

Personality Board For  
4052/4054 Graphic Computing Systems

INTRODUCTION

The 067-0942-99 Personality Board is used with the 067-0902-99 System Test Fixture to troubleshoot 4052 and 4054 circuitry. The Personality Board allows the System Test Fixture to be connected to a 4052 or 4054's ALU (Arithmetic Logic Unit) and MCP (Memory Command Processor) Boards. ROM on the Personality Board defines the functions that the System Test Fixture can perform. The firmware listing for the Personality Board ROM is included in this manual as an aid in troubleshooting the System Test Fixture. The 067-0942-99 comes with an overlay that describes the meaning of LEDs and switches on the System Test Fixture. Unlabeled switches and LEDs are not used with 4052 and 4054 systems.

INSTALLATION

The Personality Board must first be installed in the 067-0902-99 System Test Fixture and then connected to the 4052 or 4054 being tested. If not already installed, mount the Personality Board within the System Test Fixture with screws and spacers. Attach the three ribbon cables between the Personality Board and the Fixture's Controller Board.

On a 4054 the connectors on the ALU and MCP Boards are accessible after removing the 4054's top cover.

For 4052 systems, first remove the 4-board CPU Assembly from the unit. A long 5V power cable (175-2431-00) must be used between the 4052's 5V supply and the board set.

Next, connect the ribbon cables from the Personality Board to the ALU and MCP according to Table 1.

1

2

Table 1

Test Fixture and 4052/4054 Connections

Personality Board Connectors	Connector Color	4052/4054 Connectors
J801	Orange	J203 - ALU Board
J802	Yellow	J204 - ALU Board
J806	Red (2)	J272*- MCP Board
J807	Gray (2)	J264*- MCP Board
J808	Blue	J276 - MCP Board
J508	White (2 molex)	+5V and Ground** - MCP Board

\*Pin one of ten-pin cable connector lines up with pin one of MCP connector and pin one of seven-pin lines up with pin eleven of same MCP connector.

\*\* Molex connector with brown wire connects to ground pins. A different 5V source can be used instead of the system under test.

OPERATION

You can use the System Test Fixture to do several kinds of operations with a 4052 or 4054. In general the fixture can control the ALU Board by forcing the processor to a micro-code address that you key in, by forcing the data you key in onto the ALU-OUT data bus, or by controlling the ALU's clock.

There are some things to keep in mind while using the fixture:

- If a micro-code address is forced on the bus, the current program sequence may be disturbed. The micro-code address you enter must be a proper entry point in order to establish a new micro-code sequence.
- Some micro-code sequences involving I/O or Rompacks must run at normal speed to produce valid results. In order to single step through these sequences use a jumper to connect the pins of J239 (slow ALU) on the MAS Board. Be sure to remove this jumper after testing.
- Switches or LEDs that are not labelled on the System Test Fixture have no function in a 4052 or 4054 system.

LEDs

## ADDRESS SPACE

- DATA LEDs - show whether the ALU data applies to memory space A or B.
- FETCH LEDs - show whether an instruction fetch is from memory space A or B.
- MICROADDRESS LEDs - display the current ALU micro-code address.
- ALU OUT LEDs - display the data from the ALU OUT lines.
- ADDRESS LEDs - display the address in memory that is affected by a READ or WRITE operation using the fixture.

COMP SWITCH

The COMP switch (Break on COMPare) is located beneath the LED labelled COMP. The COMP switch, when on (down) stops the ALU clock if the micro-code address you key in (see COMP LOAD) is reached. If this happens the COMP LED glows. The COMP switch must be on while using the System Test Fixture to read or write in memory.

KEYS

The following keys work by controlling the ALU clock:

- STOP - When you press the STOP key the ALU clock stops on the next trailing edge. The STOP key must be pressed before the other keys on the fixture can be used.
- START - Press the START key to make the ALU clock run. After pressing START, the System Test Fixture does not control the ALU. If COMP is on and you have keyed in a micro-code address, the ALU will stop if that address is reached.

KEYS (cont)

MICROSTEP - Each time you press the MICROSTEP key, the ALU does one micro-code instruction (the ALU clock makes one cycle).

MACROSTEP - Each time you press the MACROSTEP key, the ALU does one complete 6800 type of instruction regardless of the number of micro-code instructions needed.

The following keys work by forcing the ALU to go to a given starting place and to then execute a set of micro-code instructions. The set of instructions performs the function. Some of these keys require you to enter hexadecimal data before performing the function. If you instead press a non-hexadecimal key, the previously started function is stopped and the new one started.

MICRO ADDR - The Force MICRO ADDR key places the three-digit hexadecimal address in micro-code onto the ALU's address bus. After pressing MICRO ADDR, press three hexadecimal keys to enter the address. You can then microstep, macrostep, or start the ALU.

REG - Press the REGister key and another hexadecimal key to display the contents of the selected ALU register. There are 16 internal ALU registers. Register data is displayed on the ALU OUT LEDs.

COMP LOAD - The COMP LOAD key, followed by entering three hexadecimal digits, picks a stopping point in the micro-code. If the comp switch is on when that micro-code address is reached, the ALU clock stops.

ADDR - Press the ADDRess key and four more hexadecimal keys to enter a memory address for a READ or WRITE operation. The ADDRESS LEDs show the address you entered.

READ - To use the READ key, first set the COMP switch to ON, then use ADDR to enter the address you want to read. Press READ to display the contents of the addressed location on the ALU OUT LEDs.

READ NEXT - The COMP switch must be on for READ NEXT to work. Pressing the READ NEXT key displays data on the ALU OUT LEDs from the next higher location in memory as shown in the ADDRESS LEDs.

KEYS (cont)

DATA - Press the DATA key and then four hexadecimal keys to enter data for use with WRITE functions.

WRITE - The WRITE function stores data in a memory location, reads it back, and displays it on the ALU OUT LEDs. To use WRITE, first set the COMP switch to ON, use the DATA key to enter the data you want to store, use ADDR to select the location address, and then press WRITE.

WRITE NEXT - The WRITE NEXT function increments the memory address to the next location, writes data in that location, reads it back, and displays it on the ALU OUT LEDs. To use WRITE NEXT, first set the COMP switch to ON, use the DATA key to enter the data you want to store, and press WRITE NEXT. If you wish to write the same data to successive locations, enter the data, then press WRITE NEXT repeatedly.

RESTART - Pressing the RESTART key causes the system under test to begin its restart routine as in power-up.

INIT - INITIALIZES the System Test Fixture.

FNCTN - The FUNCTION key is used with other keys to select special functions. The following functions are defined:

- FNCTN - WRITE: Writes the data entered using the DATA key to the program counter in the system under test. Enter the data first.
- FNCTN - READ: Reads the program counter into R5 (Register 5) of the ALU and displays it on the ALU OUT LEDs.
- FNCTN - WRITE NEXT: Writes the contents of R5 into the ALU's condition code register.
- FNCTN-READ NEXT: Reads the condition code register into R5 and displays it on the ALU OUT LEDs.
- FNCTN-MICROSTEP: Complete this function by entering the number of micro-code steps to be done. For example the sequence FNCTN MICROSTEP 5E causes the ALU to do 94 micro-code instructions.

5

4052/54 GENERAL FAULT ANALYSIS

The behavior of the instrument at power-on may help find any hardware malfunctions. At power-on, a number of tests are run both by special micro-code and by performing routine system initialization. The normal power-up sequence will turn off all of Front Panel lights except "POWER," will rewind the tape (if one is inserted) to the nearest beginning of a file, and will display a blinking cursor in the upper left corner of the display area. The screen will generally be flooded shortly after power is first applied. It may be necessary to use the PAGE key to erase the screen in order to observe the blinking cursor. If the blinking cursor cannot be observed, then a hardware failure exists.

The state of the front panel lights gives a little information as to the possible fault. If all lights (BUSY, I/O, and BREAK) are on, the failure could be in the ALU, MCP handshake, or I/O. If only BUSY is on, then a RAM memory error is likely. If all lights (except POWER) are off, the error is anywhere--except RAM memory is not the likely problem. ROM failure is a definite possibility. If the POWER light, as well as the other three are OFF, a blown fuse or power supply failure is likely.

If a blinking cursor is observed after power-on, but a failure in a previously successful operation subsequently occurs, then the error could be almost anywhere. If a standard error message is printed, it may be indicative of the failure: (A) Any error message associated with use of the magnetic tape may indicate a failure or needed adjustment of the tape drive or associated circuits; (B) A ROM failure may be the cause of otherwise inexplicable message; (C) A SYSTEM ERROR could be the result of a ROM failure or a firmware bug or of an intermittent RAM failure.

A memory parity error gives a SYS ERROR, "MESSAGE 41" and is non-fatal (program data are left intact). The most usual cause of the other type of SYS ERROR (without any reference to a Message Number or a Line) is an illegal stack entry resulting from a ROM error or firmware bug. This type of error is fatal. There are, at this time, no known firmware bugs which result in a SYS ERROR.

If the system locks up with none of the keyboard's keys (except the shift key's effect on screen hold-mode) operative, the failure could be anywhere. If there is apparent activity--display lights or cursor blinking, etc., then a F/W bug, ROM or RAM error may be responsible. Otherwise, an internal handshake failure could have occurred. A logic probe can be used to determine if CPU CLK is running to verify this possibility.

6

4052/54 FAULT ISOLATION USING THE MICRO SELF TEST

The micro self test is a partial test of the CPU Board assembly (4 boards). It is intended to detect those faults which, if present, prevent the system from executing firmware code such as the power-up routine in system ROM, or the firmware of the Diagnostic Rompack (067-0900-XX). The System Test Fixture can be used to monitor the micro self test and aid in troubleshooting.

The results of the self test can often be interpreted without the aid of any test equipment. There are five LED (Light Emitting Diodes) and one Front Panel display light (BUSY light) which are used by the self test. Inspection of these LEDs (one on the ALU board and four on the MCP board) is possible by removal of the instrument's top cover. The "MICRO SELF TEST (SUMMARY)" flow chart illustrates how these lights are used to indicate passage of the major sections of the self test.

NOTE: Although the major subdivisions of the self test are associated with the various boards of the CPU, it should not be assumed that failure of a given test is necessarily indicative of a failure of the associated board. The failure may involve a signal connecting to some other board.

In some cases, knowing the micro address at which the self test appears to halt will be helpful. The System Test Fixture can display the micro address. Or a logic probe (e.g., Tektronix P6401) can be used to check the micro address lines. These lines are accessible on square pins on the edge of the ALU board closest to the front of the instrument.

The self test starts each time a restart pulse occurs. A restart pulse of several hundred milliseconds is generated when power is applied to the instrument. A restart pulse can also be generated by use of the momentary contact RESTART switch of the Diagnostic Rompack or the System Test Fixture. The RESTART switch of the Diagnostic Rompack is effective independent of the state of the ENABLE/DISABLE switch. Executing the firmware instruction opcode **FF** also causes the self test to be executed.

CAUTION: Repeated power cycling of the instrument should be avoided. Instead, use the RESTART switch of the Diagnostic Rompack or System Test Fixture.

