Application Note 222-11
A SIGNATURE ANALYSIS CASE STUDY
of a 6800-Based Display Terminal

The Memorex 1377 Terminal Was Retrofit For SA Using The HP 5001A Microprocessor Exerciser.

HEWLETT PACKARD

Application Note 222-11

A SIGNATURE ANALYSIS CASE STUDY of a 6800-Based Display Terminal

This case study shows how Signature Analysis was retrofit into a Display Terminal using the HP 5001A Microprocessor Exerciser to allow troubleshooting to the component level with a Signature Analyzer.

FORWARD

ABOUT DIGITAL TROUBLESHOOTING

Microprocessors have revolutionized your product line. Your products are smarter, faster, friendlier and more competitive because they take advantage of μP-based control and computation. They are also harder to bulid, harder to test and harder to fix when they fail. Complex bus structures and timing relationships have practically obsoleted the scope/voltmeter signal tracing techniques so effective on analog products. The need to enhance the testability and serviceability of your digital products is acute. So is the need for specialized digital troubleshooting equipment.

ABOUT SIGNATURE ANALYSIS

To address these needs, Hewlett-Packard has developed the Signature Analysis technique, as well as a Signature Analyzer product line, for component-level troubleshooting of microprocessor-based products. A Signature analyzer detects and displays the unique digital signatures associated with the data nodes in a circuit under test. By comparing these actual signatures to the correct ones, a troubleshooter can back-trace to a faulty node. By designing or retrofitting S.A. into digital products, a manufacturer can provide manufacturing test and field service procedures for component-level repair, without dependence on expensive board-exchange programs.

ABOUT THIS CASE STUDY SERIES

Use of a Signature Analyzer requires that some test features be designed or retrofit into the product to be tested. This application note is one in a series of case studies aimed at assisting designers, test engineers, and others in understanding these features so that they can easily add Signature Analysis to their product. These case studies show detailed examples of these features in various digital systems based on specific microprocessors.

ABOUT THIS PUBLICATION

This is a reprint of a technical article from *Electronics* magazine. It describes how Signature Analysis testing was implemented on the 6800 microprocessor based Memorex 1377 Display Terminal. The terminal had not originally been designed with SA in mind, yet had been in production for several years. A new tool, the HP 5001A Microprocessor Exerciser, allowed SA to be easily retrofit into this existing product for faster testing and troubleshooting of the terminal on the manufacturer's production line. This article shows how the preprogrammed tests of the 5001A were selected and used to test the 6800, address and data buses, program ROM, scratchpad RAM, display RAM, and clock and timing circuits. Also included is a description of the custom program for testing the display refresh circuits in both a functional and diagnostic mode. Appended to the article is a listing of the custom program written to test the PIAs. The 5001A, combined with a special tool built by Memorex (circuit diagram shown), allowed testing of the asynchronous communications interface (IBM 3270 protocol). Results of circuit coverage and test effectiveness are covered in the article.

ABOUT OTHER PUBLILCATIONS

Application Note 222-0, "An Index to Signature Analysis Publications" lists all other application notes currently available in the AN 222 series about Signature Analysis. They cover a wide range of interests, from how to design or retrofit Signature Analysis into digital systems, to the cost reductions that can be expected in production test and field service by doing so. It also lists all data sheets for the complete line of Hewlett-Packard Signature Analysis products, plus other related publications about digital troubleshooting.

CONTENTS

Applying signature analysis to existing processor-based products

With the advent of an off-board source of stimuli for signature analysis, it is possible to retrofit this efficent test method without extensive redesign

by Robert Rhodes-Burke,* *Hewlett-Packard Co., Santa Clara, Calif.*

☐ The digital test technique known as signature analysis (SA) is now widely regarded as the best way to service digital, processor-based boards. Until recently, however, products had to be designed to meet the special requirements of SA. In particular, the digital test patterns needed to stimulate the board during SA had to be generated by an on-board processor working from a user-supplied program.

Consequently, the many processor-based products designed before the technique's introduction could not be tested by it unless they underwent expensive redesign. However, now that an inexpensive, general-purpose stimulus source—the model 5001 microprocessor exerciser—is available, existing products can be serviced with little or no modification using signature analysis.

The application of SA to the Memorex 1377 display terminal will demonstrate the ease with which the technique can be used on equipment not designed for it. Although the terminal has been in production for several years and was not originally designed for signature analysis, it is a good SA target. Large portions of its circuity can be accessed through its processor using the 5001, so that a reasonably thorough check of its operation can be performed.

Understanding the application of SA to the 1377 demands familiarity with both the technique (see "Signature analysis revisited," p. 128) and the tools of the process. A brief description of the 5001 exerciser, its capability, and general application therefore precedes an explanation of the 1377 retrofit.

The model 5001 microprocessor exerciser, or Stimpod, as it is nicknamed, is a companion to the signature analyzer. It provides the digital stimuli to the board from which the analyzer takes bit streams for translation into hexadecimal signatures. Weighing only 3 pounds and measuring 9¼ by 5½ by 1 inch, the Stimpod is portable and travels easily into the field (Fig. 1).

Each Stimpod is designed for use with a particular type of microprocessor. The 6800 is supported at present, and the 8080, 8085, and Z80 will be supported in the coming year.

To employ the Stimpod, the processor on the board under test is removed and plugged into a zero-insertion-force connector on the front panel of the 5001. A ribbon cable from the Stimpod is then connected to the board socket that has been vacated by the processor.

1. Retrofit in a box. Using signature analyzers to troubleshoot products not designed for use with them like the Memorex 1377 terminal is easy, thanks to the 5001 microprocessor exerciser (center). The checkerboard pattern (top) is produced by a 5001 program.

The processor and board can then be exercised by the 5001. Contained in the Stimpod's internal memory are 52 test programs (see Table 1) applicable to a general class of microprocessor-based designs. For testing special aspects of a design, another ZIF socket is provided on the 5001's front panel. It accepts a 2716 erasable programmable read-only memory in which the user has placed a stimuli program. Thus the 5001 lets users apply signature analysis techniques without designing in or disturbing on-board program memory.

To use one of the internal test programs, a user sets the front-panel switch to

*The author is now with Apple Computer Corp., Cupertino, Calif.

