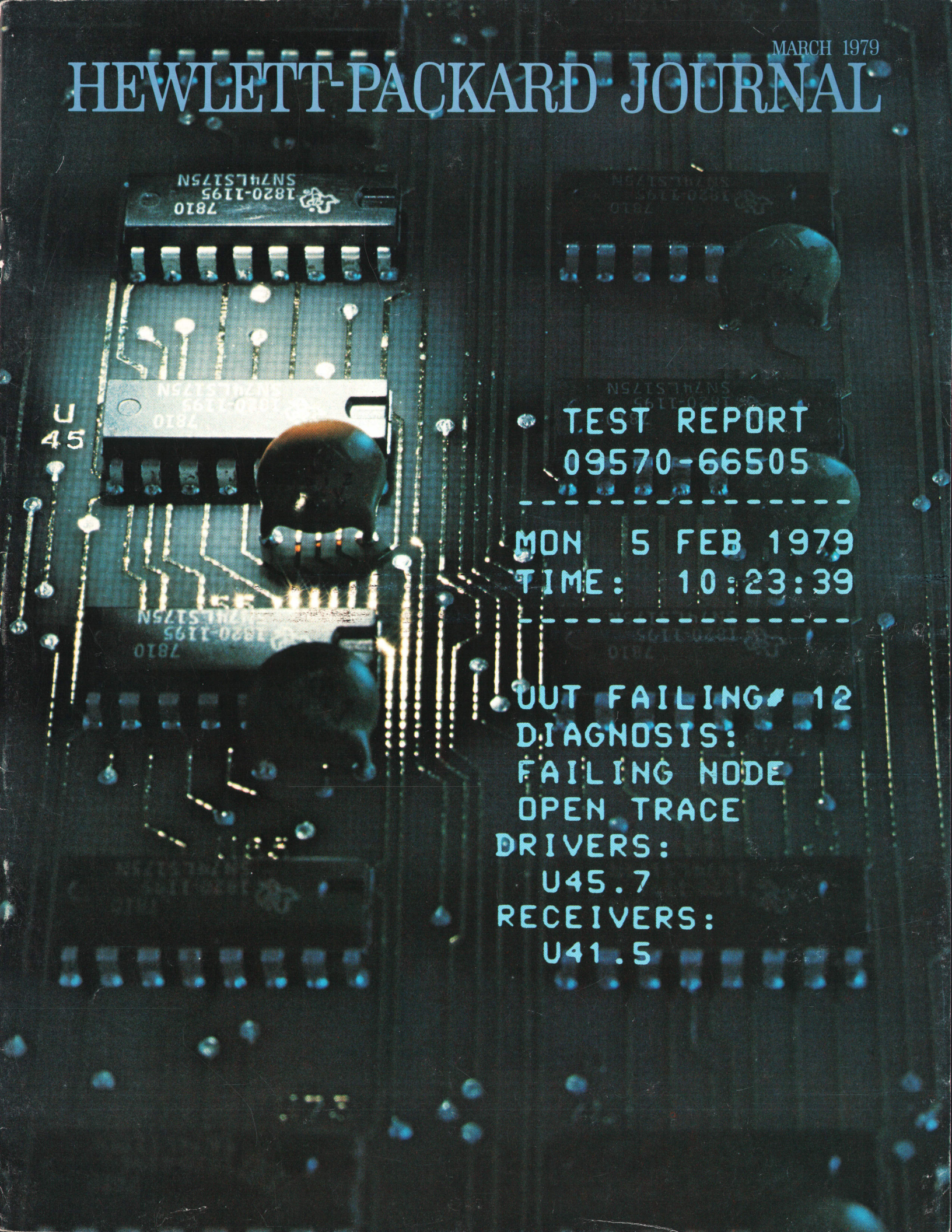
MARCH 1979

HEWLETT-PACKARD JOURNAL



Circuit Board Testing: Cost-Effective Production Test and Troubleshooting

Two new printed-circuit-board test systems find faults in complicated circuit boards quickly and efficiently to help speed production throughput.

by Peter S. Stone and John F. McDermid

PRODUCTION TEST AND TROUBLESHOOTING are rapidly becoming the major bottlenecks in the manufacture of sophisticated electronic equipment. Higher degrees of semiconductor circuit integration are packing more and more functions into electronic equipment, requiring more elaborate test procedures to assure that the equipment will operate when first turned on. Furthermore, the higher circuit density also makes the troubleshooting of faulty systems more complicated and time-consuming.

Experience indicates that when printed circuit boards have more than a certain level of complexity, every board will have at least one fault when it reaches the end of the production line. Unless each board is checked and corrected before being assembled into the final system, the system cannot be expected to function when first turned on.

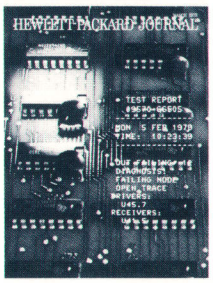
Experience also indicates that more than half the circuit board faults are process related, that is, the fault is a solder bridge, a damaged trace, or a misloaded or damaged component. Incoming inspection of components won't prevent these faults, and trying to find them after the complete system is assembled usually becomes an exercise in futility. Hence, more and more manufacturers are turning to methods of testing individual circuit boards as the most cost-effective way to conduct production test and troubleshooting.

Types of Tests

Circuit board testing can be classified as either functional or in-circuit testing. Functional testing is a test of how the circuits operate, that is, the appropriate stimuli are applied to the normal board inputs and the outputs are monitored for proper responses. In-circuit testing involves tests of individual components on the completed board. In this case, the appropriate stimulus is applied directly to each component and the measurement is made on the component.

The measurement challenge for in-circuit testing is to exclude the influence of other components connected to the component being measured. This usually requires a "bed-of-nails" fixture—one in which contacts are made to every point on the board where measurements are to be made—and some fairly sophisticated stimulus and measurement instrumentation using some form of component isolation, usually by guarding. However, the actual test is rather simple because only a small portion of the board is examined at one time, and there is minimum interaction between the component being tested and the surrounding circuits.

The challenge of functional testing, on the other hand, is to develop a set of stimuli that exercises all of the circuits on the board in such a way that faults can be detected and isolated. This task is not an easy one since fault isolation becomes very difficult as board complexity increases.



Cover: *Spot-lighting circuit and component faults (figuratively speaking) is one of the ways automatic test systems expedite final test of complex circuit boards. Two well-devised board test systems, and their very effective software, are described in this issue.*

In this Issue:

However, there is a trade-off between fixturing complexity and program development. Functional testing allows simple fixturing because electrical access to the board through its normal inputs is usually sufficient, but increased programming effort is then generally required.

This explains why Hewlett-Packard has developed two circuit-board testers. One, the DTS-70 (Fig. 1), is used primarily for functional testing and troubleshooting of complex digital circuit boards. The other, the Model 3060A (Fig. 2), is a hybrid aimed at testing both analog and digital circuits. It does in-circuit analog tests and performs both analog and digital functional tests with a digital capability that is complementary to that of the DTS-70. For example, Model 3060A can use signature analysis to test microprocessor-based boards at operating speeds. There is some overlap between the systems so that either one may be appropriate, depending on the nature of the testing requirements.

Functional Testing with the DTS-70

The growing use of medium-scale and large-scale integration makes it increasingly difficult to isolate individual digital circuits for in-circuit testing. Thus, functional testing assumes great importance for digital logic circuits.

In functional testing, an appropriate stimulus is applied to the circuit board inputs and the response is examined to decide whether the board is good or bad. The first problem in applying this technique is to find a stimulus pattern that tests the board to the level of confidence desired. The second problem is to determine what the correct response pattern should be. A third problem arises when a board is found to be bad: how to find the fault that makes the board bad.

Input patterns can be generated manually, random patterns can be used, patterns can be generated automatically using "path-sensitization" techniques, or combinations of these pattern-generating methods may be considered. The DTS-70 Digital Test System uses all of these methods. For example, it has a pseudorandom generator that can produce randomized patterns while allowing either straight binary or Grey code, looping, and manually entered sequences. A flexible automatic generator uses path-sensitization techniques to zero in on particular faults or classes of faults as directed by the test programmer.

This pattern-generating flexibility gives the test programmer the freedom to choose the most economical means of generating input patterns for the DTS-70 and to vary the pattern-generating techniques as the test program development proceeds. For example, a combination of pseudorandom and manually generated patterns might detect 70% of the faults that could occur on a particular board. The automatic generator might then be used to isolate some hard-to-find faults which, with some additional manual patterns, might bring the test effectiveness to 90% or higher.

Fig. 1. *Model DTS-70 Digital Test System gives fast and accurate go/no-go tests of complex digital circuit boards. If a board fails a test, the software presents messages to guide the operator in using the probe to find the fault.*

Board Response Simulation

The correct board responses to the chosen input patterns are calculated in the DTS-70 by means of a simulator program. Board-test simulation is a technique in which a description of the board's circuits is entered into the test system's computer, that is, the identity of each component and a description of all the circuit connections are entered (e.g., U3 pin 6 goes to U5 pin 8). From this description, the simulator calculates what the response should be to any input pattern.

Although simulation requires greater test preparation effort than some other techniques, it has a number of advantages that more than compensate for the increased effort. For example, since the programmer works from a schematic diagram, schematic verification is automatic because a known good board will fail the test if the schematic, and hence the test program, is incorrect. The simulator used in the DTS-70 also alerts the programmer to potential race and hazard problems (see the article beginning on page 13). Information about the test quality (number of detectable faults on the board versus total number of possible faults) is also part of the simulator's output.