There are two modes of operation of the self test, Normal and Debug mode. In Normal mode, any failure detected by the self test will produce a HALT condition. In the Debug mode, a failing test will, in most cases, be repeated. If, in the case of an intermittent fault, a failing test is subsequently passed, the entire self test will be restarted after finishing the current pass of the self test. Thus, even in Debug mode, an intermittent fault would have to appear good two times in succession before control

SUMMARY

LIGHTS			LIKELY CAUSE
BUSY	BREAK & I/O	POWER	
OFF	OFF	OFF	Power supply failure
ON	ON	ON	ALU, MCP Handshake, I/O
ON	OFF	ON	Memory
OFF	OFF	ON	ALU, MCP, Optional RAM, ROM or I/O, High Voltages or Display.  The high voltage, display board, and some of the I/O can be checked by typing "blind" the statement.
			PRI "G" CR  If the bell rings, the problem is in the DISPLAY function.

8

could pass to any firmware. There are two exceptions to looping on a failing test: 1. The first subtest which checks the ability of the ALU to run micro tests, and 2. the RAM quick-check. These two tests always halt when an error is detected, independent of the Normal/Debug mode.

Debug mode is established using a jumper wire to ground the TESTNRP-0 signal at J204-8.

#### LOCATION OF LIGHTS & LEDs

HALF-CARRY LED  
(DS190, ALU BOARD)

This LED is located on the right-hand side of the instrument on the side edge of the ALU board. It is the rear most of the two LEDs located there.

OVERFLOW LED  
(DS195, ALU BOARD)

The forward LED, DS195 displays the state of the overflow bit of the condition code register.

PROGRAM COUNTER LEDs

The most significant (high-order) 4 bits of the program counter are displayed in 4 LEDs located on the left-side of the instrument on the side edge of the MCP board. The most significant bit is D5 located closest to the front of the instrument.

MEMORY PARITY ERROR

This LED is located on the left-side of the instrument on the side of the MAS BOARD.

BUSY  
BREAK  
I/O  
POWER

These lights are located just to the right of the CRT face plate.

#### ACTION OF LIGHTS & LEDs DURING SELF TEST

The table on the next page shows how to interpret the results of the self test through various stages.

9

Half Carry DS190

Prog. Counter DS4, DS3, DS2, DS1

Busy

Break, I/O, & Power

Comments

During, and immediately after restart.  
Test is checking ALU.

1 X X X X 1 1 1 1  
1 X X X X 1 1 1 1  
1 X X X X 1 1 1 1  
1 0 0 0 0 1 1 1 1  
1 0 0 0 1 1 1 1  
1 0 0 1 0 1 1 1 1  
1 0 1 0 0 1 1 1 1  
1 1 0 0 0 1 1 1 1  
1 1 1 1 1 1 1 1 1  
1 0 0 0 0 1 1 1 1

Any change in the Program Counter LEDs since RESTART indicates that ALU portion of the SELF TEST is successful.

1 0 0 0 0 1 1 1 1  
1 0 0 0 1 1 1 1  
1 0 0 1 0 1 1 1 1  
1 0 1 0 0 1 1 1 1  
1 1 1 1 1 1 1 1 1  
1 0 0 0 0 1 1 1 1

During the MCP test, this sequence will appear in the program counter as a means of testing each bit of the 16 bit data path out of and into the ALU. The last two patterns occur while testing the count operation of FFFF+1 → 0000.

If the above tests are passed, then the Program Counter LEDs are used to indicate the next set of tests.

In each of the following five tests, failure at a given test is indication that all preceding tests were passed.

1 0 0 1 1 1 1 1 1

Means PC test passed. If halted (NO CPCLK), it means that there was a failure in ALU-MCP-MAS handshake during a write operation to RAM memory.

1 0 1 0 1 1 1 1 1

Ram write handshake works. If halted, read RAM handshake has failed.

1 0 1 1 0 1 1 1 1

RAM read handshake works. If halted, there was a handshake failure during a write at an I/O address.

1 0 1 1 1 1 1 1 1

Write I/O handshake works. If halted, read I/O handshake failure has occurred.

1 1 0 0 1 1 1 1 1

Read I/O handshake works. If halted, then the I/O data validity check has failed.

1 1 0 0 1 0 1 1 1

The busy light is turned off if the diagnostic Rompack is not entered and all tests so far have been passed.

The parity error is cleared and disabled, and quick check of memory begins. If this quick check fails, it halts at 0C5. Otherwise, control is passed to the power up firmware of BASIC.

1 X X X X 1 0 0 1

BASIC does a complete check of memory. If failure is detected, it will never go on to its initialization phase.

1 X X X X 0 0 0 1

BASIC's initialization in progress or completed and is blinking the cursor. Activity in some of the LEDs (indicated by "B"), including the overflow LED, DS190, at the same rate as the cursor blink can be observed.

BB 1 1 B 0 0 0 1

LEGEND: 0= LIGHT IS OFF

1= LIGHT IS ILLUMINATED

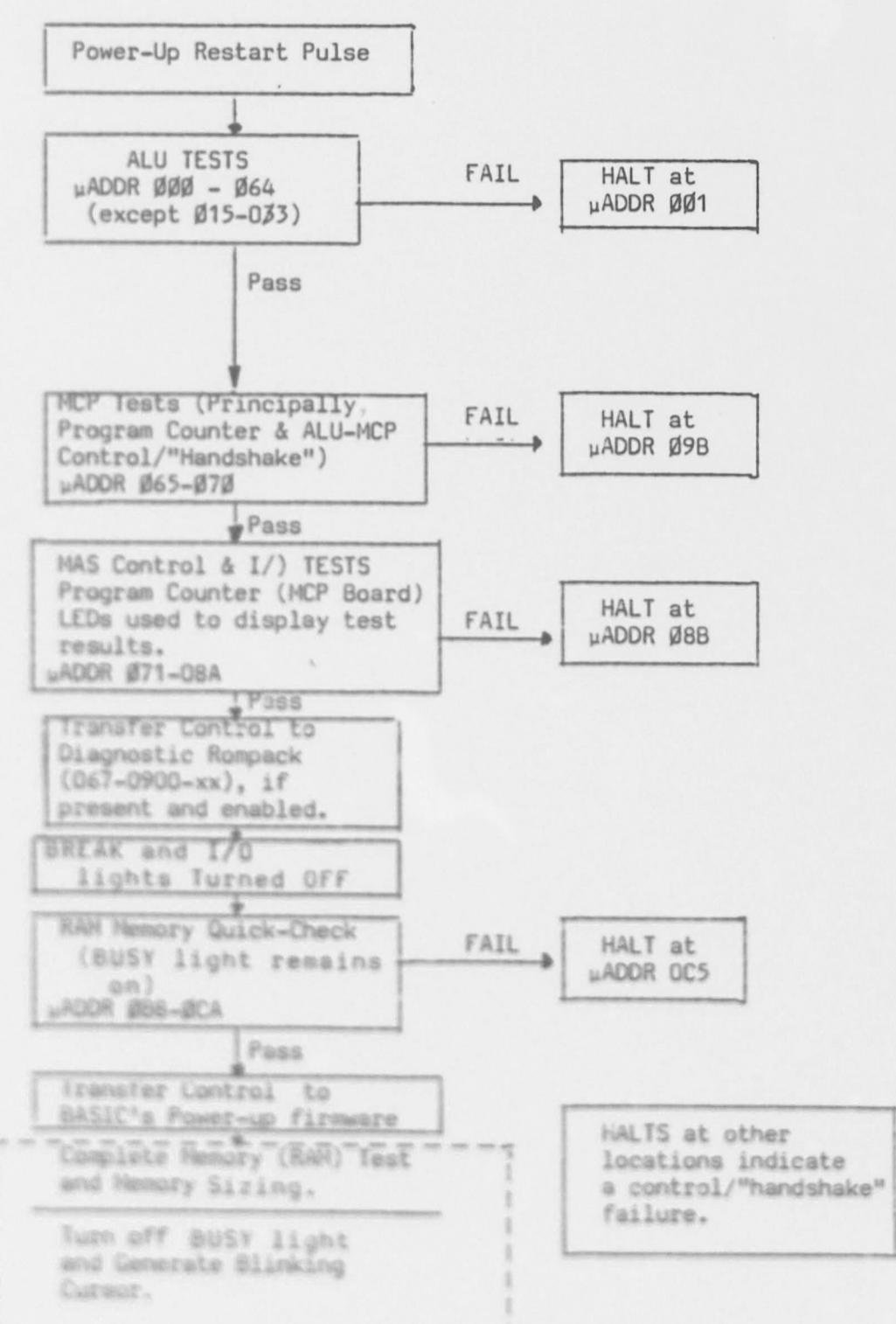
X= CONDITION OF LIGHT IS UNPREDICTABLE

OR IS RAPIDLY CHANGING

B= OBSERVABLE BLINKING ACTIVITY

10

#### MICRO SELF TEST (SUMMARY)



#### 4052/54 MICRO-CODE LISTING

The following pages give the micro-code contained in ROM on the ALU Board. The listing shows the micro-code address and ROM data according to the following format:

Line # of code listing
Micro-code address
U330 (MSB=pin 17)
U315
U300
U340
U335
U320
U305
Label used in comments

1 000 80 20 00 98 50 D1 DD \*Restart

```

        *
        *
        *
        *
        * All comments refer to the preceding line of code. When ever they
        * follow a heading, they refer to the same block as
        * the regular heading refers to.
        *
1 000 80 20 00 99 D0 01 DD#RESTART:
        *+ R13=0 indicates that no test has failed.
        *+ A branch is attempted to location 002. If it fails to branch and
        *+ executes the HALT at location 001, then a severe problem in the branch
        *+ control or sequencers of the ALU board is presumed to exist. The
        *+ half-carry bit is set and all other condition code bits except
        *+ I (interrupt mask) are cleared. "I" is whatever state the
        *+ condition code latch powers up in.
        *
        *
2 001 80 10 00 9F C0 47 00#ALUBAD:
        *+ HALT, GOTO SELF. The halt is accomplished by specifying to the ALU
        *+ that it is to wait for [MCPI] input, but no control signals are
        *+ generated to solicit any input. Just in case that doesn't work, this
        *+ statement also specifies GU TO SELW, which would have about the same
        *+ effect except that the CPUCLK would keep running.
        *+ HALTING here means its about 95% certain that the ALU board is bad.
        *+ if it HALTS here when the DEBUG JUMPER is connected, it means that
        *+ the ALU is so bad that it cannot even check itself reliably.
        *
        *
3 002 80 00 84 74 R0 40 00#OK1:
        *+ 1 -> CARRY. Tries to turn on the CARRY bit.
        *
4 003 80 10 01 9F C0 40 00:
        *+ IF CARRY (still) CLEAR THEN GO TO 001 (and halt).
        *
5 004 80 00 86 74 R0 40 00:
        *+ 0 -> CARRY. Tries to clear the CARRY bit.
        *
6 005 80 10 01 1F C0 40 00:
        *+ IF CARRY (still) SET THEN GO TO 001 (and halt).
        *
7 006 80 00 84 73 C0 40 00:
        *+ 1-> ZERO. Try to set the ZERO bit.
        *
8 007 80 10 08 9F C0 40 00:
        *+ If ZERO is not set, goto 001 (and HALT).
        *
9 008 80 00 80 73 C0 40 00:
        *+ 0->ZERO. Try to clear the ZEROP bit.
        *
10 009 80 10 08 1F C0 40 00:
        *+ If ZERO bit is set, goto 001 (and HALT).
        *
11 00A 80 00 84 74 D0 40 00:
        *+ 1->N (Negative). Set the N bit.