INT and calls up the test by number, pressing the tens and units buttons below the light-emitting-diode display. Pressing the enter button then starts the test. Putting a preprogrammed PROM in the smaller front-panel socket and moving the double-throw switch to the external position permits custom tests to be called up and run in the same manner.

Since the 5001 was designed for use with a signature analyzer, it produces the start, stop, and clock signals needed by that unit to gate the bit stream and form a signature. These signals are provided at ports on the right of the exerciser, along with a common ground and a qualifier signal that is applicable in some of the preprogrammed tests.

On the left side of the Stimpod is an eight-bit output port, a qualifier input, and a power input and ground. Using the byte-wide port, the 5001 can stimulate the input side of input/output devices on the product being tested. The qualifier line can be used to recognize when a particular device is enabled, so that the 5001 can tell the signature analyzer to collect bits going to or coming from the device. The external power ports need only be used when the exerciser's power requirements—nominally 2.75 watts, exclusive of the processor—exceed the power available directly from the board under test.

Signature analysis could be applied to most of the 1377 by the 5001 and a 5004 signature analyzer alone. The general methodology of SA application involves deriving a set of signatures for a known good board, analyzing possible faults to determine how they change the signatures, and verifying that the set of signatures thus generated is valid and unique using other boards of the same type.

Although all this was done by hand for the 1377, it can now be almost completely automated with a automatic board tester that incorporates signature analysis. Such a board tester is available from Hewlett-Packard—the 3060A with option 100. In either case, once valid signatures are obtained, the board designer or test engineer can generate a fault tree—a listing or schematic that tells what points to check next if a fault signature is found. A field technician otherwise ignorant of SA can then use this tree to troubleshoot a system.

Applying signature analysis to many microcomputer-based products is often straightforward, using only the preprogrammed tests in the 5001. The 1377, however, is a special challenge to SA application because of some unusual aspects of its design. These can be understood by examining the functional layout of the terminal.

Choice target

The digital electronics used in the model 1377 display terminal are contained on a single, 9-by-16-in. multilayered printed-circuit board (Fig. 2). The board can be functionally segmented into five separate areas: the microcomputer, the display memory, the display-refresh algorithmic state machine (ASM), the communications ASM, and the clock and timing circuitry.

It should be noted that in normal operation three machines—the microcomputer, the communications

Signature analysis revisited

Signature analysis, a patented troubleshooting technique introduced by the Hewlett-Packard Co. in 1977, is based on the principle that a good digital circuit in a known (initialized) state will produce the same output when stimulated repeatedly by the same input. If the repeated output of a device is not the one it has been designed to produce, it has failed.

While this principle is simple and fairly obvious, signature analysis implementation is a bit more complex, relying on mathematics similar to that for cyclic redundancy coding. (For a thorough explanation of the signature formation process, see *Electronics*, March 3, 1977, p. 93.) But once signature analysis techniques have been applied to a design, using them to troubleshoot it is extremely simple.

All a technician has to do, either in production or in the field or depot, is follow the time-honored technique of signal tracing. He or she checks one test point and compares the measurement result to that in a table or schematic. If it does not match, the troubleshooter checks another point in accordance with the test plan.

The point between the last bad measurement and the next good one is the failure location. It should be noted that the technician does not need to know anything about signature analysis or, for that matter, digital logic. Thus, by requiring less from the troubleshooter, the technique can greatly reduce service costs.

In the past, a repetitive digital stimulus, or bit stream, was generated by the microprocessor of the product under test, which would execute a special test program residing in an on-board memory. A signature analyzer is used to check the response at the board's various test points, or nodes.

The analyzer works by monitoring the bit stream on one line for a specified period determined by the clock rate of the circuit under test. In this process it compresses the data and translates it into a four-character hexadecimal word, or signature.

A set of signatures, one for each test node in the circuit, must be generated; the designer or test engineer generally does this by exercising a known good unit—the prototype perhaps—with routines designed to exercise each area of the board. The good signatures are recorded; then the board is analyzed for the effect of failures on the bit stream. Once this is done, the test designer can create test procedures and test-point schematics for use by production-line and field technicians.

The three key conditions for the application of signature analysis are the ability to:

- ☐ Initialize circuits that can hold two or more different states (RAMs, flip-flops, and counters, for example).
- ☐ Synthesize the timing needed for signature sampling, both in terms of framing the sequence (sample start-sample stop) and clocking the bit stream.
- ☐ Apply the stimulus.

The first of these requirements is really a basic tenet of design for testability and can only be achieved by design. Designers who fail to observe it will be faced with a major testing problem. The other two conditions, however, need no longer be absolute design imperatives, thanks to the introduction of the 5001. **-Richard W. Comerford**

TABLE 1: MICROPROCESSOR EXERCISER TEST SET

Number	Test	Address range	Qualifier
00	Microprocessor: 6800 instruction set, interrupts	–	–
01	Buses: free-run	all	–
02	RAM: read/write 6800 direct addressing range	0000 – 00FF	–
03	RAM: read/write multiple patterns (checker-	0000 – 3FFF	–
04	board, inverse, address-as-data);	4000 – 7FFF	–
05	address range is selected according	8000 – BFFF	–
06	to test number	C000 – FFFF	–
07	RAM: read/write address as data	all	–
08	RAM: read/write alternating checkerboard	all	–
09	RAM: write checkerboard, then free-run	all	–
10/11	I/O: write patterns to qualified outputs	as qualified	0/1
12/13	I/O: read stimulus from qualified inputs	as qualified	0/1
14/15	ROM: read qualified data	as qualified	0/1
16/17	ROM: bus signature (pin X_0)	as qualified	0/1
18,19	5001: self-exercise	–	–
20 – 51	ROM: read 2-K address range	*	–

*Test number indicates 2-K address range, for example, 20 is 0000 to 07FF, 51 is F800 to FFFF

ASM, and the display refresh ASM—are jointly responsible for the terminal's workings. They operate concurrently in a complex, interleaved fashion, using various bus arbitration methods to share the data and address lines and access the shared display memory. The challenge, then, was to find a way of checking the functional areas independently, yet as they really performed.

