APPLICATION NOTE 167-5
DATA DOMAIN MEASUREMENT SERIES

Trouble-shooting in the data domain is simplified by logic analyzers.

reprinted from Electronics Magazine

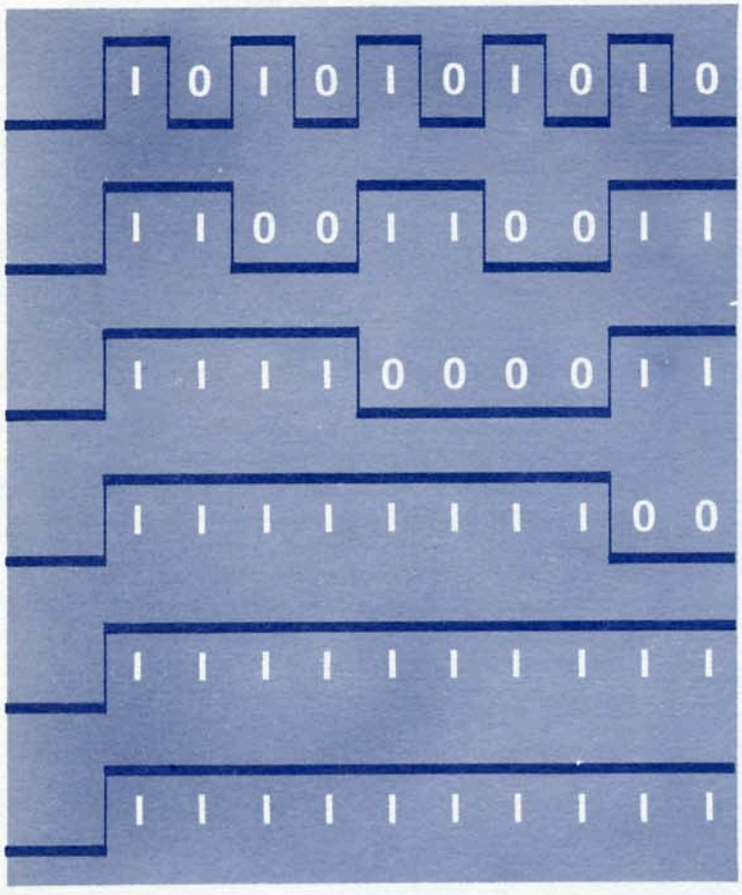


HEWLETT hp PACKARD

Troubleshooting in the data domain is simplified by logic analyzers

New instruments for checking out digital circuits pinpoint defects by monitoring bits in much the same way that oscilloscopes work in time and spectrum analyzers in frequency to find defects in analog circuits

by William Farnbach,
Hewlett-Packard Co., Colorado Springs, Colo.



☐ The increasing use of digital circuits in new products has created a concurrent need for new equipment to pinpoint and troubleshoot defects. Because more and more of these new products manipulate data, they operate in the data domain, rather than the time or frequency domains that are characteristic of analog circuitry. Instruments that analyze circuits in the time and frequency domains simply cannot cope with digital data manipulations.

Now, manufacturers are introducing instruments that operate in the data domain. Called logic analyzers, these instruments monitor bits, words, addresses, and instructions in the same way that oscilloscopes monitor time and spectrum analyzers monitor frequencies. Two examples of these logic analyzers are the Hewlett-Packard models 1600A and 1607A (Fig. 1).

These new instruments are ideal for many applications. One could be finding the defect in a processor-controlled cash register that won't do anything in its normal operating mode but initialize itself. Yet, when the machine is single-stepped, it always operates properly. Another malfunction that a data-domain instrument would easily pinpoint could be in a processor-based traffic-signal controller that causes all the lights at a six-way intersection to blink red about every two days at the beginning of the rush hour.