```

```

12 00B 80 10 02 9F C0 40 00:
        *+ If N is not set, goto to 001 (and HALT).
13 00C 80 00 80 74 C0 40 00:
        *+ 0->N. Try to clear the N bit.
14 00D 80 10 02 1F C0 40 00:
        *+ If the N bit is set, goto 001 (and halt).
15 00E 83 60 00 9F C0 40 00:
        *+ Goto to 036 to skip over the interrupt vectors and the system
        *+ test fixture code.
        *
        *
        * The following 16 branches perform priority encoding and
        * dispatching for interrupts. The alu forces a microbranch
        * to address O1X on interrupts, where 'X' is the logical 'OR' of all
        * pending interrupt conditions. A weight of 1 is given to
        * memory parity errors, a weight of 2 is given to NMI, a weight
        * of 4 to IRQ, and a spare, unused interrupt is given a weight
        * of 8. When several interrupts are pending a branch to the highest
        * priority interrupt is taken.
        *
        *
16 010 8C E0 00 9F C0 40 00:
        *+ X is 0, there are no interrupts, and we should not have gotten here,
        *+ so goto the invalid instruction trap.
17 011 90 80 00 9F C0 40 00:
        *+ X is 1, it's a parity error alone, goto to parity error trap.
18 012 8F 40 18 9F C0 40 00:
        *+ X is 2, NMI only, so goto NMI routine.
19 013 90 80 00 9F C0 40 00:
        *+ X is 3, parity error and NMI together, so take the higher-priority
        *+ parity error and goto the parity error routine.
20 014 8F 20 00 9F C0 40 00:
        *+ X is 4, IRQ alone, so goto the IRQ routine.
21 015 90 80 00 9F C0 40 00:
        *+ X is 5, IRQ and parity error, parity wins.
22 016 8F 40 18 9F C0 40 00:
        *+ X is 6, IRQ and NMI, NMI wins.
23 017 90 80 00 9F C0 40 00:
        *+ X is 7, parity error, NMI and IRQ=parity wins.
24 018 90 90 00 9F C0 40 00:
        *+ X is 8, the unused interrupt has been used, goto DSPINT to
        *+ dispose of the unused interrupt.
25 019 90 80 00 9F C0 40 00:
        *+ In the next eight instructions, the unused interrupt is
        *+ always present, but it is lower priority than any of the
        *+ other interrupts, and will never win here, so it will not
        *+ be mentioned again.
        *
        * Parity error and you-know-what, parity error wins.
26 01A 8F 40 18 9F C0 40 00:
        *+ NMI and um-um, take NMI.
27 01B 90 80 00 9F C0 40 00:
        *+ NMI and parity, take parity.
28 01C 8F 20 00 9F C0 40 00:
        *+ IRQ only, take it.

```

```

29 01D 90 60 00 9F C0 40 00:
        *+ IRQ and parity, parity always wins!
30 01E 8F 40 18 9F C0 40 00:
        *+ IRQ and NMI, NMI wins.
31 01F 90 80 00 9F C0 40 00:
        *+ X=15. IRQ, NMI, and Parity, parity wins.
        *
        *
        * Display registers for system test fixture
        *
32 020 82 00 00 9F C0 44 00#RDPU1
        *+ PASS PEGISTER R0 TO ALU OUT AND BRANCH TO SELF
33 021 82 10 00 9F C0 44 01#RDPU1:
        *+ PASS R1 TO ALU OUT, ETC.
34 022 82 20 00 9F C0 44 02#RDPU2:
        *+ PASS R2 ETC
35 023 82 30 00 9F C0 44 03#RDPU3:
        *+ PASS R3
36 024 82 40 00 9F C0 44 04#RDPU4:
        *+ PASS R4
37 025 82 50 00 9F C0 44 05#RDPU5:
        *+ PASS R5
38 026 82 60 00 9F C0 44 06#RDPU6:
        *+ PASS R6
39 027 82 70 00 9F C0 44 07#RDPU7:
        *+ PASS R7
40 028 82 80 00 9F C0 44 08#RDPU8:
        *+ PASS R8
41 029 82 90 00 9F C0 44 09#RDPU9:
        *+ PASS R9
42 02A 82 A0 00 9F C0 44 0A#RDPU10:
        *+ PASS R10
43 02B 82 B0 00 9F C0 44 0B#RDPU11:
        *+ PASS R11
44 02C 82 C0 00 9F C0 44 0C#RDPU12:
        *+ PASS R12
45 02D 82 D0 00 9F C0 44 0D#RDPU13:
        *+ PASS R13
46 02E 82 E0 00 9F C0 44 0E#RDPU14:
        *+ PASS R14
47 02F 82 F0 00 9F C0 44 0F#RDPU15:
        *+ PASS R15
        *
        * Next two instructions are used by the system test fixture
        * to write a memory location. The microcode specifies
        * R0 to memory address and R0 to (or from) memory data but the
        * test fixture disables the alu outputs and substitutes its
        * own values for data and address.
        *
48 030 82 10 00 9F C0 44#RDNEXT1:
        *+ R0 to write address
49 031 82 00 00 9F C0 44#RDNEXT2:
        *+ R0 to memory data

```

```

50 032 83 30 40 9F CC 44 00#READ1:
        *+ R0 to memory address
51 033 00 00 00 77 CA 47 00#RDNEXT2:
        *+ data from memory
        *
        * Continue general alu arithmetic diagnostics
        *
52 034 80 10 38 1F C0 40 00#GA2:
        *+ ...which sets R13 to -1 to indicate that at least one failure
        *+ was detected. Then fall through to repeat these tests.
53 035 83 60 00 9F C0 D1 DD#:
        *+ ...
54 036 83 70 00 9B 50 D1 00#GA0:
        *+ Start here. Clear R0 by subtracting it from itself. Set condition codes.
55 037 R3 40 08 98 40 CC 00#GA1:
        *+ System test fixture can be used to start this test here.
        *
56 038 83 40 01 1F C0 40 00:
        *+ If the 'zero' condition is not true, go to GA2 (error).
57 039 83 40 02 9F C0 40 00:
        *+ Subtract 1 from R0 (result should be negative) and set cond. codes.
58 03A 83 40 08 12 50 C4 00:
        *+ Carry should be cleared, if not go to GA2 (error).
59 03B 83 40 08 9F C0 40 00:
        *+ If not negative result, goto to GA2 (error).
60 03C 83 40 02 1F C0 40 00:
        *+ If 'zero' is set goto GA2 (error).
61 03D 83 40 01 9F C0 40 00:
        *+ Add 1 back to R0 and set condition codes.
62 03E 83 F0 00 9F C0 40 00#SHORT0:
        *+ If result is not zero now, goto GA2 (error).
63 03F AA AA 80 77 C0 C7 00#SHORT1:
        *+ If result is negative, goto GA2 (error)
64 040 55 55 80 77 C0 C7 10#:
        *+ If carry is not set, goto GA2 (error).
        *
        * The following tests check for stuck or lin-to-line shorts
        * on ALU in.
65 041 80 00 00 77 C0 C1 11#:
        *+ Load an alternate I-O pattern in R0 and its complement
        * in R1.
66 042 80 00 00 72 00 71 01#:
        *+ Double R1 to shift it left.
67 043 84 80 08 1F C0 40 00#:
        *+ EOR R1 and R0 and set cond. codes. Result should be zero.
68 044 80 10 38 1F C0 40 00#:
        *+ If result is zero, goto to register tests.
69 045 83 F0 00 9F C0 D1 DD#:
        *+ Error was detected. If debug jumper is not on, goto ALUBAD.

```

```

    o:+ Otherwise, set R13 to -1 and repeat the test.
    o
    o
70 046 80 10 38 1F C0 40 000REGBAD:
    o:+ Register test failures come here. If debug jumper is on
    o:+ repeat the test, otherwise, goto ALUBAD.
71 047 84 80 00 9F C0 D1 DDe
    o
72 048 88 C0 00 8F C0 40 000REG1:
    o:+ Call a subroutine to put 0000 in R0, 1111 in R1, 2222 in R2,
    o:+ 4444 in R4, 8888 in R8, and FFFF in all other registers (except R13).
73 049 80 00 00 77 C0 09 100
    o:+ EUR R0 and R1.
74 04A 11 11 80 72 00 75 010
    o:+ Result should be 1111.
75 04B 84 60 00 9F C0 40 000
    o:+ If not, goto REGBAD above.
    o:+ Perform the test with R1 and R2, R2 and R4,
    o:+ R4 and R8, and R8 and R0. Any failures indicate
    o:+ a possible short in the register A&B lines of the
    o:+ 2901, a bad 2901, or a bad microcode bit.
    o
76 04C 88 C0 00 8F C0 40 000
77 04D 80 C0 00 77 C0 09 210
78 04E 33 33 80 72 00 75 020
79 04F 84 60 00 9F C0 40 000
    o
80 050 88 C0 00 AF C0 40 000
81 051 80 00 00 77 C0 D9 420
82 052 66 66 80 72 00 75 040
83 053 84 60 00 9F C0 40 000
    o
84 054 88 C0 00 8F C0 40 000
85 055 80 00 00 77 C0 09 840
86 056 CC CC 80 72 00 75 080
87 057 84 60 00 9F C0 40 000
    o
88 058 88 C0 00 8F C0 40 000
89 059 80 00 00 77 C0 09 080
90 05A 88 88 80 72 00 75 000
91 05B 84 60 00 9F C0 40 000
    o
    o
    o:+ The "P" (condition code) register is outside the 2901's.
    o:+ Test the ALU's ability to communicate with it.
    o
92 05C 85 D0 00 9E 90 51 550P92:
    o:+ Generate a 0 (RS=RS) and stuff it in P.
    o:+ Test fixture can start here.
    o
93 05D 00 00 81 72 C0 47 000P11:
    o:+ Pass P back to the 2901's and set the zero bit
    o:+ according to the result.
    o
94 05E 86 30 08 9F C0 40 000
    o:+ Go to error: if the zero condition is not set now.
    o
95 05F 80 00 84 74 C0 40 000
    o:+ Force the int. sk on.