Other aspects of the 1377's design were challenging, too. The use of two programmable interface adapters (PIAs) in the microcomputer section required a somewhat custom approach, since the designation of these devices' ports for input or output is determined and programmed to fit a specific application. In the 1377, the PIAs scan the terminal's keyboard and configuration switches, as well as drive its audio output (bell) and status indicator. Data from the PIAs can directly control some of the functions of the ASMs, such as selecting different segments of the display RAM or inhibiting the display or its reset. Therefore, it was essential to find a way to check the operation thoroughly.

The most elaborate part of the board is the timing and clock circuitry. Like other clock circuits, it generates timing waveforms independently of the microprocessor. On the 1377, it generates 11 phased clocks—more than usual—which are in turn used by the other functional blocks for timing generation. Despite this circuitry's complexity, SA checked it 100% with the least difficulty.

Of the other four functional blocks, the microcomputer also was 100% tested with SA. But the display memory block could be only 90% tested and the display-refresh ASM could be only 80% tested because of the communications ASM.

The communications ASM operates asynchronously with respect to the rest of the system, so that it could not be checked using the 5001 and the 5004 alone. However, a special test tool designed previously by Memorex had been used for some time to check this portion of the terminal. With some adaptation of such a tool (to be described later), even this asynchronous segment could be tested with signature analysis techniques.

The microcomputer section was (and in general is) the logical place to begin applying signature analysis. It is the area in which the 5001 exercises most direct control over the system operation. Further, the elements of the block are extremely common and hence can be easily checked out with the 5001's preprogrammed tests.

The 5001 was connected to the 1377 as described earlier, and the 5004's start, stop, and clock inputs were connected to the Stimpod. The 5004 was set to operate from the rising edge of the clock and start signals and the falling edge of the stop signal. For most of the tests performed, this was the only setup needed.

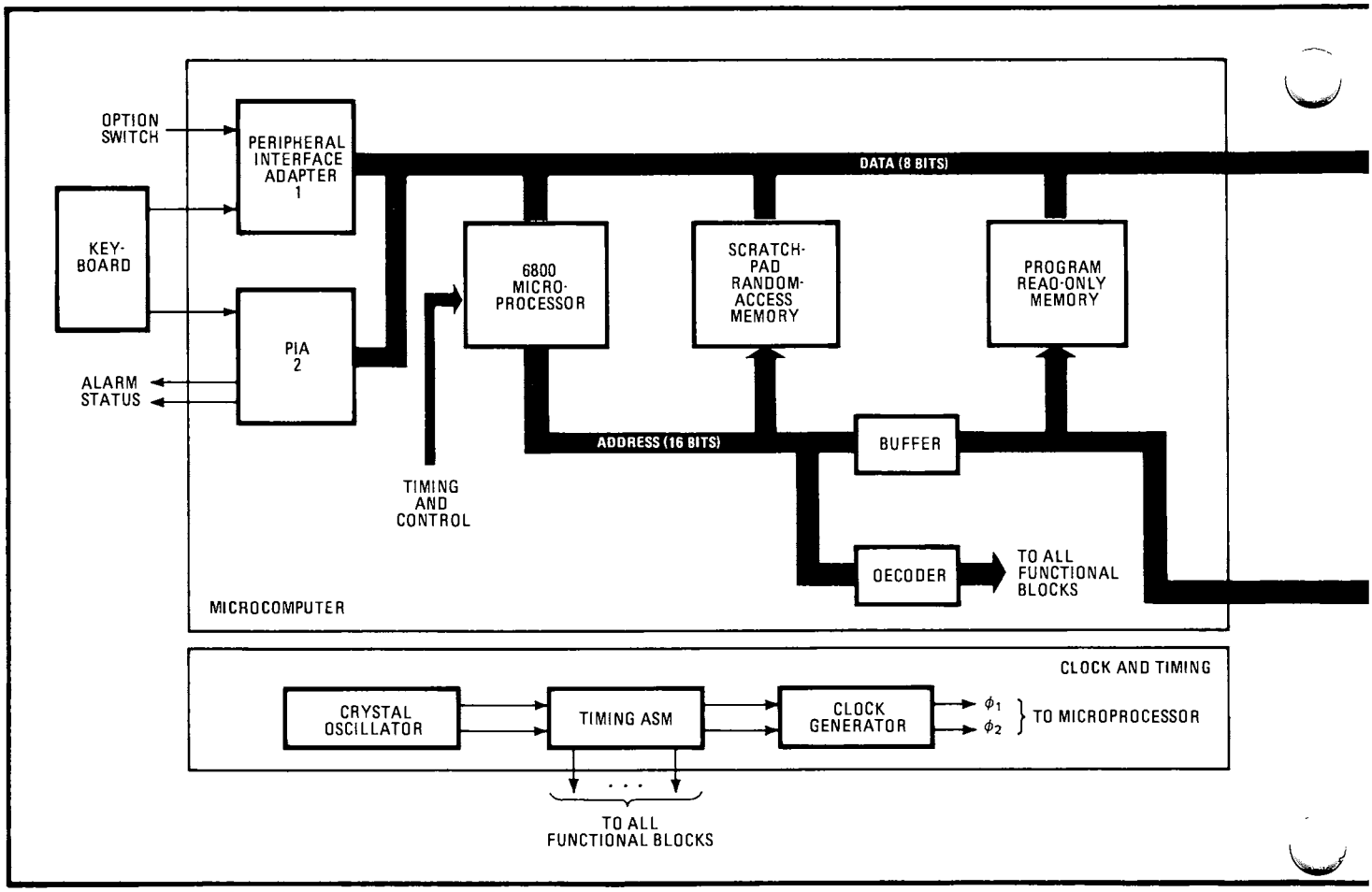
Test 00 (see Table 1) was the first performed. It checks the 6800's functionality by running it through its entire instruction set, except the wait-for-interrupt instruction, while it is isolated from the board. Other instructions in the test sequence serve to verify the interrupt line operation and the processor's ability to service interrupts.

Checking out the kernel

While the test is running, the functioning of the kernel—the processor, its clock and power—can be verified by taking a single signature. If the kernel is operating, holding the 5004's data probe to the processor's +5-volt supply produces a bit stream of 1s, which results in a signature of $28PH_{16}$. This provides an 80% confidence level that the processor is good.

To provide an even higher confidence level (95%), each pin of the 6800 can be probed while the test is run. This will result in the signatures shown in Fig. 3 and takes about 30 seconds to perform. Unless other tests indicate that the processor could be at fault, however, this check is not absolutely necessary.