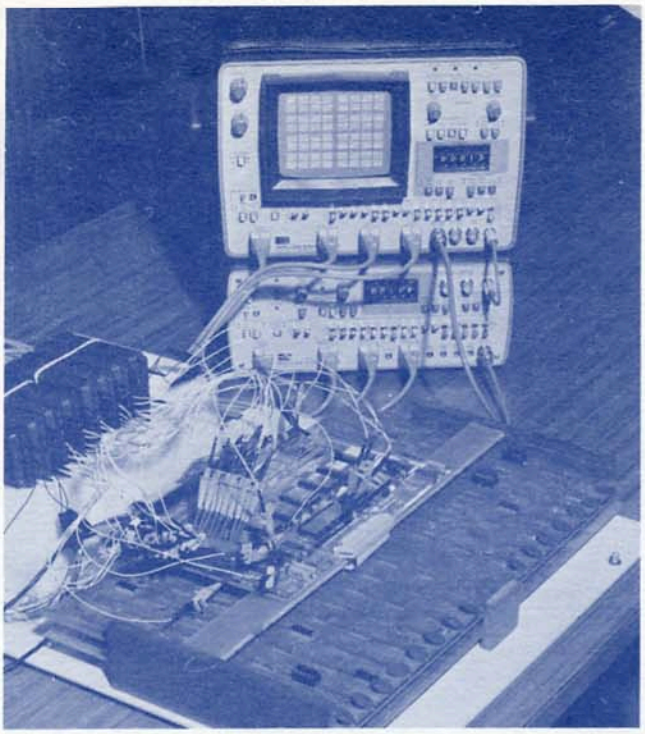
When a repair technician arrives, the program counter in the processor is pointing to, say, 37416—undefined territory. No random-access or read-only memory location in the controller corresponds to that address. After the technician takes the processor back to the shop for repair, it repeats the error every couple of days. The problem could be solved fairly simply if the point in the program from which the processor jumped could be determined.

Both of the machines described above can be classified as data manipulators. The processor sends some data—the program address—to a ROM or RAM and receives some data back—the next instruction to be executed. To define the operation of such machines, the repairman needs the data transactions, program addresses, instructions for the central processing unit, and input/output controls, as well as the sequence of program instructions.

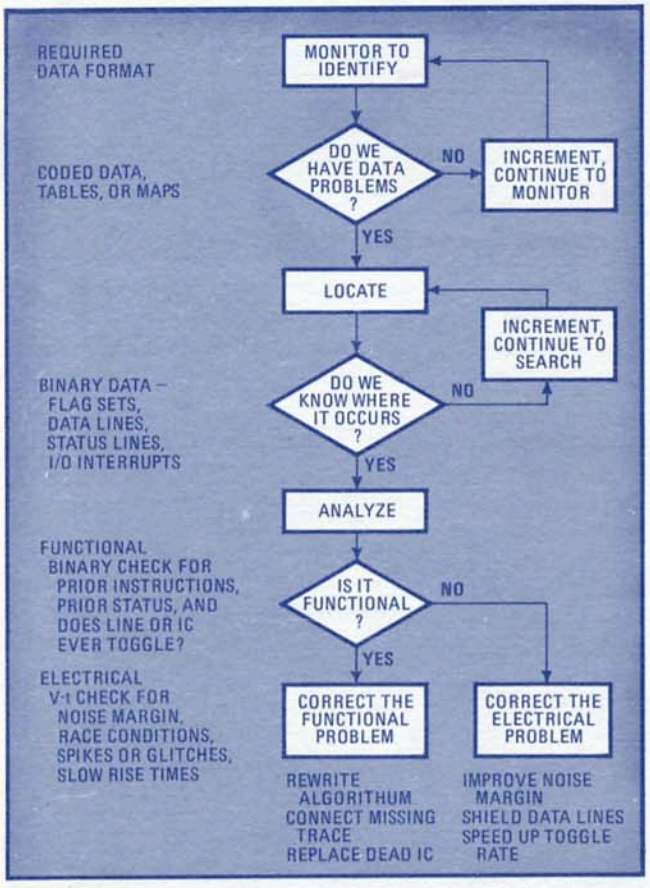
Analysis of these data transactions is necessary for data manipulators to be tested, debugged, and repaired. This analysis requires working in the data domain, and

This article, which discusses the application of data-domain instruments to solving problems in digital circuits, is the second of a two-part series. The first installment, written by Charles House of Hewlett-Packard Co., defined the data domain and explained its importance in digital design and troubleshooting. That article appeared in the May 1 issue.

Reprinted from Electronics, May 15, 1975, copyright 1975 by McGraw Hill Inc.



1. Analyzing the data domain. These two new logic-state analyzers are the Hewlett-Packard models 1600A (top) and 1607A (bottom).



2. Troubleshooting pattern. The steps required to find and correct an error in a data-domain product is represented by this flow chart. Much of this analysis involves functional parameters, which can usually be displayed clearly on a logic-state analyzer.

it requires instrumentation especially designed for the task.

Data-domain instruments—generally classified as logic analyzers—are useful in analyzing data manipulators because they present in a readable format the information that is most important in understanding data manipulators: the sequence of program flow, and data transactions. The three basic classes of logic analyzers are logic-timing analyzers, logic-state analyzers, and logic-trigger generators.

Considering data-domain instruments

Most logic-timing analyzers display digital signals as replicas of voltage-versus-time functions and are most useful for analyzing parametric faults in digital systems. Logic-state analyzers are similar, but they display signals in binary form—either 1s or 0s—on a cathode-ray tube or on and off states of lamps. They are most suitable for functional checks. Logic-trigger generators are used to trigger displays on oscilloscopes upon receipt of specified data words.