```

```

    o:+
    o
    o
96 060 00 00 81 77 C0 C7 500
    o:+ Get P back in RS.
    o
97 061 00 14 80 72 C0 75 050
    o:+ XOR RS with hex 14 and set condition codes.
    o
98 062 86 50 00 1F C0 40 000
    o:+ If RS was 14 (result of XOR is zero) go to next test (PC).
    o
99 063 80 10 38 1F C0 40 000PBA0:
    o:+ Error was detected. If debug jumper is not on, goto ALUBAD.
    o
100 064 85 C0 00 9F C0 D1 DDe
    o:+ Otherwise, repeat the test.
    o
    o
    o:+ ALU tests have determined that the ALU is reasonably free of
    o:+ faults. Testing of the rest of the boards begins.
    o
101 065 86 60 04 9D 00 40 000PCTEST:
    o:+ Begin test of PC on MCP. Dummy goto next statement so
    o:+ test fixture can start the test here.
    o
102 066 00 01 80 74 40 C7 500PC0:
    o:+ Load a single bit into RS.
    o
103 067 80 00 00 77 C6 44 050PC1:
    o:+ Transfer RS to the PC.
    o
104 068 00 00 00 72 04 75 050
    o:+ EUR RS and PC. Result should be zero.
    o
105 069 86 P0 08 9B 40 C1 550
    o:+ If the result is not zero, goto PCBAD, Shift RS left
    o:+ to ripple the single bit through all 16 PC bits.
    o
106 06A 86 70 01 9F C0 40 000
    o:+ Go repeat the test with the next bit if no bit was
    o:+ shifted into the carry bit.
    o
107 06B FF FF 80 77 C6 47 000
    o:+ Continue PC test. Load -1 (all ones) into PC.
    o
108 06C 80 00 00 77 C2 47 000
    o:+ Increment the PC by 1 via the increment circuitry
    o:+ on the MCP.
    o
109 06D 00 00 00 72 04 47 000
    o:+ Read the PC back. It should be zero now.
    o
110 06E 87 20 08 1F C0 40 000
    o:+ Is PC is now zero, go to memory tests.
    o
111 06F 89 80 38 1F C0 40 000PCBAD:
    o:+ PC increment or load error. If debug jumper is on,
    o:+ repeat the test otherwise goto MPCBAD.
    o
112 070 86 60 00 9F C0 D1 DDe
    o:+ Any hangup problems in the above tests indicate an ALU-MCP
    o:+ handshake problem.
    o
    o
    o:+ Begin test of memory function.
    o
113 071 07 20 00 9F C0 40 000
    o:+ Dummy goto next statement for system test panel start.
    o
114 072 30 00 80 77 C6 47 000MEMHRT:

```

```

    o:+ Store 3 in top 4 bits of PC (LED's on MCP board). PC is
    o:+ determined by preceding tests to be reliable enough to
    o:+ use as progress indicator for the following tests. Note
    o:+ value in PC LED's if a hang occurs because of ALU-MCP-MAS
    o:+ handshake problems.
    o
115 073 00 01 80 A7 00 47 000PUMP:
116 074 40 90 C0 77 CE 47 000
117 075 00 00 40 6F C1 47 000
    o:+ The above three instructions are a loop executed 15 times.
    o:+ Data of zeroes is written to memory address 4000 to pump
    o:+ up internal bias voltages in memory chips. handshake problems
    o:+ may cause the loop to hang.
    o
118 076 87 70 00 4F C0 40 000
    o:+ Dummy goto for system test fixture startup.
    o
119 077 50 00 80 77 C6 47 000MEMRD:
    o:+ Put 5 in PC LED's to show progress.
    o
120 078 40 00 00 77 CC 47 000
    o:+ Put 4000 to memory write address.
    o
121 079 00 00 00 77 C1 47 000
    o:+ Read memory.
    o
122 07A 50 00 80 77 C6 47 000
    o:+ Put 6 to PC to indicate previous read did not hang.
    o
    o
    o:+ Begin tests of I/O
    o
123 07B 67 C0 00 9F C0 40 000
    o:+ Dummy goto next statement for system test fixture.
    o:+ Test I/O by rippling a single 1 bit thru the data
    o:+ direction register of a PIA.
    o
124 07C FF 0F 80 77 CE 47 000I0J0:
    o:+ Clear the control register of PIA at FF0F
    o
125 07D 00 00 80 77 C8 47 000
126 07E 00 01 80 77 C0 C7 000
    o:+ Initialize the loop with a single '1' bit in R0.
    o
127 07F FF 0F 80 77 CE 47 000I011:
128 080 80 00 00 77 C8 44 000
    o:+ Write R0 to the data direction register.
    o
129 081 70 00 80 77 C6 47 000
    o:+ Put 7 in the PC LED's to show we haven't hung yet.
    o
130 082 FF 0E 80 77 CC 47 000
131 083 00 00 00 72 C0 75 000
    o:+ Read the PIA data direction bank and 'OR' it with RS
    o
132 084 90 00 80 77 C6 47 000
    o:+ Put 9 in PC LED's.
    o
133 085 88 90 08 91 C0 40 000
    o:+ If the PIA and RS didn't match, goto IUBAD.
    o
134 086 80 00 00 72 C0 C1 000
    o:+ Shift RS, the test bit, left one place.
    o
135 087 87 F0 08 9F C0 40 000
    o:+ If there's still a bit in RS (lower 8 bits) continue the test.
    o
136 088 89 C0 00 9F C0 40 000
    o:+ Test passed, goto DIAGCOUNT to skip over some stuff.
    o
137 089 88 80 38 1F C0 40 000IDBAD:
    o:+ Goto IDBAD1 if debug jumper not connected.

```

```

    o:+
    o
    o
138 08A 87 C0 00 9F C0 D1 DDe
    o:+ Otherwise, repeat the I/O test.
    o
139 08B 08 80 00 9F C0 47 000IDBAD1:
    o:+ Hang here on I/O errors. See PC LED's to determine
    o:+ how far the test got before failing.
    o
    o
    o:+ The following lines are a subroutine used to put
    o:+ standard values in the 2901 registers for the register
    o:+ test described above.
    o
140 08C 00 00 80 77 C0 C7 000SETREG:
141 08D 11 11 80 77 C0 C7 100
142 08E 22 22 80 77 C0 C7 200
143 08F 44 44 80 77 C0 C7 400
144 090 88 88 80 77 C0 C7 800
    o
145 091 FF FF 80 77 C0 C7 300
146 092 80 00 00 77 C0 C4 530
147 093 80 00 00 77 C0 C4 630
148 094 80 00 00 77 C0 C4 730
149 095 80 00 00 77 C0 C4 930
150 096 80 00 00 77 C0 C4 A30
151 097 80 00 00 77 C0 C4 B30
152 098 80 00 00 77 C0 C4 C30
153 099 80 00 00 77 C0 C4 E30
154 09A 80 00 00 07 C0 C4 F30
    o:+ End of subroutine
    o
    o
155 09B 09 80 00 9F C0 47 000MCPBAD:
    o:+ Hang here on MCP failures that are not caused by
    o:+ handshake problems.
    o
    o
156 09C FF 41 80 77 CE 47 000DIAGCONT:
    o:+ Initialize the bank switch PIA (which may not be a PIA)
    o:+ and clear SIS (the debug interrupt vector switch).
    o
157 09D 00 00 80 77 C0 C7 F00
158 09E FF 40 80 77 CE 47 000
159 09F 38 04 80 77 C0 47 000
160 0A0 FF 40 80 77 CE 47 000
161 0A1 00 20 80 77 C0 47 000
    o:+ Set the bank switch to the leftmost slot
    o
162 0A2 FF 00 80 77 CE 47 000
163 0A3 00 00 80 77 C0 47 000
164 0A4 FF 0C C0 77 CE 47 000

```

```

165 0A5 00 04 80 77 CA 47 00*          *;+ Initialize the PIA used for the 4052/4054 identity
166 0A6 00 00 00 76 00 40 00*          *;+ determination.
167 0A7 00 0C C0 77 CC 47 00*          *
168 0A8 00 00 00 77 CA C7 00*          *
169 0A9 0B 52 80 72 00 75 00*          *;+ Look for "DB52" in location C of the rompack
170 0AA 0B 06 08 9F C0 40 00*          *;+ to see if it's the diagnostic rompack.
171 0AB 00 F0 80 77 C0 E5 00*          *;+ Go to the regular restart at NODEBUG1 if diagnostic
172 0AC 00 0E C0 77 CC 47 00*          *;+ rompack is not in place.
173 0AD 00 00 00 77 CA C7 00*          *
174 0AE 00 F0 80 76 00 47 00*          *
175 0AF 00 00 00 7F C0 44 00*          *;+ Set accumulator A to "F0" if any errors occurred.
176 0B0 00 00 00 72 00 44 0D*          *;+ (As noted in R13).
177 0B1 00 00 00 0F C0 40 00*          *;+ Got the entry point to the diagnostic rompack
178 0B2 0F 2b 80 77 C0 47 00*          *;+ from location 000E and go there.
179 0B3 00 00 00 77 C0 47 00*          *
180 0B4 0F 2A C0 77 CE 47 00*          *
181 0B5 F0 04 80 77 CA 47 00*          *
182 0B6 0F 2A 80 77 CE 47 00*          *
183 0B7 00 00 00 77 C0 47 00*          *;+ Turn off the busy light and begin a more complete test
184 0B8 0A 90 00 0F 16 D1 55*          *;+ of the basic (32K) ram.
185 0B9 00 00 00 77 DE 44 05*          *;+ This is the regular (no diagnostic rompack)
186 0BA 00 02 00 77 CA 05 55*          *;+ restart point. Test R13 to see if any error occurred during
187 0BB 7F FE 80 72 00 75 05*          *;+ microdiagnostics.
188 0BC 0F 90 00 0F C0 40 00*          *;+ If any errors were found, go to RESTART and repeat
189 0BD 0B 80 00 0F D0 D1 55*          *;+ all test.
190 0BE 00 00 00 77 DC 44 05*          *;+ HEM1:
191 0BF 00 00 00 72 0A F5 65*          *;+ Read the locations and compare to what should
192 0C0 0C 60 00 1F C0 40 00*          *;+ be stored.
193 0C1 00 00 00 77 D6 44 05*          *;+ Go to HL... : no error is found.
194 0C2 00 00 00 77 C6 44 00*          *;+ Write out failing address to PC
195 0C3 00 00 00 77 C0 D1 DD*          *;+ Write failing bits to PC (bad bits are ones).
196 0C4 0C 60 38 9F C0 40 00*          *;+ Put -1 in R13 to indicate at least one error occurred.
197 0C5 0C 50 00 9F C0 47 00*          *;+ If the debug jumper is in place, continue
198 0C6 00 03 00 77 C0 C5 55*          *;+ with the test.
199 0C7 7F FE 80 72 00 75 05*          *
200 0C8 0B 80 00 0F C0 40 00*          *;+ Increment the address, test for end, and go back
201 0C9 00 00 00 72 00 44 00*          *;+ to HEM2 if not finished.
202 0CA 0C 50 00 9F C0 40 00*          *
203 0CB FE C4 74 CC 47 00*          *;+ Test R13 to see if any errors occurred. If so, goto HEM3.
204 0CC 00 00 00 77 4A C7 40*          *
205 0CD 00 00 00 7F C0 44 00*          *;+ Fetch the firmware restart address from FEFE and start BASIC.
206 0CE 10 D0 81 9F C0 C7 40*          *;+TRAP:
207 0CF 00 00 00 7F C0 C7 00*          *;+PULACONT:
208 0D0 00 00 00 7F C0 C7 10*          *;+PULACONT:
209 0D1 00 00 00 7F C0 C7 44 00*          *;+PSHACONT:
210 0D2 00 00 00 7F C0 C7 44 01*          *;+PSHBCONT:
211 0D3 00 00 00 77 CA C7 40*          *;+RTSCONT:
212 0D4 00 00 00 7F D6 83 34*          *
213 0D5 00 00 00 77 CC 44 04*          *;+HEGCONT:
214 0D6 00 00 00 70 9C CF 50*          *
215 0D7 00 00 00 77 C0 44 04*          *;+STUFFIT:
216 0D8 00 00 00 7A C0 44 05*          *
217 0D9 00 00 00 77 CC 44 04*          *;+COMCONT:
218 0DA 0D 70 00 99 08 FF 50*          *
219 0DB 00 00 00 77 CC 44 04*          *;+DECCONT:
220 0DC 0D 70 00 98 C0 D7 50*          *
221 0DD 00 00 00 77 CC 44 04*          *;+INCCONT:
222 0DE 0D 70 00 98 D8 C7 50*          *
223 0DF 00 00 00 77 CC 44 04*          *;+LSRCONT:
224 0EO 0D 70 00 98 09 47 50*          *
225 0E1 00 00 00 77 CC 44 04*          *;+RORCONT:

```

22

21

```

226 0E2 0D 71 00 9A 89 47 50*          *;+ASHCONT:
227 0E3 00 00 00 77 CC 44 04*          *;+ASHCONT:
228 0E4 00 00 00 70 88 C7 50*          *
229 0E5 0D 72 00 49 81 44 55*          *
230 0E6 00 00 00 77 CC 44 04*          *;+ASLCONT:
231 0E7 00 00 00 77 C0 C7 50*          *
232 0E8 0D 76 00 98 80 C1 55*          *
233 0E9 00 00 00 77 CC 44 04*          *;+RULCONT:
234 0EA 00 00 00 77 C0 C7 50*          *
235 0EB 0D 70 00 98 A0 C1 55*          *
236 0EC 00 00 00 76 88 47 00*          *;+TSTCONT:
237 0ED 00 00 00 79 58 51 00*          *;+CLKACONT:
238 0EE 00 02 00 73 A1 44 00*          *;+ASRACONT:
239 0EF 00 00 00 7A C0 44 00*          *;+SETZA:
240 0F0 00 02 00 73 A1 44 11*          *;+ASPCONT:
241 0F1 00 00 00 7A C0 44 01*          *;+SETZB:
242 0F2 FE F9 80 77 C0 C7 50*          *;+IMU:
243 0F3 0F 60 00 0F C0 40 00*          *
244 0F4 FE FC 00 77 C0 C7 50*          *;+HWINT:
245 0F5 00 00 00 83 77 C0 40 00*          *;+CLINT:
246 0F6 00 00 00 61 77 00 C7 00*          *;+STACK:
247 0F7 00 00 00 84 74 D0 40 03*          *
248 0F8 00 00 00 77 C0 44 06*          *
249 0F9 00 00 00 77 C0 C7 40*          *
250 0FA 00 00 00 77 C0 44 05*          *
251 0FB 00 00 00 77 C0 C7 70*          *
252 0FC 00 00 00 77 C0 C7 03*          *
253 0FD 00 00 00 77 C0 44 04*          *
254 0FE 00 00 00 77 C0 44 03*          *
255 0FF 00 00 00 77 C0 44 02*          *
256 100 00 04 80 77 DE 40 03*          *
257 101 00 00 00 77 C0 44 00*          *
258 102 00 05 80 77 DE 40 03*          *
259 103 00 00 00 77 C0 44 01*          *
260 104 00 07 80 77 D0 CD 33*          *
261 105 00 00 00 73 40 44 07*          *
262 106 91 A0 02 1F C0 40 00*          *
263 107 00 00 00 76 80 50 00*          *
264 108 8J 00 00 7F C0 44 07*          *
265 109 FE F4 80 77 C0 C7 50*          *;+OSPIINT:
266 10A 0F 60 00 0F C0 40 00*          *
267 10B FE F2 80 77 C0 C7 50*          *;+HWFAIL:
268 10C 0F 50 00 0F C0 40 00*          *
269 10D FE FA 80 77 C0 C7 50*          *;+SWICONT:
270 10E 0F 60 00 0F C0 40 00*          *
271 10F 00 00 00 77 C0 C7 50*          *;+RTICONT:
272 110 00 00 00 77 DC C4 33*          *
273 111 00 00 00 77 C0 C7 10*          *
274 112 00 00 00 77 DC C4 33*          *
275 113 00 00 00 77 C0 C7 00*          *
276 114 00 00 00 77 DC C4 33*          *

```

```

277 115 00 00 00 77 C0 C7 20*          *
278 116 00 02 C0 77 CC C9 23*          *
279 117 00 00 00 77 C0 C7 00*          *
280 118 00 00 00 77 C0 C7 00*          *
281 119 00 00 00 77 C0 C7 00*          *
282 11A FE 40 80 77 C0 47 00*          *;+ALTVECT:
283 11B 30 20 80 77 C0 47 00*          *
284 11C 00 02 00 76 92 40 00*          *
285 11D FE F2 C0 77 DC 40 00*          *
286 11E 00 00 00 77 C0 C7 00*          *
287 11F 00 00 00 76 80 50 00*          *
288 120 00 00 00 7F C0 44 00*          *
289 121 00 00 00 77 21 C0 44 00*          *;+DSRCONT:
290 122 00 00 00 77 21 C0 44 00*          *;+DSRCONT:
291 123 00 00 00 77 21 C0 44 00*          *
292 124 00 00 00 77 C0 44 00*          *
293 125 00 00 00 77 C0 44 00*          *
294 126 00 C0 40 78 70 00 00*          *;+DSRCONT:
295 127 00 3F 00 77 21 C0 44 00*          *;+DSRCONT:
296 128 00 C0 00 77 21 C0 44 00*          *
297 129 00 00 00 76 80 50 00*          *
298 12A 00 00 00 77 21 C0 44 00*          *
299 12B 00 00 00 76 80 50 00*          *
300 12C 00 00 00 76 80 50 00*          *
301 12D 00 00 00 76 80 50 00*          *;+DSRCONT:
302 12E 00 00 00 76 80 50 00*          *;+DSRCONT:
303 12F 00 00 00 76 80 50 00*          *;+DSRCONT:
304 130 00 00 00 76 80 50 00*          *;+DSRCONT:
305 131 00 00 00 76 80 50 00*          *;+DSRCONT:
306 132 00 00 00 76 80 50 00*          *;+DSRCONT:
307 133 00 00 00 76 80 50 00*          *;+DSRCONT:
308 134 00 00 00 76 80 50 00*          *;+DSRCONT:
309 135 00 00 00 76 80 50 00*          *;+DSRCONT:
310 136 20 00 00 77 21 C0 44 00*          *;+DSRCONT:
311 137 00 00 00 76 80 50 00*          *;+DSRCONT:
312 138 00 00 00 76 80 50 00*          *
313 139 00 00 00 76 80 50 00*          *;+DSRCONT:
314 13A 00 00 00 76 80 50 00*          *;+DSRCONT:
315 13B 00 00 00 76 80 50 00*          *;+DSRCONT:
316 13C 00 00 00 76 80 50 00*          *;+DSRCONT:
317 13D 00 00 00 76 80 50 00*          *;+DSRCONT:
318 13E 00 00 00 76 80 50 00*          *;+DSRCONT:
319 13F 00 00 00 76 80 50 00*          *;+DSRCONT:
320 140 00 00 00 76 80 50 00*          *;+DSRCONT:
321 141 00 00 00 76 80 50 00*          *;+DSRCONT:
322 142 00 00 00 76 80 50 00*          *;+DSRCONT:

```

23

24

323 143 00 00 50 72 0A C7 20\*LDXXCC0\*:  
 324 144 00 00 50 72 0A 44 03\*STSAC7W:  
 325 145 00 00 50 72 0A 44 02\*STXXCON:  
 326 146 00 00 50 72 48 44 01\*STAAXCON:  
 327 147 00 00 50 72 48 44 00\*STAAXCON:  
 \*  
 328 148 00 FF 80 77 C0 E5 00\*  
 329 149 00 00 40 77 CC 11 20\*  
 330 14A 00 00 50 7A 6A C7 20\*  
 \*  
 331 14B 00 FF 80 77 C0 F5 00\*LDUXCON:  
 332 14C 00 00 40 77 CC 41 20\*  
 333 14D 00 00 50 7A 48 C7 00\*  
 \*  
 334 14E 00 FF 80 77 C0 E5 00\*LDUXCON:  
 335 14F 00 00 00 77 CC 41 20\*  
 336 150 00 00 50 7A 48 C7 10\*  
 \*  
 337 151 00 FF 80 77 C0 E5 00\*STAXCON:  
 338 152 00 00 00 77 CC 41 20\*  
 339 153 00 00 50 7A 48 14 01\*  
 \*  
 340 154 00 FF 80 77 C0 E5 00\*JMPXCON:  
 341 155 00 00 50 7F C0 11 20\*  
 \*  
 342 156 95 00 00 8F C0 10 00\*FPSHCONT:  
 343 157 95 00 00 8F C0 10 00\*  
 344 158 00 00 50 7F C0 00 00\*  
 \*  
 345 159 00 00 00 77 CC 05 00\*  
 346 15A 00 00 00 77 CC 27 00\*  
 347 15B 00 00 04 C0 77 CC 05 00\*  
 \*  
 348 15C 00 00 00 77 CC 27 00\*  
 349 15D 00 00 02 C0 77 CC 05 00\*  
 350 15E 00 00 00 77 CC 27 00\*  
 351 15F 00 00 00 77 CC 44 00\*  
 352 160 00 00 00 07 CC 27 00\*  
 \*  
 353 161 00 00 40 79 E5 00\*FPSHCONT:  
 354 162 00 00 00 77 C0 00 00\*  
 355 163 00 00 03 C0 77 00 00 00\*  
 356 164 00 00 00 77 CC 44 00\*  
 357 165 00 00 05 C0 77 00 00 00\*  
 358 166 00 00 00 77 CC 44 00\*  
 359 167 00 00 07 C0 77 00 00 00\*  
 360 168 00 00 00 77 CC 44 00\*  
 361 169 00 00 08 80 77 0F 00 00\*  
 362 16A 00 00 0C 80 77 CC 47 00\*  
 363 16B 00 00 00 07 00 2D 33\*  
 \*  
 364 16C 00 00 00 73 01 44 00\*FPSHICONS:  
 365 16D 00 00 40 80 77 CC 25 00\*  
 366 16E 00 00 00 7B 00 71 20\*  
 367 16F 15 90 00 87 CC 27 00\*  
 368 170 96 10 00 8E 00 44 00\*  
 369 171 00 00 80 77 CC 45 00\*  
 370 172 00 00 50 7F C0 00 00\*  
 \*  
 371 173 00 00 00 77 00 C0 33\*PULCONT:

420 1A4 80 00 00 77 C0 C4 D0\*  
 421 1A5 00 00 80 D7 C0 C7 E0\*  
 422 1A6 A2 30 00 8F C0 40 00\*FDIVCON:  
 423 1A7 80 00 00 77 D0 D1 01\*  
 424 1A8 04 01 80 77 C0 C5 00\*  
 425 1A9 80 00 00 73 43 44 00\*  
 426 1AA 9C 70 08 1F C0 40 00\*  
 427 1AB 00 00 80 77 C0 C7 50\*  
 428 1AC 98 40 00 8F 20 40 00\*  
 429 1AD 80 00 00 77 20 C0 C6 60\*  
 430 1AE 00 01 80 77 20 C0 C7 40\*  
 431 1AF 00 00 8F C0 40 00\*  
 432 1B0 80 00 00 77 C0 C4 00\*  
 433 1B1 00 01 80 77 C0 C7 40\*  
 434 1B2 98 00 00 8F C0 40 00\*  
 \*  
 435 1B3 9F E0 00 9F C0 C4 C4\*  
 \*  
 436 1B4 00 01 60 77 C0 C7 40\*DIVI:  
 \*  
 437 1B5 00 00 00 73 50 C9 69\*DSUB:  
 438 1B6 00 00 00 73 60 C9 7A\*  
 439 1B7 00 00 00 73 60 C9 8B\*  
 \*  
 440 1B8 00 01 00 72 81 44 11\*  
 441 1B9 00 00 34 74 90 D9 15\*  
 \*  
 442 1BA 80 00 00 70 A0 C1 11\*FROLB:  
 443 1BB 80 00 00 73 60 C1 44\*  
 444 1BC 80 00 01 57 C0 C4 14\*  
 \*  
 445 1BD 80 00 00 73 40 C1 66\*REENT:  
 446 1BE 80 00 00 73 60 C1 77\*  
 447 1BF 80 00 00 73 60 C1 88\*  
 \*  
 448 1C0 80 01 00 72 81 44 11\*  
 449 1C1 9B 50 01 1F C0 C4 51\*  
 \*  
 450 1C2 80 00 00 73 40 C1 69\*  
 451 1C3 80 00 00 73 60 C1 7A\*  
 452 1C4 80 00 00 73 60 C1 88\*  
 453 1C5 80 01 00 72 81 44 11\*  
 454 1C6 9B A0 00 9F C0 E1 15\*  
 \*  
 455 1C7 80 00 00 73 40 E5 00\*DIVBYZ:  
 456 1C8 00 09 00 77 C0 C5 33\*  
 457 1C9 A1 C0 84 9C 80 40 00\*  
 \*  
 458 1CA A2 30 00 8F C0 40 00\*FSUBCON:  
 459 1CB 80 00 00 77 C0 F5 00\*  
 460 1CC 9C E0 00 9F C0 40 00\*  
 \*  
 461 1CD A2 30 00 8F C0 40 00\*FADDCON:  
 \*  
 \*  
 462 1CE 80 00 00 77 C0 C4 C9FADD1:  
 463 1CF 80 00 00 77 C0 C4 DA\*  
 \*