Simulation also provides a solution to the problem of uninitialized memory elements. Boards may not always power up in the same state, or the state may vary depending on the test history. If a comparison technique were used in which the state of a known good board is memorized, some good boards might fail the test since they might not power up to the same state as the prototype board. The DTS-70 simulator, however, keeps track of which memory elements have not yet been initialized by the input test pattern, and instructs the hardware to ignore responses involving these elements.

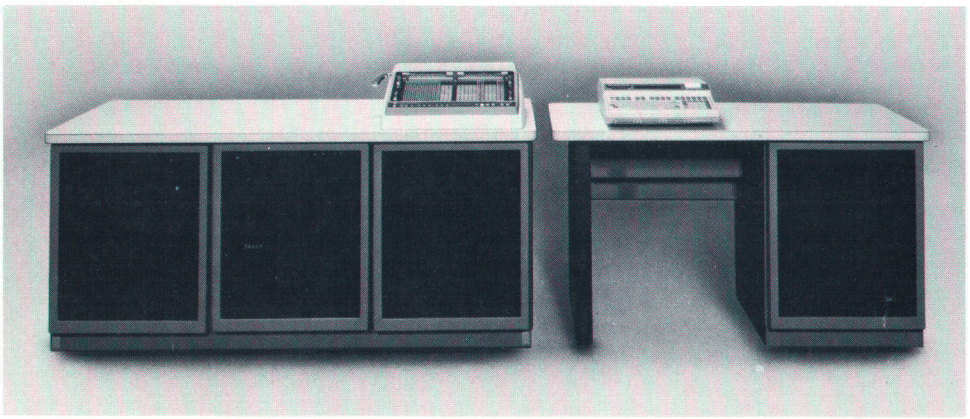


Fig. 2. *Model 3060A Board Test System uses a bed-of-nails test fixture to measure the impedance of individual components on a circuit board, and then performs both analog and digital functional tests on the board. The software automatically generates error messages that contain specific information about faults on boards that fail to pass.*

A major benefit of simulation is that it provides all information necessary for troubleshooting a board. The state of every node in the board for each test pattern input is contained in the simulator state file. Thus, all the fault signatures and other information necessary for backtracing to a fault are readily available. Also, since a known good board is not required for test generation, a simulator can be used during a board's design phase to verify the design and improve its testability.

The DTS-70 System

The DTS-70 system hardware consists of a test station (Model 9571A,) and a standard HP 1000 Computer system.[1] The interface to the circuit board under test is through plug-in digital circuit cards in the test station. Each card has 15 channels and up to 24 cards can be installed, giving a maximum of 360 I/O pins for the interface. The actual connection to the circuit board under test is through an adapter that applies the power supply and test signal channels to the appropriate pins on the circuit board's edge connector.

There are three types of plug-in driver/comparator interface cards: TTL, CMOS, and programmable. The TTL and CMOS cards have high-low and threshold logic levels compatible with their respective logic families. The programmable cards can be set for logic levels ranging between −16 and +16 volts with a current capability of 16 mA per channel. A relay switch card is also available for interfacing analog measurement equipment such as counters and/or digital multimeters. An HP-IB* I/O port provided with the system simplifies the integration of additional instruments.

The HP 1000 Computer System supplied with the DTS-70 has the 21MX E-Series Computer with a 2645A CRT Terminal and a 7906A Disc Drive. It uses the standard RTE-IV operating system software[2] and so may be used for other multiterminal computer applications while being used for testing and/or test development. Many peripherals, such as line printers, magnetic tape drives, and so on, can be supported by the system.

Although the DTS-70 may be easily configured to meet a wide range of customer requirements, it is supplied in a standard configuration, adequate for most applications, that supports with a single HP 1000 Computer System up to three 9571A Test Stations, and seven additional terminals for concurrent test-development programming.

*HP-IB: Hewlett-Packard interface bus, HP's implementation of IEEE 488-1975 and identical ANSI MC1.1.

The software supplied with the DTS-70 system has two major parts: TESTAID and FASTRACE. TESTAID is the simulator program that converts the board-description language, which is simple enough to be typed in directly from a schematic, to a form usable by the computer. The IC library supplied with TESTAID contains more than 1500 popular SSI and MSI circuits and documentation describing the modeling techniques is supplied so the user can model unique ICs that are not in the library.

The simulation process predicts the outputs for both good and bad boards for the set of input patterns that the test programmer selects, and stores the predicted responses for each node on the board in its data base.

The FASTRACE portion of the software takes the data generated by TESTAID together with the test station configuration data and generates a test-data file. FASTRACE is then used to apply the set of test patterns to a board and measure the responses, comparing them to those predicted by TESTAID. If the board does not check out correctly. FASTRACE uses the information in the test-data file to direct the operator to place the system's probe on particular circuit board nodes and thus backtrace to the point of failure. Because of the prompts supplied by FASTRACE, little training is required to operate the system.

The DTS-70 system has been used successfully to test boards containing more than 220 medium-scale integrated circuits.

Board Testing with the 3060A

The HP Model 3060A Board Test System is intended for use on analog and hybrid analog/digital circuit boards. It is a third-generation system that evolved from eight years' experience with two previous automatic test systems developed for in-house use.

Most production-line failures in circuit boards are caused by solder splashes, misloaded components, and faulty components. Hence, the primary interface between the 3060A System and a board being tested is the bed-of-nails fixture. This gives the observability needed for checking between every pair of circuit nodes.

The approach taken for circuit-board testing with the Model 3060A is the following:

1. As a first priority, the impedance between each node pair is tested to find solder splashes and defective circuit board traces. Any detected faults are corrected before proceeding. This prevents possible damage to compo-

nents when power is first applied to the board.
2. Next, impedance measurements are made on all passive components and some active components. The results are compared to expected values.
3. Stimulus/response measurements are then made on active components, such as op amps, that the in-circuit method does not test.
4. Digital stimulus/response measurements are made on digital-to-analog interfaces and some small-scale integrated circuits.
5. Finally, signature analysis, an optional feature, is used to verify the performance of medium- and large-scale integrated circuits, such as RAMs, ROMs, and processors.

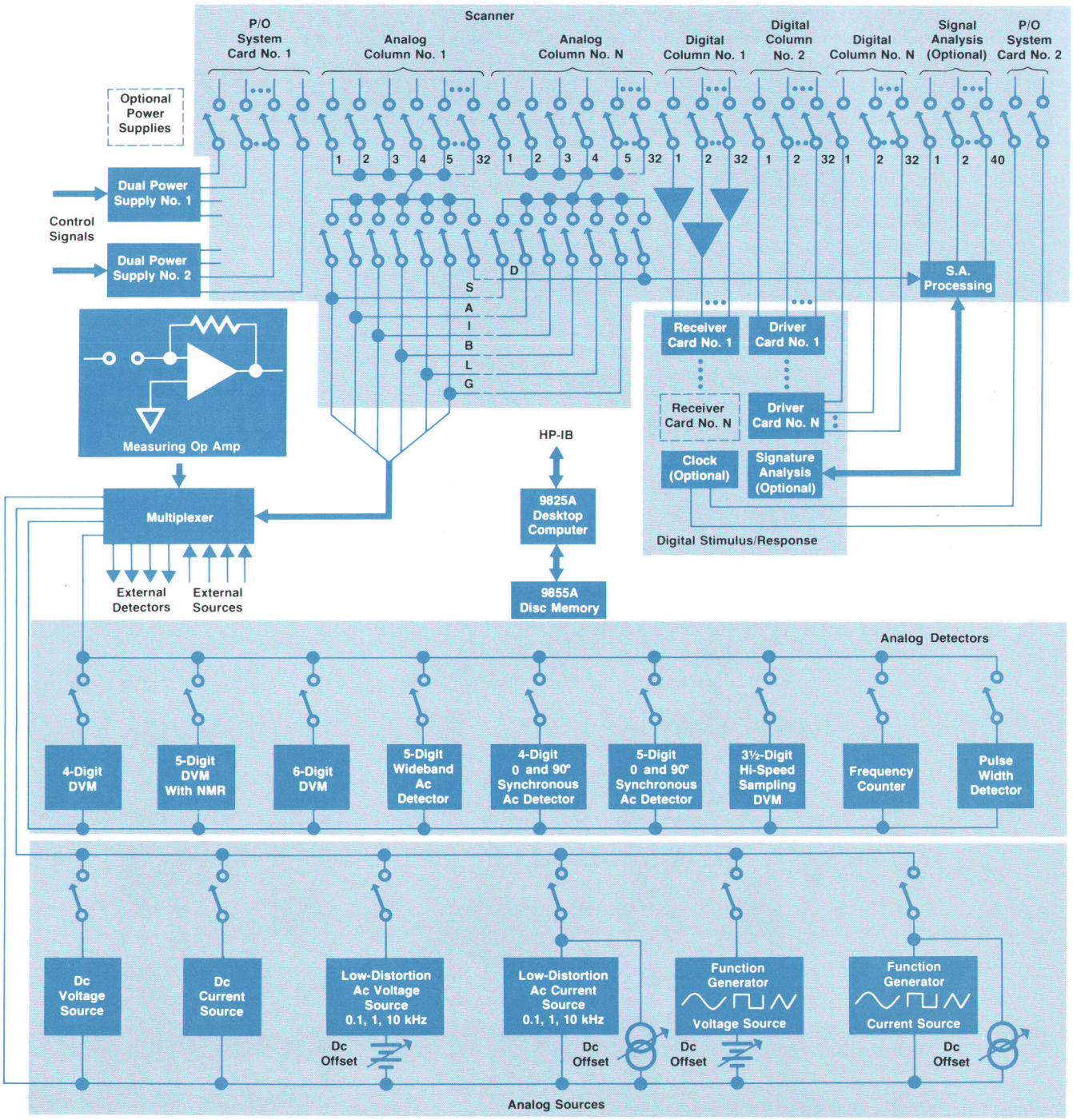


Fig. 3. *The capabilities of the Model 3060A Board Test System are symbolized in this block diagram. Connections between the scanner pins shown at top and the pertinent nodes on a circuit board under test are made through a patch panel and flexible leads to the appropriate contact pins in the bed-of-nails fixture.*

To perform all these measurements, the 3060A system has a collection of software-controlled electronic tools, symbolized in the functional block diagram of Fig. 3. Although there are ten microprocessors distributed among these tools, they all operate under control of an HP Model 9825 Desktop Computer. The use of software and firmware enhancements made it possible to develop a set of measurement functions that is much larger than that provided by the collection of electronic tools.

