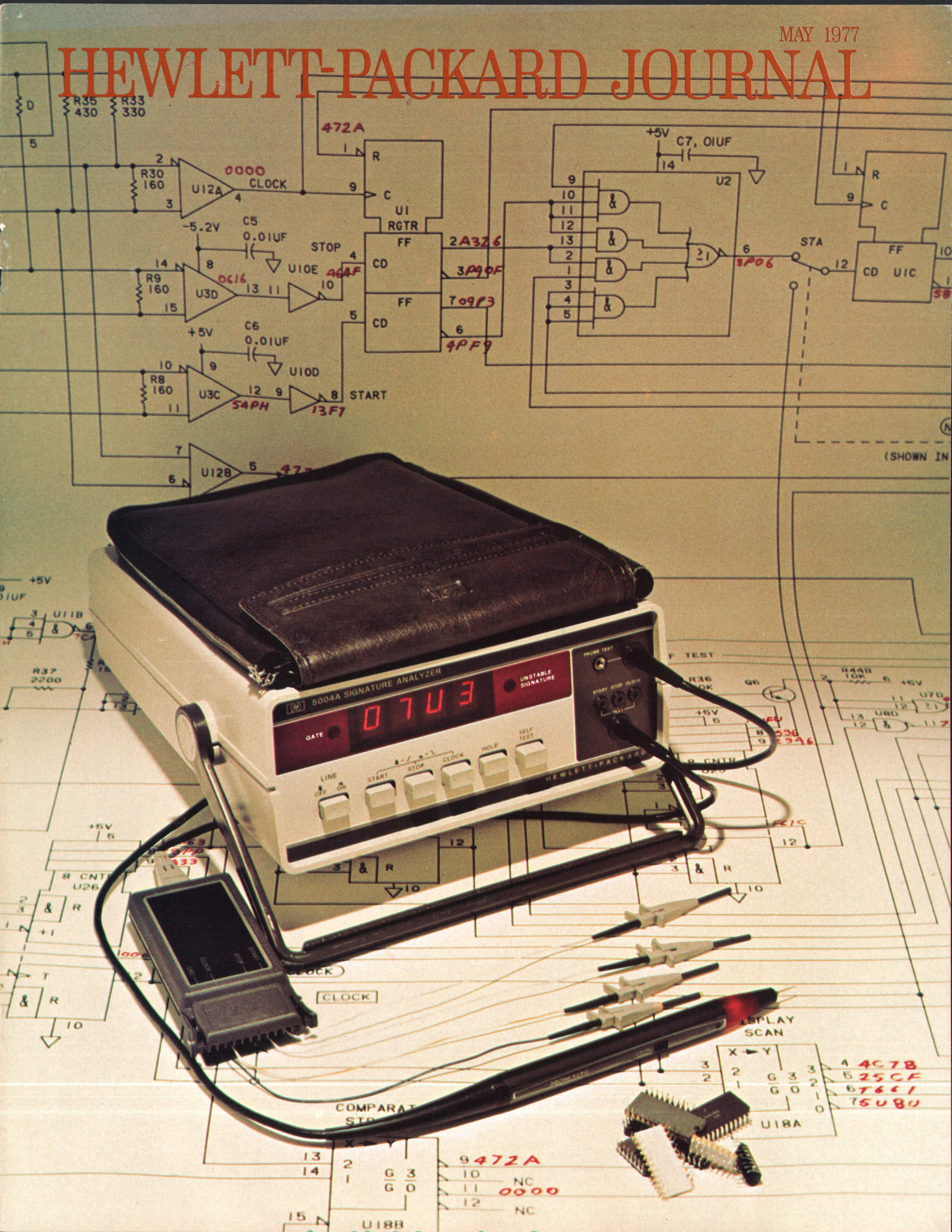
MAY 1977

HEWLETT-PACKARD JOURNAL



Signature Analysis: A New Digital Field Service Method

In a digital instrument designed for troubleshooting by signature analysis, this method can find the components responsible for well over 99% of all failures, even intermittent ones, without removing circuit boards from the instrument.

by Robert A. Frohwerk

WITH THE ADVENT OF MICROPROCESSORS and highly complex LSI (large-scale integrated) circuits, the engineer troubleshooting digital systems finds himself dealing more with long digital data patterns than with waveforms. As packaging density increases and the use of more LSI circuits leaves fewer test points available, the data streams at the available test points can become very complex. The problem is how to apply some suitable stimulus to the circuit and analyze the resulting data patterns to locate the faulty component so that it can be replaced and the circuit board returned to service.

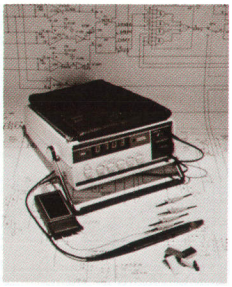
The search for an optimal troubleshooting algorithm to find failing components on digital circuit boards has taken many directions, but all of the approaches tried have had at least one shortcoming. Some simply do not test a realistic set of input conditions, while others perform well at detecting logical errors and stuck nodes but fail to detect timing-related problems. Test systems capable of detecting one-half to two-thirds of all possible errors occurring in a circuit have been considered quite good. These systems tend to be large, for factory-based use only, and computer-driven, requiring program support and software packets and hardware interfaces for each type of board to be tested. Field troubleshooting, beyond the logic-probe capability to detect stuck nodes, has been virtually neglected in favor of board exchange programs.

The problem seems to be that test systems have too often been an afterthought. The instrument designer leaves the test procedure to a production test engineer, who seeks a general-purpose solution because he lacks the time to handle each case individually.

Obviously it would be better if the instrument designer provided for field troubleshooting in his original design. Who knows a circuit better than its original designer? Who has the greatest insight as to how to test it? And what better time to modify a circuit to accommodate easy testing than before the circuit is in production?

New Tools Needed

But here another problem arises: what do we offer the circuit designer for tools? A truly portable test instrument, since field troubleshooting is our goal, would be a passive device that merely looked at a circuit and told us why it was failing. The tool would provide no stimulus, require little software support, and have accuracy at least as great as that of computer-driven factory-based test systems.



Cover: *Those strange-looking strings of four alphanumeric characters on the instrument's display and the schematic diagram are* signatures, *and the instrument is the 5004A Signature Analyzer, a troubleshooting tool for field repair of digital systems. With a failing system operating in a self-stimulating test mode, the service person probes various test points, looking for incorrect signature displays that can point to faulty components.*

In this Issue:

If a tester provides no stimulus, then the circuit under test must be self-stimulating. Whereas this seemed either impossible or at best very expensive in the past, a self-stimulating circuit is not out of the question now. More and more designs are microprocessor-oriented or ROM-driven, so self-stimulus, in the form of read-only memory, is readily available and relatively inexpensive.

By forcing a limitation on software, we have eliminated the capability to stop on the first failure and must use a burst-mode test. Another restriction we will impose is that the device under test must be synchronous, in the sense that at the time the selected clock signal occurs the data is valid; not an unfair condition by any means, and it will be justified in the article beginning on page 15.

There are only a few known methods for compressing the data for a multiple-bit burst into a form that can be handled easily by a portable tester without an undue amount of software. One method used in large systems is transition counting. Another method, a much more efficient data compression technique borrowed from the telecommunications field, is the cyclic redundancy check (CRC) code, a sort of checksum, produced by a pseudorandom binary sequence (PRBS) generator.

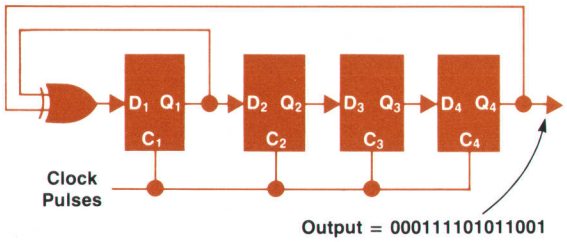
A troubleshooting method and a portable instrument based on this concept turns out to be the answer we are seeking. We call the method signature analysis and the instrument the 5004A Signature Analyzer. The instrument is described in the article on page 9. Here we will present the theory of the method and show that it works, and works very well.

Pseudorandom Binary Sequences

A pseudorandom binary sequence is, as implied, a pattern of binary ones and zeros that appears to be random. However, after some sequence length the pattern repeats. The random-like selection of bits provides nearly ideal statistical characteristics, yet the sequences are usable because of their predictability. A PRBS based upon an n-bit generator may have any length up to 2^n-1 bits before repeating. A generator that repeats after exactly 2^n-1 bits is termed maximal length. Such a generator will produce all possible n-bit sequences, excluding a string of n zeros. As an example, let us take the sequence: 000111101011001. This is a fifteen-bit pattern produced by a four-bit maximal-length generator ($15=2^4-1$). If we were to wrap this sequence around on itself, we would notice that all possible non-zero four-bit patterns occur once and only once, and then the sequence repeats.

To construct a PRBS generator we look to the realm of linear sequential circuits, which is where the simplest generators reside mathematically. Here there exist only two types of operating elements. The first is a modulo-2 adder, also known as an exclusive-OR gate. The other element is a simple D-type flip-flop, which being a memory element behaves merely as a time delay of one clock period. By connecting flip-flops in series we construct a shift-register as in Fig. 1, and by taking the outputs of various flip-flops, exclusive-ORing them, and feeding the result back to the register input, we make it a feedback shift register that will produce a pseudorandom sequence. With properly chosen feedback taps, the sequence will be maximal length. The fifteen-bit sequence above was produced by the generator in Fig. 1, with the flip-flops initially in the 0001 state since the all-zero state is disallowed. The table in Fig. 1 shows the sequence in detail. The list contains each of the sixteen ways of arranging four bits, except four zeros.

If we take the same feedback shift register and provide it with an external input, as in Fig. 2, we can overlay data onto the pseudorandom sequence. The overlaid data disturbs the internal sequence of the generator. If we begin with an initial state of all zeros and supply a data impulse of 1000..., the result is the same sequence as in Fig. 1 delayed by one clock period.



	Cycle	Q_1	Q_2	Q_3	Q_4	$D_1 = Q_1 \oplus Q_4$
Initial State:	0	1	0	0	0	1
	1	1	1	0	0	1
	2	1	1	1	0	1
	3	1	1	1	1	0
	4	0	1	1	1	1
	5	1	0	1	1	0
	6	0	1	0	1	1
	7	1	0	1	0	1
	8	1	1	0	1	0
	9	0	1	1	0	0
	10	0	0	1	1	1
	11	1	0	0	1	0
	12	0	1	0	0	0
	13	0	0	1	0	0
	14	0	0	0	1	1
Begins To Repeat:	15	1	0	0	0	1

Fig. 1. *Signature analysis is a troubleshooting technique that makes use of the cyclic redundancy check (CRC) code, a sort of checksum, produced by a pseudorandom binary sequence (PRBS) generator. Shown here is a feedback shift register that generates a 15-bit PRBS. The outputs of the four flip-flops go through all possible non-zero four-bit patterns and then the sequence repeats.*

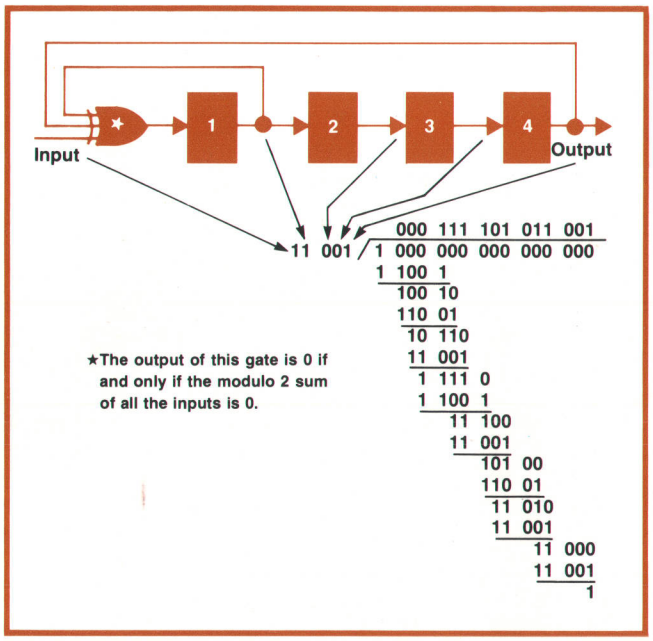


Fig. 2. *When the feedback shift register of Fig. 1 is provided with an external input, data can be overlaid on the PRBS generated by the circuit. Feeding data into a PRBS generator is the same as dividing the data by the characteristic polynomial of the generator.*

Shift Register Mathematics

A shift register may be described using a transform operator, D, defined such that $X(t) = DX(t-1)$. Multiplying by D is equivalent to delaying data by one unit of time. (Recall that we are concerned only about synchronous logic circuits.) In Fig. 2 the data entering the register is the sum of samples taken after one clock period and four clock periods along with the input data itself. Thus, the feedback equation may be written as $D^4X(t) + DX(t) + X(t)$ or simply X^4+X+1.

It happens that feeding a data stream into a PRBS generator is equivalent to dividing the data stream by the characteristic polynomial of the generator. For the particular implementation of the feedback shift register considered here the characteristic polynomial is X^4+X^3+1, which is the reverse of the feedback equation. Fig. 2 shows the register along with longhand division of the impulse data stream (100...). Keep in mind that in modulo-2 arithmetic, addition and subtraction are the same and there is no carry. It can be seen that the quotient is identical to the pattern in Fig. 1 and repeats after fifteen bits (the "1" in the remainder starts the sequence again).