464 1D0 07 FF 80 77 C0 E5 40\*  
 465 1D1 07 FF 80 77 C0 E5 51\*  
 466 1D2 80 30 00 73 50 C9 45\*  
 467 1D3 9E C0 08 1F C0 C4 E8\*  
 468 1D4 90 E0 02 9F C0 04 01\*  
 \*  
 469 1D5 80 03 00 77 C0 C4 10\*  
 470 1D6 80 00 00 77 C0 C2 00\*  
 471 1D7 80 00 00 77 C0 C4 C6\*  
 472 1D8 80 00 00 77 C0 C4 69\*  
 473 1D9 80 00 00 77 C0 C4 D7\*  
 474 1DA 80 00 00 77 C0 C4 7A\*  
 475 1DB 80 00 00 77 C0 C4 E8\*  
 476 1DC 80 00 00 77 C0 C4 8B\*  
 \*  
 \*  
 477 1DD 8U 00 00 77 D0 D4 44\*  
 \*  
 478 1DE 00 3U 80 73 50 40 04\*F00:  
 479 1DF 9F E0 01 1F C0 40 00\*  
 \*  
 480 1E0 0F F0 80 73 40 65 04\*SHIFTAS:  
 481 1E1 9E 60 08 1F C0 40 00\*  
 482 1E2 09 10 80 77 D0 CD 44\*  
 483 1E3 80 00 00 77 C0 C4 67\*  
 484 1E4 80 00 00 77 C0 C4 78\*  
 485 1E5 9E 00 00 9F D0 D1 88\*  
 486 1E6 80 00 00 73 40 44 04\*SINGA:  
 487 1E7 9E C0 08 1F C0 40 00\*SINGA:  
 488 1E8 80 04 00 73 01 44 04\*SINGA:  
 489 1E9 80 05 00 72 81 44 77\*  
 490 1EA 80 05 00 72 81 44 66\*  
 491 1EB 9E 70 00 98 40 CC 44\*  
 \*  
 492 1EC 80 00 00 73 40 71 10\*ADDSUB:  
 493 1ED 9F 80 02 1F C0 40 00\*  
 494 1EE 80 00 00 73 40 C1 C4 SSSAHE:  
 495 1EF 80 00 00 73 60 C1 D7\*  
 496 1F0 80 00 00 73 60 C1 E8\*  
 497 1F1 9F E0 01 9F C0 40 09\*  
 \*  
 \*  
 498 1F2 80 05 00 72 81 44 6E\*  
 499 1F3 80 05 00 72 81 44 00\*  
 500 1F4 80 05 00 72 81 44 C0\*  
 501 1F5 9F E0 00 9F D0 C4 00\*  
 \*  
 502 1F6 80 00 00 73 50 C9 C6 DIFF:  
 503 1F7 80 00 00 73 60 C9 D7\*  
 504 1F8 80 00 00 73 60 C9 E8\*  
 505 1F9 9F E0 01 1F C0 40 00\*  
 \*  
 506 1FA 80 00 00 77 C0 F5 00\*

```

507 1FB 80 00 00 73 50 D4 CC*
508 1FC 80 00 00 73 60 D4 DD*
509 1FD 80 00 00 73 60 D4 EE*
      *
510 1FE 00 09 80 77 C0 C5 33=NORM1;
511 1FF 00 00 80 77 C0 C7 10*
512 200 80 00 00 73 40 44 0E=NORM1;
513 201 A0 80 08 9F C0 40 00*
514 202 00 10 80 77 C0 C5 11*
      *
515 203 00 30 80 70 90 4D 01*
516 204 A0 D0 08 1F C0 40 00*
517 205 80 00 00 77 C0 C4 ED*
518 206 80 00 00 77 C0 C4 DC*
519 207 A0 00 00 9F D0 D1 CC*
      *
520 208 A1 00 02 1F C0 40 00=NORM2;
521 209 80 00 00 77 D0 C4 11*
522 20A 80 00 00 73 40 C1 CC*
523 20B 80 00 00 73 60 C1 DD*
524 20C A0 80 00 98 60 C1 EE*
      *
525 20D 00 00 80 73 40 C7 00=NORM3;
526 20E A3 30 00 8F C0 40 00=NSTUFF;
527 20F 80 00 50 7F C0 40 00*
      *
528 210 80 00 00 77 D0 C9 01=NORM4;
529 211 78 00 80 73 40 65 00*
530 212 A1 50 08 9F C0 40 00*
531 213 80 00 00 73 40 44 00*
532 214 A0 E0 80 98 C0 40 00*
      *
533 215 40 00 80 73 40 65 00=NORM45;
534 216 A1 80 09 1F C0 40 00*
535 217 00 00 80 73 40 C7 00*
536 218 80 00 84 74 40 C4 ED*
537 219 80 00 00 77 C0 C4 DD*
538 21A A2 00 00 97 C0 C4 C0*
      *
539 21B 80 00 80 73 40 ES 00=NORM5;
540 21C 07 FF 80 77 C0 DD 00=MAXANS;
541 21D FF FF 84 74 40 C7 ED*
542 21E 80 00 80 73 C0 C4 DE*
543 21F 80 00 00 77 C0 C4 CE*
544 220 A3 30 00 8F C0 40 00=FAULT;
545 221 FE F6 80 77 C0 C7 50*
546 222 BF 60 00 9F C0 40 00*
      *
547 223 00 02 C0 77 CC 45 03=FLDADS;
548 224 00 00 00 77 C0 C7 00*
549 225 00 04 C0 77 CC 45 03*
      *
550 226 00 00 00 77 C0 C7 80*
551 227 00 05 C0 77 CC 45 03*
552 228 00 00 00 77 C0 C7 80*

```

```

553 229 00 08 C0 77 CC 45 03*
554 22A 00 00 00 77 C0 C7 90*
555 22B 00 09 C0 77 CC 45 03*
556 22C 00 00 00 77 C0 C7 10*
557 22D 00 00 C0 77 CC 45 03*
558 22E 00 00 00 77 C0 C7 80*
559 22F 00 0F C0 77 CC 45 03*
560 230 00 00 00 77 C0 C7 70*
561 231 00 11 C0 77 CC 45 03*
562 232 00 00 00 D7 C0 C7 60*
      *
563 233 00 02 C0 77 CE 45 03=FSTOP;
564 234 80 00 00 77 CA 44 00*
565 235 00 04 C0 77 CE 45 03*
566 236 80 00 00 77 CA 44 0E*
567 237 00 06 C0 77 CE 45 03*
568 238 80 00 00 77 CA 44 0D*
569 239 00 08 C0 77 CE 45 03*
570 23A 80 00 00 D7 CA 44 0C*
      *
571 23B 00 02 C0 77 CC 45 03=FNCOUN;
572 23C 00 00 00 77 CA C7 00*
573 23D 00 04 C0 77 CC 45 03*
574 23E 00 00 00 77 CA C7 E0*
575 23F 00 06 C0 77 CC 45 03*
576 240 00 00 00 77 CA C7 D0*
577 241 00 08 C0 77 CC 45 03*
578 242 00 00 00 77 CA C7 C0*
579 243 A0 00 00 9F D0 D1 11*
      *
      *
      *
      *
580 244 00 70 80 77 C0 E5 40=STROCONT;
581 245 00 0C 80 A7 .0 47 00*
582 246 80 00 20 68 01 44 44*
583 247 A7 E0 00 8F C0 40 00*
584 248 00 04 C0 77 DC 4D 02*
585 249 00 00 00 77 CA C5 44*
586 24A 00 04 C0 77 CE 45 02*
587 24B 80 00 00 77 CA 44 04*
588 24C 00 0F 80 77 C0 E5 40*
589 24D A7 E0 00 8F C0 40 00*
590 24E 00 02 C0 77 DC 4D 02*
591 24F 00 00 00 77 CA C5 44*
592 250 00 00 C0 77 CE 45 02*
593 251 80 00 00 77 CA 44 04*
594 252 80 00 40 77 CC 44 02=VECCONT;
595 253 28 70 00 8F C0 C7 40*
596 254 80 00 00 77 C0 C4 54*
597 255 00 02 C0 77 CC 45 02*
598 256 00 00 00 77 CA C7 60*
599 257 00 04 C0 77 CC 45 02*
600 258 28 70 00 8F C0 C7 40*
601 259 00 00 C0 77 CC 45 02*
602 25A 00 00 00 77 CA C7 80*
603 25B 00 02 80 77 C0 C7 80*
604 25C 80 00 00 73 50 C9 45*
605 25D A6 00 02 9F C0 C4 74*

```

```

606 25E 00 02 80 77 D0 CD BB*
607 25F 80 00 00 77 D0 D4 77*
608 260 80 00 00 73 50 C9 06=VEC2;
609 261 A6 40 02 9F C0 40 00*
610 262 00 04 80 77 C0 C5 BB*
611 263 80 00 00 77 D0 D4 88*
612 264 00 12 80 77 C0 C5 66=VEC3;
613 265 80 00 00 73 50 51 78*
614 266 A6 90 02 98 40 CC 48*
615 267 80 00 00 77 D0 C4 BB*
616 268 80 00 00 73 40 CC 47*
617 269 80 00 177 CE 4C 03=YMAJOR;
618 26A A7 C0 OC 1F CA 44 04*
619 26B 80 02 80 77 DE 4D 03*
620 26C 80 00 00 77 C0 44 DB*
      *
621 26D 80 00 00 73 40 19 87=VECNORM;
622 26E A7 C0 08 1F C0 40 00*
623 26F 08 00 80 73 40 66 00*
624 270 A7 30 08 9F C0 40 00*
625 271 80 00 00 77 C0 C1 77*
626 272 A6 D0 00 9F C0 C1 88*
      *
627 273 00 04 C0 77 DE 4D 03=VSTOP;
628 274 80 00 00 77 C0 44 07*
629 275 00 06 C0 77 DE CD 33*
630 276 A7 80 00 8F C0 88 38*
631 277 80 00 50 7A 40 44 00*
      *
632 278 80 00 40 77 CE 4C 03=YPSHI;
633 279 80 00 00 77 CA 44 05*
634 27A 00 03 C0 77 DE CD 33*
635 27B 80 00 00 D7 CA 88 36*
      *
636 27C A7 80 84 8C 40 40 00=VPOINT;
637 27D 80 00 50 7F C0 40 00*
      *
638 27E 80 00 00 77 C0 C4 C4=SCALE;
639 27F 80 00 00 77 C0 C1 44*
640 280 80 00 00 77 C0 C1 44*
641 281 00 01 80 70 80 65 01*
642 282 A8 40 08 9F C0 C1 44*
643 283 80 00 00 77 C0 C1 4C*
      *
644 284 00 02 80 70 80 65 01=SCALE1;
645 285 80 00 08 57 C0 C1 CC*
646 286 80 04 00 D3 01 41 4C*
      *
647 287 80 00 00 77 C0 44 04=XFORM;
648 288 30 0A 80 A7 C0 47 00*
649 289 80 06 20 6B 81 44 44*
650 28A 80 00 00 D7 D0 C0 44 44*
      *
651 28B 80 00 40 77 CC 44 04=PSHRCONT;
652 28C 00 00 00 77 C0 C7 50*
653 28D 80 00 40 77 DC C4 33*
654 28E 00 00 00 77 CA C7 60*
655 28F 80 00 40 77 CE 44 05*
656 290 80 00 00 77 CA 44 06*