Since, to a large extent, troubleshooting a digital system involves the examination of a system's functional behavior (Fig. 2), logic-state analyzers are the most commonly required instruments.

Logic-state analyzers must handle program flow and data transactions as nearly as possible the same way the device under test processes the data. This requirement is analogous to the way oscilloscopes must show time-domain signals with minimum nonlinearity, overshoot, or bandwidth degradation, and spectrum analyzers must show frequency-domain signals with a minimum of noise, harmonic distortion, or amplitude distortion.

Defining logic-state analyzers

To handle the data as the system under test does, the logic-state analyzer must meet five requirements:

- The data must be read and presented in binary form (Fig. 3). This format is much more easily read than the equivalent time-domain representation (Fig. 4). The threshold, or division between a 1 and a 0, should be as close to the threshold of the logic used in the machine under test as possible. Methods of pointing out such information as the presence of glitches or providing some dual-threshold arrangement for detecting indeterminate logic levels fall in the realm of the time domain and are needed only after errors in the data transactions are located by a logic-state analyzer.
- The logic-state analyzer should have enough inputs so that the entire data word can be monitored at once. If only part of the word can be monitored, the data cannot be completely defined. Such a shortcoming would be the data-domain equivalent of an oscilloscope trace that goes off-screen vertically, or a spectrum analyzer that covers only half of the frequencies of interest.
- Incoming data must be read into the logic-state analyzer in the same way it is read by the system under test. This means that the data must be clocked into the logic-state analyzer by the same clock—at the same time and with the same slope—that clocks the data in the system under test. In a typical data manipulator, data begins changing almost immediately after a clock edge

and becomes stable again shortly before the next clock edge. This requires that a logic-state analyzer have zero data-hold time (the period data must be held after the arrival of the clock-pulse edge) and the smallest possible data set-up time (the period data must be available before arrival of the clock pulse).

- The logic-state analyzer should change the system under test as little as possible. Test gear should not alter the performance of the system being tested, and the device should be tested in its normal operating mode. A processor won't run the same in a single-step mode as it does on a 5-megahertz clock. Likewise, it is a major alteration to a processor program if outputs, such as print or display commands, are added to trace the program. Such alterations are analogous to operating a high-speed linear amplifier at low speed so that it can be tested with an oscilloscope of limited bandwidth or operating a transmitter at 10% of full power so that a spectrum analyzer will not be damaged by the large input.
- Finally, the logic-state analyzer must be designed to connect to the system to be tested. It is impossible to connect 16 standard oscilloscope probes to one 24-pin dual in-line package. Even if a logic clip were used, the weight of the 16 probes would drag the clip off the integrated circuit. The probes must be far smaller than standard oscilloscope probes. In fact, probes should be small enough to connect to every lead in a row of DIP leads spaced on 0.1-inch centers, the standard spacing.

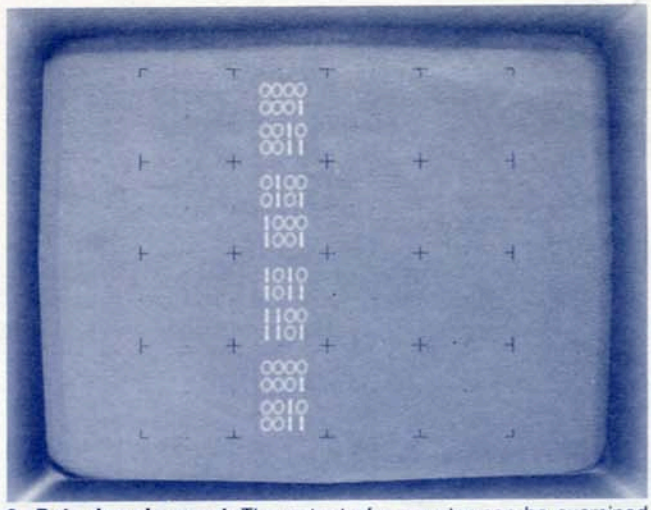
Generating usable displays

A logic-state analyzer must also be designed to make the display readable and useful. To present a useful data-sequence table, the logic-state analyzer must have some sort of trigger arrangement for selecting a particular segment of the data for viewing. Since the information displayed is in binary form, the obvious trigger would be some data pattern or data word. When that predetermined trigger matches a data word, the storage of data should be started or stopped. In tracing program flow, for example, the trigger word may be the address of a subroutine entry point. In troubleshooting a digitally controlled printer, the trigger word may be the code for a letter "A" that is always misprinted.