Because the shift register with exclusive-OR feedback is a linear sequential circuit it gives the same weight to each input bit. A nonlinear circuit, on the other hand, would contain such combinatorial devices as AND gates, which are not modulo-2 operators and which would cancel some inputs based upon prior bits. In other words a linear polynomial is one for which $P(X+Y) = P(X) + P(Y)$. Take the example of Fig. 3, where the three different bit streams X, Y, and X+Y are fed to the same PRBS generator. Notice that the output sequences follow the above relationship, that is, $Q(X+Y) = Q(X) + Q(Y)$. Also, notice that Y is a single impulse bit delayed in time with respect to the other sequences and the only difference between X and X+Y is that single bit. Yet, Q(X+Y) looks nothing like Q(X). Indeed, if we stop after entering only twenty bits of the sequences and compare the remainders, or the residues in the shift register, they would be: $R(X+Y) = 0100$, $R(X) = 0111$.

Error Detection by PRBS Generator

Looking at this example in another manner, we can think of X as a valid input data stream and X + Y as an erroneous input with Y being the error sequence. We will prove later that any single-bit error, regardless of when it occurs, will always be detected by stopping the register at any time and comparing the remainder bits (four in this case) with what they should be. This error detection capability is independent of the length of the input sequence. In the example of Fig. 3, R(X+Y) differs from the correct R(X), and the effect of the error remains even though the error has disappeared many clock periods ago.

Let us stop for a moment to recall our original goal. We are searching for a simple data compression algorithm that would be efficient enough to be usable in a field service instrument tester. As such it was to require only minimal hardware and software support.

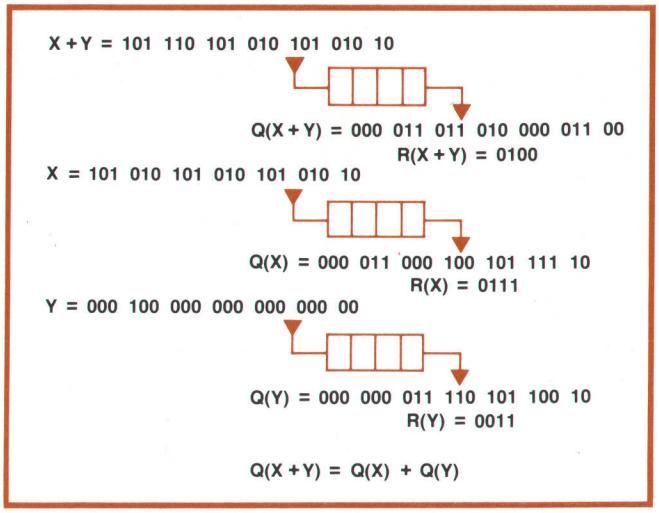


Fig. 3. *Three different input data sequences fed to the same PRBS generator produce very different output sequences even though the input sequences differ by only one bit. If the generators are stopped at some time and the patterns remaining in the flip-flops are compared, they are also different. These remainder patterns are called* signatures. *They show the effects of an error sequence Y added to a data stream X even when the error occurs only once in a long measurement window.*

We have now found such an algorithm. If the circuit designer arranges his synchronous circuit so as to provide clock and gate signals that produce a repeatable cycle for testing, then the feedback shift register is the passive device that we need to accumulate the data from a node in the instrument under test. By tracing through an instrument known to be good, the designer merely annotates his schematic, labeling each test point with the contents of the shift register at the end of the measurement cycle, and uses this information later to analyze a failing circuit. Because this PRBS residue depends on every bit that has entered the generator, it is an identifying characteristic of the data stream. We have chosen to call it a *signature*. The process of annotating schematics with good signatures as an aid in troubleshooting circuits that produce bad signatures has been termed *signature analysis*.

Errors Detected by Signature Analysis

We have claimed that any single-bit error will always be detected by a PRBS generator. But how about multiple errors? Also, our goal was to maintain error detection capability at least as good as existing methods. Earlier mention was made of transition counting, which appears to be the only other method that could easily be made portable. To show how signature analyis stands up against transition counting requires a mathematical discussion of the error detection capabilities of these methods. Take first the PRBS.

Assume X is a data stream of m bits, P is an n-bit PRBS generator, P^{-1} its inverse ($P^{-1}P = 1$), Q is a quotient and R the remainder.

$$P(X) = Q(X)\cdot 2^n + R(X). \qquad (1)$$

Take another m-bit sequence Y that is not the same as X and must therefore differ by another m-bit error sequence E such that

$$Y = X + E.$$

Now,

$$P(Y) = Q(Y)\cdot 2^n + R(Y)$$

so,

$$P(X+E) = Q(X+E)\cdot 2^n + R(X+E).$$

But all operators here are linear, so

$$P(X) + P(E) = Q(X)\cdot 2^n + Q(E)\cdot 2^n + R(X) + R(E).$$

Subtracting (or adding, modulo 2) with equation 1 above,

$$P(E) = Q(E)\cdot 2^n + R(E). \qquad (2)$$

However, if Y is to contain undetectable errors,

$$R(Y) = R(X).$$

It follows that

$$R(Y) = R(X+E) = R(X) + R(E) = R(X),$$

$$R(E) = 0.$$

Substituting into equation 2,

$$P(E) = Q(E)\cdot 2^n,$$

and all undetectable errors are found by

$$E = P^{-1}Q(E)\cdot 2^n. \qquad (3)$$

For a single-bit error

$$E = D^a(1)$$

where D is the delay operator, a is the period of the delay, and "1" is the impulse sequence 1000... Substituting into (3),

$$D^a(1) = P^{-1}Q(D^a(1))\cdot 2^n.$$

D commutes with other linear operators, so

$$D^a(1) = D^aP^{-1}Q(1)\cdot 2^n$$

$$1 = P^{-1}Q(1)\cdot 2^n$$

$$P(1) = Q(1)\cdot 2^n.$$

But by the original assumptions,

$$P(1) = Q(1)\cdot 2^n + R(1)$$

and by addition

$$R(1) = 0.$$

However, it has been shown by example that $R(1) \neq 0$. Therefore, $E \neq D^a(1)$ and the set of undetectable errors E does not include single-bit errors; in other words, a single-bit error is always detectable. (An intuitive argument might conclude that a single-bit error would always be detected because there would never be another error bit to cancel the feedback.)

To examine all undetectable errors as defined by equation 3, it helps to consider a diagrammatical representation, Fig. 4, of:

$$E = P^{-1}Q(E)\cdot 2^n.$$

Since X, Y, and E are all m-bit sequences, it follows that $Q\cdot 2^n$ must be an m-bit sequence containing n final zeros. Q therefore contains (m−n) bits. Hence, there are 2^{m-n} sequences that map into the same residue as the correct sequence, and there are $2^{m-n}-1$ error sequences that are undetectable because they leave the same residue as the correct sequence. 2^m sequences can be generated using m bits and only one of these is correct, so the probability of failing to detect an error by a PRBS is

$$\text{Prob (PRBS, fail)} = \frac{\text{Undetectable Errors}}{\text{Total Errors}} = \frac{2^{m-n}-1}{2^m-1}.$$

For long sequences, large m,

$$\text{Prob (PRBS, fail)} \approx 1/2^n.$$

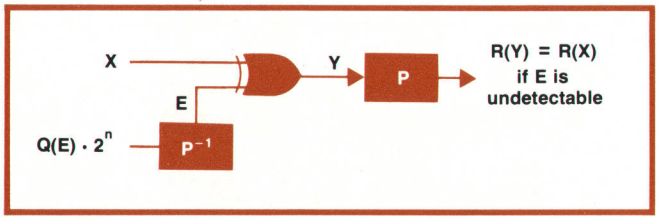


Fig. 4. *A diagrammatical representation of errors undetectable by signature analysis. For long data sequences the probability of not detecting an error approaches $1/2^n$, where n is the number of flip-flops in the feedback shift register.*

In summary, a feedback shift register of length n will detect all errors in data streams of n or fewer bits, because the entire sequence will remain in the register, R(X) = P(X). For data streams of greater than n bits in length, the probability of detecting an error using a PRBS is very near certainty even for generators of modest length. The errors not detected are predictable and can be generated by taking all m-bit sequences with n trailing zeros and acting upon such sequences by the inverse of the n-bit PRBS generator polynomial P, that is

$$E = P^{-1}(Q \cdot 2^n).$$

Furthermore, such error detection methods will always detect a single-bit error regardless of the length of the data stream. It can also be proved that the only undetectable error sequence containing two errors such that the second cancels the effect of the first is produced by separating the two errors by exactly 2^n-1 zeros.[1] The one sequence of length n+1 that contains undetectable errors begins with an error and then contains other errors that cancel each time the original error is fed back.

Errors Detected by Transition Counting

It appears that signature analysis using a PRBS generator is a difficult act to follow, but let us give transition counting a chance. A transition counter assumes an initial state of zero and increments at each clock time for which the present data bit differs from the previous bit. With a transition counter the probability of an undetected error, given that there is some error, is:

$$\text{Prob (Trnsn, Fail)} = N_u/N_t,$$

where N_u = number of undetected errors and N_t = total number of errors. But

$$N_u = \sum_{r=0}^{m} p_{ur}$$

where p_{ur} = Prob (undetected errors given r transitions). However,

$$p_{ur} = N_{ur} \cdot p_r$$

where N_{ur} = number of undetected errors given r transitions, and p_r = Prob (counting r transitions). Reducing further,

$$N_{ur} = N_r - N_c,$$
$$p_r = N_r/N_s,$$

where N_c = number of ways of counting correctly (=1), N_s = total number of m-bit sequences, and N_r = number of ways of counting r transitions:

$$N_r = \binom{m}{r} = \frac{m!}{r!(m-r)!}$$

The binomial coefficient $\binom{m}{r}$ expresses the number of ways of selecting from m things r at a time. Looking back to the original denominator,

$$N_t = N_s - N_c.$$

Putting all of this together,

$$\text{Prob (Trnsn, Fail)} = \frac{\sum_{r=0}^{m} (N_r - N_c)(N_r/N_s)}{N_s - N_c}.$$

Or,

$$\text{Prob (Trnsn, Fail)} = \frac{\sum_{r=0}^{m} \left[\binom{m}{r} - 1\right] \binom{m}{r}/2^m}{2^m - 1}$$
$$\approx 1/\sqrt{m\pi}.$$

This is the probability of a transition counter's failing to detect an error in an m-bit sequence.

A similar argument finds the probability of the specific case where a single-bit error is not detected by a transition counter. There are 2^m sequences of m bits and any one of the m bits can be altered to produce a single-bit error, so that there are $m \cdot 2^m$ possible single-bit errors. To determine how many undetected single-bit errors exist, we must look at how to generate them.

Upon considering the various ways of generating single-bit errors that are undetectable, a few observations become obvious. We can never alter the final bit of a sequence, because that would change the transition count by plus or minus one, which would be detected. The only time we can alter a bit without getting caught is when a transition is adjacent to a double bit; that is, flipping the center bit in the patterns 001, 011, 100, or 110 will not affect the transition count. In other words, the transition count for ...0X1... and ...1X0... does not depend on the value of X.

Since our transition counter assumes an initial 0 state, the first bit of the sequence, regardless of its state, can be flipped without affecting the transition count, provided that the second bit is a one. In this case only the second of m bits is predetermined, i.e., $b_2 = 1$, and there are 2^{m-1} ways of completing the sequence. Any bit other than the first or last, that is, the m−2 bits from b_2 through b_{m-1}, can be altered without affecting the transition count if the bit in question is flanked by a zero on one side and a one on the other. For a given bit b_i we have free choice of m−1 bits, since as soon as we select b_{i-1} then b_{i+1} is forced to the opposite state. There are $(m-2) \cdot 2^{m-1}$ of these midstream errors. Adding the 2^{m-1} sequences where b_1 can be changed we have a total of $(m-1) \cdot 2^{m-1}$ sequences containing single-bit errors that cannot be detected by a transition counter. But earlier we showed that the total number of single-bit errors was $m \cdot 2^m$, hence the probability of failing to detect a single-bit error is

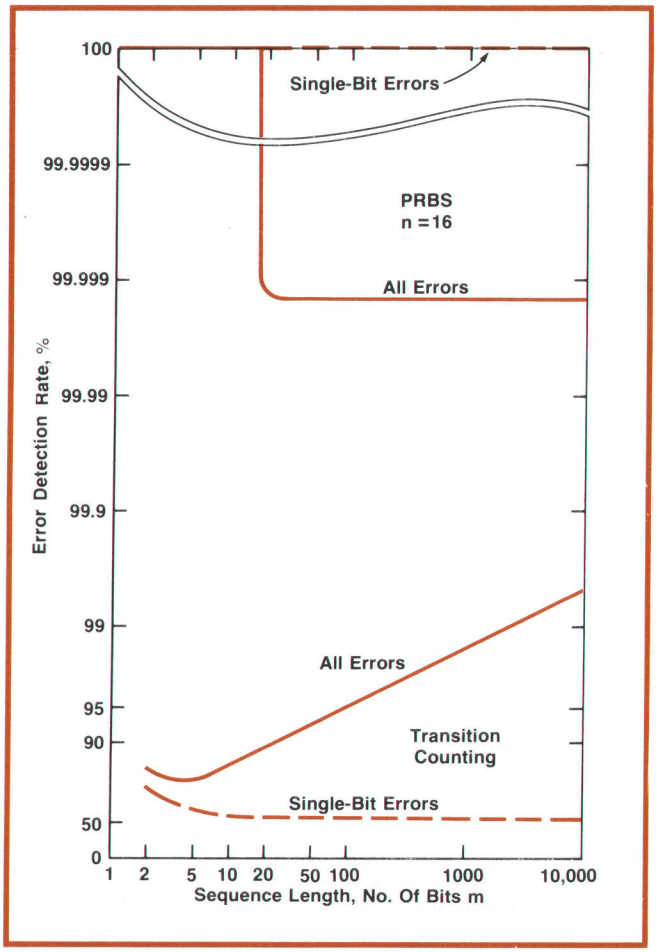


Fig. 5. *Probability of detecting errors for signature analysis and transition counting as a function of the length of the data sequence. n=16 for the PRBS generator.*

$$\text{Prob (Trnsn, Fail, single-bit)} = \frac{(m-1)\cdot 2^{m-1}}{m\cdot 2^{m}} = \frac{m-1}{2m}$$

$$\approx 1/2.$$

It may be noted that the failure rate is actually somewhat higher, because a counter of limited length will overflow for long sequences, leaving some ambiguity. It can be shown that because of this overflow an n-bit transition counter will never detect more than $1/2^n$ of all errors.