Versatile Hardware

The contacts on the bed-of-nails fixture connect to the measuring instruments through the system scanner. A flexible lead from each contact is inserted in the appropriate pin in a standard patch panel. For analog measurements, each pin connects through a system of relays to one or more of six system buses. These buses are labelled S, I, G, A, B, and L.

For stimulus-response measurements, the stimulus is applied to the S and G buses and the response is measured on the I and L buses. During impedance measurements, the component is connected between the S and I buses and the G bus is used to guard out the effects of other components. The A, B, and L buses provide remote sensing for the other buses, such as when making four-terminal impedance measurements.

For digital pattern-testing, the patch-panel pins are not switched to the buses but each one connects directly through a relay to a digital driver or a digital receiver.

Analog stimulus signals are generated within a unit known as the analog stimulus/response unit (ASRU). The response to a stimulus may also be measured by the ASRU or by the HP Models 3455A and 3437A Digital Voltmeters that are included with the system, the 3455A[3] to give 6½-digit resolution on dc measurements and 5½-digit resolution on true rms ac measurements, and the 3437A[4] to make sampled voltage measurements (>4000 reading/s) for determining quantities such as slew rate, transient response, and total harmonic distortion. Where this array of equipment may not be sufficient, up to four external sources and four detectors may be connected and controlled through the HP interface bus.

Digital stimuli originate in the digital stimulus/response unit (DSRU), which also contains the digital receivers.

Up to eight floating power supplies can be included in the system for powering the circuit card under test. Optional features include a programmable clock on a card that plugs into the scanner. This clock has a burst mode that generates 1 to 10^6 clock cycles per burst. Also available is a signature analysis card that enables collection of signatures at specified nodes on the board under test in response to dynamic digital patterns applied to the board or stored in ROM on the board, (see page 29). Microprocessor-based and other sequential-logic circuit boards, which are not testable by application of static digital patterns, can be tested with the signature analysis option.

All of these tools are combined through various firmware enhancements to generate special tests for shorts and opens and to test diodes, transistors, and junction FETs. Additional firmware enhancements remove the effects of relay-contact thermal potentials, offset voltage and bias currents in the measuring circuit, gain and offset errors in the detectors, and magnitude and phase errors that are introduced by source loading and finite gain in the measuring operational amplifier. Remote sensing with enhancements removes the residuals of the system, thereby increasing the range of components and circuit configurations that can be measured.

Fast Test Program Development

The Model 9825A Desktop Computer that controls operation of the system is a standard unit that enables many additional peripherals, a CRT terminal, and a slave flexible disc drive, to be integrated into the system. For use with the 3060A Board Test System, the 9825A's HPL language has been augmented by 39 high-level board-test statements that automate complete measurement cycles.

ABBREVIATED SPECIFICATIONS
HP Model DTS-70 Board Test System

TESTAID Simulator

PRIMITIVE MODELING ELEMENTS

AND	Amplifier	ROM
OR	Inverter	RAM
NAND	Connector	Shift Register
NOR	D-Latch	Delay
XOR	Decoder	Inverting Delay

PATTERN GENERATION: Manual and automatic. Automatic includes path-sensitizing generator and pseudorandom generator.
FAULTS SIMULATED: Stuck at one and stuck at zero for node faults, pin faults and input/output faults.
SIZE LIMITATIONS (general guidelines only):
PRIMITIVE ELEMENTS (in an expanded circuit description): 11,000.
FAULT SIGNATURE FILE: 256K bytes.
SIGNAL OUTPUTS FROM EACH PRIMITIVE ELEMENT: 255.
MAXIMUM SIZE OF PRIMITIVE MEMORY ELEMENTS (RAM or shift register): 64K bits.

Digital Test Unit (DTU)

CAPACITY: 12 driver/comparator and measurement switch cards in any combination. Each card has 15 UUT pins. Optional extender adds 12 more cards.
DRIVER COMPARATOR CARDS: 3 types: TTL, CMOS, and programmable (±16V).
MEASUREMENT SWITCH CARD: Enables external stimulus/response to be connected to 15 DTU pins per card.
TEST RATES: Typically 2000 patterns per second when toggling 75 DTU pins between drive and compare. 12,000 per second when toggling pins on one card and delays are not present.
UUT PCB SIZE: Up to 38.7 cm wide by 33 cm long (15¼×13 in). Longer cards may be inserted without using insertion levers.
UUT POWER SUPPLIES:
DUAL SUPPLY: 0 to +25V and 0 to −25V, 2A max.
HIGH CURRENT SUPPLY: 0 to +20V, 10A max.
(System can accommodate up to five supplies.)
FASTRACE PROBE:
THRESHOLD LEVEL: −12.8 to +12.775V.
PULSE WIDTH: 30 ns minimum.
INPUT IMPEDANCE: 100k ohms.
REPAIR TICKET PRINTER: 5×7 dot matrix characters on 5.7 cm (2¼ inches) wide thermally-sensitive paper.
TEMPERATURE: 0 to 50°C, 10 to 85% relative humidity (non-condensing).
POWER REQUIREMENTS: 120/230V ac ± 10%, single phase 56-63 Hz, 240VA max.

System 1000 Controller

Includes 21MX Series Computer with 256k bytes of memory, Model 7906A 20M byte disc memory, and 2645A Display Station. Software includes RTE-IV operating system and FORTRAN-IV and assembly language compilers. Supports three DTS-70 Test Units and seven programming terminals.
TEMPERATURE: 10 to 38°C. 20 to 80% relative humidity, non-condensing.
POWER REQUIREMENTS: 115V/60 Hz ±10%, single phase, 1665 W.
PRICE IN U.S.A.: Basic DTS-70 System, $89,250. 15-pin TTL card, $425; 15-pin CMOS card, $500; 15-pin programmable card, $1100.
MANUFACTURING DIVISION: LOVELAND INSTRUMENT DIVISION
815 Fourteenth Street, S.W.
Loveland, Colorado 80537 U.S.A.

ABBREVIATED SPECIFICATIONS
HP Model 3060A Board Test System

(Basic scanner configuration has 384 analog, 32 digital driver, and 32 digital receiver contact pins, expandable to 1024/128/256, and 12 general-purpose relays.)

In-Circuit Passive Components Test

RESISTANCE

SOURCE: 100 mV dc.
MAXIMUM SPEED: 30 scanned readings/second.

ACCURACY (% of reading):	±1%	±0.25%/±1%	±1%
Connection	4-term.	4-/2-terminal	2-term.
Accuracy Mode	Enhanc. Line Rej.	Enhanc. Line Rej./None	Enhanc. Line Rej.
	0.1Ω	250Ω	$10^6\Omega$ $10^7\Omega$

(Measurements of $>10^7\Omega$ are possible using higher voltage sources.)

RESISTANCE, ACCURACY ENHANCED

SOURCE: 1 V dc.
MAXIMUM SPEED: Approximately 3.5 scanned readings/second.
CONNECTION: 6-terminal.
ACCURACY MODE: Enhancement, line rejection.

ACCURACY (10kΩ): ≤±2% of reading, verified by confirmation software.
GUARDING RATIOS: Source (Rx/R1), 10^3
Input (Rx/R2), 10^3
Product (Rx/R1R2), 10^6.

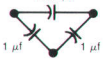
CAPACITANCE

SOURCE: 100 mV rms ac.
MAXIMUM SPEED: 21 scanned readings/second.
ACCURACY (% of reading):

	±2% + 10 pF	±2%	>2%
Connection	2-term.	2-term.	4-term.
Accuracy Mode	None	None	Enhanc. Line Rej.
	10 pF	500 pF 1 μF	10^5 μF

CAPACITANCE, ACCURACY ENHANCED

SOURCE: 1V rms ac.
SPEED: Approximately 3.5/second.
CONNECTION: 6-terminal.
ACCURACY MODE: Enhancement, line rejection.



ACCURACY (1000 pF): ±3%, verified by confirmation software.

INDUCTANCE, UNGUARDED

SOURCE: 100 mV rms ac.
MAXIMUM SPEED: 10 scanned readings/second.
ACCURACY FOR Q ≥1 (% of reading):

	±2%	±2%
Connection	4-terminal	2-terminal
Accuracy Mode	Enhancement Line Rejection	None
	25 μH 50 mH	100 H

ACCURACY, LOW Q (% of reading): 5% for 10 mH with Q = 1/2 using 4-terminal connection, accuracy enhancement, and line rejection.

SHORTS/OPENS

THRESHOLD RANGE: Programmable from 5Ω to 125 kΩ. Default value: 10Ω.
TEST CURRENT: 15 μA to 50 mA depending on threshold.
SPEED: 8.5 ms/node + 140 ms. Programmable wait to charge capacitors.