Once the kernel's operation is verified, the next step is to check that the address lines and decoders are operational so that the Stimpod can access other circuits for test. Another preprogrammed test, 01, is used to do this



2. Shared facilities. In the 1377, the microprocessor, display refresh ASM, and the communications ASM share buses and the display memory. A major challenge is finding a way to independently analyze how these blocks work through the shared facilities.

check. In this free-run test, the CPU runs through its entire address repertoire, the analyzer's data probe is placed on each of the address lines, and a signature is obtained for each (Table 2). These signatures will be the same for any 6800 processor tested in this way.

Using the same test, signatures can be obtained at the outputs of the address decoders, but since their output is product-dependent, the signatures will vary from one design to another. After the ability of the processor, and thus the Stimpod, to access the various components of the system has been verified, they can be checked. The most logical area to investigate next is one closest to the processor, such as the memory in which the operating program is contained.

One-signature checking

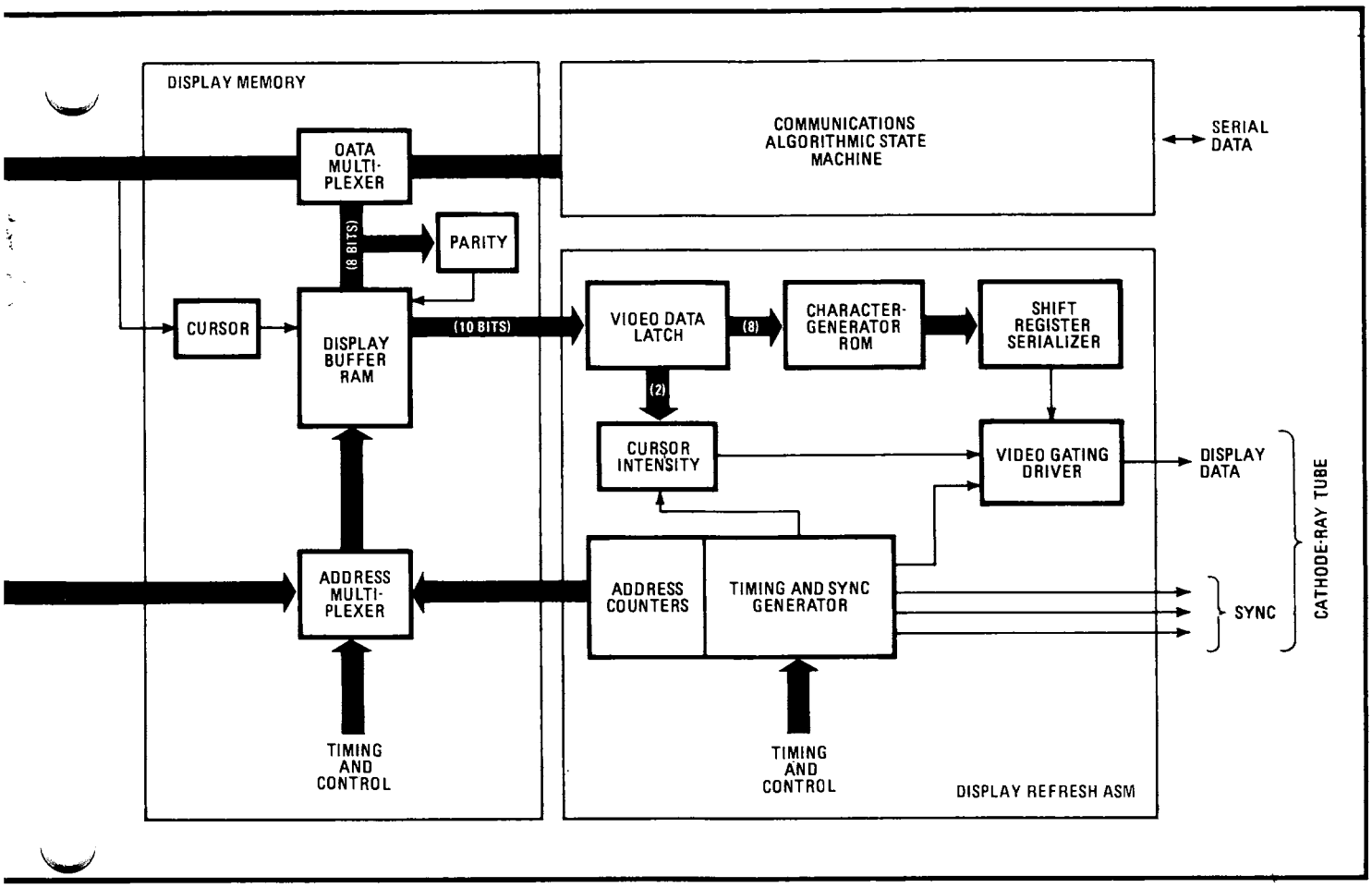
Depending on the product design, the program memory—read-only memory, or ROM—can be checked with a single signature or one signature for each memory device in it. This is done using either preprogrammed test 16 or 17, depending on whether the memory-enable line is active high or low, respectively.

In both tests, the entire contents of the enabled memory are read by the 5001 and formed into a single bit stream, which is output through port X_0 on the Stimpod's left side. By placing the data probe of the signature analyzer on this port, a signature that is unique to the content of the memory being tested can be taken.

Up to three ROMs contain the operating program in the 1377. To check each individually, the 5001's Q-in is tied to a chip-enable line and the contents of the chip are read by the Stimpod; test 16 does this since the 1377's chip-enable lines are active low. Were a block-enable line used in the design, the entire chip set could be checked with a single signature. Then, if an incorrect signature resulted, the contents of each chip could be checked individually.

The 128-byte scratchpad random-access memory was checked next, using test 02, read-write 6800 direct-addressing range. This test uses a sliding-1 pattern with a reverse background fill, which detects address interdependency. After the Stimpod writes the pattern, it reads it and the values written and read determine the clocking signal it supplies to the signature analyzer. Thus, by placing the signature analyzer on the +5-V supply line as in the microprocessor test, a signature is produced that reflects the number of good cells in the RAM. RAMs with less than 256 bytes can be checked using this test; other tests are provided for larger RAMs.