Signature Analysis versus Transition Counting

We can now plot the probabilities of detecting any error using a transition counter versus a PRBS generator (see Fig. 5). It is interesting to note that the transition count method looks worst on single-bit errors, exactly where the feedback shift register never fails. Overall the transition counter looks pretty good, detecting at least half of all errors, but even a one-bit shift register could do that. The four-bit PRBS generator used in earlier examples will always detect better than $(100-100/2^4)=93\%$ of all errors. It seems conclusive that the PRBS method puts on a good performance, and if we want it to do better we merely add one more bit to the register to halve the rate of misses.

How Close Do We Want to Get?

We set out to find a means of instrument testing at least as good as present computer-based methods. These existing systems generally perform as well as the engineer who adapts them to the circuit under test. The task of adapting a circuit to be tested by signature analysis is very much the same as adapting to any other tester—engineering errors are assumed constant. If the PRBS technique is used for backtracing to find faulty components in field service, then the largest remaining block of human error is the ability of the service person to recognize a faulty signature.

It seems that a four-character signature is easily recognized, while the incidence of correct pattern recognition falls off with the addition of a fifth character. We tried this on a statistically small sample of people and found it to be so. Electronically, four hexadecimal characters is sixteen bits. A few bits more or less is not likely to complicate a shift register, but it would have an adverse effect on the user. Sixteen bits gives a detector failure rate of less than sixteen parts per million (one in 65,535), adequate for most purposes, so we settled on a four-character signature.

Since the signature offers no diagnostic information

Last In → A	B	C	D ← First In	Display
0	0	0	0	0
1	0	0	0	1
0	1	0	0	2
1	1	0	0	3
0	0	1	0	4
1	0	1	0	5
0	1	1	0	6
1	1	1	0	7
0	0	0	1	8
1	0	0	1	9
0	1	0	1	A
1	1	0	1	C
0	0	1	1	F
1	0	1	1	H
0	1	1	1	P
1	1	1	1	U

Fig. 6. *In the HP 5004A Signature Analyzer, n=16 and the remainder, or signature, is displayed as four non-standard hexadecimal characters. Each character represents the outputs of a group of four flip-flops as shown here.*

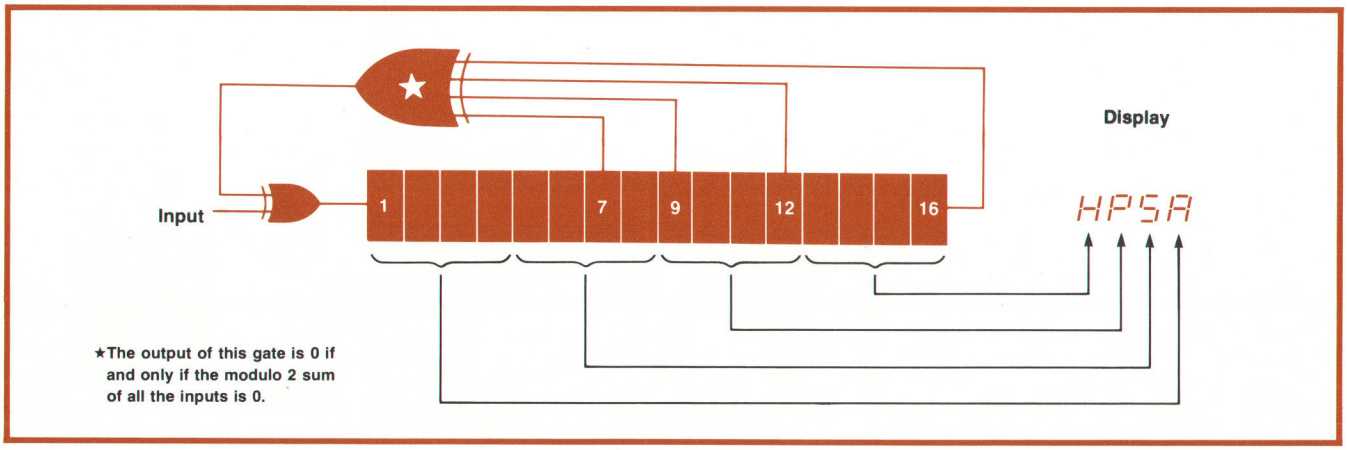


Fig. 7. *The 16-flip-flop PRBS generator used in the 5004A Signature Analyzer.*

whatsoever, but is purely go/no-go, the character set was not restricted, except to be readable. Numbers are quite readable but there are not enough of them. Another consideration was that for an inexpensive tool, seven-segment displays are desirable. The chosen character set (Fig. 6) is easily reproduced by a seven-segment display and the alpha characters are easily distinguishable even when read upside down. A further psychological advantage of this non-standard ("funny hex") character set is that it does not tempt the user to try to translate back to the binary residue in search of diagnostic information.

Register Polynomial

We have decided on a four-character display for a sixteen-bit register, but it remains to select the feedback taps to guarantee a maximal length sequence. It happens that this can be done in any of 2048 ways.[2] The computer industry uses two:

$$\text{CRC-16} = X^{16}+X^{15}+X^2+1,$$

and

$$\text{SDLC(or CCITT-16)} = X^{16}+X^{12}+X^5+1.$$

But each of these is reducible:

$$\text{CRC} = (X+1)\,(X^{15}+X+1),$$

and

$$\text{SDLC} = (X+1)\,(X^{15}+X^{14}+X^{13}+X^{12}+X^4+X^3+X^2+X+1).$$

The X+1 factor was included in both to act as a parity check; it means that all undetectable error sequences will have even parity. This is apparent by looking at the original polynomials and noting that they each have an even number of feedback taps, so an even number of error bits is required to cancel an error. For our purposes this clustering of undetectable errors seems undesirable. We would like a polynomial that scatters the missed errors as much as possible. For this reason we would also like to avoid selecting feedback taps that are evenly spaced or four or eight bits apart because the types of instruments, microprocessor-controlled, that we will most frequently be testing tend to repeat patterns at four and eight-bit intervals. The chosen feedback equation is:

$$X^{16}+X^{12}+X^9+X^7+1,$$

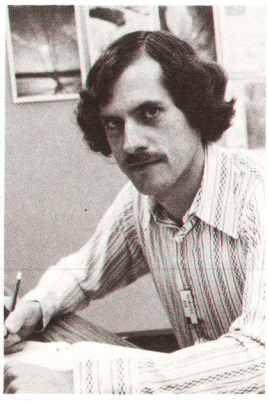
which corresponds to the characteristic polynomial

$$P(X) = X^{16}+X^9+X^7+X^4+1.$$

This is an irreducible maximal length generator with taps spaced unevenly (see Fig. 7). Our relatively limited experience with this PRBS generator has shown no problems with regard to the selection of feedback taps. The test of time will tell; even the CRC-16 generator seems to have fallen out of favor with respect to that of SDLC after having served the large-computer industry for well over a decade.

References

1. W.W. Peterson, and E.J. Weldon, Jr., "Error-Correcting Codes," The MIT Press, Cambridge, Massachusetts, 1972.
2. S.W. Golomb, "Shift-Register Sequences," Holden-Day, Inc., San Francisco, 1967.



Robert A. Frohwerk

Bob Frohwerk did the theoretical work on signature analysis. He received his BS degree in engineering from California Institute of Technology in 1970, and joined HP in 1973 with experience as a test engineer for a semiconductor manufacturer and as a test supervisor and quality assurance engineer/manager for an audio tape recorder firm. He's a member of the Audio Engineering Society. Bob was born in Portland, Oregon. Having recently bought a home in Los Altos, California, he and his wife are busy setting up a workshop for Bob's woodworking and metal sculpture, putting in a large organic garden, and planning furniture projects and a solar heating system. Bob also goes in for nature photography, sound systems, and meteorology (he built his wife's weather station).

Easy-to-Use Signature Analyzer Accurately Troubleshoots Complex Logic Circuits

It's a new tool for field troubleshooting of logic circuits to the component level.

by Anthony Y. Chan

THE NEW HEWLETT-PACKARD Model 5004A Signature Analyzer (Fig. 1) was designed to meet the need for field troubleshooting of digital circuits to the component level. The basic design goal was to implement the signature analysis technique described in the preceding article in a compact, portable instrument with inputs compatible with the commonly used logic families (TTL and 5V CMOS).

The 5004A is a service tool. It receives signals from the circuit under test, compresses them, and displays the result in the form of digital signatures associated with data nodes in the circuit under test. The signature analyzer does not generate any operational signal for circuit stimulus, depending instead on the circuit being tested to have built-in stimulus capability. The analyzer is capable of detecting intermittent faults. Its built-in self-test function increases user confidence and its diagnostic routine allows quick, easy troubleshooting with another 5004A if the instrument fails.

The signature analyzer's data probe is also a logic probe similar to the HP 545A Logic Probe.[1] The lamp at the probe tip turns bright for a logic 1, turns off for a logic 0, and goes dim when the input is open-circuited or at a bad level (third state).

What's Inside

Fig. 2 is a block diagram of the signature analyzer. During normal operation, the level detectors receive trains of start, stop, and clock control signals from the circuit under test and transmit them to the edge select switch. The edge select switch allows the user to

Fig. 1. *Model 5004A Signature Analyzer is a new tool for field-troubleshooting of digital circuits to the component level. (The circuits must be designed for signature analysis and must have built-in stimulus.) The 5004A gets start, stop, and clock inputs via its pod, shown here in the foreground, and data inputs via its data probe, which doubles as a logic probe.*

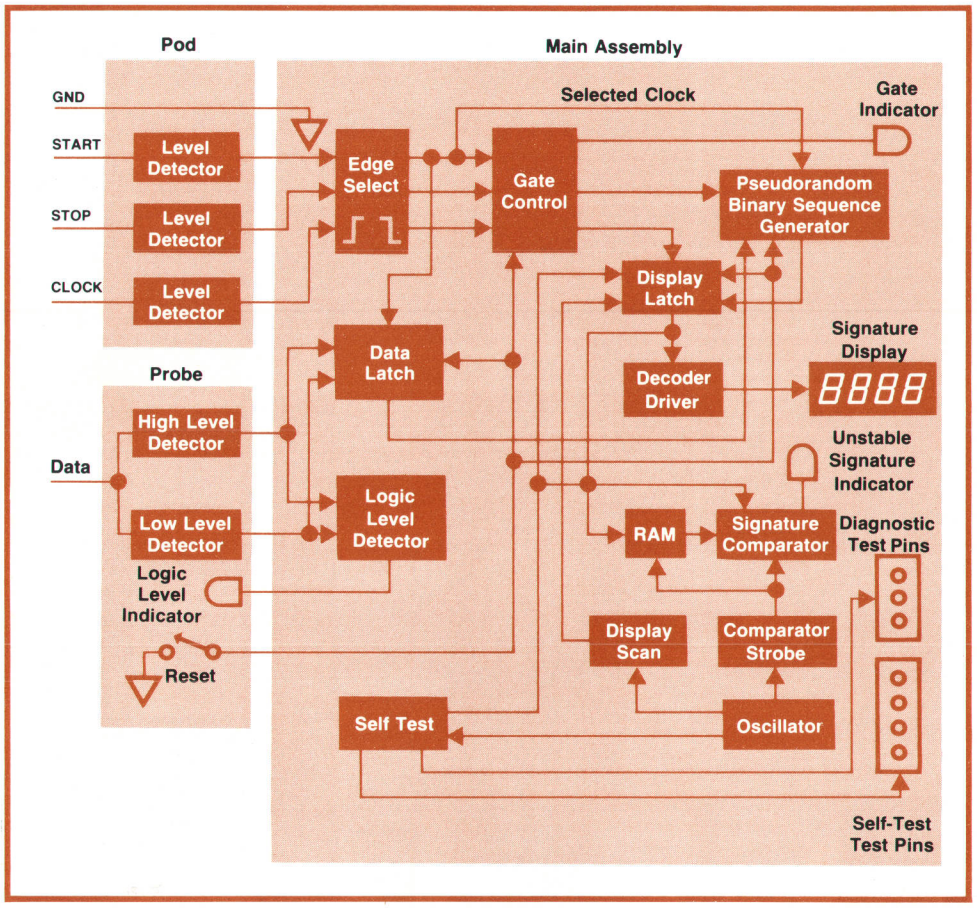


Fig. 2. *Model 5004A Signature Analyzer block diagram. The last 16 bits remaining in the PRBS generator when the stop signal occurs are loaded into the display latch and displayed as four hexadecimal characters.*

choose the polarity of signal transitions that the instrument will respond to. The gate control receives the selected control signals from the edge select switch and generates a gated measurement window (gate on) for the pseudorandom generator; it also turns the gate light on. The measurement window is the period between valid start and stop signals, and its length is measured in clock cycles (see Fig. 3). The minimum possible window length is one clock cycle.