In-Circuit Semiconductor Component Tests

DIODES (standard and Zener)

Measured parameter is voltage across diode at specified current.
RANGE: 0-14V (Zeners with breakdowns ≤100V can be tested with transfer tests).
CONDITIONS SELECTED BY BTL (Programmer can specify different conditions):
REFERENCE CURRENT: 10 mA.
REFERENCE RESISTOR IN MEASURING OP AMP: 100Ω.
SOURCE AMPLITUDE: 1V dc.
SOURCE ACCURACY: ±(1% of reading + 0.03V).
SPEED: 35 readings/second.

TRANSISTORS (NPN or PNP)

Measured parameter is small signal beta with transistor biased on.
SMALL SIGNAL BETA (CURRENT GAIN):
RANGE: 0.65 to 100 or 6.5 to 1000, selected by BTS software.
TEST CURRENT: 0 to 140 mA peak. Ac and dc current can be applied simultaneously to the transistor and its biasing network. Programmable from 0 to 140 mA peak.

TRANSISTORS (Field Effect)

PINCHOFF VOLTAGE TEST FOR DEPLETION MODE FETS:
Pinchoff Voltage = Vgs at Itest.
Vgs at Itest = (1 ± 0.05) × (voltage returned).
Itest = (1 ± 0.05) × Vgs/10^7).
RANGE: 0 to 14.2V.
SOURCE VOLTAGE SELECTED BY BTL: 0.1V.
SPEED: 35 scanned readings/second.
Idss AND R-on TEST:
Idss RANGE: 1nA to 50mA (constant-current test).
R-on RANGE: 2Ω to 100 kΩ (resistance test).
Vds≈Source amplitude.
Vgs≈200 μV.

CURRENT (useful for measuring leakage)

RANGE: 1nA to 50 mA.
SOURCE: Programmable from 0 to ±14.2V dc; default value: 0.1V dc.
SPEED: Approximately 30 readings/second.

Analog Transfer Tests

(All analog capabilities can be used simultaneously with the digital functional capability for combined analog-digital circuits.)

Stimulus/Source

DC VOLTAGE

AMPLITUDE: −14.2 to +14.2V.

Range	Accuracy (open circuit)	Resolution
0 to 0.15V	±(0.08% of setting + 0.55 mV)	10 μV
.15 to 1.5V	±(0.08% of setting + 0.65 mV)	100 μV
1.5 to 15V	±(0.08% of setting + 2.0 mV)	1 mV

OUTPUT IMPEDANCE: 2Ω
CURRENT LIMIT: 0 to 142 mA.

AC VOLTAGES (PRECISION)

FREQUENCY: 100 Hz, 1 kHz, 10 kHz (within ±0.3%).
AMPLITUDE: 0 to 10V rms.

Range	Accuracy	Resolution
0 to 0.1V rms	±(0.22% of setting + 1.0 mV)	10 μV rms
0.1 to 1.5V rms	±(0.22% of setting + 1.2 mV)	100 μV rms
1.5 to 10V rms	±(0.22% of setting + 2.5 mV)	1 mV rms

OUTPUT IMPEDANCE: <3Ω.
TOTAL HARMONIC DISTORTION: ≤−60 dB.
DC OFFSET: −12.8 to +12.7, accurate within ± (0.8% of setting + 0.1V).
CURRENT LIMIT: 0 to 142 mA.

FUNCTION GENERATOR (VOLTAGE)

FUNCTIONS: Sine, square, triangle.
FREQUENCY: 10 Hz to 100 kHz, accurate within ±(5% of setting + 1 Hz).
AMPLITUDE: 0 to 14.2V peak.
ACCURACY (open circuit): ±(4% of setting + 0.005V).
RESOLUTION

Range: square, triangle V peak	sine V rms	Resolution
0 to 0.15V	0 to 0.1V	0.64 mV
0.15 to 1.5V	0.1 to 1.0V	6.4 mV
1.5 to 14.2V	1 to 10V	64 mV

OUTPUT IMPEDANCE: Approximately 50Ω.
SQUARE WAVE SYMMETRY: >98%.
TRIANGLE LINEARITY: <30 mV deviation from a straight line at 100 Hz, 10 kHz.
SINE WAVE DISTORTION: ≤−33 dB.
DC OFFSET: −12.8 to +12.7V, accurate within ±(0.8% of setting + 0.1V).
CURRENT LIMIT: 0 to 142 mA.

DC CURRENT

RANGE: 0 to 142 mA.
ACCURACY: ±(0.5% of setting +0.03 mA).

Resolution	Range	Output Impedance
100 μA	0 to 1.5 mA	Approximately 100 kΩ
1 mA	1.5 to 15 mA	Approximately 10 kΩ
10 mA	15 to 142 mA	Approximately 1 kΩ

VOLTAGE LIMIT: ±14.2V ±(current setting × 100).

AC CURRENT (PRECISION)

FREQUENCY: 100 Hz, 1 kHz, 10 kHz (within ±0.3%).
RANGE: 0 to 100 mA.
ACCURACY: ±(0.5% of setting + 0.05 mA).

Resolution	Range
100 μA	0 to 1 mA rms
1 mA	1 to 10 mA rms
10 mA	10 to 100 mA rms

OUTPUT IMPEDANCE:

	100 Hz	1 kHz	10 kHz
0 to 1 mA rms	100 kΩ	65 kΩ	6.5 kΩ
1 to 10 mA rms	10 kΩ	6.5 kΩ	650 Ω
10 to 100 mA rms	1 kΩ	650 Ω	65 Ω

VOLTAGE LIMIT: 0 to 14.2V peak −(current setting × 100).

FUNCTION GENERATOR (CURRENT)

FUNCTIONS: Sine, square, triangle.
FREQUENCY: 10 Hz to 10 kHz.
ACCURACY: ±(5% of setting + 1 Hz).
AMPLITUDE: 0 to 142 mA peak or 0 to 100 mA rms.
ACCURACY (short circuit): ±(4% of setting + 0.05%).

Resolution	Range	Output Impedance
100 μA	0 to 1.5 mA peak	6.5 kΩ
1 mA	1.5 to 15 mA peak	650 Ω
10 mA	15 to 142 mA peak	65 Ω

VOLTAGE LIMIT: 0 to 14.2V peak −(current setting × 100).

Response/Detectors

DC VOLTAGE

ACCURACY (% of reading + V):

	4½ digit	5½ digit
0.1V range:	±(0.2% + 0.0004V)	±(0.05% ± 0.00035V)
1 V range:	±(0.15% + 0.0005V)	±(0.022% ± 0.00035V)
10V range:	±(0.15% + 0.005V)	±(0.02% ± 0.001V)

OVERRANGE: 50% (automatic upranging).
INPUT IMPEDANCE: 10^{10} Ω shunted by <950 pF.
COMMON-MODE REJECTION: 53 dB from dc to 60 Hz with 1 kΩ imbalance in low lead.
NORMAL MODE REJECTION (5½ digit): >40 dB at 50 or 60 Hz ± 1%.

3455A DIGITAL VOLTMETER

RESOLUTION: 6½ digits.
RANGE: 0.1 to 100V in decade steps, autoranging.

AC VOLTAGE

SYNCHRONOUS DETECTORS: Measure real part (X) and imaginary part (Y) of detected signals with respect to phase of precision sources (100 Hz, 1 kHz, 10 kHz).
ACCURACY: (A% of reading + B% of Y* or X + C(mV) + 0.2% of dc component).**

Range	0.1V			1V			10V		
	A	B	C	A	B	C	A	B	C
100 Hz	.2	.2	.2	.5	.2	1.5	.5	.2	15
1 kHz	.2	.34	.2	.5	.34	1.5	.5	.34	15
10 kHz	.2	3.4	.2	.5	3.4	2.25	.5	3.4	22.5

*When X is measured, use Y; when Y is measured, use X.
**Dc offset at phase detector input, if any.

3455 DIGITAL VOLTMETER

RESPONSE: True rms.
FREQUENCY RANGE: 30 Hz to 1 MHz.
RANGE: 1, 10, 100V, autoranging.

3437A DIGITAL VOLTMETER: Digitizes repetitive signals and low-frequency transients for analysis by computer (9825A).
MAXIMUM READING RATE: >4000 per second.
INPUT BANDWIDTH: Approximately 40 kHz on the 0.1 V range, 1 MHz on the 1 and 10V ranges.

FREQUENCY COUNTER

RANGE: 1 Hz to 5 MHz dc coupled; 30 Hz to 5 MHz ac coupled.
BANDWIDTH: 2 MHz at 3-dB point; 20 dB/decade roll off.
ACCURACY: ±[0.02% of reading + (10^{-7}/gate time or 10^{-7} × measured frequency, whichever is smaller)].
GATE TIMES: 1 ms to 1 s in decade steps. Gate time automatically extends to include integral number of input cycles.
TRIGGER LEVEL: −12.8 to +12.7V.
SENSITIVITY: 0.3V to 500 kHz.

PULSE WIDTH (measured from rising to falling or falling to rising edges):

RANGE: 300 ns to 1 s.
ACCURACY: 0.02% of reading + 200 ns.
INPUT CHARACTERISTICS: 3-dB point >2 MHz.
EQUIVALENT INPUT CIRCUIT: Approximately 100 kΩ shunted by 950 pF.
THRESHOLD (rising or falling edge): −12.8 to +12.8V.
SENSITIVITY: 0.3V to 500 kHz.

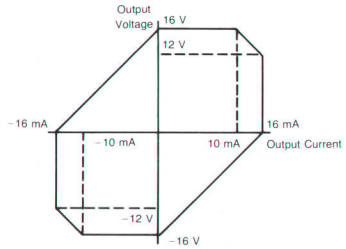
Digital Transfer Tests

DIGITAL RECEIVERS

THRESHOLD: −15 to +15V.
STATIC ACCURACY: 0.15V.
RESOLUTION: 8 mV.
INPUT IMPEDANCE: 400 kΩ 10% shunted by 15 pF.
CROSSTALK: 10 pF between lines (minimum crosstalk is 30 dB).