```

```

657 291 80 00 40 77 DE 83 34*
658 292 00 02 80 77 CA 45 05*
659 293 B3 00 00 9F C0 40 00*
      *
660 294 80 00 40 77 CC 44 04=RTRNCONT;
661 295 00 00 00 77 CA C7 50*
662 296 00 02 C0 77 DC CD 55*
      *
663 297 00 00 00 77 CA C7 60*
664 298 80 00 40 77 CE 44 04*
665 299 80 00 00 77 CA 44 05*
666 29A B1 60 00 9F C6 44 06*
      *
667 29B 80 00 50 TF CA 88 32=PSHRCONT;
668 29C 00 00 00 77 CA C7 20=PULXCONT;
669 29D 80 00 50 7F D0 C4 33*
      *
670 29E 28 80 09 9F C4 C7 50=BHICONT;
671 29F 80 00 50 TF C0 40 00*
      *
672 2A0 28 80 09 1F C4 C7 50=BLSCONT;
673 2A1 80 00 50 7F C0 40 00*
      *
674 2A2 28 80 01 9F C4 C7 50=BCCCONT;
675 2A3 80 00 50 7F C0 40 00*
      *
676 2A4 28 80 01 1F C4 C7 50=BCSCONT;
677 2A5 80 00 50 7F C0 40 00*
      *
678 2A6 28 80 08 9F C4 C7 50=BNENCONT;
679 2A7 80 00 50 7F C0 40 00*
      *
680 2A8 28 80 08 1F C4 C7 50=BEQCONT;
681 2A9 80 00 50 7F C0 40 00*
      *
682 2A0 28 80 03 9F C4 C7 50=BYVCONT;
683 2A8 80 00 50 7F C0 40 00*
      *
684 2AC 28 80 03 1F C4 C7 50=BSVCONT;
685 2AD 80 00 50 7F C0 40 00*
      *
686 2AE 2B 80 02 9F C4 C7 50=BLPLCONT;
687 2AF 80 00 50 7F C0 40 00*
      *
688 2B0 2B 80 02 1F C4 C7 50=BLTCONT;
689 2B1 80 00 50 7F C0 40 00*
      *
690 2B2 2B 80 04 9F C4 C7 50=BGECONT;
691 2B3 80 00 50 7F C0 40 00*
      *
692 2B4 2B 80 04 1F C4 C7 50=BLTCONT;
693 2B5 80 00 50 7F C0 40 00*
      *
694 2B6 2B 80 0C 9F C4 C7 50=BGTCONT;
695 2B7 80 00 50 7F C0 40 00*

```

737 2F6 2F 70 00 9F C4 C7 40eLNUPI6:  
     0;+ Transfer the PC, which was loaded with the  
     0;+ desired address to test, to a work register in 2901  
     0;  
 738 2F7 80 00 40 77 2E 44 00eLP16CONT:  
     0;+ Transfer work register to memory write address.  
 739 2F8 80 00 00 77 CA 44 030:  
     0;+ Write RS (previously loaded with desired data) to memory  
 740 2F9 80 00 40 77 CC 44 040:  
     0;+ Transfer work register to memory read address.  
 741 2FA 2F 73 00 9F CA C7 600:  
     0;+  
     0;  
     0;  
     0;  
     0;+ The following is an 8-bit version of the same test.  
 742 2FB 2F C0 00 9F C4 C7 40eLOCPI6:  
     0;+ Transfer PC to work register.  
     0;  
 743 2FC 80 00 00 77 2E 44 04eLP8CONT:  
     0;+ Transfer work register to memory write address.  
 744 2FD 80 00 00 77 CA 44 050:  
     0;+ Write RS to memory.  
 745 2FE 80 00 00 77 CC 44 060:  
     0;+ Transfer work register to read address.  
 746 2FF 2F C0 00 9F C4 C7 600:  
     0;+  
     0;  
     0;  
     0;  
     0;  
     0;  
     0;  
 747 300 80 00 00 4F C0 40 00eTEST:  
 748 301 80 00 50 7F C0 40 00eNOP1:  
 749 302 80 00 50 7F C0 40 00eNOP2:  
 750 303 80 00 50 7E 4C 10 00eSFIA:  
 751 304 8C E0 00 9F C0 40 00eDUMP:  
 752 305 8C E0 00 9F C0 40 00e  
 753 306 12 70 01 9F C0 C7 40eTAPI:  
 754 307 12 60 81 9F C0 C7 00eTPA1:  
 755 308 80 00 50 79 D0 C4 22eINAI:  
 756 309 80 00 50 79 C0 CC 22eDKX:  
 757 30A 80 00 00 7C 40 40 00eCLV:  
 758 30B 80 00 00 7C 60 40 00eSEV:  
 759 30C 80 00 00 7C 80 40 00eCLC:  
 760 30D 80 00 00 7C 80 40 00eSEC:  
 761 30E 80 00 00 7C C0 40 00eCLII:  
 762 30F 80 00 00 7C C0 40 00eSEI:  
 763 310 80 00 50 79 50 C9 01eSDAI:  
 764 311 80 01 50 79 50 31 10eCBAI:  
 765 312 80 00 50 7E 80 44 00eTAPI:  
 766 313 80 00 00 D1 7F C0 C7 00eTPAII:  
 767 314 80 00 00 50 7F C0 C5 22eADXI:  
 768 315 80 00 00 50 7F C0 C5 33eASPI:  
 769 316 80 00 00 50 7A 40 C4 10eTABI:  
 770 317 80 00 00 50 7A 40 C4 01eTBAI:  
 771 318 80 00 50 7F 00 40 00eSDAI:  
 772 319 8C E0 00 9F C0 40 00eDAAI:  
 773 31A 94 80 00 9F C0 40 00eNLDXI:  
 774 31B 80 00 50 78 40 C1 01eABAII:

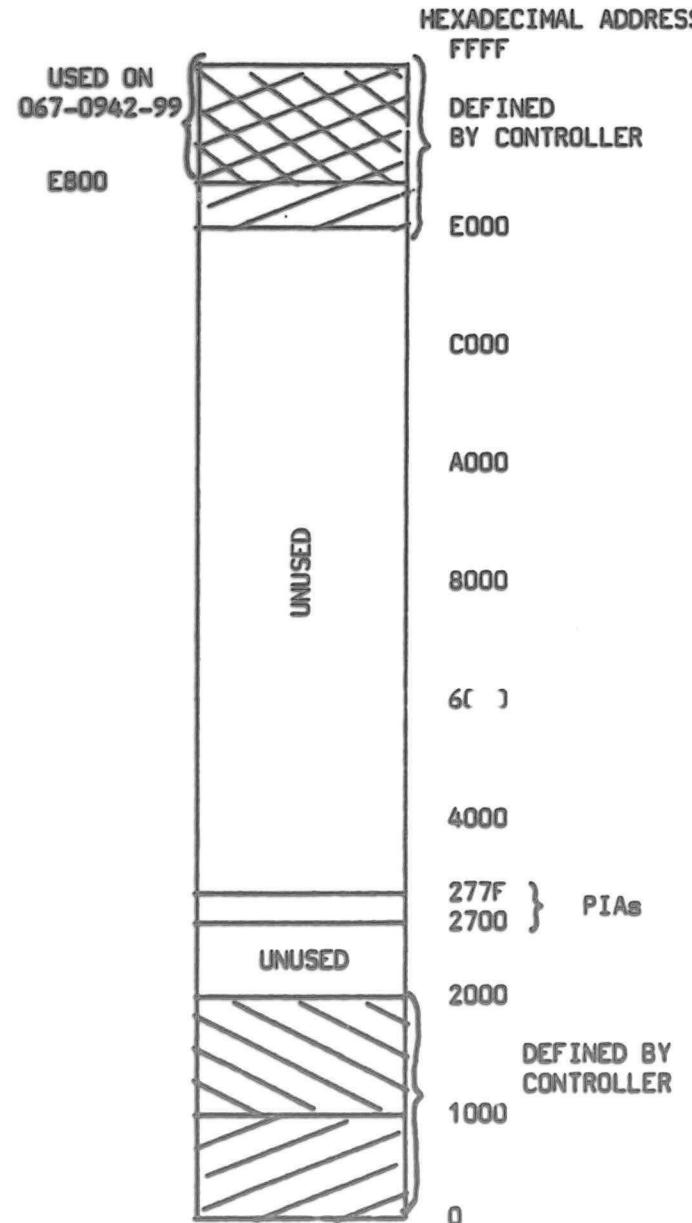
776	B1C	94	80	90	97	C0	60	0000HLDKX;		
777	B1D	94	80	90	97	C0	60	0000HLDKX;		
778	B1E	95	10	90	97	C0	60	0000HSTAKX;		
779	B1F	95	10	90	97	C0	60	0000JMPAKX;		
780	B20	AB	A0	90	97	C0	C7	0000BABA;		
781	B21	AB	A0	90	97	C0	60	0000BBB;		
782	B22	AB	A0	90	97	C0	C7	0000B4IX;		
783	B23	AB	A0	90	97	C0	C7	0000BLBX;		
784	B24	AB	A0	90	97	C0	C7	0000BCCX;		
785	B25	AB	A0	90	97	C0	C7	0000BCBX;		
786	B26	AB	A0	90	97	C0	C7	0000BDE;		
787	B27	AB	A0	90	97	C0	C7	0000BEE;		
788	B28	AB	A0	90	97	C0	C7	0000BVC;		
789	B29	AB	A0	90	97	C0	C7	0000BVS;		
790	B2A	AB	A0	90	97	C0	C7	0000BPLX;		
791	B2B	AB	A0	90	97	C0	C7	0000B4IX;		
792	B2C	AB	A0	90	97	C0	C7	0000BCE;		
793	B2D	AB	A0	90	97	C0	C7	0000BLTX;		
794	B2F	AB	A0	90	97	C0	C7	0000BGT;		
795	B30	AB	A0	90	97	D0	23	0000TRK;		
796	B31	AB	A0	90	90	7F	D0	C4	130-LMS;	
797	B32	AC	F0	60	97	DC	C4	130-PULAS;		
798	B33	AB	A0	90	97	DC	C4	130-PULAS;		
799	B35	AB	A0	90	90	7F	C0	C2	130-BLBS;	
800	B36	AB	A0	90	90	7F	C0	C2	130-FTKS;	
801	B36	AB	A0	10	90	97	DC	C4	130-PSMAS;	
802	B37	AB	A0	20	90	97	DC	C4	130-PSMS;	
803	B38	AB	A0	50	97	C0	C4	0000JMP14;		
804	B39	AB	A0	30	90	97	DC	C4	130-CHTS;	
805	B3A	AB	A0	60	90	97	C0	C7	0000FPMSP;	
806	B3B	AB	A0	90	90	97	DC	C4	130-RTI;	
807	B3C	AB	A0	60	90	97	DC	C4	0000FPMK;	
808	B3D	AB	A0	60	90	97	DC	C4	0000FPMK;	
809	B3E	AB	A0	90	97	DC	C4	0000FPMK;		
810	B3F	AB	A0	90	97	DC	C4	0000FPMK;		
811	B3G	AB	A0	90	90	78	90	D4	0000ELGA;	
812	B3I	AB	A0	90	90	78	90	D4	0000FPMK;	
813	B3J	AB	A0	90	90	97	DC	C4	0000FPMK;	
814	B3K	AB	A0	90	90	78	90	FC	0000CDFA;	
815	B3L	AB	A0	90	90	90	90	D4	0000ELFA;	
816	B3M	AB	A0	90	90	97	C0	C5	420-FPULK;	
817	B3N	AB	F1	00	90	90	S1	44	0000RUMA;	
818	B3P	AB	E0	00	90	90	90	44	0000ASPA;	
819	B3Q	AB	E0	00	90	78	90	C1	0000ASLA;	
820	B3R	AB	E0	00	90	78	90	C1	0000ASLA;	
821	B3A	AB	E0	00	90	78	C0	C3	0000DECA;	
822	B3B	AB	97	30	90	97	C0	C7	0000FPULX;	
823	B3C	AB	80	00	90	78	D0	C4	0000IMCA;	
824	B3D	AB	80	00	90	78	90	44	0000TSTA;	
825	B3E	AB	97	B0	00	97	C0	44	0000FDUPX;	
826	B3F	AB	98	00	90	78	90	C7	0000CLRA;	
827	B3G	AB	80	00	90	78	90	D4	110-MEGS;	
828	B3I	AB	98	20	00	97	C0	60	0000FSLP;	
829	B3J	AB	9C	D0	00	97	C0	60	0000FADD;	
830	B3L	AB	98	00	90	50	79	90	FC	110-CUMB;
831	B3M	AB	9F	10	00	98	90	91	44	110-LSRM;
832	B3N	AB	A0	00	90	97	C0	60	0000FBUD;	
833	B3P	AB	9F	11	00	98	S1	44	110-RORP;	
834	B3Q	AB	9F	00	00	98	90	44	010-ASRS;	

