****           V(4) specifies ending offset ( 0000H to 0FFFFH).
* ****           (TEST A SPECIFIED RANGE OF MEMORY)
* **** if V(0) = 2 then starting page = 1,
* ****           starting offset = 0000H,
* ****           V(3) specifies ending page ( 1 to 0FH ),
* ****           V(4) specifies ending offset ( 0000H to 0FFFFH).
* ****           (TEST ALL OF MEMORY FROM PAGE 1 TO SPECIFIED ENDING)
*
*
*test 0 to 0CH<cr>
0000H Data Bus Ripple Memory Test           "PASSED"
0001H Address Bus Ripple Memory Test       "PASSED"
0002H Address Memory Test                   "PASSED"
0003H Refresh Memory Test                   "PASSED"
0004H March Memory Test                     "PASSED"
0005H Byte/Word Transfer Test               "PASSED"
0006H CPU Test                              "PASSED"
0007H Timer Count Test                     "PASSED"
0008H Timer Interrupt Test                  "PASSED"
0009H Software Interrupt Test               "PASSED"
000AH PROM Checksum Test                    "PASSED"
000BH Failsafe Timer Test                   "PASSED"
000CH Short Moving Inversion Memory Test    "PASSED"
*
* **** this test (tests 0 to 0CH) takes ~10 minutes at 8 MHz clock
*
*debug = 1<cr>
*test 0 to 0CH<cr>

```

Figure 4. ECON86 Tests 0 Through C Example

---

```

0000H Data Bus Ripple Memory Test
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
0000H Data Bus Ripple Memory Test           "PASSED"
0001H Address Bus Ripple Memory Test
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
0001H Address Bus Ripple Memory Test       "PASSED"
0002H Address Memory Test
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
  Block 0000:0000, length 2000 words
  Block 0000:4800, length 5000 words
0002H Address Memory Test                   "PASSED"
0003H Refresh Memory Test
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
0003H Refresh Memory Test                   "PASSED"
0004H March Memory Test
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
  Block 0000:0000, length 2000 words
  Block 0000:4800, length 5000 words
0004H March Memory Test                     "PASSED"
0005H Byte/Word Transfer Test
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
  Block 0000:0000, length 2000 words
  Block 0000:4800, length 5000 words
0005H Byte/Word Transfer Test               "PASSED"
0006H CPU Test
0006H CPU Test                             "PASSED"
0007H Timer Count Test
0007H Timer Count Test                     "PASSED"
0008H Timer Interrupt Test
0008H Timer Interrupt Test                 "PASSED"
0009H Software Interrupt Test
0009H Software Interrupt Test             "PASSED"
000AH PROM Checksum Test
000AH PROM Checksum Test                  "PASSED"
000BH Failsafe Timer Test
000BH Failsafe Timer Test                 "PASSED"

```

---

Figure 4. ECON86 Tests 0 Through C Example (Cont'd.)

```

000CH Short Moving Inversion Memory Test
  Block 1000:0000, length 8000 words
    One fill, Backward
  Block 2000:0000, length 8000 words
    One fill, Backward
  Block 3000:0000, length 8000 words
    One fill, Backward
  Block 0000:0000, length 2000 words
  Block 0000:4800, length 5000 words
000CH Shorting Moving Inversion Memory Test                "PASSED"
*
* **** this test (tests 0 to 0CH) also takes ~10 minutes at 8 MHz clock
*
*summary<cr>
0000H Data Bus Ripple Memory Test           0000 FAILED IN 0002 TRIALS
0001H Address Bus Ripple Memory Test       0000 FAILED IN 0002 TRIALS
0002H Address Memory Test                  0000 FAILED IN 0002 TRIALS
0003H Refresh Memory Test                  0000 FAILED IN 0002 TRIALS
0004H March Memory Test                    0000 FAILED IN 0002 TRIALS
0005H Byte/Word Transfer Test              0000 FAILED IN 0002 TRIALS
0006H CPU Test                             0000 FAILED IN 0002 TRIALS
0007H Timer Count Test                     0000 FAILED IN 0002 TRIALS
0008H Timer Interrupt Test                 0000 FAILED IN 0002 TRIALS
0009H Software Interrupt Test              0000 FAILED IN 0002 TRIALS
000AH PROM Checksum Test                   0000 FAILED IN 0002 TRIALS
000BH Failsafe Timer Test                  0000 FAILED IN 0002 TRIALS
000CH Short Moving Inversion Memory Test    0000 FAILED IN 0002 TRIALS
000DH Long Moving Inversion Memory Test    *** IGNORED ***
*
*recognize 0DH<cr>
*test<cr>
0000H Data Bus Ripple Memory Test
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
0000H Data Bus Ripple Memory Test                "PASSED"
0001H Address Bus Ripple Memory Test
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
0001H Address Bus Ripple Memory Test                "PASSED"
0002H Address Memory Test
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
  Block 0000:0000, length 2000 words
  Block 0000:4800, length 5000 words
0002H Address Memory Test                          "PASSED"

```

Figure 4. ECON86 Tests 0 Through C Example (Cont'd.)

---