DIGITAL DRIVERS (Drive and receive levels can be separately configured):

LOGIC LOW RANGE: −16 to +16V.
LOGIC HIGH RANGE: −16 to +16V.
STATIC ACCURACY: 0.125V.
RESOLUTION: 8mV.
SLEW RATE: 40V/μs into 50 pF load.
MAXIMUM SKEW BETWEEN BITS OUTPUTTED: 150 ns.
CROSSTALK: 10 pF between lines (minimum crosstalk is 30 dB).
MAXIMUM CURRENT:



DIGITAL CLOCK (Opt 007): 1, 10, 100, 1000 kHz, ±16V. Burst mode: up to 999.999 counts.
SIGNATURE ANALYSIS (Opt 008) for at-speed μP testing. Threshold range: ±15V.

General

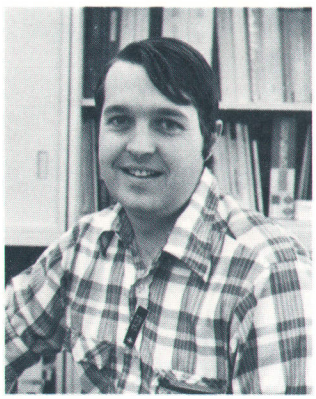
POWER SUPPLIES (for unit under test): Two programmable dual 20V/3A dc supplies are standard; up to four are optional. Also, 100/120/220/240V ac.
EXTERNAL INPUTS AND OUTPUTS: 4 source inputs and 4 detector outputs.
OPERATING TEMPERATURE: 0 to 40° C; 5 to 80% relative humidity at 40°C.
POWER REQUIREMENTS: 200-240V, 3-phase, 50-60 Hz, 12-18A.
PRICE IN U.S.A.: Standard configuration, $74,000; Option 007, $300; Option 008, $2500.
MANUFACTURING DIVISION: LOVELAND INSTRUMENT DIVISION
815 Fourteenth Street, S.W.
Loveland, Colorado 80537, U.S.A.

Software supplied with the Model 3060A includes an in-circuit program generator that automatically generates a complete in-circuit test program. It sets up the optimum test condition for each in-circuit measurement, analyzes the accuracy of the measurement, and then generates a wiring layout for the test fixture. The data input for this program, which can be entered by people who have no technical training, is a straightforward listing of the components to be measured and the nodes to which they connect.

The basic in-circuit measuring scheme and other details of the software and electronic tools are described in other articles in this issue.

References

1. R.K. Juncker, "Higher-Performance HP 1000 Computer Systems," Hewlett-Packard Journal, October 1978.
2. E.J. Wong and C.M. Manley, "RTE-IV: The Megaword-Array Operating System," Hewlett-Packard Journal, October 1978.
3. A.G. Gookin, "A Fast-Reading, High-Resolution Voltmeter that Calibrates Itself Automatically," Hewlett-Packard Journal, February 1977.
4. J.E. McDermid, J.B. Vyduna, and J.M. Gorin, "A High-Speed System Voltmeter for Time-Related Measurements," Hewlett-Packard Journal, February 1977.



Peter S. Stone

Pete Stone did it all in one stretch: BSEE, MSEE, and PhD degrees at the University of Minnesota. He then worked for a medical equipment firm for a year before joining Hewlett-Packard in 1972 to work on automatic systems. He is currently project manager in the circuit test lab. Married, with one small child and another on the way, he still finds time for flying (he's 1/3-owner of a Cherokee 180) and ham radio.

John E. McDermid

John McDermid earned his BSEE degree at the University of Idaho and his MSEE degree at the University of Alberta at Calgary, Canada. He joined HP in 1969, investigating high-speed A-to-D converters and then designing the dc portions of the Model 3490A Digital Voltmeter. He was project manager for the 3437A Voltmeter and is now project manager in circuit-board test. John enjoys fishing and hunting (with a bow and arrow) with his wife, son, 10, and daughter, 6.

Rapid Digital Fault Isolation with FASTRACE

by William A. Groves

FASTRACE IS A COMBINED hardware and software package provided with the DTS-70 Digital Printed Circuit Board Test System. FASTRACE can test and troubleshoot most digital circuits, including those with intermittencies and inaccessible nodes. The only requirement is that the test file (stimulus/response data) must be developed using TESTAID (see article, page 13).

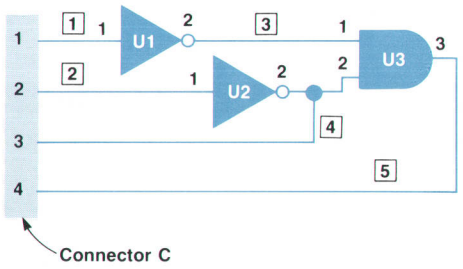
As explained in the article on page 2, the DTS-70 consists of an HP 1000 Computer System and a 9571A Test Station. The test station is responsible for all hardware interfacing to the printed circuit board (PCB), including digital inputs/outputs, PCB power supplies, the operator-guided probe, and optional HP-IB-compatible instruments.

The heart of the test station is the digital test unit (DTU). The DTU contains up to 360 bidirectional digital input/output pins. These pins are connected to a test adapter through four 90-pin connectors. Contact with the PCB is provided by a customer-supplied connector that is bolted to the opposite side of the test adapter and wired to the 90-pin connectors. Additional space is available on the test adapter for mounting components (level shifters, Schmitt triggers, etc.) needed for proper electrical interfacing.

The edge connector signals from the PCB are sufficient for determining whether a PCB is good or bad. However, if a PCB is defective, access to the signals internal to the PCB (signals not on the edge connector) must be provided for fault isolation. The operator-guided probe provides this connection. The probe is a digital comparator similar to a DTU input pin, except that the probe can also detect an open-circuit condition. FASTRACE will tell the operator, "Probe not making contact," if an open circuit is detected.

Creating the Test File

Before using FASTRACE, a PCB test file must be created by TESTAID. To use TESTAID, the test programmer must supply two sets of information concerning the PCB. The first set of information is a description of the types and interconnections of all integrated circuits (ICs) on the PCB. In Fig. 1 the IC labeled U3 must be specified as having part number 7409. Also, part of the interconnection description indi-



Type	Interconnection Device Pin	Connects To
U1=7404	C.1	U1.1
U2=7404	C.2	U2.1
U3=7409	U1.2	U3.1
	U2.2	U3.2, C.3
	U3.3	C.4

Pattern	Input Pattern [1] [2]	Internal Node States [3]	Output Pattern [4] [5]
No. 1	0 1	1	0 0
No. 2	1 0	0	1 0
No. 3	0 0	1	1 1

Topology

Node No.	Is Affected by (Predecessors)
[1]	Nothing
[2]	Nothing
[3]	[1]
[4]	[2]
[5]	[3], [4]

Fig. 1. *FASTRACE relies on a test file created using TESTAID. The programmer supplies TESTAID with the types and interconnections of every IC on a printed circuit board. TESTAID generates topology tables listing the predecessors of every node. The programmer also supplies input patterns and TESTAID calculates the board's responses.*

cates that U1 pin 2 connects to U3 pin 1. This interconnection data is referred to as the PCB's topology. TESTAID then generates topology tables (see Fig. 1) listing the predecessors, or preceding nodes, for every node on the PCB. These tables are used for backtracing from a failing node.

The other set of information needed by TESTAID is a set of input patterns. An input pattern is simply a collection of ones and zeros to be applied in parallel to the PCB's inputs. Although TESTAID has an automatic method for developing input patterns, this mechanism will not be considered here.

After the topology is described and the input patterns are defined, TESTAID calculates all of the PCB responses. Since TESTAID is a three-valued simulator (0,1, and X or unknown), nodes fed by uninitialized memory elements are calculated as "X". This assures that the calculated responses are correct even though memory elements may power-up in either state. In Fig. 1 the states for nodes 3, 4, and 5 will be calculated automatically given the input patterns for nodes 1 and 2.

In addition to calculating the PCB responses, TESTAID predicts the test effectiveness of the input pattern set. To determine test effectiveness, TESTAID constructs a fault model for predicting the behavior of defective PCBs. One fault model used by TESTAID is the single-fault stuck node model.* This means that TESTAID assumes that a PCB can fail by having any single node solidly stuck at 1 or stuck at 0 (SA1/SA0). In the circuit of Fig. 1, there are five nodes. Therefore, according to the single-stuck-node fault model, this circuit can fail ten different ways—five nodes either SA1 or SA0.

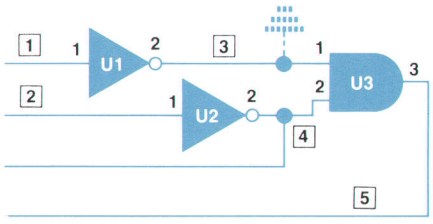
After TESTAID has determined all the ways a PCB can fail, response data is calculated for the good PCB and for each faulty PCB. Thus in Fig. 1, TESTAID predicts the responses for 11 PCBs (1 good, 10 faulty). When the output pattern of a faulty PCB is predicted to be different from the output pattern of the good PCB, the fault is said to be "detected." This means that when the input pattern set is applied to a PCB with this fault, the fault will be propagated to the output edge connector and will be detected as a failure by the DTU. Test effectiveness is determined by the ratio of detected faults to all possible faults. If an input pattern set supplied to TESTAID detects five of the faults in Fig. 1, then the test effectiveness is 50%.