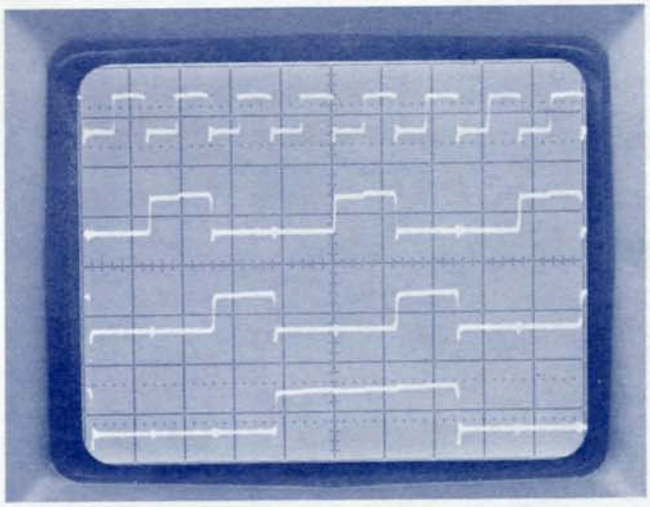
To make the data-sequence table easy to read, the logic-state analyzer should associate the maximum positive voltage with a 1 and the maximum negative level with a 0, or vice versa, so that the technician does not have to make this translation from voltage to data. In addition, the common way to write a sequence of data flow is with the most significant bit on the left and the least significant bit on the right with the first word at the top and each succeeding word under the word before it, as the table in Fig. 5 is written.

The data-sequence table can also be made easier to read by grouping the rows and columns of data into blocks with intervening spaces. The columns might be grouped into blocks of three if the data is to be read as octal or blocks of four if the data is to be read as hexadecimal or binary coded decimal. The rows should also be broken into groups by spaces to make it easier to follow one row across the screen.

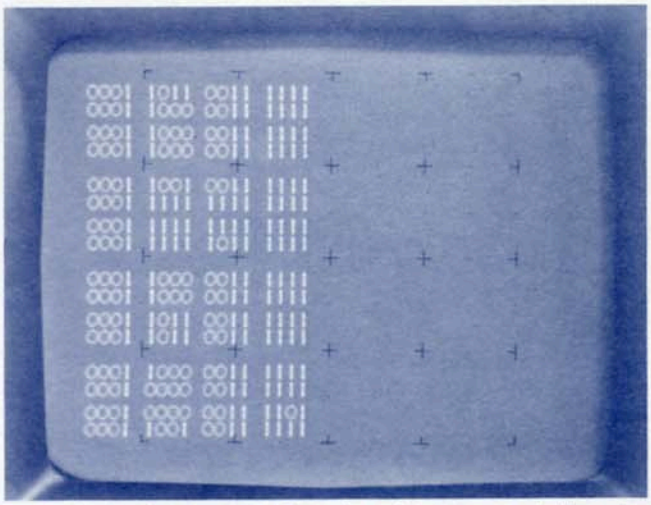
The Hewlett-Packard 1600A and 1607A are two new



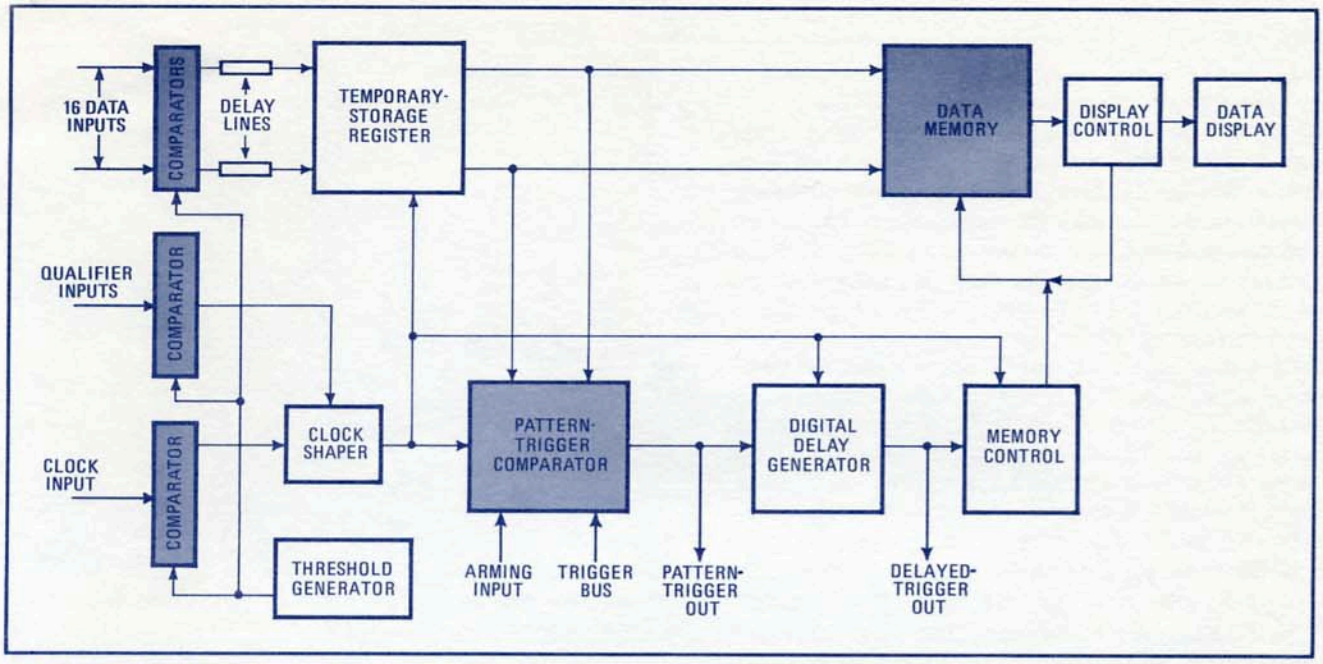
3. Data-domain count. The output of a counter can be examined easily if it is displayed as binary digits.