The data probe translates voltages measured at circuit nodes into three logic states (logic 1, logic 0, and bad state) and transfers them to the data latch. The latch further translates the data, from three logic levels to two, at selected clock edges.* At each clock time, the data latch will pass a 1 or 0 level but will remain latched to the previous state if the input is in the bad state. The data latch may be the end of the road for some data because the pseudorandom generator accepts data only during the measurement window (gate on). Once data enters the pseudorandom generator, it is shifted in synchronism with the clock until the end of the measurement window. The last 16 bits remaining in the generator at gate-off time are loaded into the display latch and then output in the form of four non-standard hexadecimal characters—the signature. The display latch keeps the signature on until the end of the next measurement window, when the display is updated with new information. The signature will be stable as long as the measurement window and the data received within the window are repeatable.

*The probe recognizes three logic states instead of only two because of its logic-probe function.

Importance of Setup and Hold Times

Frequency response is one of the most important parameters in a test instrument. In the case of the

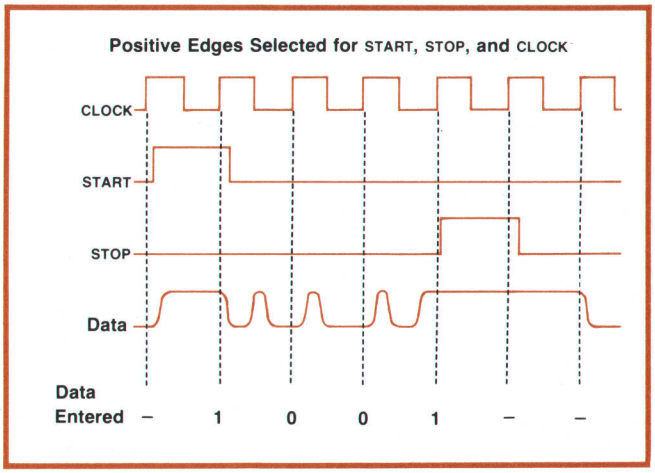


Fig. 3. *The measurement window is an integral number of clock cycles. One cycle is the minimum length.*

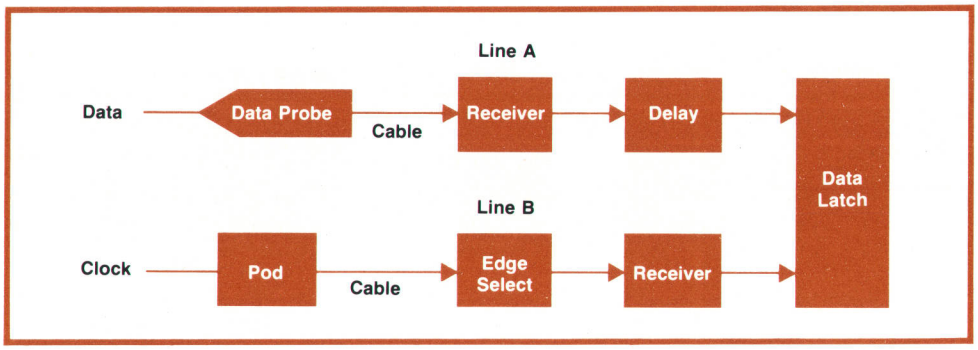


Fig. 4. *Delays through probe and pod channels are matched so that hold time (the time that input data must remain stable after a clock edge) is non-positive. Setup time (the time that input data must be stable before a clock edge) is typically 7 ns.*

signature analyzer, two other factors, data setup time and hold time, are very important as well. Data setup time is the interval for which data must be stable before the selected clock edge occurs. Hold time is the interval for which data must remain stable after the selected clock edge. Assuming a signature analyzer requiring 30 ns setup time and 10 ns hold time is used to test a circuit, then the logic of the circuit under test must be stable for at least 30 ns before the active clock edge and the logic must remain stable for 10 ns after the clock edge; otherwise, ambiguous readings may result. The setup and hold times limit the speed of the analyzer.

It is not easy for a high-speed circuit to guarantee that its logic will remain stable for some period of time after every active clock edge. The 5004A design goal was to be able to operate with reasonably short data setup time and non-positive hold time to minimize ambiguities.

Data and clock signals are received and transmitted to the data latch through the data probe, receiver, edge select switch, and cables (Fig. 4). There is one time delay for the data signal going through the data probe, cable, and receivers (line A), and another time delay for the clock pulses going through the wire and edge select switch into the data latch (line B). Every component, and therefore the time delays, may differ from unit to unit because of manufacturing tolerances. To guarantee a non-positive hold time, eliminate race conditions, and be reproducible in a production environment, the minimum delay of line A must be equal to or longer than the maximum delay of line B ($t_{Amin} \geq t_{Bmax}$). Also desired is a minimum setup time $T_s = t_{Amax} - t_{Bmin}$.

One way to achieve short setup time is to have identical circuitry in the data and clock channels, so propagation delays cancel each other. Circuitry in the data probe is very similar to that in the pod. The receivers in both channels are identical and share the same IC chip. The edge select switch in the clock line has very little delay. The symmetry results in a good match between the two signal lines, but to insure that $t_{Amin} \geq t_{Bmax}$, there is a delay circuit in line A, and the cable length of line B is shorter. Thus it is possible to guarantee hold time less than 0 ns and setup time less than 15 ns (7 ns typical).

Input Impedance

Since the 5004A is a test instrument, it is important that its inputs do not load or condition the circuit under test. It is generally true that high input impedance reduces loading. But, how high can the input impedance be before other effects cause problems?

Let's study a few cases of high-impedance input in a synchronous device (Fig. 5). Fig. 5a shows the result of the input data's changing from a logic 1 to a third state at clock 1. The input voltage is pulled toward ground by the pull-down resistance. The difference between the solid line and the dotted line is the difference of node capacitance (C) and pull-down resistance (R) tolerances. Depending on the clock rate and the RC difference, the result at clock 2 can be either a logic 0 or a third state. The same thing in reverse happens in case B. In case C, the input data is changing to an intermediate level (~1.4V). When the data changes from a logic 1 to the third state, the input is pulled towards 1.4V. The result at clock 2 is in the third state no matter what the RC time constant is.

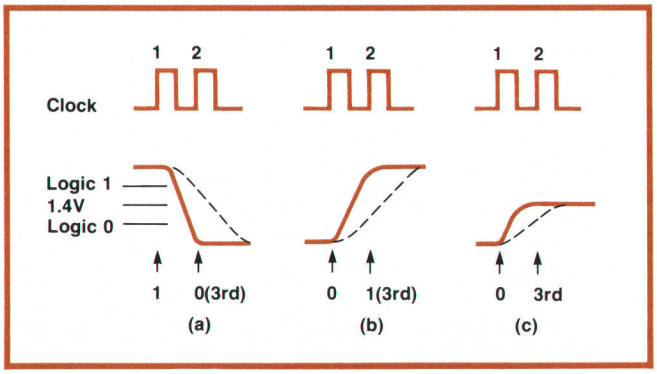


Fig. 5. *5004A input impedance is returned to 1.4V to eliminate ambiguities caused by input RC tolerances. Here are three possible results of the input data's changing from a 0 or 1 at clock 1 to a third state at clock 2. The solid and dotted lines are for different values of input RC. In (a), the input at clock 2 can be seen as a logic zero or a third state, depending on the value of RC. (b) is the reverse of (a). In (c), with the input returned to 1.4V, the result is always a third state.*

There are two other advantages to returning the input impedance to 1.4V. First, there is less voltage swing, and almost equal swings for both logic 1 and logic 0 states. Second, the logic probe open-input requirement is met.

Very high input impedance may cause problems even for a non-clocked device. It introduces threshold errors because of the offset bias current of the input amplifier, and the leakage current of the three-state bus might change the measured voltage. After study and calculation, we chose 50 kΩ to 1.4V as the input impedance and return voltage for the 5004A inputs. 50 kΩ is large enough not to load TTL and most 5V CMOS logic families, and small enough not to cause excessive offset voltage with typical leakage currents on a three-state bus.

Construction

The 5004A Signature Analyzer is constructed in a lightweight, rugged case. A hand-held data probe and a small rectangular pod are connected to the instrument by cables (Fig. 1). Inside the main case are the edge select circuit, gate control, data latch, pseudorandom generator, display latch, signature displays, signature comparator, self-test stimulus generator, and power supply shown in Fig. 2. All the electronics and mechanical components are mounted on a single printed circuit board assembly sandwiched inside two shells held together with four screws. On the front panel are four large seven-segment displays. A light to the left of the display shows gate (measurement window) activity while one on the right indicates unstable signature. Six pushbutton switches control power on/off, start, stop, and clock edge polarities, a hold mode for single cycle events or freezing the signature, and a self-test mode. Start, stop, clock, and data test sockets on the right-hand side of the front panel are for self-test and diagnostic setup. A soft pouch mounted on top of the instrument stores the data probe, pod, and necessary accessories when not in use.

Data Probe

The active data probe is a hand-held probe. Its main function is to accept tip logic information with minimum tip capacitance. The input signal is connected to two comparators through voltage dividers and an RC network (see Fig. 6). The voltage divider R1-R2 is terminated at 1.4V, which guarantees an open input at a bad level and eliminates the potential ambiguity, discussed earlier, resulting from RC tolerances.

Input overload protection is provided by on-chip clamp diodes and the external network R1 and CR1. C1 provides a bypass for fast transitions. R3, R5, and R6 set up voltage references for comparators A and B. Two comparators are needed to measure the three logic states—1, 0, and bad level. The high-speed differential-in/differential-out comparators translate the input voltage into digital signals and transmit them to the main instrument through twisted pairs. A single-contact pushbutton switch on the data probe resets the pseudorandom generator, state control, and displays.

Pod

The thin, rectangular pod houses three identical

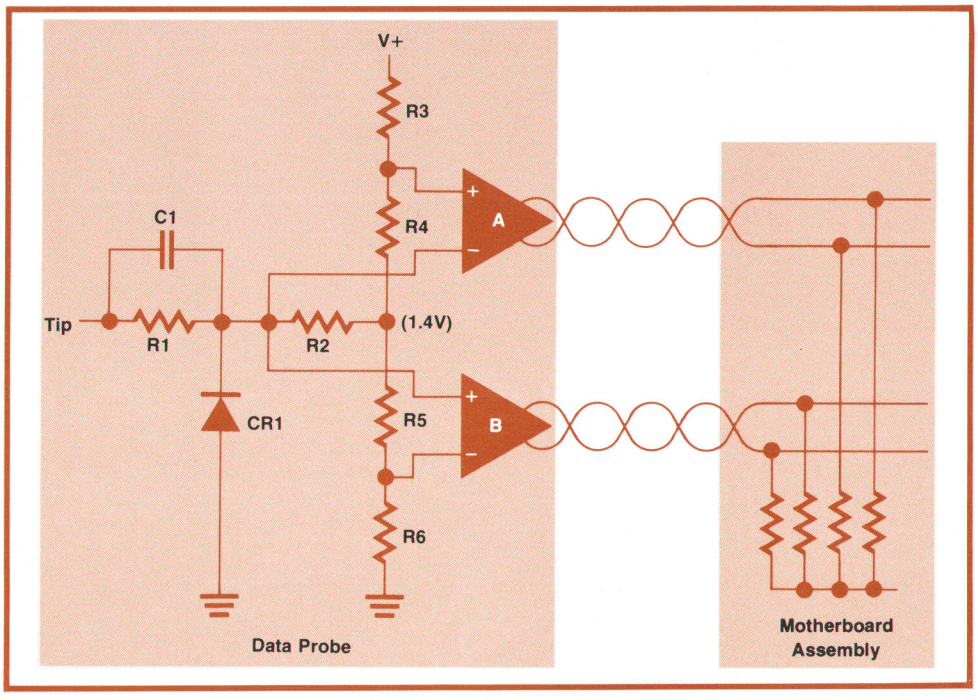


Fig. 6. *Simplified 5004A data probe schematic.*

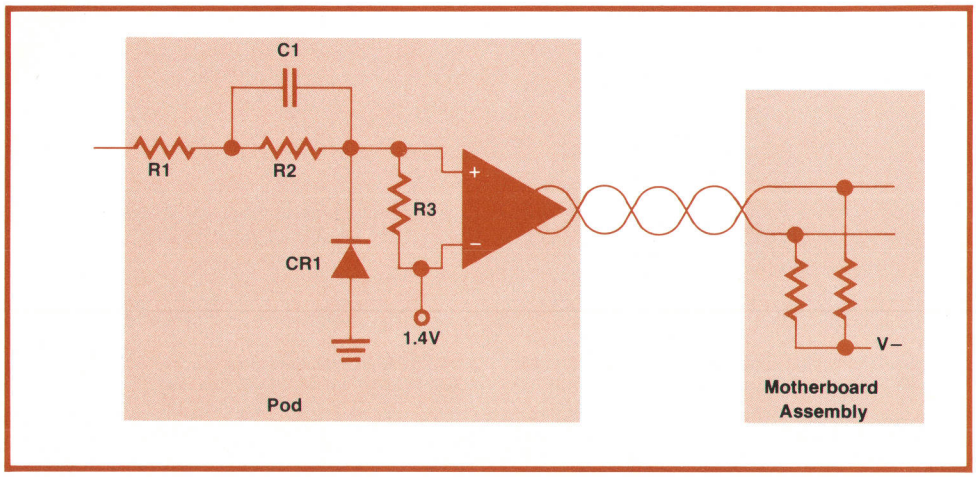


Fig. 7. *Simplified schematic of one channel of the 5004A input pod.*

channels for start, stop, and clock control inputs. The input wires can be directly plugged into any 0.03-inch round socket or connected to a "grabber" that can hook onto almost any test pin and is particularly good for IC pins. Each of the control channels is very similar to the circuitry in the data probe to match propagation delays. In fact, delay match is the major function of the pod.

In each channel of the pod is a comparator of the same type as those in the data probe. One of the comparator inputs is connected to voltage divider R1-R2-R3 while the other input is connected to 1.4V along with R3; this increases input hysteresis and sets the input threshold (Fig. 7). The impedance of R1-R2-R3 is the same as the impedance of the data probe (50 kΩ); termination at the same voltage level further improves matching.