In the 1377, PIA_2 falls within the direct address range of the 6800. Thus, the PIA was partially exercised using test 02, when it was responding as though there were two extra scratchpad RAM cells when it was operating properly. If an incorrect signature appeared, the PIA-enable



line was grounded and the test rerun to see if the fault was actually in the RAM or the PIA.

With the completion of the scratchpad RAM test, the entire microcomputer block with the exception of the PIAs was completely checked. To test the PIAs, it was determined that a custom program would be needed. Rather than change the test setup and insert a custom programmable ROM, it was decided to save time and continue testing whatever could be checked with the preprogrammed tests and the present setup. The custom PIA test will be described later, along with a discussion of other custom tests.

The display memory, or display buffer RAM, in the 1377 is a 4,098-by-10-bit store composed of 10 1-K RAMs. This is divided into two pages, each consisting of 2,049 10-bit words. As the microprocessor can handle only bytewide data, two accesses (and therefore two addresses) are needed to obtain the data set; one address maps to the cursor portion of the stored data, while the other maps to the character portion.

Since the display memory address range is therefore large, two preprogrammed tests were first used to check its operation: test 03, which covers the address range from 0000_{16} to $3FFF_{16}$, and test 04, from 4000_{16} to $7FFF_{16}$. Before these tests were run, the display refresh was disabled by grounding a single pin, in order to prevent it from interfering with the 5001's control of the display RAM.

As shown in Table 1, these tests cause a number of different patterns to be read from and written to the RAM, thoroughly checking its operation. While the tests are run, the analyzer's data probe is set on each of the data-in and data-out pins of the two-port RAMs. The test resulted in only 20 signatures being generated for the entire display RAM.

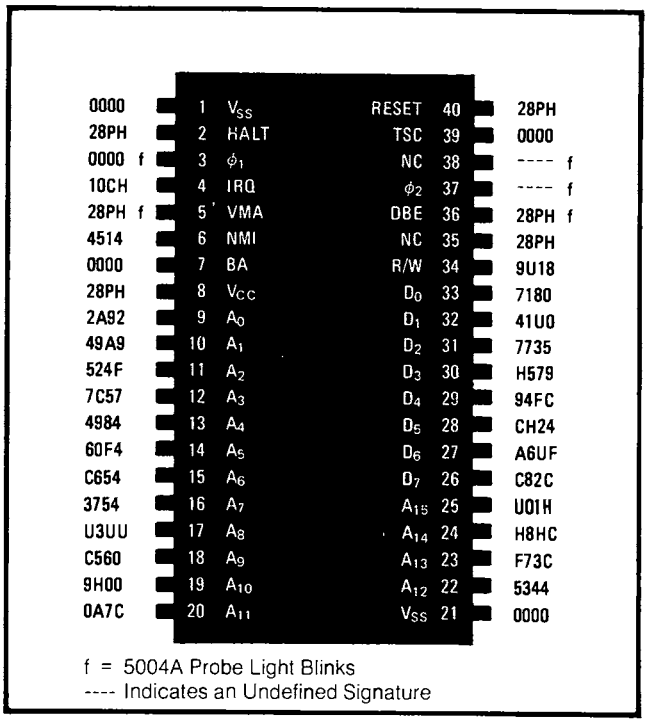
Though the preprogrammed routines were well suited to testing the display memory, it was subsequently found that a custom program needed to test the display refresh circuitry could also check display-memory operation, and was more effective in that it resulted in the need for even fewer signatures. This test will be described later, together with other custom programs.

Clock and timing

As previously noted, the clock and timing circuitry of the 1377 terminal is elaborate and takes up a considerable portion of the board's electronics. Even so, it was a relatively straightforward matter to derive a set of signatures for it.

Since the circuit is self-stimulating, it runs asynchronously and independently of the processor. For this reason, the start, stop, and clock inputs to the 5004 were moved from the Stimpod to the circuit itself; it was not necessary to reset the edges for these inputs.

Carrying out the tests at the clock rate of the circuit was very desirable; taking signatures in this way is simpler and the results are more easily interpreted. The 1377's clock rate is very high—greater than 17 mega-



3. 6800 pin signatures. Running the 5001's test 00 on a good 6800 yields the above set of signatures. A good signature at pin 2 is enough to indicate that the processor kernel is operational; other signatures can be taken if further testing indicates a problem.

hertz—but with option H02, the 5004 is able to test at rates up to 18 MHz. With HP's latest signature analyzer, the 5005 [*Electronics*, Nov. 20, 1980, p. 44], rates of up to 20 MHz can now be checked.

Before checking the circuit, two preprogrammed tests were run on the Stimpod. Test 09 was run first to fill the display RAM with a checkerboard pattern; this predictable pattern allowed some stable signatures to be taken in the area of the character generator. Before any signatures were taken, however, test 32 was run. This test tells the processor to read a 2-K address range that does not affect the shared buses and thus limits its activity to an area removed from the one to be checked. Thus, timing circuits outside the actual clock generator itself could also be investigated.

The 1377's clock and timing circuit is a long, chain-like circuit with a fair amount of feedback. It was therefore necessary to move outside the loop to verify correct inputs. Though it may be necessary to break a loop when extensive feedback exists, it is not absolutely required. When provision is made for resetting the counter chain, as it was in the 1377, it is often not necessary to break it, particularly if the loop contains only two or three elements. Isolating a fault to such a loop is then sufficient, permitting the faulty element to be easily found with the signature analyzer's built-in logic probe.

As noted previously, two areas of the board required some custom approaches to generating signatures: the PIAs and the display-refresh circuitry. Both tests involved programming a 2716 erasable PROM, but this was the only thing necessary for testing the display-refresh ASM.

*See Appendix A

TABLE 2: 6800 ADDRESS BUS SIGNATURES FOR TEST 01

Line	Signature	Line	Signature
A_0	UUUU	A_8	7791
A_1	FFFF	A_9	6321
A_2	8484	A_{10}	37C5
A_3	P763	A_{11}	6U28
A_4	1U5P	A_{12}	4FCA
A_5	0356	A_{13}	4868
A_6	U759	A_{14}	9UP1
A_7	6F9A	A_{15}	0002

For the PIA test, jumpers were needed to ensure complete exercise of the devices; that was not due to any deficiency in the test, but rather to the layout and partitioning of the design itself. Accessing several lines connected to the device required looping back to other lines in order to thoroughly stimulate the device, and this looping would have been required for other test techniques as well.