4. Time domain. The time-domain representation of a counter output yields more information about circuit operation than the equivalent data-domain pattern, but is more difficult to interpret if only logic-state information is required.



5. Arranging the sequence. To make a data-sequence table easier to read, a logic-state analyzer should display the information with the most significant bit on the left and the first word at the top.



6. Looking inside. A logic analyzer, controlled by a pattern-trigger comparator, displays input data that has been held in a data memory.

logic-state analyzers. How they meet the requirements for data-domain instrumentation is illustrated by the block diagram of Fig. 6.

Sixteen data inputs are fed to comparators, which translate the signals into binary form. The threshold for the comparators may either be a 1.5-v transistor-transistor-logic level or a variable threshold from −10v to +10 v. For convenience, there is a test point on the front panel where the variable threshold voltage may be measured accurately.

The digitized data then passes through delay lines to a temporary-storage register. The delay lines ensure that the data hold time is zero. With these delay lines and the other delays in the data and clock paths, the data set-up time becomes 20 nanoseconds. The data is clocked into the temporary storage register by the system's clock. The clock shaper, which enables the active clock edge to be selected by a push button, inhibits the storage and display of data if the data on two qualifier inputs does not match the data pattern previously selected by two qualifier-selector switches.

After the data has been clocked into the temporary-storage registers, it is compared to the trigger word set on pattern-trigger switches. These switches, as well as the qualifier-selector switches, have three positions: high, off or "don't care," and low. When a match occurs, a pattern trigger is sent to the digital delay generator and the pattern-trigger output is also available for triggering an oscilloscope.

The arming input is normally disabled; however, if the arming function is active, then the arming input must receive a positive-going edge before a trigger can be generated and delivered to either the delay generator or pattern-trigger output. The arming signal usually is supplied by the trigger output from another 1600A or 1607A. In this mode, two data patterns must occur in sequence before a trigger can be generated. The trigger bus allows the pattern-trigger comparator of a 1600A to be wire-ANDed with the trigger comparator of a 1607A to generate a 32-bit-wide pattern-trigger comparator.

Once the delay generator has received a pattern trigger, it counts off the number of qualified input clocks set by the delay thumbwheels on the front panel. When the preset number of clocks has passed, the delay generator provides a delayed trigger to the memory control and to the delayed trigger output.

When the trigger reaches the memory control, it either initiates or halts the storage of data in the data memory, according to the setting of the trigger-function controls. If the trigger initiates data storage, the next 16 words of data are stored for display.

If the trigger ends the data storage, the memory control rejects triggers until 16 data words are stored in memory so that the data in memory represents the 16 words preceding the receipt of a trigger. In this mode, the logic-state analyzer presents pretrigger information so that conditions leading up to an error can be displayed. After data storage has been completed, control of the data memory is transferred to the display control.

The display control arranges the data into an easily read format for display. This formatting includes dividing the rows of bits into 3- or 4-bit bytes, depending on the byte control, for easy grouping into octal, hex, or BCD characters. Another control allows unused bits to be blanked from the screen. In the repetitive mode, the data is displayed for some time, then control of the data memory is returned to the memory control so that new data may be stored. The length of the display may be set from the front panel for as long as about 5 seconds.