036	359	00	00	90	70	00	C1	11-000000:	
037	36A	00	00	90	70	00	C0	11-000000:	
038	36B	90	00	90	70	00	C0	00-000000:	
039	36C	00	00	90	70	00	C4	11-000000:	
040	36D	00	00	90	70	00	C4	01-TSTD:	
041	36E	9A	00	00	97	C0	40	00-CPD1:	
042	36F	00	00	90	70	00	C7	10-CLAD:	
043	36G	00	00	90	97	C0	C5	02-000000:	
044	36I	A3	00	00	97	C0	40	00-FWDN:	
045	36J	A8	00	00	97	C0	C7	00-PMDT:	
046	36L	90	00	90	97	C0	C9	02-COPR:	
047	36M	90	70	00	97	C0	C9	02-LDR:	
048	36S	A9	40	00	97	C0	C7	00-ATRN:	
049	36T	B0	10	00	97	C0	C9	02-000000:	
050	36Y	9C	30	00	97	C0	C9	03-ASRE:	
051	36Z	9C	60	00	97	C0	C9	02-000000:	
052	369	9C	90	00	97	C0	C9	03-000000:	
053	36A	00	00	90	97	C0	C9	03-000000:	
054	36B	A9	80	00	97	C0	CC	13-000000:	
055	36C	00	00	90	97	C0	C9	03-100000:	
056	36D	00	00	90	97	C0	40	02-000000:	
057	36E	00	00	90	97	C0	40	03-000000:	
058	36F	00	00	90	97	C0	40	03-000000:	
059	370	00	50	00	97	C0	C7	00-000000:	
060	371	A4	40	00	97	C0	40	00-000000:	
061	372	A5	20	00	97	C0	40	00-000000:	
062	373	90	90	00	97	C0	C7	00-000000:	
063	374	00	70	00	97	C0	C7	00-LDR:	
064	375	A9	C0	00	97	DC	C4	13-000000:	
065	376	9C	10	00	97	C0	C7	00-000000:	
066	377	00	30	00	97	C0	C7	00-000000:	
067	378	00	60	00	97	C0	C7	00-000000:	
068	379	00	90	00	97	C0	C7	00-000000:	
069	37A	00	00	90	97	C0	C7	00-000000:	
070	37B	00	20	00	97	C0	40	00-000000:	
071	37C	00	00	90	97	C0	C7	00-000000:	
072	37D	00	00	90	70	00	C7	00-TSTD:	
073	37E	00	00	90	70	00	C7	00-000000:	
074	37F	00	00	90	70	00	S1	00-CLAD:	
075	380	00	00	90	70	00	CD	00-000000:	
076	381	00	00	90	70	00	CD	00-000000:	
077	382	00	00	90	70	00	CD	00-000000:	
078	383	00	00	90	97	C0	40	00-000000:	
079	384	00	00	90	70	00	CD	00-000000:	
080	385	00	00	90	70	00	CD	00-000000:	
081	386	00	00	90	70	00	C7	00-000000:	
082	387	00	00	90	97	C0	40	00-000000:	
083	388	00	00	90	70	00	CD	00-000000:	
084	389	00	00	90	70	00	CD	00-000000:	
085	38A	00	00	90	70	00	CD	00-000000:	
086	38B	00	00	90	70	00	CD	00-000000:	
087	38C	93	40	00	90	90	40	03-CPD1:	
088	38D	93	10	00	90	97	C0	40	03-000000:
089	38E	00	00	90	70	00	C7	00-LDR:	
090	38F	00	20	00	97	C0	40	00-000000:	
091	389	00	00	90	70	00	CD	00-000000:	
092	391	00	00	90	70	00	CD	00-000000:	
093	392	00	00	90	70	00	CD	00-000000:	
094	393	00	50	00	97	C0	40	00-000000:	

955	3D0	00 00 00 00 00 00 00 00	90 79 50 CD	11-0SUBBD:
956	3D1	00 00 00 00 00 00 00 00	90 79 50 40	01-0CPBBD:
957	3D2	00 00 00 00 00 00 00 00	90 79 70 CD	11-0SBCBD:
958	3D3	0C E0 00 00 00 00 00 00	90 9F C0 40	00-000:
959	3D4	00 00 00 00 00 00 00 00	90 7A 40 45	11-0ANDBD:
960	3D5	00 00 00 00 00 00 00 00	90 7A 40 05	01-0RITBD:
961	3D6	00 00 00 00 00 00 00 00	90 7A 40 40	C7 10-0LDAD:
962	3D7	00 00 00 00 00 00 00 00	90 7A 40 40	01-0STABD:
963	3D8	00 00 00 00 00 00 00 00	90 7A 40 40	F5 11-0CRBBD:
964	3D9	00 00 00 00 00 00 00 00	90 7B 00 00	C5 11-0ADCBD:
965	3DA	00 00 00 00 00 00 00 00	90 7A 40 00	40 11-0GRABD:
966	3DB	00 00 00 00 00 00 00 00	90 7B 00 00	40 11-0ACRBD:
967	3DC	FF FF 00 00 00 00 00 00	7F C0 00 00	C7 F0-0SBUG:
968	3DD	00 00 00 00 00 00 00 00	90 7F C0 00	C7 F0-0CSBUG:
969	3DE	00 00 00 00 00 00 00 00	90 7A 00 00	C7 20-0LDIBD:
970	3DF	00 00 00 00 00 00 00 00	90 7A 00 40	02-0STXBD:
971	3E0	93 90 00 00 00 00 00 00	9F CC 00 00	05 02-0SUBBX:
972	3E1	93 A0 00 00 00 00 00 00	9F CC 00 00	05 02-0CPBBD:
973	3E2	93 B0 00 00 00 00 00 00	9F CC 00 00	05 02-0SBCBX:
974	3E3	AC 20 00 00 00 00 00 00	9F CC 00 00	00-00VRL:
975	3E4	93 C0 00 00 00 00 00 00	9F CC 00 00	05 02-0ANDBX:
976	3E5	93 D0 00 00 00 00 00 00	9F CC 00 00	05 02-0ITABX:
977	3E6	93 E0 00 00 00 00 00 00	9F CC 00 00	05 02-0LDABX:
978	3E7	94 60 00 00 00 00 00 00	9F CC 00 00	05 02-0STABX:
979	3E8	93 F0 00 00 00 00 00 00	9F CC 00 00	05 02-0EURDX:
980	3E9	94 00 00 00 00 00 00 00	9F CC 00 00	05 02-0ADCBX:
981	3EA	94 10 00 00 00 00 00 00	9F CC 00 00	05 02-0CRABX:
982	3EB	94 20 00 00 00 00 00 00	9F CC 00 00	05 02-0ASDBX:
983	3EC	AC F0 00 00 00 00 00 00	9F CC 40 40	00-00VRL:
984	3ED	AD F0 00 00 00 00 00 00	9F CC 00 00	C7 40-0HAD:
985	3EE	94 30 40 00 00 00 00 00	9F CC 00 00	05 02-0LDXX:
986	3EF	94 50 40 00 00 00 00 00	9F CC 00 00	02-0STXX:
987	3F0	00 00 50 70 00 00 00 00	90 3D CD 40	11-0SUDB:
988	3F1	00 00 50 70 00 00 00 00	90 40 40 40	01-0CPBD:
989	3F2	P0 00 50 70 00 00 00 00	79 70 CD	11-0SBCB:
990	3F3	AD C0 00 00 00 00 00 00	90 9F C0 00	C7 50-0CPATCH:
991	3F4	00 00 50 70 00 00 00 00	7A 40 40 40	11-0ANDB:
992	3F5	00 00 50 70 00 00 00 00	7A 40 40 05	01-0RITB:
993	3F6	00 00 50 70 00 00 00 00	7A 40 40 40	C7 10-0LDAB:
994	3F7	00 00 50 70 00 00 00 00	7A 40 40 44	01-0STAB:
995	3F8	00 00 50 70 00 00 00 00	7A 40 40 40	11-0EONB:
996	3F9	00 00 50 70 00 00 00 00	7A 40 40 40	11-0ADCB:
997	3FA	00 00 50 70 00 00 00 00	7A 40 40 40	DD 11-0GRAB:
998	3FB	00 00 50 70 00 00 00 00	7B 40 40 40	11-0ASBD:
999	3FC	8C E0 00 00 00 00 00 00	9F C0 40 40	00-00:
1000	3FD	AD 90 00 00 00 00 00 00	9F C0 40 40	40-0PATCH:
1001	3FE	00 00 50 7A 00 00 00 00	7A 00 00 40	20-0LDX:
1002	3FF	00 00 50 7A 00 00 00 00	7A 00 40 44	02-0STX:

**067-0902-00/067-0942-99 MEMORY MAP**

The ROM on the 067-0942-99 Personality Board defines some memory space in the 067-0902-99 fixture. Figure 1 shows the resulting memory map.



**Figure 1.** 067-0942-99  
Memory Map

**FIRMWARE FOR 067-0942-99**

If the 067-0942-99 fails to operate correctly, the following firmware listing may be useful in finding circuit faults.