The custom program developed for the PIAs is of a general nature and could be used for other such devices. It is a 150-byte program and copies of it will be made available this year by Hewlett-Packard.* The reason it is not included in the preprogrammed tests is that, though general, it requires that the user specify the address and the port functions (input or output) specific to his or her application. Given that information, the program is simple to adapt to a design.

Testing display-refresh

Two levels of display-refresh testing—functional and diagnostic—were provided for the 1377 with one custom program. This program initialized the display RAM with all possible display characters and some nondisplayed control sequences. It was, in effect, similar to the "quick brown fox . . ." type of test often used in data communications testing.

For both tests, the framing and clock signals input to the signature analyzer were taken from the display-refresh ASM, since it runs asynchronously with respect to the microprocessor. The clock signal for the functional test was taken from the dot-clock that controls the video gating drivers. Using this clock, the analyzer can verify that about 40% of the terminal's circuits are functional with only one signature, which is taken by placing the analyzer's data probe on the display-refresh ASM's video output. This quick and easy go/no-go test also provides a full CRT display.

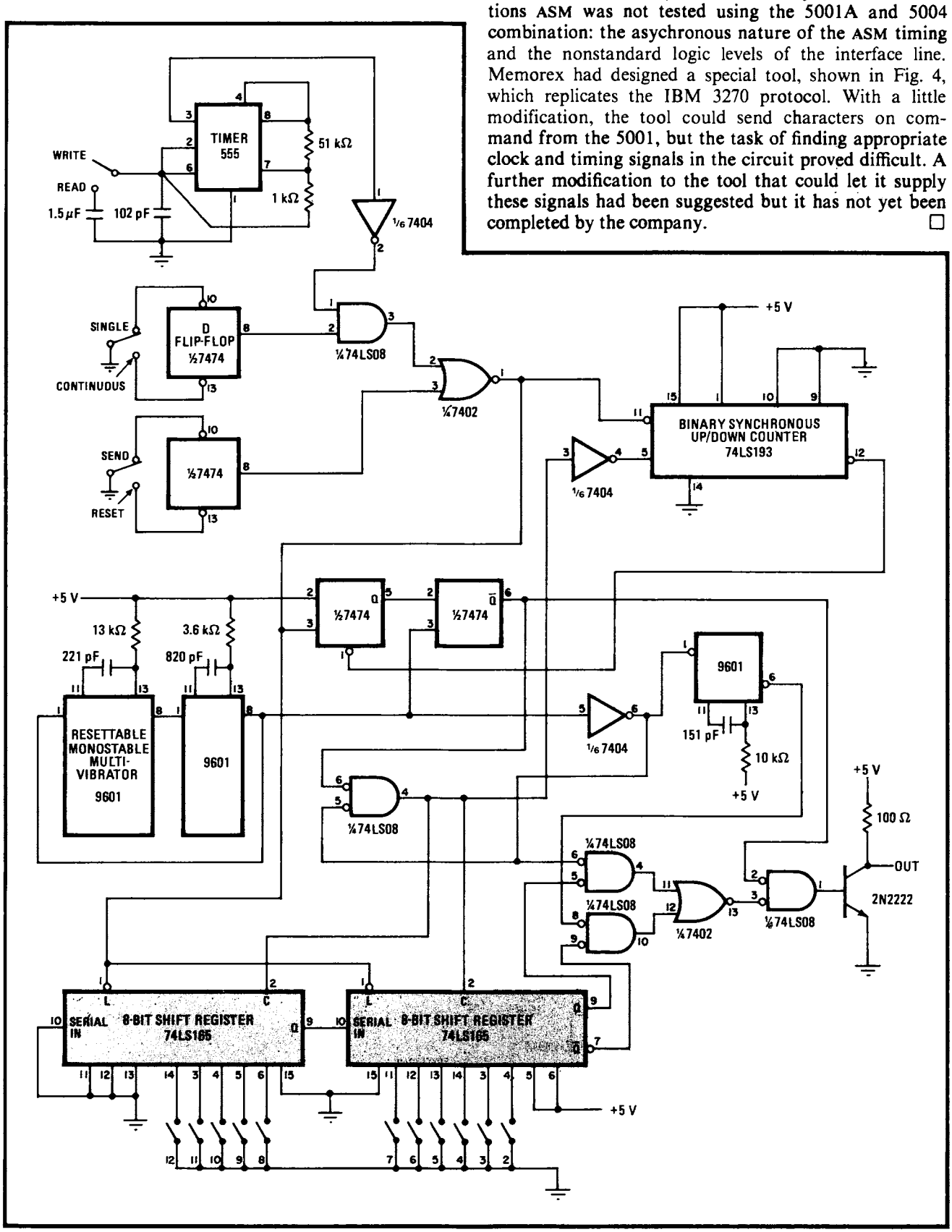
It should be noted that with this custom program both the microcomputer and display-refresh ASM are running concurrently, just as they would in normal interleaved operation. For the diagnostic check of the display-refresh ASM, the signature analyzer's clock is moved to the character clock, which is synchronous with direct memory accesses performed by that ASM. Thus, by using this clock to check activity on the system buses, signatures are generated only when the display-refresh is using them—the interleaved operation is separated for testing.

Both the functional and diagnostic approaches used in

4. Serial stimuli. For checking out the serial interface (the communications ASM) of the 1377, Memorex has been using the in-house tester, which simulates IBM 3270 protocol. Setting toggle switches causes two 8-bit registers (shaded) to create a test word.

the 1377 could easily be applied to other products. They may, in fact, prove invaluable in testing the increasing number of processor-based CRT products.

There were basically two reasons why the communications ASM was not tested using the 5001A and 5004 combination: the asychronous nature of the ASM timing and the nonstandard logic levels of the interface line. Memorex had designed a special tool, shown in Fig. 4, which replicates the IBM 3270 protocol. With a little modification, the tool could send characters on command from the 5001, but the task of finding appropriate clock and timing signals in the circuit proved difficult. A further modification to the tool that could let it supply these signals had been suggested but it has not yet been completed by the company. □



APPENDIX A

CUSTOM PIA (PARALLEL INTERFACE ADAPTER) TEST PROGRAM LISTING

This is a listing of a custom stimulus program written for use with the HP 5001A Microprocessor Exerciser. The program stimulates the two PIAs in the Memorex 1377 Display Terminal. Signature analysis is then used to verify operation of the PIAs and surrounding circuits and troubleshoot when necessary. The two PIAs are exercised independently. Jumpers are used to connect several outputs to inputs in order to thoroughly exercise the devices. The program takes about 150 bytes of code, slightly larger than required due to its generality.