Protection against input overload is provided by internal clamp diodes in the comparator IC and by external network R1, R2, and CR1. R1 damps the ringing generated by the inductance of the input wire. C1 provides a bypass for fast transitions.

Unstable Signature

An intermittent fault is one of the biggest problems in electronic repair. The fault comes and goes, and in most cases does not stay long enough for positive detection. Signature analysis can detect such faults if they occur within a measurement window. However, the operator may not receive the message if the measurement cycle time is too short.

The random-access memory (RAM) in the main assembly of the 5004A continuously writes and reads the display information from the display latch at the display scan rate. During each scan cycle, the signature comparator compares the signature stored in the RAM with the one in the display latch, and turns on the unstable signature light on the front panel when any difference exists. This light is stretched for 100 ms to allow recognition. The comparison is done on a sampled basis and not each time a new signature is developed, so the unstable signature detector works most of the time, but not 100%. Errors occurring in a very short measurement cycle may not always be detected by the relatively slow-scanning comparator.

Hold Mode

The hold mode works closely with the stop signal. If the hold switch on the front panel is pushed in, the hold mode will be entered at the end of the measurement window, freezing the signature display and preventing the gate control from starting a new cycle. Hold mode is particularly useful for testing single-shot events like the start-up sequence of a system.

Self Test

It is important for a user to know that a test instrument is in good working condition before it is used to test anything. The 5004A has a built-in self-test function that gives a quick, accurate check of the instrument. Pressing the self-test switch on the front panel energizes the self-test ROM, which interrupts the display update and generates a special programmed stimulus of start, stop, clock, and data signals to the test sockets on the front panel. With the start, stop, and clock control inputs connected to the corresponding test sockets and the data probe to the data test socket on the front panel, and with positive edges selected for the start, stop, and clock inputs, the signature analyzer performs the self test. When a good working 5004A is tested, its gate light flashes, the unstable signature light blinks, the logic light at the data probe tip flashes, and the signature displays 3951, 2P61, 8888 and then repeats. Pushing the hold switch in turns the gate light off and the signature displays 8888, 3951, or 2P61. The self-test routine tests the entire instrument except the clock edge select circuit and the ground wire at the pod input.

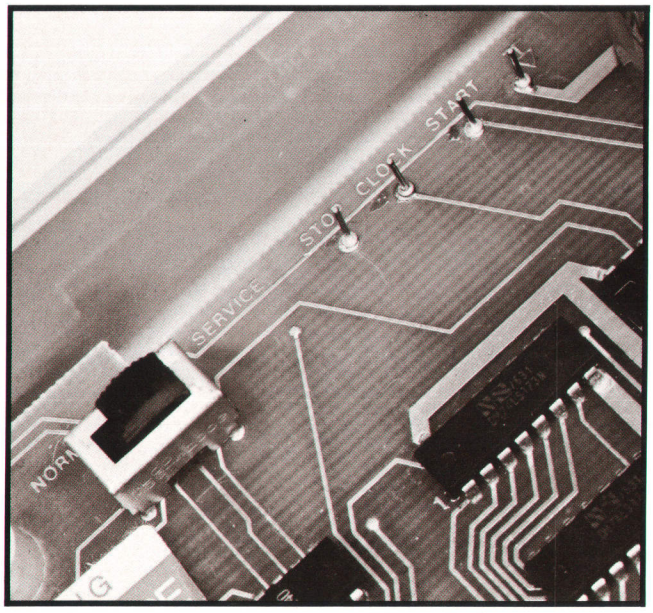


Fig. 8. *The 5004A Signature Analyzer is designed for troubleshooting with another 5004A should the self test reveal a fault. Sliding the* NORM/SERVICE *switch to the* SERVICE *position opens the three feedback loops in the instrument.*

Diagnostic Routine

When an unexpected result during self test indicates a fault, troubleshooting and repair are required. The 5004A was designed with signature analysis in mind. It can be tested with another 5004A. The instrument's top cover can be easily removed by removing four hold-down screws on the bottom and loosening two heat sink mounting screws on the back. All the components in the instrument's main case are then exposed for testing. The failing instrument is placed in self-test mode and the start, stop, clock, and ground inputs of a known good 5004A are connected to the test sockets located on the left side of the printed circuit board in the main case. Probing the circuit nodes with the data probe, reading the signatures on the analyzing 5004A, and comparing them with those printed on the schematic is an easy and almost error-free way of determining the quality of a circuit node. Once a faulty node is found, the source of the problem can be easily located with standard backtracing techniques.

When the fault is in a feedback loop, any single fault will cause all the nodes within the loop to appear bad. To pinpoint the fault, the loop must be opened. There are three feedback loops in the signature analyzer, and a slide switch (NORM/SERVICE) on the left of the main printed circuit board is provided for opening them (Fig. 8). Sliding the switch to the SERVICE position opens all three loops.

The diagnostic routine works on the entire instrument except the power supply, the ECL circuits in the data probe and pod, and their interface circuits.

SPECIFICATIONS
HP Model 5004A Signature Analyzer

DISPLAY:
SIGNATURE: Four-digit hexadecimal.
Characters 0,1,2,3,4,5,6,7,8,9,A,C,F,H,P,U.
GATE UNSTABLE INDICATORS: Panel lights. Stretching: 100 ms.
PROBE-TIP INDICATOR: Light indicates high, low, bad-level, and pulsing states. Minimum pulse width: 10 ns. Stretching: 50 ms.
PROBABILITY OF CLASSIFYING CORRECT DATA STREAM AS CORRECT: 100%.
PROBABILITY OF CLASSIFYING FAULTY DATA STREAM AS FAULTY: 99.998%.
MINIMUM GATE LENGTH: 1 clock cycle.
MINIMUM TIMING BETWEEN GATES (from last STOP to next START): 1 clock cycle.
DATA PROBE:
INPUT IMPEDANCE:
50 kΩ to 1.4V, nominal. Shunted by 7 pF, nominal.
THRESHOLD:
Logic one: 2.0V +.1, −.3
Logic zero: 0.8V +.3, −.2
SETUP TIME: 15 ns, with 0.1V over-drive. (Data to be valid at least 15 ns before selected clock edge.)
HOLD TIME: 0 ns (Data to be held until occurrence of selected clock edge.)
GATING INPUT LINES:
START, STOP, CLOCK INPUTS:
Input Impedance: 50 kΩ to 1.4V, nominal. Shunted by 7 pF, nominal.
Threshold: 1.4V ±.6 (.2V hysteresis, typical).
START, STOP INPUTS:
SETUP TIME: 25 ns. (START, STOP to be valid at least 25 ns before selected clock edge.)
HOLD TIME: 0 ns. (START, STOP to be held until occurrence of selected clock edge.
CLOCK INPUT:
MAXIMUM CLOCK FREQUENCY: 10 MHz.
MINIMUM CLOCK TIME IN HIGH OR LOW STATE: 50 ns.
OVERLOAD PROTECTION: All inputs ±150V continuous, ±250V intermittent, 250Vac for 1 min.
OPERATING ENVIRONMENT: Temperature: 0-55°C; Humidity: 95% RH at 40°C; Altitude: 4,600 m.
POWER REQUIREMENTS: 100/ 120 Vac, +5%, −10%, 48-440 Hz.
220/ 240 Vac, +5% −10%, 48-66 Hz.
15 VA Max.
WEIGHT: Net: 2.5 kg, 5.5 lbs. Shipping: 7.7 kg, 17 lbs.
OVERALL DIMENSIONS: 90 mm high × 215 mm wide × 300 mm deep (3½ in × 5½ in × 12 in). Dimensions exclude tilt bale, probes, and pouch.
PRICE IN U.S.A.: $990.
MANUFACTURING DIVISION: SANTA CLARA DIVISION
5301 Stevens Creek Boulevard
Santa Clara, California 95050 U.S.A.

Anthony Y. Chan

Tony Chan received his BSEE degree from the University of California at Berkeley in 1969 and his MSEE degree from California State University at San Jose in 1974. With HP since 1973, he developed the IC chip for the 546A Logic Pulser and designed the 5004A Signature Analyzer. Before joining HP, he designed linear and digital ICs for four years. A native of Hong Kong, Tony now lives in Sunnyvale, California. He's married and has two children. Having just finished remodeling his home, he's now taking on landscaping and furniture-building projects. He likes working with his hands, especially on wood and automobiles, and enjoys an occasional game of tennis.

Signature Analysis—Concepts, Examples, and Guidelines

Guidelines for the designer are developed based on experience in attempting to retrofit existing products for signature analysis and the successful application of signature analysis in a new voltmeter.

by Hans J. Nadig

THE POWER OF SIGNATURE ANALYSIS as a field troubleshooting technique is amply demonstrated by the analysis presented in the article on page 2 of this issue. The technique can even pinpoint the 20% or so of failures that are "soft" and therefore difficult to find, taking 70-80% of troubleshooting and repair time. Soft failures include those that occur only at certain temperatures or vibration levels. They may be related to noise performance or marginal design, such as race conditions that occur only when the power supply voltage is low but still within specifications. Or they may occur only when the user gives the machine a certain sequence of commands.

Signature analysis is applicable to complex instruments using microprocessors and high-speed algorithmic state machines. Yet it is simple enough so that the user of a product may be able to apply it nearly as well as more highly trained field service personnel.

Having recognized the power of signature analysis, we first attempted to apply it to existing products, including computers, CRT terminals, and the digital portions of microwave test equipment. We soon recognized that either the circuits had to be altered or the signature analysis approach would be no better than earlier methods. After some experience we were able to define rules for making a product compatible with signature analysis. These rules, summarized on page 18, are guidelines for the designer. Following them helps assure that a product will be simple and inexpensive to troubleshoot by the signature analysis method.

How We Got Started

A good way to demonstrate the advantages of signature analysis and the requirements for applying it successfully is to describe what happened when we first tried it a few years ago.

With a prototype signature analyzer we set out to apply the technique to various Hewlett-Packard instruments. We first attacked a CRT terminal with microprocessor control and ROM and RAM storage, including some dynamic memories.

The built-in self-test mode of the terminal displayed a certain test pattern on the CRT and flashed the cursor at a 2½-Hz rate. Taking advantage of this self test as a stimulus, and using the most significant address bit to start and stop the measurement, we soon recognized that these signals did not provide a stable measurement window. Some portions of the terminal operated on an interrupt basis, so the number of clock periods varied within the START-STOP window. Needless to say, the data stream changed, too. Next we concentrated our efforts on one section, the memory. To test it, we wanted to force the microprocessor into a mode in which all the memory locations were addressed, but to do this, we were forced to cut the data bus. Fortunately, we could separate the microprocessor from the data bus by using an extender board and cutting the lines there. Grounding a few lines and pulling some other lines high caused the microprocessor to repeatedly execute one instruction that automatically incremented the address each cycle, effectively stepping up through the whole address field. Fig 1 shows how this can be done for an 8080 microprocessor.

At this point we realized that the microprocessor, the clock, and the power supply were the heart of the product. We decided to call this the "kernel" (Fig. 2). By verifying the proper operation of these parts first we could then expand and test additional portions of the circuitry. With the free-running microprocessor exercising the control and address lines, we were able to test the address bus, the ROMs, and the data bus. As a START and STOP signal the most significant bit of the address bus was used, allowing us to check all the ROM locations. Since a number of RAMs were also affected, we applied a grounded jumper wire to force the enable line to the RAMs low; this was necessary to get stable signatures, since the RAMs did not contain a defined data pattern, and if addressed, randomly altered the information on the data bus.

Here, then, were our first lessons: *feedback loops cause problems unless opened; circuits not related to*

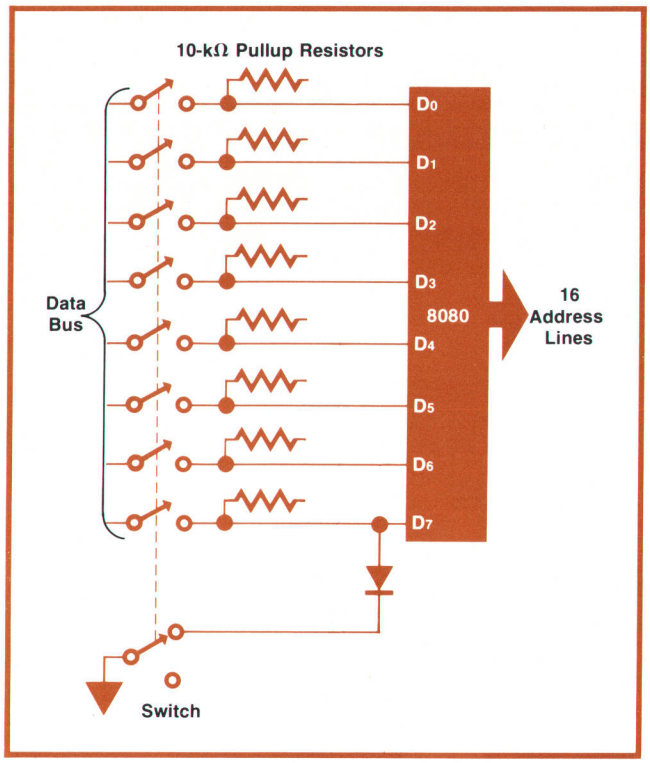


Fig. 1. *To be an effective troubleshooting technique, signature analysis must be designed into a product. For example, for a test of the address lines of a microprocessor, there should be a switch that opens the data bus and forces the microprocessor to free run. The address lines can then be checked and can also be used as control inputs to the signature analyzer.*

the test must be disabled.