Besides providing the user with a measure of test effectiveness, TESTAID also supplies FASTRACE with failure information, or fault signatures, for every fault that was detected. Each fault signature is a prediction of the failing pattern number (the number of the input pattern that produces the failure) and the failing output pins for every fault.

Production Testing

Before placing the PCB test into production, all of the good PCB response data generated by TESTAID must be verified by FASTRACE. This is done by connecting a known good board to the tester and requesting FASTRACE to "verify." FASTRACE then applies the set of input patterns and verifies that the PCB responses match TESTAID's predictions. FASTRACE also prompts the operator to place the guided probe sequentially on each internal node to verify these signals. This step occurs only once, assuming that the PCB passes. All TESTAID predictions for the good PCB are thus verified and the PCB is ready for production testing.

The process of determining whether a PCB is good or bad is referred to as a GO/NOGO test. To perform this test, only the input/output pattern data is necessary. Internal node



Defective PCB (Node [3] SA0)

[1] [2]	[3]	[4] [5]
0 1	0*	0 0
1 0	0	1 0
0 0	0*	1 0*

*Indicates Incorrect Signal Value

Fig. 2. *In this combinatorial circuit, node three fails pattern one, and the tester observes the failure on pattern three, when an output pin fails. The operator can backtrace to the fault site by measuring only the expected node states for pattern three.*

*Actually there are several fault models, but for this article, only the stuck node model is considered.

data is not needed. For quick access, all test data is kept in a virtual memory system (see box, page 17). To execute the GO/NOGO test, pattern data is retrieved from the virtual memory system and applied to the DTU. A typical GO/NOGO test executes in less than 50 ms. Several complete GO/NOGO tests are normally executed to ensure that the PCB is not intermittent. If a PCB fails the GO/NOGO test, FASTRACE determines the reason for the failure.

When the GO/NOGO test fails, FASTRACE notes the failing pattern number and the list of failing output pins (one or more). This information is known as the PCB syndrome. Note the similarity between fault signatures and syndromes. Fault signatures are predictions made by TESTAID concerning the effect a fault will have on the PCB's output pattern. A syndrome is the actual measurement of the failing PCB's output pattern.

As described earlier, TESTAID generates a list of fault signatures corresponding to each of the single SA1 or SA0 node faults. One obvious solution to fault-isolating the PCB would be simply to list these failure predictions. However, there are two major problems. First, TESTAID assumes that a defective PCB contains only one fault (single-fault model). If more than one fault exists, the faults may interact with one another and make TESTAID's predictions invalid.

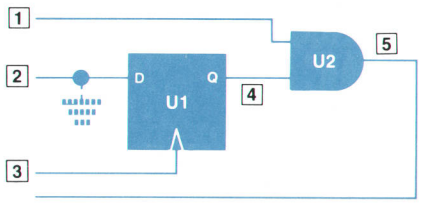
Second, to predict the behavior of a defective PCB, it is necessary to construct a fault model. While the SA1/SA0 node fault model (and others) that TESTAID uses is an excellent model for determining test effectiveness, there is no guarantee that the fault model precisely matches the actual PCB fault. If the fault model is similar to but not the same as the actual PCB fault, then the model is good for determining whether the PCB will pass or fail; however, the failure predictions will not be precise. Despite these limitations, fault signatures frequently point to the approximate locations of PCB faults.

The other method for fault-isolating a failing PCB is to direct the operator-guided probe to begin on a failing PCB output pin and to trace the failing signal path back to the PCB fault. This operation is known as backtrace.

FASTRACE combines both methods—fault signatures and backtrace. If a fault signature matches the PCB failure syndrome, then FASTRACE begins backtrace at the faulty node predicted by TESTAID. If this node fails, then backtrace continues until a diagnosis is reached. Since the fault signatures often point to the approximate fault site, typically the backtrace operation will be faster than if begun on a failing output pin. If the node predicted by TESTAID passes, then backtrace begins on a failing output pin, since in this case TESTAID's prediction is totally incorrect. In either case, the PCB failure is analyzed through the guided probe operation, so the user is assured of an accurate diagnosis based on physical measurement, not computational prediction.

Combinatorial and Sequential Circuits

The simplest type of circuit to fault-isolate is a combinatorial circuit. A combinatorial circuit is one whose output pattern depends only on the current input pattern and is independent of history. Generally, this means circuits with no feedback or memory. In Fig. 2, for example, node three fails pattern one; however, the tester will not observe the failure until pattern three when node five (output pin) fails. If the tester stops on pattern three, the guided probe can backtrace to the fault site by measuring only the expected node states for pattern three.



U1=Positive-Edge-Triggered D-Type FF (7474)

Good PCB

1	2	3	4	5
0	1	0	X**	0
0	1	1	1	0
1	0	0	1	1

**X=Unknown

Defective PCB (Node 2 SA0)

1	2	3	4	5
0	0*	0	X	0
0	0*	1	0*	0
1	0	0	0*	0*

*Indicates Incorrect Signal Value

Topology

Node No.	Is Affected by (Predecessors)
1	Nothing
2	Nothing
3	Nothing
4	2,3
5	1,4

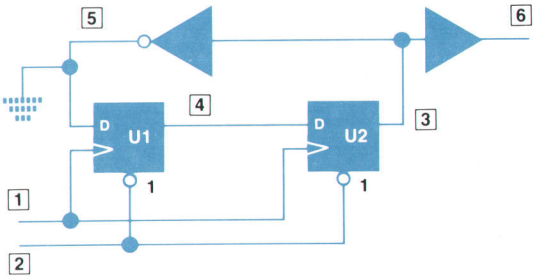
Fig. 3. *In this sequential circuit, node two fails pattern one and the tester observes the failure on pattern three, but the value of node two is correct at that time. To find the fault site by backtracing in a sequential circuit, the tester must reapply the set of input patterns each time a new node is probed.*

Fault isolation would be a simple task if all circuits were combinatorial. However, nearly all circuits are sequential. A sequential circuit is one whose output pattern depends on the current input pattern and also on previous input patterns (history). In other words, sequential circuits contain memory.

Fig. 3 shows a sequential circuit with a SA0 fault on node two. Again, node two fails the first pattern, but the tester does not detect a failure until node five (output pin) fails pattern three. Notice that on pattern three the value of node two (the faulty node) is correct. If the tester stopped on pattern three and if backtrace checked only the node states expected on pattern three then nodes four and five would fail while nodes one through three would pass. The obvious but incorrect conclusion would be that node four is the faulty node.

In this situation, the fault has caused the memory element U1 to be in the wrong state on patterns two and three. Since in a sequential circuit a node state is dependent on the input pattern and on the node's history, it is necessary to know the node's history as well as its present state to determine whether it is functioning properly.

To verify a node's history, we need to reapply the set of input patterns to the PCB each time the probe measures a new node. To check node four in Fig. 3, input patterns one through three are reapplied. The probe checks the node state value as each pattern is applied. At node four the probe detects a failure on pattern two. When the input pattern set is reapplied with the probe on node two, the probe detects a



U1 and U2 Are Positive-Edge-Triggered D-Type FFs (7474)

Defective PCB (Node 5 SA0)

Pattern	1	2	3	4	5	6
1	0	0	0	0	0*	0
2	0	1	0	0	0*	0
3	1	1	0	0*	0*	0
4	0	1	0	0*	0*	0
5	1	1	0*	0*	0	0*
6	0	1	0*	0*	0	0*
7	1	1	0*	0	0	0*
8	0	1	0*	0	0	0*
9	1	1	0	0	0*	0

*Indicates Incorrect Signal Value

Fig. 4. *A synchronous circuit with a feedback loop. In such a circuit, a node is classified as failing only if its state is incorrect on a pattern number equal to or lower than the "lowest failing" pattern number—the lowest-numbered pattern on which any node fails.*

failure on pattern one. Since node two is not affected by any other node, node two is diagnosed as bad. The significant point to remember is that for sequential circuits the input pattern set must be reapplied to the circuit for each probe position to determine whether or not nodes are correct.

Feedback Loops

Another complication in the fault-isolation process is that many circuits contain numerous topological feedback loops. A topological feedback loop is defined as a group of two or more nodes in which each node can affect every other node. Figs. 4 and 5 demonstrate two types of feedback loops, synchronous and asynchronous.

A synchronous circuit contains a master clock signal that is used to propagate signals from one node to another. An asynchronous circuit has no master timing information. In an asynchronous feedback loop, the signals may propagate around an entire loop one or more times before stabilizing.

In Fig. 4, when node five is stuck at 0, the tester observes the failure on node six, pattern five. Backtrace eventually encounters the failing loop of nodes three, four, and five. Even though all three nodes are failing and affect each other, it is obvious that node five is the defective node. Node five is failing pattern one and no other node is failing pattern one; therefore, node five must be defective. In this situation, the fault is propagated completely around a feedback loop. However, the feedback loop can be broken by observing the "lowest failing" pattern number.

In this example, since node six failed pattern five, it is only necessary to reapply patterns one through five when node three is probed. Any failures occurring after the first failure (pattern five) are superfluous. Eventually, node five is probed and fails pattern one. Since its predecessor (node three) passes on pattern one, node five is isolated as the defective node.