If a longer display is desired, the single-sample mode may be selected. In this mode, the stored data is displayed until a manual reset button is pushed. When the incoming clock rate is less than about 30 hertz, the display control and memory control can share the memory so that data can be displayed as it comes in. This sharing capability eliminates the need to have 16 clock

pulses before any data can be displayed.

The data inputs for the 1600A and 1607A come through a set of four probes (Fig. 7). Three of these probes contain the 16 data channels and two qualifiers. The fourth probe contains the clock input. The inputs may be connected directly to the input leads on a DIP with a grabber, or the grabber may be removed to expose a pin socket that can be connected to a logic clip or standard wire-wrap pin. This arrangement simplifies connection of a large number of leads in a small space.

The self-contained 1600A has a couple of more capabilities than the 1607A. When a 1607A is not connected to the 1600A, the 16 display channels reserved for the 1607A may be used to store data previously captured by the 1600A so that it can be compared to data taken later. In that mode, data may be transferred from the data memory to an auxiliary memory so that subsequent data captured in the data memory can be compared to the stored data. If desired, the data in the data memory can be held indefinitely whenever it does not match the data in the auxiliary memory.

Mapping data flow

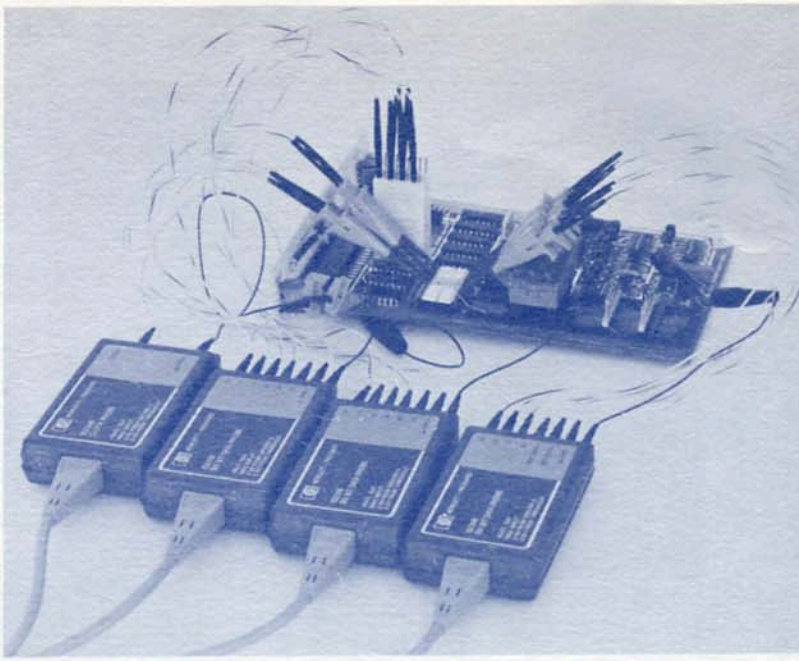
Another advantage of the 1600A is the map display. In the map mode, the CRT displays an array of 2^{16} dots instead of a table of 1s and 0s. Each dot represents one possible combination of the 16 input lines so that any input word is represented by an illuminated dot. An input of all 0s is at the upper left corner of the CRT. An input of all 1s is at the lower right corner of the screen. The dots are interconnected so that the sequence of data transactions can be observed. The interconnecting line gets brighter as it moves toward a new point, thereby showing the direction of data flow.

The map display facilitates observance of the over-all operation of a machine in a repetitive loop. A properly functioning two-decade counter, for example, would be represented by the map of Fig. 8. If the counter were not functioning properly, its representation might be like the map of Fig. 9. In Fig. 9, the counter skips from state 59 (0101 1001) to state 70 (0111 0000) without passing through states 60 through 69. The map simplifies examination of the functional behavior of such circuits and pinpointing of errors.

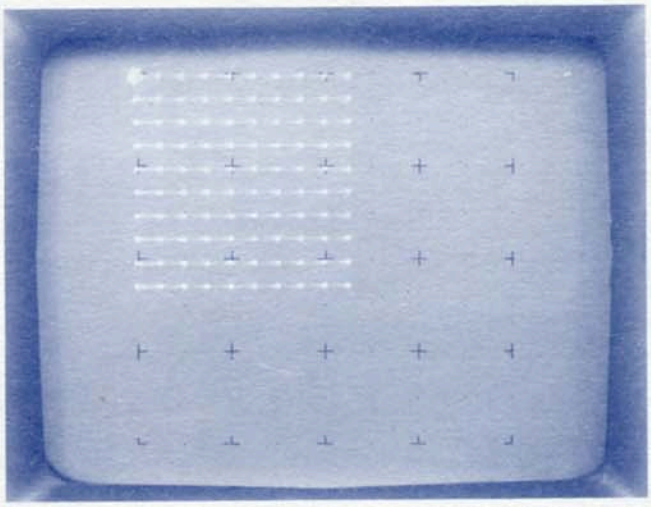
To look at a data table of the counter shown in Fig. 9 or to trigger an oscilloscope where the improper state transition occurs, the 1600A can be switched into the table mode, or the pattern-trigger output of the analyzer can be connected to the external-trigger input of an oscilloscope.