Synchronous Operation Necessary

It would be ideal if one setup would allow troubleshooting most parts of an instrument. The synchronization signal with the highest rate would be connected to the clock input of the signature analyzer.

Our display terminal uses a number of different frequencies, from 21 MHz down to 1¼ Hz. A ripple counter divides the frequencies down. Trying to characterize the divider chain showed us unstable signatures for every node after the first stage. The reason was that the circuits operated asynchronously with as much as 500 ns skew from the first to the last stage. Lowering the clock frequency to about 2 MHz by removing the crystal from the oscillator, we were able to define stable signatures for the counter chain. However, one measurement lasted 10 seconds, and to verify whether it was stable or not we had to have at least two complete measurements. An alternative to reducing the clock frequency was a new test setup for the slower parts of the divider. However, it is always wise to minimize the number of necessary setups.

So we learned that *synchronous operation is essential for high-speed testing.* Fortunately, this is easy to accomplish in most microprocessor designs, even those with the newer types of microprocessors that use asynchronous handshake lines to gate information in and out. Although it might seem at first that signature analysis is not applicable in such a case, the problem of asynchronous operation can be eliminated if the handshake lines are used to clock the data into the analyzer. Also, when this is done third-state conditions are no longer a problem because at the time of the "data valid" signal the data is either high or low. Thus a seemingly asynchronous system can behave as a synchronous system as seen by the troubleshooting tool, the signature analyzer.

Need for Designed-In Capability

An interesting possibility is that of measuring all the possible fault conditions at a central node by inducing faults into a good circuit and recording the corresponding signatures at the central node in a signature fault table. Testing the central node then tells the whole story of whether the instrument is in working condition or not. If it fails, the fault table indicates where the error is and sometimes even which part has to be replaced.

In the case of the terminal, an ideal central node seemed to be the video signal. Every data and control line is ultimately concentrated in one node containing all the information to scan a dot across the screen. Using the terminal's self-test feature as a stimulus, we chose the new-frame trigger signal as our START and STOP inputs. But for some reason we could not get stable signatures, which meant that the data stream between the two gate signals was not the same for each frame.

The culprits were two signals that occurred at a much slower rate than the 60 Hz for the frames. The blinking signals for the 2½-Hz cursor and for the 1¼-Hz enhancement were changing the characteristic data stream for the frames. Not until we disabled the signal generators for the blinking did we get stable signatures.

If parts of the circuit are being disabled the comprehensiveness of the test is reduced. In this case the designer could have provided the necessary setup to do a complete test. But after the design is frozen without signature analysis in mind it is hard to apply it successfully. If the window for signature analysis is selected so that the slowest blinker is the trigger for START and STOP, it is possible to create a stable signature or, in other words, a repeatable data stream.

The characterization of the RAM required special attention. A defined pattern had to be loaded into the memories before useful signatures were obtained at the outputs. By using several jumpers to enable the write cycle and the ROM outputs, then switching into

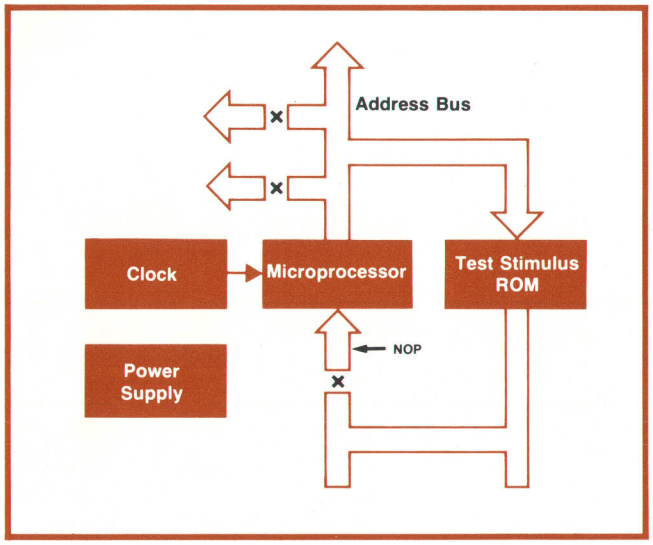


Fig. 2. *Signature analysis test procedures should verify the operation of key portions—the "kernel"—of a product first, then use the kernel to test other circuits. A typical kernel might consist of microprocessor, clock, power supply, and one or more read-only memories.*

a read mode for the RAM and disabling the ROM, we effectively loaded the content of the ROMs into the RAMs. After that, valid readings were obtainable and it was possible to trace down a bad RAM component.

Having tested and characterized about 30% of the digital boards, we next concentrated our efforts on the large display memory. Testing this dynamic memory was not easy, because it went through an asynchronous refresh cycle every 2 ms. Even adding more jumper wires, we had to admit finally that without cutting leads or altering the circuit we would not get any satisfactory results.

Looking at the CRT terminal with the oscillator crystal removed, with a cut-up extender board and jumper wires clipped into the circuit here and there, we learned the most important lesson: *signature analysis capability has to be designed into the circuit.*

After that a number of additional products were tested and the message remained the same: retrofitting is not an effective approach. On the other hand, it became clear that the additional effort to make the circuit signature-analysis-compatible is indeed very small if done at an early stage of the product development. Early, in this case, means the breadboard stage.

Thoughts on Implementation

The versatility of the signature analysis concept is impressive. As long as the data is valid at the selected clock edge, and the stimulus is repeatable, many parameters can be selected. The window length, or the number of bits in the data stream, can be of any value (100,000 bits is not unusual). Any suitable signal can be selected as the clock input, enabling the designer to make seemingly asynchronous circuits look as if they were synchronous. A major advantage is that everything happens at normal speed.

The implementation of signature analysis into a product is similar to designing a microprocessor into a product. In the latter case, the designer has to learn the instructions, and has to understand the advantages and the limitations of the microprocessor. Because of the learning curve, the first application will most likely take more time than later designs. Also, there is no cookbook approach to a microprocessor design because there are no two situations alike. The designer makes decisions based on the evaluation of power consumption, cost, size, reliability, and so on.

The same is true for the implementation of signature analysis. The design engineer must understand the function of each component and create a test stimulus that tests each function totally. Simply exercising a node may not be enough. Even a component as simple as an AND gate may have stable and correct input and output signatures and still be bad, as shown in Fig. 3. Similar cautions apply to any test method, of course. The designer must be careful to *test completely the function of the smallest replaceable part.*

Serviceability is an additional algorithm. If the service algorithm is taken into consideration at an early stage of the development, the application will be easier, and the additional cost for hardware, test program memory space, and development time will be offset by shorter test times in production. Also, the warranty service and repair costs will be much lower. Later in an actual example, we will see how even the

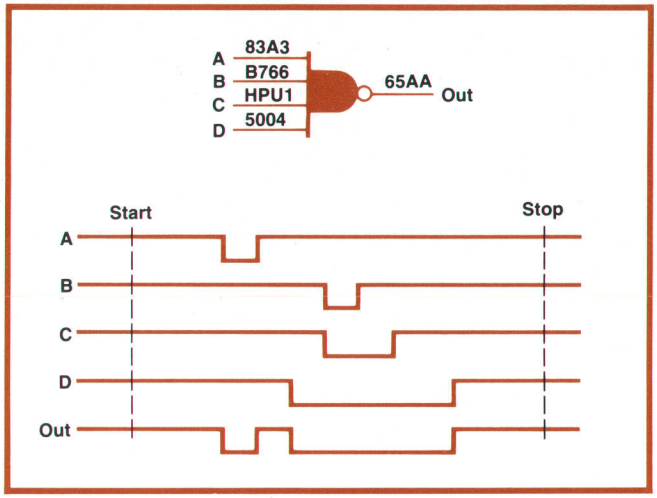


Fig. 3. *The test stimulus should test each component thoroughly. Otherwise a circuit, such as this AND gate, can have stable and correct signatures at each input and output node and still be bad. For example, input B or C might have an open bonding wire inside the IC, but in this case the error is masked by input D. Careful test stimulus design avoids this.*

Designer Guidelines for Applying Signature Analysis to Microprocessor-Based Products

General Guidelines

- Make a full commitment to use signature analysis at the definition stage of the product.
- Evaluate the trade-offs, such as increased factory costs versus lower test time in production and lower warranty and repair cost in the field. Other factors that might influence the decision are warranty cost goal, profit, cost of field repair, acceptable downtime, the cost of alternative service procedures like board exchange programs, the topography of the service organization, and the extra cost for customs if the parts have to be shipped back and forth across country borders.
- Familiarize yourself with the signature analysis service philosophy and allow some extra time for the design.
- Start to prove the basic working of signature analysis at the breadboard stage, before laying out printed circuit boards.
- Team up with the service engineer who will write the manual for the product. Do it at an early stage, before the first prototype is finished.
- If you hope for some benefit for production testing, get the production engineer involved during your definition of the test stimulus and the method of connecting the signature analyzer.
- As a design engineer, be aware that the volume of the necessary documentation can be minimized by selecting the appropriate partitioning of the tested sections in the product.

Technical Rules

- The stimulus for troubleshooting comes from within the product. The self-test stimulus can frequently be used.
- Provide if possible a free-running repeatable stimulus for continuous cycling.
- Tested nodes are to be in a valid and repeatable state at the time of the selected clock edge for triggering.
- Provide easy access to the START, STOP, and CLOCK test points.
- Feedback loops must be capable of being opened. Only an open-ended test allows backtracing.
- The test program or stimulus should exercise within the START-STOP window all the functions that are used in the instrument, although it is not necessary to perform a meaningful operation.
- Provide a controlled test stimulus to interrupt lines, open connectors, and signals that are normally asynchronous.

Additional Guidelines

- Verify the heart or kernel of the instrument first. The kernel may consist of the power supply, the clock generator, and the microprocessor. Then, use this central part to create the stimulus for the peripheral circuits.
- ROMs may be used to write the stimulus program.
- Divide the circuit into well defined portions. Several test setups may be necessary.
- Avoid the use of circuits with non-repeatable delays (e.g. one-shot multivibrators) within the test loop.
- Avoid, if possible, the third-state condition of a three-state node during the measurement cycle.

factory cost can be lowered in spite of needing some extra components, because the whole circuit could be placed on a single large board, while for the traditional board swap service approach, the circuit would have been divided into a number of easy-to-replace subassemblies, which would have required more connectors and hardware to hold the boards in place.

Signature Analysis and the Service Engineer

How would a service engineer use signature analysis if a product failed? The assumption is made that the signature analysis method is designed into the product. Instructions on the schematic or in the service manual show how to switch the product into the diagnostic mode and how to connect the signature analyzer to the device under test. Each node on the schematic is marked with a signature (Fig. 4). With the aid of the schematic the service engineer first reads the output signatures of the device under test. If they are bad, he traces back to a point in the circuit where a good signature appears at the input side of a component and a bad one at the output side. This is called backtracing.

Some understanding of the components in a digital schematic is essential. The direction of the data flow is important but no special knowledge about the actual function of the assembly is required. So, one advantage of the signature analysis service method is that less training is needed to learn to do fault tracing. We can even go a step further and develop a troubleshooting tree without the use of a schematic. A picture of the physical board with signatures at the pins of IC's or components may be used instead (Fig. 5). This way the technician is not required to know whether the circuit he is testing contains a complex storage device or simply a gate. One suggestion is to print the signatures onto the printed circuit board itself, with arrows indicating the signal flow. Another is to print a test template that is attached to the component side of the circuit board when service is required (see Fig. 6). Holes in the appropriate locations, signatures, and other instructions printed on the template guide the service person to the faulty node.

The 3455A Voltmeter—an Example

The first HP instrument using signature analysis is the 3455A Digital Voltmeter[1] (Fig. 7). The digital portion of this instrument is quite extensive. It is microprocessor controlled and contains an elaborate self-test program stored in ROM. If the self test fails, a jumper inside the enclosure is removed, breaking feedback loops and enabling the self-test program, which is then used to troubleshoot the instrument.

Signature analysis influenced other factors that make this voltmeter easier to troubleshoot down to the component level. The entire digital portion is on a

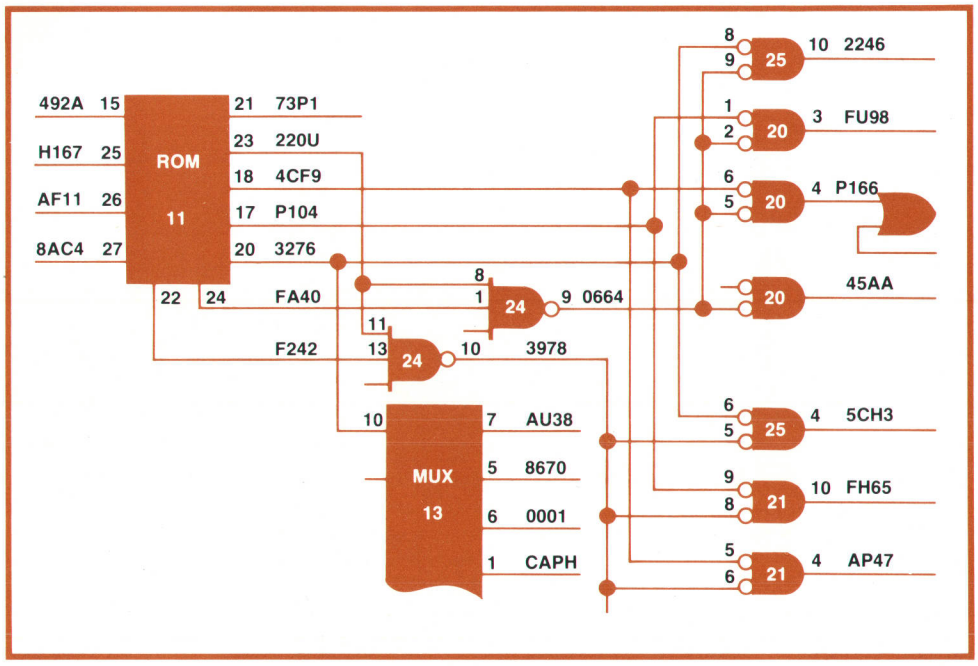


Fig. 4. *An example of an annotated schematic, showing correct signatures at various circuit nodes.*

single board. The elimination of connectors and a multitude of smaller subassemblies not only reduced production cost, but also made all the parts easily accessible for testing without special extender boards.