Asynchronous Feedback

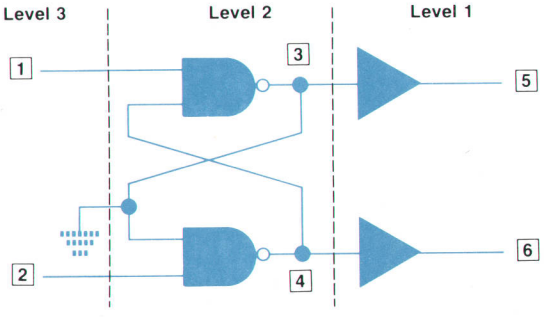
The asynchronous topological feedback loop complicates the backtrace algorithm considerably. Notice in Fig. 5 that four nodes are failing pattern one. By inspection, we know that nodes 5 and 6 could not be defective (assuming a single fault) since nodes 3 and 4 cannot be affected by nodes 5 and 6. Since nodes 3 and 4 are failing the same pattern number and affect each other (feedback loop), the best diagnosis we can obtain is that either node 3 or node 4 is defective. This example was trivial enough to solve by inspection. However, for large complex asynchronous loops, and to automate the procedure, the concept of leveling must be introduced.

In Fig. 5, for example, the first level of the circuit consists of the output nodes 5 and 6. The next level contains nodes that can affect lower levels (level one) and each other but cannot be affected by nodes in lower levels. This leveling continues until the circuit's input nodes are reached.

For fault isolation we are concerned about leveling only when several nodes are failing the same pattern number (lowest failing). During backtrace, leveling begins with the first probe. Additional nodes are entered as long as they fail the current lowest failing pattern number. If a node is encountered that fails a pattern lower than the lowest failing pattern, then the existing level structure is cleared and the new node is entered as the "seed" of a new structure.

Backtrace continues until all predecessors to the topmost level of nodes pass. When this occurs, the nodes in the topmost level are defective. If there is only one node in the top level, then the diagnosis is a single defective node. If more than one node exists in the top level then an asynchronous feedback loop of nodes has been encountered and cannot be broken by the lowest failing pattern number.

Fig. 6 illustrates the concept of leveling. In this example, seven nodes are failing the same pattern number. This collection of nodes could be surrounded by an arbitrarily complex series of nodes; however, all other nodes either pass



Defective PCB (Node 3 SA0)

1	2	3	4	5	6
0	1	0*	1*	0*	1*
1	1	0*	1*	0*	1*
1	0	0	1	0	1

*Indicates Incorrect Signal Value

Fig. 5. *An asynchronous circuit with a feedback loop. Here the concept of leveling is employed to backtrace to a failing node or group of nodes.*

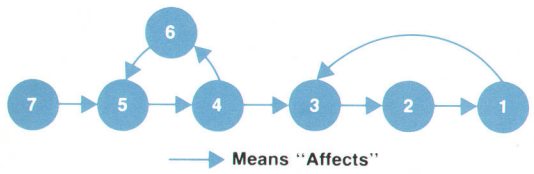


Fig. 6. *An illustration of leveling. Here seven nodes are failing the same pattern number. If all predecessors of node seven pass, then node seven is isolated as a single defective node. If node seven passes, the diagnosis would be three failing nodes (4, 5, 6) in an unbreakable feedback loop.*

completely or fail a higher pattern number. Initially, node one is found to be failing the lowest failing pattern number; therefore, any existing level structure is cleared and node one is entered into the first level. Node two is then entered as level 2. When node three is entered, it is discovered that a node in a lower level (node 1) affects node three; therefore, all levels down to the level containing node 1 are joined together. Thus, as each new failing node (failing the same pattern) is encountered, a higher level is created with the new node, but before the new node is entered, the structure is searched to ensure that no predecessors exist in lower levels. If one is found, then the new node and all levels above and including the predecessor's level are joined together.

This process continues until all predecessors of the top level pass. In Fig. 6, node 7 is entered as the highest level, since none of the nodes in lower levels can affect it. When all predecessors to node 7 are found to pass, node 7 is isolated as a single defective node. If node 7 had passed, then the diagnosis would be three failing nodes (4, 5, 6) in an unbreakable feedback loop.

Using the leveling concept and appropriate algorithms, fault isolation for any type of circuit becomes possible.

Intermittent Symptoms

In the examples so far, the following assumptions have been made:

1. The PCB failure is solid, not intermittent.
2. The operator probes the correct point when requested.
3. All nodes possess continuity.
4. All nodes are physically accessible to the operator for probing.

As mentioned earlier, every defective PCB has an associated failure syndrome. The syndrome is the lowest failing pattern number and the list of failing output pins. An intermittent PCB is one whose syndrome is changing as multiple GO/NOGO tests are run. There are two reasons for PCB intermittency. The more commonly understood reason is that the physical fault is marginal (bad level, hairline shorts) and is changing its characteristics as its environment changes (temperature, vibration).

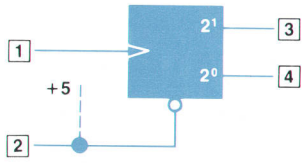
However, many PCBs exhibit intermittent responses when solid SA1/SA0 faults are present. Formally, this occurs when the predicted output of a faulty PCB is indeterminate (X) while the predicted output of a good PCB is known (0 or 1). This commonly occurs when a fault blocks the initialization sequence for one or more memory elements.

Fig. 7 shows a two-bit binary counter circuit. The two tables indicate the predicted node state values for the good PCB and also for a PCB with node 2 SA1. The predicted states for nodes 3 and 4 in the faulty table are X (unknown). Since the initialization signal (node 2) was SA1, the counter states cannot be predicted. The counter states will be strictly a function of the power-up characteristics of the counter. As the input pattern set is applied four times, four completely different syndromes will occur. The PCB responses are cyclic and repeat themselves every four pattern set applications.

Another example of this intermittency is shown in Fig. 8. When node 2 is SA1, a 1's hazard appears on node 5. Thus the flip-flop may or may not respond to the glitch. This circuit may appear intermittent (passing and failing) as the PCB's temperature changes, even though the actual fault (node 2) is solidly SA1.

During the backtrace operation, the pattern set is reapplied as each new node is checked. Obviously, if the PCB is changing its response, the backtrace operation can become totally confused. To detect this condition, FASTRACE initially runs several GO/NOGO tests. If the PCB is intermittent, the most commonly occuring syndrome is selected (the selection process is actually more complex than indicated here).

During backtrace, as the pattern set is reapplied to the PCB, nodal responses are ignored if the syndrome does not



Good PCB

1	2	3	4
0	0	0	0
0	1	0	0
1	1	0	1
0	1	0	1
1	1	1	0
0	1	1	0
1	1	1	1
1	0	0	0

Predicted States For Defective PCB (Node 2 SA1)

1	2	3	4	
0	1*	X	X	
0	1	X	X**	
1	1	X	X	
0	1	X	X	
1	1	X	X	
0	1	X	X	*Will Fail
1	1	X	X	**May Fail
1	1*	X	X	(Don't Know)

Defective PCB Measured Response
Assume Nodes 3 4 Power Up to 00

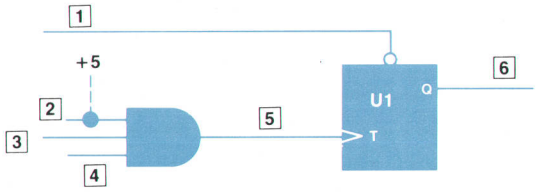
Pattern	First Pattern Set Application				Second Pattern Set Application				Third Pattern Set Application				Fourth Pattern Set Application			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	0	1*	0	0	0	1*	1*	1*	0	1*	1*	0	0	1*	0	1*
2	0	1	0	0	0	1	1*	1*	0	1	1*	0	0	1	0	1*
3	1	1	0	1	1	1	0	0*	1	1	1*	1	1	1	1*	0*
4	0	1	0	1	0	1	0	0*	0	1	1*	1	0	1	1*	0*
5	1	1	1	0	1	1	0*	1*	1	1	0*	0	1	1	1	1*
6	0	1	1	0	0	1	0*	1*	0	1	0*	0	0	1	1	1*
7	1	1	1	1	1	1	1	0*	1	1	0*	1	1	1	0*	0*
8	1	1*	1*	1*	1	1*	1*	0	1	1*	0	1*	1	1*	0	0

Syndrome:

Failing Pattern:	8	1	1	1
Failing Output:	3, 4	3, 4	3	4

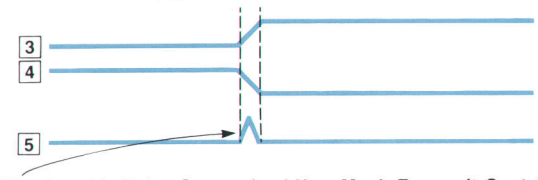
Note: Failing Pattern Implies "Lowest Failing" Output Pattern

Fig. 7. *Intermittent responses may occur when solid faults are present, as when a fault blocks the initialization sequence for one or more memory elements, such as this binary counter. FASTRACE runs several tests and selects the most commonly occurring fault syndrome for use during backtrace.*



U1 Is a Positive-Edge-Triggered Toggle FF with an Asynchronous Preset.

Transient (Dynamic) Response With Node 2 SA1



Whether this Pulse Occurs (and How Much Energy it Contains) is a Function of Many Variables—Signal Skew, Propagation Delay, Age of Components, Temperature, Etc.

Fig. 8. *Intermittent faults may also be caused by race conditions. Here, when node two is stuck at a logic 1, a one's hazard appears on node five. The flip-flop may or may not respond to the pulse. Again,* FASTRACE *selects the most common fault syndrome.*

match the initially selected syndrome. The pattern set is continuously reapplied until the original syndrome returns. For example, the circuit of Fig. 7 repeats the same syndrome every fourth pattern set application; therefore, for every node that is probed, three sets of nodal responses are ignored because the syndrome is not correct.