The word that triggers the logic-state analyzer is represented in the map mode by a circle, as shown in Figs. 8 and 9. The position of this trigger-pattern cursor can be controlled through the trigger-pattern-selector switches on the front panel of the 1600A.

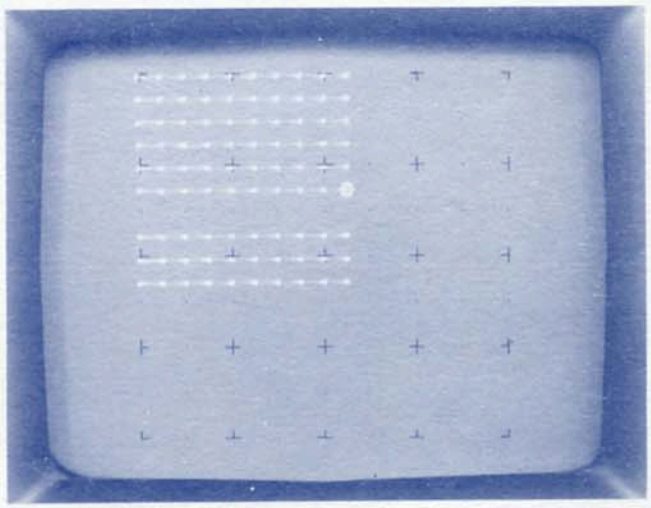
Figure 10 is a block diagram of the 1600A in the map mode. This diagram is similar to Fig. 6. The pattern-trigger comparator and delay generator are not used to control the storage of data in the data memory, but they do still generate pattern and delayed-trigger outputs. The other difference is the way the display is generated. The display controller connects bits 8 through 15 to a



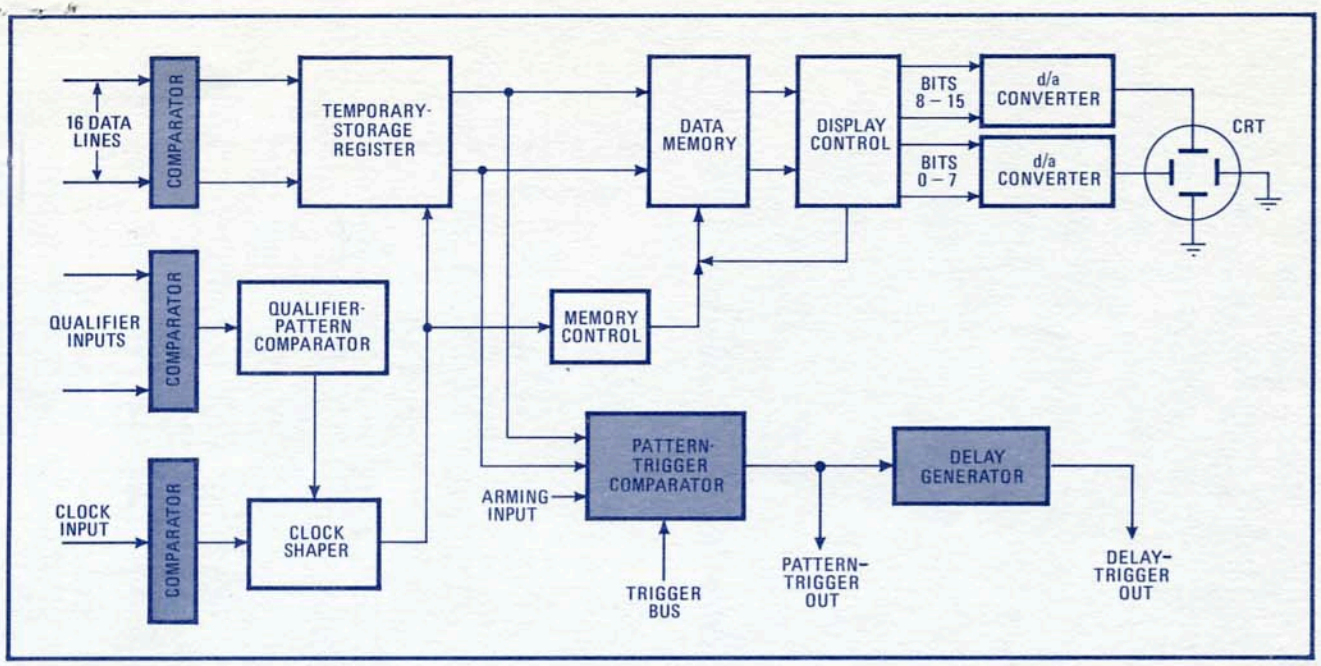
7. Probing for data. High-density multiline digital systems require the kind of compact probes shown here.