```

0003H Refresh Memory Test
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
0003H Refresh Memory Test                                     "PASSED"
0004H March Memory Test
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
  Block 0000:0000, length 2000 words
  Block 0000:4800, length 5000 words
0004H March Memory Test                                     "PASSED"
0005H Byte/Word Transfer Test
  Block 1000:0000, length 8000 words
  Block 2000:0000, length 8000 words
  Block 3000:0000, length 8000 words
  Block 0000:0000, length 2000 words
  Block 0000:4800, length 5000 words
0005H Byte/Word Transfer Test                               "PASSED"
0006H CPU Test
0006H CPU Test                                             "PASSED"
0007H Timer Count Test
0007H Timer Count Test                                     "PASSED"
0008H Timer Interrupt Test
0008H Timer Interrupt Test                               "PASSED"
0009H Software Interrupt Test
0009H Software Interrupt Test                           "PASSED"
000AH PROM Checksum Test
000AH PROM Checksum Test                                 "PASSED"
000BH Failsafe Timer Test
000BH Failsafe Timer Test                               "PASSED"
000CH Short Moving Inversion Memory Test
  Block 1000:0000, length 8000 words
    One fill, Backward
  Block 2000:0000, length 8000 words
    One fill, Backward
  Block 3000:0000, length 8000 words
    One fill, Backward
  Block 0000:0000, length 2000 words
  Block 0000:4800, length 5000 words
000CH Short Moving Inversion Memory Test                   "PASSED"
000DH Long Moving Inversion Memory Test
  Block 1000:0000, length 8000 words
    Zero fill, Forward
    One fill, Forward
    Zero fill, Backward
    One fill, Backward
  Block 2000:0000, length 8000 words
    Zero fill, Forward
    One fill, Forward
    Zero fill, Backward
    One fill, Backward

```

---

Figure 4. ECON86 Tests 0 Through C Example (Cont'd.)



```
Block 3000:0000, length 8000 words
  Zero fill, Forward
  One fill, Forward
  Zero fill, Backward
  One fill, Backward
Block 0000:0000, length 2000 words
Block 0000:4800, length 5000 words
000DH Long Moving Inversion Memory Test           "PASSED"
*
* **** this test (tests 0 to 0DH) takes ~45 minutes at 8 MHz clock,
* **** thus 0DH takes ~35 minutes at 8 MHz clock.
*
*list<cr>
*exit<cr>
=>
```

---

Figure 4. ECON86 Tests 0 Through C Example (Cont'd.)

---





## REQUEST FOR READER'S COMMENTS

Intel's Technical Publications Departments attempt to provide publications that meet the needs of all Intel product users. This form lets you participate directly in the publication process. Your comments will help us correct and improve our publications. Please take a few minutes to respond.

Please restrict your comments to the usability, accuracy, readability, organization, and completeness of this publication. If you have any comments on the product that this publication describes, please contact your Intel representative. If you wish to order publications, contact the Intel Literature Department (see page ii of this manual).

1. Please describe any errors you found in this publication (include page number).

---

---

---

---

---

---

2. Does the publication cover the information you expected or required? Please make suggestions for improvement.

---

---

---

---

---

3. Is this the right type of publication for your needs? Is it at the right level? What other types of publications are needed?

---

---

---

---

---

---

4. Did you have any difficulty understanding descriptions or wording? Where?

---

---

---

---

5. Please rate this publication on a scale of 1 to 5 (5 being the best rating).

NAME \_\_\_\_\_ DATE \_\_\_\_\_  
TITLE \_\_\_\_\_  
COMPANY NAME/DEPARTMENT \_\_\_\_\_  
ADDRESS \_\_\_\_\_  
CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP CODE \_\_\_\_\_  
(COUNTRY)

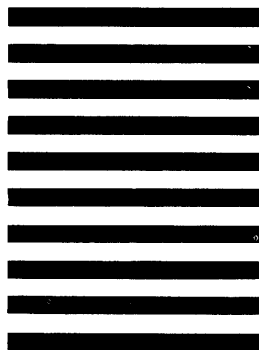
Please check here if you require a written reply.

**WE'D LIKE YOUR COMMENTS ...**

This document is one of a series describing Intel products. Your comments on the back of this form will help us produce better manuals. Each reply will be carefully reviewed by the responsible person. All comments and suggestions become the property of Intel Corporation.



NO POSTAGE  
NECESSARY  
IF MAILED  
IN U.S.A.



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 1040 SANTA CLARA, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation  
Attn: Technical Publications M/S 6-2000  
3065 Bowers Avenue  
Santa Clara, CA 95051

Printed in U.S.A.

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 (408) 987-8080