This program is not a preprogrammed routine of the 5001A Microprocessor Exerciser. Please note that each line of each PIA port can be programmed as an input or an output. It is not possible to know that information in advance. This custom program configures these ports to match the circuits of the Memorex terminal. We wrote it in a general manner so that it could be easily adapted to the unique way you configure your PIAs. You will need to modify several lines of this program to do this. These lines have been identified in the listing with the symbol "<<<". We've included this program listing as an example of the size and simplicity of typical custom programs that can be written for the 5001A. For information on how to write custom programs for the 5001A Microprocessor Exerciser, see the "Operating and Programming Manual".

```
                ************************************************
              * THIS IS A SAMPLE PROGRAM FOR THE PIA TEST IN THE MEMOREX
              * 1377 DISPLAY TERMINAL. THE PROGRAM IS INTENDED TO
              * RESIDE IN THE CUSTOM ROM SOCKET OF THE HP-5001A
              * MICROPROCESSOR EXERCISER
              *
FFFF          INPUT    EQU    $FFFF  AND READ   * IN THE EXERCISER
FFFF          CONTROL  EQU    $FFFF  AND WRITE  * IN THE EXERCISER
FFFE          USERBUS  EQU    $FFFE  AND READ   * IN THE EXERCISER
FFFE          OUTPUT   EQU    $FFFE  AND WRITE  * IN THE EXERCISER
              *
1000          PIA1     EQU    $1000  IDENTIFIES BASE ADDRESS OF PIA  1
0080          PIA2     EQU    $0080  IDENTIFIES BASE ADDRESS OF PIA  2
              ************************************************
0000                   ORG    $0000
              *
              ************************************************
              * THIS PROGRAM MUST BE ENTERED AT THE "PIATEST" ENTRY
              * POINT WITH THE INDEX REGISTER CONTAINING THE BASE
              * ADDRESS OF THE PIA TO BE TESTED. ENSURE THAT THE JUMP AT
              * THE LAST LINE CAUSES THE PROGRAM TO EXECUTE AT A RANGE
              * OF ADDRESSES THAT OVERLAY THE PREVIOUSLY TESTED ROM IN
              * YOUR PRODUCT. NOTE TOO THAT THE INPUT/OUTPUT CON-
              * FIGURATION BYTES WILL REQUIRE CHANGE IN YOUR PRODUCT.
              ************************************************
003D  CE TEST1    LDX    $1000  SET BASE ADDRESS OF PIA  1 <<<<<<<<
003E  10
003F  00
```

```
0040  7E          JMP     PIATEST+$6800   GO TO TEST OF PIA <<<<<<<<
0041  68
0042  49
            ********************************************************
0043  CE TEST2   LDX       $0080  SET BASE ADDRESS OF PIA  2 <<<<<
0044  00
0045  80
0046  7E          JMP      PIATEST+$6800  GO TO TEST OF PIA <<<<<<<<
0047  68
0048  49
            ********************************************************
0049  86 PIATEST LDA  A     $1C    START=STOP=0, EMIT SA
                                       READ CLOCKS
004A  1C
004B  B7          STA  A   CONTROL
004C  FF
004D  FF
004E  86          LDA  A     $1F    START=STOP=1, EMIT SA RD AND
                                       WR CLK
004F  1F
0050  B7          STA  A   CONTROL
0051  FF
0052  FF
0053  4F          CLR  A             CLEAR DDRA ACCESS BIT=0
0054  A7          STA  A     1,X     FOR A-SIDE OF PIA
0055  01
0056  86          LDA  A     $FF     SET DDR TO ALL OUTPUTS <<<<<<<<<<
0057  FF
0058  A7          STA  A     0,X     ON A-SIDE OF PIA
0059  00
005A  86          LDA  A     $04     SET    DDRA ACCESS BIT=1
005B  04
            *
            * NOW WE HAVE ACCESS TO DATA REGISTER ON THE A-SIDE
            *
005C  A7          STA  A     1,X     FOR A-SIDE OF PIA
005D  01
005E  4F          CLR  A
005F  A7 STIMA1   STA  A     0,X     WRITE PATTERN TO OUTPUTS
0060  00
0061  E6          LDA  B     0,X     READ BACK RESULTING LEVELS
0062  00
            * NOTE THAT THE A-SIDE IS UNBUFFERED, AND RESULTING DATA
            * IS THE SAME AS ACTUALLY SEEN BY THE OUTPUT OF THE PIA
            *
0063  4C          INC  A             INCREASE PATTERN BY ONE
0064  26          BNE      STIMA1    CONTINUE UNTIL 256 PATTERNS WRITTEN
0065  F9
0066  20          BRA       BSIDE
0067  02
0068  20 BACKPIA  BRA      PIATEST
0069  DF
            * NOW THE B-SIDE
```

```
006A  86 BSIDE   LDA  A    #01   SET X0, CLEAR X1 BITS IN
006B  01
006C  B7         STA  A  OUTPUT  OUTPUT LATCH ON 5001A
006D  FF
006E  FE
006F  4F         CLR  A          CLEAR DDRB ACCESS BIT=0
0070  A7         STA  A    3,X   FOR PIA  B-SIDE
0071  03
0072  86         LDA  A    #7F   SET DATA DIRECTION 7 OUT 1 IN <<<
0073  7F
0074  A7         STA  A    2,X   FOR PIA   B-SIDE
0075  02
0076  86         LDA  A    #04   SET DDRB ACCESS BIT=1
0077  04
0078  A7         STA  A    3,X   FOR PIA   B-SIDE
0079  03