8. Mapping. A two-decade counter can be represented by a map of data states. Here, the counter sequence is from zero (upper left) through all intermediate steps to 99 (lower right), then back to zero.



9. Pinpointing a malfunction. Functional faults in a counter show up clearly in a map display. The gap between the sixth and seventh lines displayed here means that the counter skips past 10 normally-required states—60 through 69. No other faults are evident.



10. Operating in the map mode. The pattern-trigger comparator and delay generator are not used to control the storage or data in the 1600A data memory when the unit is in the map mode. However, they are still available for generating pattern and delayed-trigger outputs.

digital-to-analog converter connected to the vertical deflection plates, and bits 0 through 7 to a d-a converter connected to the horizontal plates. This divides the screen into a 2^8 by 2^8 matrix of dots.

Either an HP 1600A or 1607A logic-state analyzer may be used to pinpoint the malfunctions previously cited in the cash register and the traffic-light controller. To troubleshoot the cash register, the technician connects the logic-state analyzer to the program-counter (ROM-address) lines, turns off all of the trigger-word switches, and captures a set of data in the single-sample mode. This data indicates what the cash register is doing. The repairman chooses one of the words displayed as a trigger pattern; then returns the analyzer to the repetitive-sample mode and begins delaying through the loop until the program jumps back on itself.

The program calls for a jump from the initializing routine in locations 0 through 137 to the start of the keyboard scan at 4052. But instead of jumping to 4052, the program jumps to 52. The reason for this does not become apparent until an oscilloscope is triggered from the logic-state analyzer. Because the pull-up on the 4000 bit on the processor is bad, that address line comes up so slowly that the ROM interprets the address as 52 and returns to that instruction. Since the instruction at 52 is a jump to 57, the machine then jumps to 57, altering the program counter permanently.

However, when the processor is single-stepped, the faulty address line has time to rise, and the jump to 4052 is executed correctly. Although this malfunction can be detected easily by either the 1600A or the 1607A, the former's mapping capability facilitates the initial location of the defect. After the probes are connected, the service technician uses the map to examine the faulty loop, and uses the trigger-word locator to trigger the table mode a couple of states before the faulty jump. The analyzer can then be switched to the table mode, and troubleshooting can proceed as before.

The trouble in the traffic-signal controller is even more easily diagnosed by a logic-state analyzer. The trigger word is set to address 37416 and the mode to end the data storage on the trigger word. The analyzer is then set to compile a single-sample table, connected to the program-counter bus, and left alone.

Whenever the traffic controller malfunctions, the logic-state analyzer will store and display the 15 words of program execution preceding the total jump to 37416. This information will remain on display until the technician gets back to see what is happening.

The traffic controller always jumps to 37416 from 2173. This instruction should be an AND instruction in a part of the program that changes the traffic light's timing cycles to accommodate the evening rush. In particular, 2173 is only called if one of the lights is yellow at exactly 4:40 p.m., when the rush hour timing is initiated. Since this occurs about every two days, the fault only shows up then.

A quick examination of the program instructions and the program counter shows that the instruction at 2173 is a JUMP instead of an AND because of a faulty bit in the program ROM. Once the processor encounters the JUMP, it interprets the next two instructions as a jump address and flies off to 37416. Since the processor interprets the unprogramed or missing ROM, as a HALT, it promptly halts at 37416 until restarted.

Logic-state analyzers make quick work of troubleshooting data manipulators by pinpointing the errant data transactions. Even if additional time-domain analysis is required to find the glitch, race, or other malfunction, the logic-state analyzer shortens the search by putting out a trigger at or near the fault. The instrument also eliminates a large amount of probing by providing an exact record of the error so that only those nodes that could cause the error need to be probed. □

HEWLETT hp PACKARD

For More Information, Call Your Local HP Sales Office or, in U. S., East (201) 265-5000. Midwest (312) 677-0400. South (404) 434-4000. West (213) 877-1282. Or, Write: Hewlett-Packard, 1501 Page Mill Road, Palo Alto, California 94304. In Europe, Post Office Box 85, CH-1217 Meyrin 2, Geneva, Switzerland. In Japan, YHP, 1-59-1, Yoyogi, Shibuya-Ku, Tokyo, 151.

5952-1975

PRINTED IN U.S.A.