Some extra design time, a few more ROM locations, and the extra jumper wire were the price paid for serviceability. A cost evaluation verified that the production cost was lowered. The extra design time amounted to approximately 1% of the overall development time.

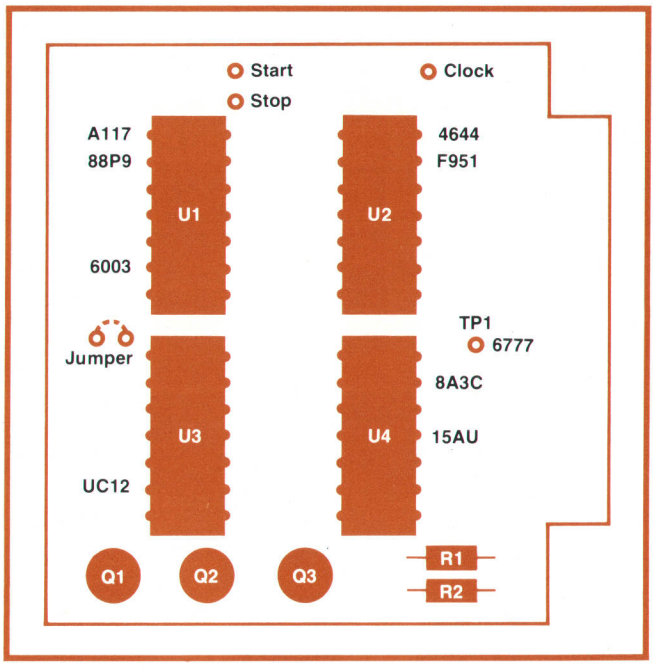


Fig. 5. *A service manual may use a picture or drawing of the board being tested, showing proper signatures at various test points. A troubleshooting tree in the manual guides the service person, who need not know the function of each component. A board overlay or template may also be used.*

Besides the design engineer, the service engineer who wrote the service manual made an important contribution to the successful application of signature analysis. He learned the internal algorithms of the product almost as well as the designer. Because there was no precedent to fall back on, he used a number of innovative ideas, which have been well accepted by the field engineers.

The service manual guides the service person to the fault within a very short time. The manual contains a troubleshooting tree that, combined with annotated schematics and graphs of board layouts, leads directly to the bad node. In some cases the manual gives instructions as to which IC to replace. In other cases the use of a logic probe, which may be the 5004A Signature Analyzer's data probe, may be required. A current sensor helps to find short circuits between traces or to ground and is particularly helpful if a long bus line should fail. A portion of the 3455A Voltmeter troubleshooting tree is shown in Fig. 8.

The first test checks the kernel, which consists of the microprocessor, the clock circuit, the power supply, and a number of external gates. After proper functioning of the kernel is verified, the test setup is changed (one control input of the signature analyzer is moved to another pin) and the remaining portions of the circuit are tested.

A special portion of the ROM control loads and reads the RAMs. Some asynchronous portions require a third test setup. Again, the connection of the START wire is simply moved to the next pin designated for

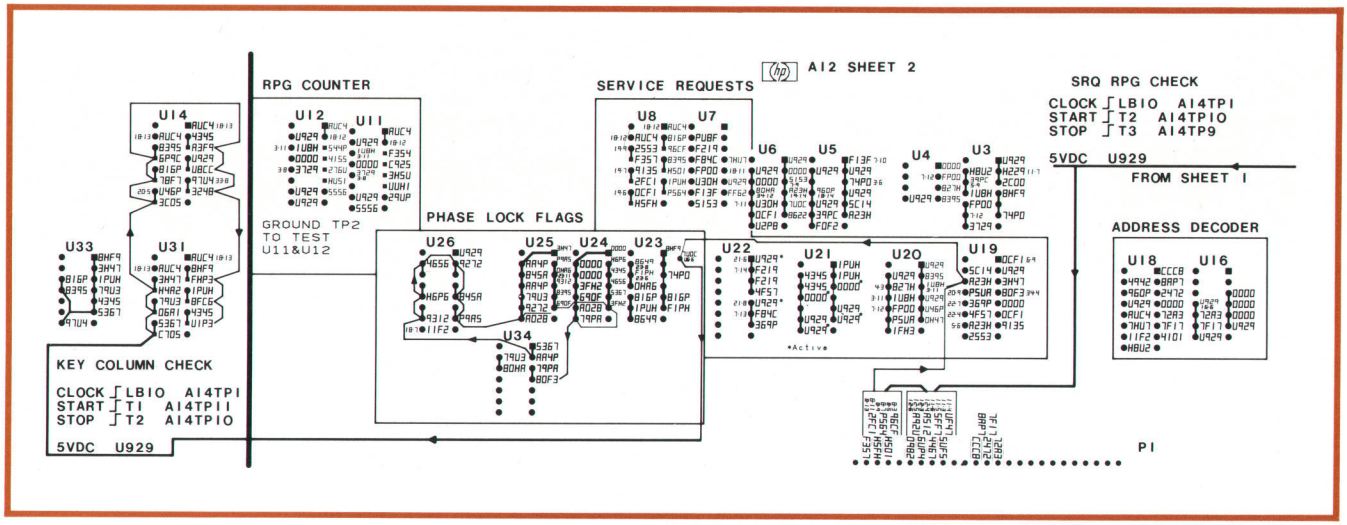


Fig. 6. *A template for signature analysis troubleshooting. The template is attached to the component side of the circuit board. Holes allow probe access to the test points. If the test point is not a source, the origin of the signal (IC and pin number) is listed next to the correct signature.*

this purpose and troubleshooting can continue.

It is obvious that proper documentation is essential. The 3455A manual shows, for each test setup, a picture of the board. Only the signatures related to that particular test are given. This helps to direct the effort towards the important areas on the board. Interrupt signals are simulated by the ROM program so they occur repeatedly at the same spot within a window and stable signatures result.

Fig. 7. *Model 3455A System Voltmeter is the first HP instrument designed for troubleshooting with the 5004A Signature Analyzer.*

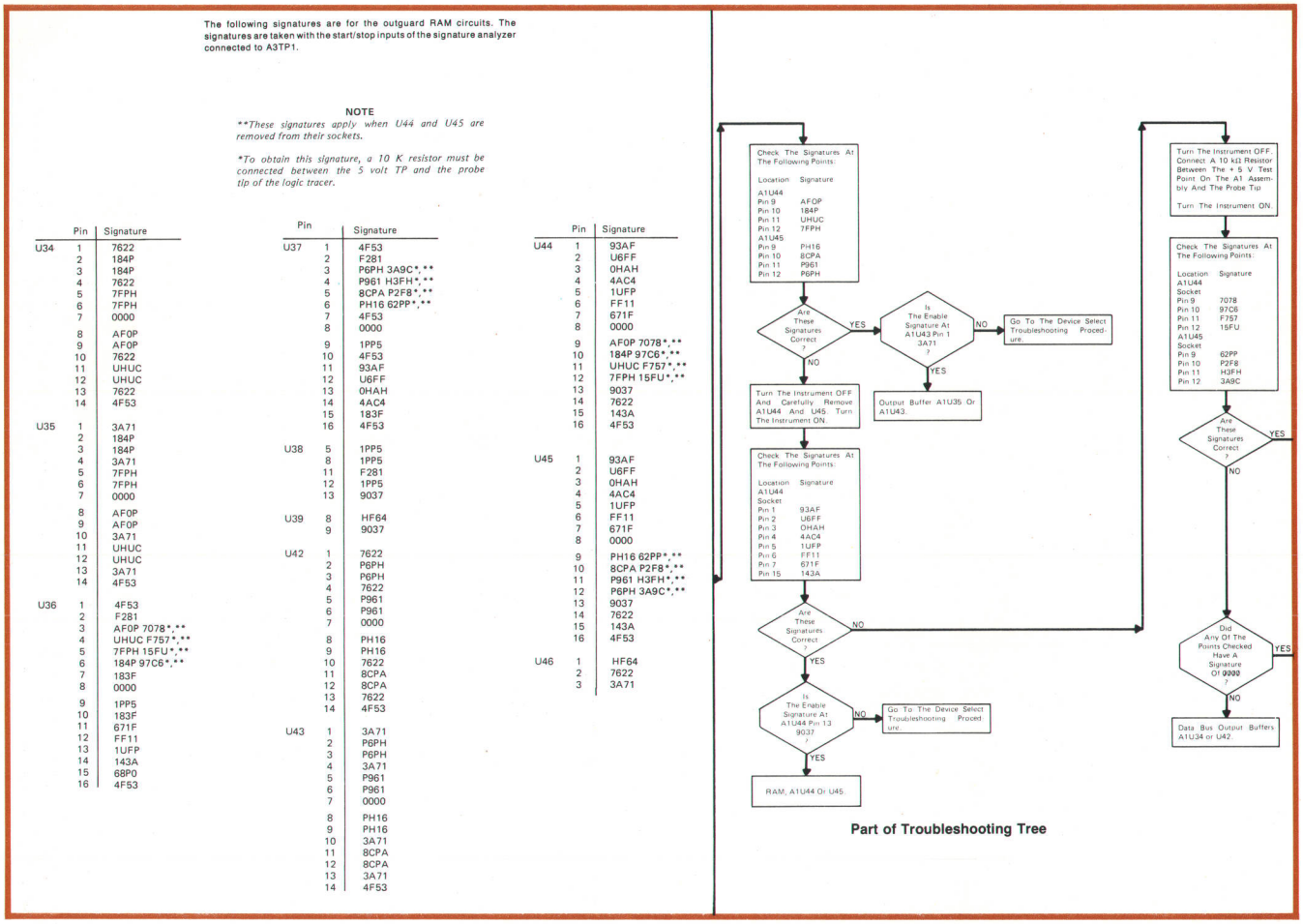
When all the signatures seem to be bad, the question arises whether the test setup itself is correct. The 5004A Signature Analyzer's self-test feature can be used to check it for proper operation. Each test setup can then be tested by touching V_{cc} with the 5004A probe. If this characteristic signature is correct, it means that the START and STOP channels are triggered at the correct moments and that the number of clock pulses within the measurement window is correct. It also tells the user that the switches on the signature analyzer are set correctly, and that all the jumpers, switches, and control buttons in the voltmeter are set to the right position. Thus the confidence level is very high at the beginning of a test routine.

A conclusion drawn from this application is as follows: success is assured if the service engineer works closely with the design engineer. This also saves time at the end of the development phase because the service engineer is fully aware of the new product's internal operation. It also forces the designer to think about serviceability.

The fact that signature analysis is built into the 3455A Voltmeter not only made serviceability but also final testing on the production line much easier. The signature analyzer is now a standard piece of equipment on the production line.

Acknowledgments

The 5004A Signature Analyzer is a result of the effort of many engineers in HP's Colorado Springs and Santa Clara Divisions. Loveland Instrument Division and Santa Rosa Division added a great deal to definition of the needs and conditions of the new



The following signatures are for the outguard RAM circuits. The signatures are taken with the start/stop inputs of the signature analyzer connected to A3TP1.

NOTE

**These signatures apply when U44 and U45 are removed from their sockets.

*To obtain this signature, a 10 K resistor must be connected between the 5 volt TP and the probe tip of the logic tracer.

	Pin	Signature
U34	1	7622
	2	184P
	3	184P
	4	7622
	5	7FPH
	6	7FPH
	7	0000
	8	AF0P
	9	AF0P
	10	7622
	11	UHUC
	12	UHUC
	13	7622
	14	4F53
U35	1	3A71
	2	184P
	3	184P
	4	3A71
	5	7FPH
	6	7FPH
	7	0000
	8	AF0P
	9	AF0P
	10	3A71
	11	UHUC
	12	UHUC
	13	3A71
	14	4F53
U36	1	4F53
	2	F281
	3	AF0P 7078*,**
	4	UHUC F757*,**
	5	7FPH 15FU*,**
	6	184P 97C6*,**
	7	183F
	8	0000
	9	1PP5
	10	183F
	11	671F
	12	FF11
	13	1UFP
	14	143A
	15	68P0
	16	4F53

	Pin	Signature
U37	1	4F53
	2	F281
	3	P6PH 3A9C*,**
	4	P961 H3FH*,**
	5	8CPA P2F8*,**
	6	PH16 62PP*,**
	7	4F53
	8	0000
	9	1PP5
	10	4F53
	11	93AF
	12	U6FF
	13	0HAH
	14	4AC4
	15	183F
	16	4F53
U38	5	1PP5
	8	1PP5
	11	F281
	12	1PP5
	13	9037
U39	8	HF64
	9	9037
U42	1	7622
	2	P6PH
	3	P6PH
	4	7622
	5	P961
	6	P961
	7	0000
	8	PH16
	9	PH16
	10	7622
	11	8CPA
	12	8CPA
	13	7622
	14	4F53
U43	1	3A71
	2	P6PH
	3	P6PH
	4	3A71
	5	P961
	6	P961
	7	0000
	8	PH16
	9	PH16
	10	3A71
	11	8CPA
	12	8CPA
	13	3A71
	14	4F53

	Pin	Signature
U44	1	93AF
	2	U6FF
	3	0HAH
	4	4AC4
	5	1UFP
	6	FF11
	7	671F
	8	0000
	9	AF0P 7078*,**
	10	184P 97C6*,**
	11	UHUC F757*,**
	12	7FPH 15FU*,**
	13	9037
	14	7622
	15	143A
	16	4F53
U45	1	93AF
	2	U6FF
	3	0HAH
	4	4AC4
	5	1UFP
	6	FF11
	7	671F
	8	0000
	9	PH16 62PP*,**
	10	8CPA P2F8*,**
	11	P961 H3FH*,**
	12	P6PH 3A9C*,**
	13	9037
	14	7622
	15	143A
	16	4F53
U46	1	HF64
	2	7622
	3	3A71

Part of Troubleshooting Tree

Fig. 8. *An example of a troubleshooting chart from the 3455A Voltmeter service manual. The chart tells which part to replace under certain conditions.*

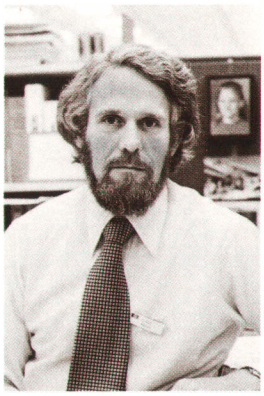
microprocessor-based instruments relating to troubleshooting and service.