                 *
                 * NOW THE ACCESS BIT IS SET, WE CAN ACCESS THE DATA
                 * REGISTER
                 *
007A  4F         CLR  A          RESET PATTERN TO ZERO
007B  A7 STIMB1  STA  A    2,X   FOR PIA   B-SIDE
007C  02
007D  4C         INC  A          NEXT PATTERN
007E  26         BNE     STIMB1 UNTIL ALL 256 POSSIBLE ARE WRITTEN
007F  FB
                 *
                 * HERE WHEN ALL PATTERNS WRITTEN, READ THE INPUT PIN
                 *
0080  A6         LDA  A    2,X   FOR PIA   B-SIDE
0081  02
0082  C6         LDA  B    #03   GET INPUT BIT HIGH, LEAVE RESET HIGH
0083  03
0084  F7         STA  B  OUTPUT
0085  FF
0086  FE
0087  A6         LDA  A    2,X   FOR PIA   B-SIDE
0088  02
                 *
                 * NOW GO ON TO TEST CA1,CA2,CB1,CB2
0089  20         BRA     TESTCAB
008A  02
008B  20 PIABACK BRA     BACKPIA
008C  DB
                 *
                 * HERE WITH CA1 AND CA2 TIED TOGETHER IN THE KEYBOARD
                 * CONNECTOR, SINCE THE KEYBOARD IS NOT PRESENT.
                 *
008D  86 TESTCAB LDA  A    #34   SET CA2 LOW FIRST, NO EFFECT
008E  34
008F  A7         STA  A    1,X
0090  01
```

```
0091  4C          INC  A          NOW CONFIGURE CA1 FOR FALLING EDGE
0092  A7          STA  A    1,X
0093  01
0094  A6          LDA  A    1,X   READ STATUS ON A-SIDE OF PIA
0095  01
0096  C6          LDA  B    #3D   FORCE A RISING EDGE ON CA1 VIA CA2
0097  3D
0098  E7          STA  B    1,X   WRITE TO CONTROL REG A-SIDE
0099  01
009A  A6          LDA  A    1,X   READ STATUS ON A-SIDE OF PIA
009B  01
009C  C6          LDA  B    #35   FORCE A FALLING EDGE ON CA1 VIA CA2
009D  35
009E  E7          STA  B    1,X   WRITE TO CONTROL REG A-SIDE
009F  01
00A0  A6          LDA  A    1,X   READ STATUS ON A-SIDE OF PIA
00A1  01
00A2  C6          LDA  B    #36   PROGRAM FOR POS EDGE INTERRUPT ON IRQA
00A3  36
00A4  E7          STA  B    1,X   WRITE TO CONTROL REG A-SIDE
00A5  01
00A6  A6          LDA  A    1,X   READ STATUS ON A-SIDE OF PIA
00A7  01
00A8  C6          LDA  B    #3E   FORCE A RISING EDGE ON CA1 VIA CA2
00A9  3E
00AA  E7          STA  B    1,X   WRITE TO CONTROL REG A-SIDE
00AB  01
00AC  A6          LDA  A    1,X   READ STATUS ON A-SIDE OF PIA
00AD  01
00AE  C6          LDA  B    #36   FORCE A FALLING EDGE ON CA1 VIA CA2
00AF  36
00B0  E7          STA  B    1,X   WRITE TO CONTROL REG A-SIDE
00B1  01
00B2  A6          LDA  A    1,X   READ STATUS ON A-SIDE OF PIA
00B3  01
00B4  C6          LDA  B    #37   PROGRAM FOR NO PULL ON IRQA
00B5  37
      *                           IF INTERRUPT OCCURS
00B6  E7          STA  B    1,X   WRITE TO CONTROL REG A-SIDE
00B7  01
00B8  A6          LDA  A    1,X   READ STATUS ON A-SIDE OF PIA
00B9  01
00BA  C6          LDA  B    #3F   FORCE A RISING EDGE ON CA1 VIA CA2
00BB  3F
00BC  E7          STA  B    1,X   WRITE TO CONTROL REG A-SIDE
00BD  01
00BE  A6          LDA  A    1,X   READ STATUS ON A-SIDE OF PIA
00BF  01
```

```
00C0  C6          LDA  B     #37    FORCE A FALLING EDGE ON CA1 VIA CA2
00C1  37
00C2  E7          STA  B     1,X    WRITE TO CONTROL REG A-SIDE
00C3  01
00C4  A6          LDA  A     1,X    READ STATUS ON A-SIDE OF PIA
00C5  01
            *
            * HERE WHEN DONE WITH SIDE A CA1 AND CA2
            *
            *
00C6  86          LDA  A     #34    SET CB2 LOW FIRST, NO EFFECT
00C7  34
00C8  A7          STA  A     3,X
00C9  03
00CA  4C          INC  A            NOW CONFIGURE CA1 FOR FALLING EDGE
00CB  A7          STA  A     3,X
00CC  03
00CD  A6          LDA  A     3,X    READ STATUS ON B-SIDE OF PIA
00CE  03
00CF  C6          LDA  B     #3D    FORCE A RISING EDGE ON CA1 VIA CB2
00D0  3D
00D1  E7          STA  B     3,X    WRITE TO CONTROL REG B-SIDE
00D2  03
00D3  A6          LDA  A     3,X    READ STATUS ON B-SIDE OF PIA
00D4  03
00D5  C6          LDA  B     #35    FORCE A FALLING EDGE ON CA1 VIA CB2
00D6  35
00D7  E7          STA  B     3,X    WRITE TO CONTROL REG B-SIDE
00D8  03
00D9  A6          LDA  A     3,X    READ STATUS
00DA  03
00DB  4F          CLR  A            CLEAR DDRA ACCESS BIT=0
00DC  20          BRA     PIABACK   REPEAT TEST
00DD  AD
```

HEWLETT
PACKARD

For more information, call your local HP Sales Office or nearest Regional Office: **Eastern** (201) 265-5000; **Midwestern** (312) 255-9800; **Southern** (404) 955-1500; **Western** (213) 970-7500; **Canadian** (416) 678-9430. Ask the operator for instrument sales. Or write Hewlett-Packard, 1501 Page Mill Road, Palo Alto, CA 94304. **In Europe:** Hewlett-Packard S.A., 7, rue du Bois-du-Lan, P.O. Box, CH 1217 Meyrin 2, Geneva, Switzerland. **In Japan:** Yokogawa-Hewlett-Packard Ltd., 29-21, Takaido-Higashi 3-chome, Suginami-ku, Tokyo 168.

02-5952-7622

APRIL 1981

PRINTED IN USA