In addition to the authors of the articles in this issue, key contributors include David Kook, who managed to pack the 5004A into an existing plastic case, and Kuni Masuda, who designed the front panel to give it a well-balanced appearance. Gary Gitzen did the breadboarding and designed the unstable signature feature.

It is also a pleasure to acknowledge the many valuable inputs from Dan Kolody and George Haag in Colorado Springs, Kamran Firooz and David Palermo in Loveland, Jan Hofland who is now with the Data Systems Division, Ed White in our own marketing department, and Dick Harris who brought the instrument into production. Gary Gordon as section manager was instrumental in getting the algorithm implemented in a service tool.

Reference

1. A. Gookin, "A Fast-Reading, High-Resolution Voltmeter that Calibrates Itself Automatically," Hewlett-Packard Journal, February 1977.



Hans J. Nadig

Hans Nadig is signature analysis project manager at HP's Santa Clara Division. Holder of an MS degree in electrical engineering from the Federal Institute of Technology in Zürich, he's been with HP since 1967, contributing to the design of instruments for Fourier analysis and serving as a project manager in the IC laboratory. Born in Berne, Switzerland, Hans served two years in the Swiss Army. He's a rock climber, a qualified instructor and guide in high-altitude mountaineering, and a campaigner for environmental protection, especially through power conservation, in local government circles. He also participates in HP's career counseling program for high-school students, and enjoys photography and gardening. Hans is married, has two daughters, and lives in Saratoga, California.

Personal Calculator Algorithms I: Square Roots

A detailed description of the algorithm used in Hewlett-Packard hand-held calculators to compute square roots.

by William E. Egbert

BEGINNING WITH THE HP-35,[1,2] all HP personal calculators have used essentially the same algorithms for computing complex mathematical functions in their BCD (binary-coded decimal) microprocessors. While improvements have been made in newer calculators,[3] the changes have affected primarily special cases and not the fundamental algorithms.

This article is the first of a series that examines these algorithms and their implementation. Each article will present in detail the methods used to implement a common mathematical function. For simplicity, rigorous proofs will not be given, and special cases other than those of particular interest will be omitted.

Although tailored for efficiency within the environment of a special-purpose BCD microprocessor, the basic mathematical equations and the techniques used to transform and implement them are applicable to a wide range of computing problems and devices.

The Square Root Algorithm

This article will discuss the algorithm and methods used to implement the square root function.

The core of the square root algorithm is a simple approximation technique tailored to be efficient using the instruction set of a BCD processor. The technique is as follows:

$\sqrt{x}$ is desired

1. Guess an answer a
2. Generate a^2
3. Find $R = x - a^2$
4. If the magnitude of R is sufficiently small, $a = \sqrt{x}$.
5. If R is a positive number, a is too small.
   If R is a negative number, a is too big.
6. Depending on the result of step 5, modify a and return to step 2.

The magnitude of R will progressively decrease until the desired accuracy is reached.

This procedure is only a rough outline of the actual square root routine used. The first refinement is to avoid having to find a^2 and $x - a^2$ each time a is changed. This is done by finding a one decade at a time. In other words, find the hundreds digit of a, then the tens digit, the units digit, and so on. Once the hundreds digit is found, it is squared and subtracted from x, and the tens digit is found. This process, however, is not exactly straightforward, so some algebra is in order.

The following definitions will be used:

x = the number whose square root is desired

a = most significant digit(s) of $\sqrt{x}$ previously computed

b = the next digit of $\sqrt{x}$ to be found

j = the power of 10 associated with b

$R_a = x - a^2$, the current remainder

a_j = the new a when digit b is added in its proper place. $a_j = a + (b \times 10^j)$ (1)

R_b = the portion of remainder R_a that would be removed by adding b to a. $R_b = a_j^2 - a^2$ (2)

For example, let $x = 54756$. Then $\sqrt{x} = 234$.

Let $a = 200$.

b = the digit we are seeking (3, in this case)

j = 1 (the 10's digit is being computed)

$R_a = 54756 - (200)^2 = 14756$.

Note that a_j and R_b will vary with the choice of b.

The process of finding $\sqrt{x}$ one decade at a time approaches the value of $\sqrt{x}$ from below. That is, at any point in the computation, $a \leq \sqrt{x}$. Consequently, $R_a \geq 0$.

With this in mind it is easy to see that for any decade j, the value of b is the largest possible digit so that

$$R_a - R_b \geq 0$$

or

$$R_b \leq R_a. \quad (3)$$

Using equations 1 and 2 we have

$$R_b = [a + (b \times 10^j)]^2 - a^2.$$

Expanding and simplifying,

$$R_b = 2ab \times 10^j + (b \times 10^j)^2. \quad (4)$$

Inserting (4) into (3) yields the following rule for finding digit b.

Digit b is the largest possible digit so that

$$2ab \times 10^j + (b \times 10^j)^2 \leq R_a \quad (5)$$

When the digit that satisfies equation 5 is found, a new a is formed by adding $b \times 10^j$ to the old a, the decade counter (j) is decremented by 1, and a new R_a is created; the new R_a is the old R_a minus R_b.

Continuing the previous example,

$x = 54756$

$j = 1$

$a = 200$

$x - a^2 = R_a = 14756$

Applying equation 5 to find b:

b	$R_b = 2ab \times 10^j + (b \times 10^j)^2$	$R_a - R_b$
0	0	14756
1	4100	10656
2	8400	6356
3	12900	1856
4	17600	−2844

Thus $b=3$, since $b=4$ causes overdraft, i.e., $R_a - R_b < 0$. The new $a = 200 + 3 \times 10^1 = 230$. The new $R_a = 1856$, the new $j=0$. With these new parameters, the units digit can be found.

This process may seem vaguely familiar, which is not surprising since upon close inspection it turns out to be the (usually forgotten) scheme taught in grade school to find square roots longhand. Of course, trailing zeros and digits are not written in the longhand scheme.

To make this process efficient for a calculator, still another refinement is needed.

$(b \times 10^j)^2$ can be expressed as a series, using the fact that the square of an integer b is equal to the sum of the first b odd integers. Thus,

$$(b \times 10^j)^2 = b^2 \times 10^{2j}$$

$$= \sum_{i=1}^{b} (2i-1) \times 10^{2j}$$

For example,

$$(3 \times 10^j)^2 = 1 \times 10^{2j} + 3 \times 10^{2j} + 5 \times 10^{2j}$$

$$= 9 \times 10^{2j}$$

Thus $2ab \times 10^j + (b \times 10^j)^2$ can be expressed as:

$$2ab \times 10^j + (b \times 10^j)^2 = \sum_{i=1}^{b} 2a \times 10^j + (2i-1) \times 10^{2j}$$

or

$$R_b = \sum_{i=1}^{b} 2a \times 10^j + (2i-1) \times 10^{2j} \qquad (6)$$

Now comes a key transformation in the square root routine. It was shown earlier how inequality 3 will give the value b for the next digit of a. Since multiplying both sides of an inequality by a positive constant does not change the inequality, equations 3 and 6 can be multiplied by the number 5.

$$5R_b \leq 5R_a$$

$$5R_b = \sum_{i=1}^{b} 10a \times 10^j + (10i-5) \times 10^{2j} \qquad (7)$$

b becomes the largest digit so that $5R_b \leq 5R_a$. The new $5R_a$ is equal to the old $5R_a$ minus $5R_b$.

These transformations may seem useless until we examine a few examples of the last term of the right side of (7) for various values of b.

$$10a \times 10^j + 05 \times 10^{2j}, b=1$$
$$10a \times 10^j + 15 \times 10^{2j}, b=2$$
$$10a \times 10^j + 25 \times 10^{2j}, b=3$$

Notice that the two-digit coefficient of 10^{2j} consists of $(b-1)$ and a 5. These two digits will be expressed as $(b-1)|5$ in succeeding equations. $10a$ is formed by a simple right shift and does not change between terms. If the sum defined in equation 7, as b is incremented by 1, is subtracted from $5R_a$ until overdraft occurs, the digit in the next-to-last digit position is b. Best of all, it is in the exact position to form the next digit of a without further manipulation. Redoing the previous example may help clarify matters.

$R_a = 14756$

$j = 1$

$a = 200$

$5R_a = 73780$

b	$10a \times 10^j + (b-1)\|5 \times 10^{2j}$	$5R_a - 5R_b$
1	20500	53280
2	21500	31780
3	22500	9280 new $5R_a$
4	23500	−14220 overdraft
	new value of a digits	

Notice that when overdraft occurs the new value of a is already created and the new value of $5R_a$ can be found by restoring the previous remainder.

Decrementing the value of j would cause, in effect, $(10a \times 10^j)$ to shift right one place, and $(b-1)|5 \times 10^{2j}$ to shift right two places. The result is that the final 5 shifts one place to the right to make room for a new digit. Continuing with the same example,

$5R_a = 9280$

$a = 230$

$j = 0$

b	$10a\times10^{j}+(b-1)\vert5\times10^{2j}$	$5R_a-5R_b$	
1	2305	6975	
2	2315	4660	
3	2325	2335	
4	2335	0	remainder
5	2345	−2345	overdraft
	final $a = \sqrt{x}$		

For ease of understanding, the preceding example treated a large positive number. A number in the calculator actually consists of a mantissa between 1 and 10 and an exponent. The problem is to find the square root of both parts of this argument. Happily, if the input exponent is an even number, the portion of the answer resulting from it turns out to be the exponent of the final answer and is simply the input exponent divided by 2. Thus to find $\sqrt{x}$, the exponent of x is first made even and the mantissa shifted to keep the number the same. The exponent of $\sqrt{x}$ is found by dividing the corrected input exponent by 2. The method described above is then used to find the square root of the shifted input mantissa, which (after possibly being shifted) can be between 1 and 100. The result will then be between 1 and 10, which is the range required for the mantissa of $\sqrt{x}$.

During the process of finding $\sqrt{x}$ the remainder R_a progressively decreases. To avoid losing accuracy, this remainder is multiplied by 10^j after finding each new digit b. This avoids shifting a at all, once the square root extraction process begins. A 12-digit mantissa is generated, which insures accuracy to ±1 in the tenth digit of the mantissa of $\sqrt{x}$.

In summary, the computation of $\sqrt{x}$ proceeds as follows:

1. Generate exponent of answer.
2. Multiply mantissa by 5 to create original $5R_a$
3. With an original a of 0, use the method described above to find 12 b digits to form the mantissa of the answer.
4. Round the mantissa and attach the exponent found previously.
5. Display the answer.

The calculator is now ready for another operation.

References

1. T.M. Whitney, F. Rodé, and C.C. Tung, "The 'Powerful Pocketful': An Electronic Calculator Challenges the Slide Rule," Hewlett-Packard Journal, June 1972.
2. D.S. Cochran, "Algorithms and Accuracy in the HP-35," Hewlett-Packard Journal, June 1972.
3. D.W. Harms, "The New Accuracy: Making $2^3=8$," Hewlett-Packard Journal, November 1976.

William E. Egbert

Bill Egbert is a project leader at HP's Corvallis, Oregon Division. He produced this series of algorithm articles as part of his work on the HP-67 and HP-97 Programmable Calculators. He was project leader for the HP-67 and did microprogramming for both calculators. Bill received his BSEE degree from Brigham Young University in 1973 and his MSEE from Stanford University in 1976. He's been with HP since 1973. Born in Fallon, Nevada, he's married, has two small children, and lives in Corvallis.

Hewlett-Packard Company, 1501 Page Mill Road, Palo Alto, California 94304

HEWLETT-PACKARD JOURNAL

MAY 1977 Volume 28 • Number 9

Technical information from the Laboratories of Hewlett-Packard Company

Hewlett-Packard Central Mailing Department
Van Heuven Goedhartlaan 121
Amstelveen-1134 The Netherlands
Yokogawa-Hewlett-Packard Ltd., Shibuya-Ku
Tokyo 151 Japan

Editorial Director • Howard L. Roberts
Managing Editor • Richard P. Dolan
Art Director, Photographer • Arvid A. Danielson
Illustrator • Susan E. Wright
Administrative Services, Typography • Anne S. LoPresti
European Production Manager • Dick Leeksma

Bulk Rate
U.S. Postage
Paid
Hewlett-Packard
Company

CHANGE OF ADDRESS: To change your address or delete your name from our mailing list please send us your old address label (it peels off). Send changes to Hewlett-Packard Journal, 1501 Page Mill Road, Palo Alto, California 94304 U.S.A. Allow 60 days.

HP Archive

On-line curator: John Miles, KE5FX

jmiles@pop.net