Another assumption made so far is that the operator places the probe on the correct point when requested. During the course of a day's probing, even the most experienced operator will occasionally probe the wrong point. To minimize operator misprobes, all pertinent IC pins are probed in sorted ascending order before moving to the next IC. Thus pin-to-pin and chip-to-chip movement is minimized. Each node is probed at least twice in different locations along the trace. If a node responds differently at different locations then one or more of three conditions has occurred:

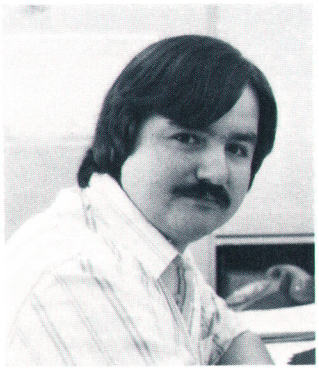
1. The node contains an open trace.
2. The operator has misprobed.
3. The node is intermittent (even though the output syndrome is constant).

When this situation occurs, each point on the node is reprobed. If an open trace exists, then the second set of measurements will be the same as the first set (but different from each other). If a misprobe has occurred (or a node is intermittent) then the second set of measurements will differ from the first set.

Another problem often encountered during backtrace is that a point to be probed is inaccessible. For example, edge connector signals may be on the bottom side of the board. To keep from having to probe these points, the test programmer can declare any point on the PCB inaccessible. During backtrace, FASTRACE will not probe inaccessible points. When a diagnosis is reached, if any inaccessible point could also be the fault, this information is listed on the test report as "not checked." If all points on the PCB are declared inaccessible, a diagnosis will still be reached and all inaccessible nodes that could be faulty will be listed as not checked.

Acknowledgments

Early work on FASTRACE was done by Bill Haydamack and Bob Aiken. George Booth made a major contribution with the probing algorithm, and Ken Parker provided helpful advice and counsel.



William A. Groves

Bill Groves joined HP in 1973 as a computer systems analyst, then transferred to Loveland Instrument Division where he designed the FASTRACE software for the DTS-70. Bill studied electrical engineering at Johns Hopkins University and is working toward his MSEE degree at Colorado State University. Born in Rahway, New Jersey, Bill is married, has three children, and lives in Loveland, Colorado. Bill enjoys waterskiing, boating, fishing, and ice skating.

Software Simulator Speeds Digital Board Test Generation

by Kenneth P. Parker

TESTAID IS A GROUP of software programs for HP 1000 Computer Systems. It is used to develop high-quality production test programs for the DTS-70 Digital Test System for printed-circuit boards. The DTS-70, in conjunction with the FASTRACE software package (see preceding article), is capable of verifying the per-

formance of a logic circuit and isolating the cause of a malfunction. It is TESTAID's primary responsibility to produce the data base for FASTRACE. However, TESTAID is also gaining acceptance as a design aid, to help a logic designer spot potential flaws in a design before it is committed to production. This can be helpful in integrated circuit design.

The central program in TESTAID is SIMUL, a logic simulator. It operates on circuit description data provided by other TESTAID programs. It is used typically in an interactive mode, in which the user specifies the stimulus to the circuit (input patterns), and SIMUL propagates the effects of these patterns through the logic and displays the result. On request, it will also analyze the behavior of the circuit in the presence of circuit malfunctions (faults) and provides the user with a measure of test effectiveness. In other words, if the pattern set being developed is intended for use as a circuit test, SIMUL determines how effective this test will be at exposing potential circuit malfunctions.

Simulation Techniques

Over the past fifteen years many techniques for logic simulation have been developed. The earliest and most straightforward simulators were known as compiled code simulators, which meant that the simulators were executable programs made up of the circuit equations. Once compiled, this kind of program can simulate the circuit's logic at full CPU execution rates. However, it must execute all the equations for each input pattern. This can take a long time. Other problems with compiled code simulators are that the circuit equations must be properly ordered to simulate input-to-output propagation (tedious by hand and not trivial to program), and no feedback loops are allowed (they must be found and broken). Finally, a problem arises because these equations are evaluated as Boolean equations (this is called two-valued simulation). Having only 0 and 1 as allowable signal values, which does one assume for the value of an uninitialized memory element?

These difficulties led to new developments, namely multiple-valued and table-driven simulation techniques. SIMUL is a three-valued table-driven simulator. The three values are 0, 1, and X, where X represents unknown. SIMUL thinks of a circuit as a table of gates, signals, and connectivity relations as in the example in Fig. 1. Assume that ABC=000 has been applied. Then E=1, F=1, G=1, and D=1. If we wish to change input A to 1, SIMUL uses the following algorithm:

1. Find input pin in table (gate 5).
2. Change its output signal to 1.
3. Queue the evaluation of its successor gates (gate 2).
4. If the queue is empty, STOP.
5. Take a gate out of the queue.
6. Fetch its input signal values.
7. Noting gate type (NAND, OR, etc.) calculate output.
8. If output is different from before, store it and schedule successors in queue. If gate already exists in queue, do not enter it again.
9. Go to step 4.

Using this algorithm, the gates evaluated will be:

5	(input pin)	output changes to 1
2	(NAND)	output changes to 0
4	(OR)	no output change

Note that not all gates are evaluated and that the order of their evaluation is by cause and effect. A table-driven simulator is slower in a gate-by-gate comparison with a compiled code simulator, but can often complete a pattern calculation more quickly when only a small subset of gates is actually evaluated.

Because of the third value, X, a ternary algebra must be used to evaluate a logic circuit. Consider, for example, the three most basic circuit elements, AND, OR, and NOT.

SIMUL represents 0, X, and 1 with two bits: 0 is 00, X is 01, and 1 is 11. The computer has AND, OR, and NOT instructions (they operate on a full word) plus an instruction for swapping bytes (8 bits) in a word (16 bits). SIMUL arranges the two bits of data in split-bit format, that is, the right bit in the right byte (right justified) and the left bit in the left byte (right justified). Then the following assembly code, which executes very quickly, will perform ternary AND, OR, and NOT calculations.

AND:	LDA	SIGA	Operand A
	AND	SIGB	ANDed with B
	STA	SIGC	Stored in C
OR:	LDA	SIGA	Operand A
	IOR	SIGB	ORed with B
	STA	SIGC	Stored in C
NOT:	LDA	SIGA	Operand A
	CMA		Complement
	ALF,ALF		Swap Bytes
	STA	SIGB	Stored in B

More complicated functions, such as EXCLUSIVE OR or random-access memory, can also be coded. These basic functions are known as primitives. All circuit descriptions including integrated circuit models from TESTAID libraries are automatically expanded to the primitive level.

Simulating Timing

To simulate a complex circuit, a simulation system must allow the user to model the circuit in some reasonable fashion. We have already seen some TESTAID primitives

"Gate"	Type	Inputs	Output	Successors
1	NAND	B,C	F	2,3
2	NAND	A,F	E	4
3	NAND	C,F	G	4
4	OR	E,G	D	8
5	Input Pin	—	A	2
6	Input Pin	—	B	1
7	Input Pin	—	C	1,3
8	Output Pin	D	—	—

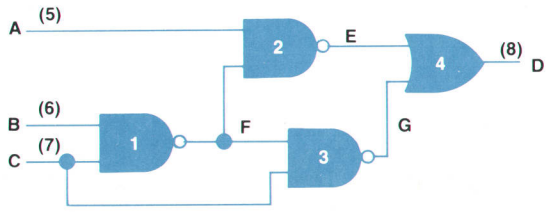


Fig. 1. *SIMUL, a logic simulator, is the central program in the TESTAID software package. SIMUL thinks of a circuit as a table of gates, signals, and connectivity relations.*

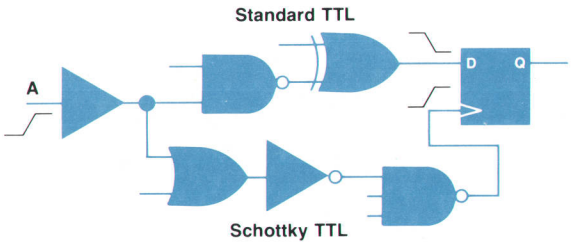


Fig. 2. *SIMUL assumes each gate has zero delay, but allows the programmer to specify units of delay for various circuits. Here the designer has used some slow components to resolve a race. This can be specified by inserting delays.*

that might allow a user to describe the logic of a circuit, but what about timing information? All hardware logic takes time to operate and signals take time to propagate. In some instances, these delays are not important to the circuit function, but in others, they are fundamental to proper performance. Some simulation systems allow the user to specify detailed delays and propagation characteristics for each gate, but the penalty is that even small circuits can be prohibitively expensive to simulate because of the computational complexity of this approach. Others remove this complexity by simplifying the delay characteristics of their primitives. One such approach is to assume a unit of delay for each primitive. This assumption is intuitive and easy to implement. However, this assumption may lead to incorrect simulation (see Fig. 2). In the circuit of Fig. 2, signal A fans out and its effects arrive at a flip-flop by two paths of different length, one path for data and another for clock. A unit delay assumption will result in the flip-flop being clocked after the data arrives, latching a zero. However, in the hardware realization the clock arrives before the data, thus latching a 1, since the clock path is composed of higher-speed Schottky parts (it is common for designers to do this). This disparity can be very hard to uncover when such circuitry is buried in a maze of surrounding circuitry. In other situations, paths may exist that the designer was unaware of. Or if, say, one path is 14 gates long and the other 18 gates long, a unit delay simulator happily resolves timing in favor of the shorter path even though the accumulated uncertainty of the delay in either path grossly exceeds the average delay differential.

SIMUL has a somewhat counter-intuitive delay assumption. It assumes each gate has zero delay. In this way, all potential timing problems, such as those in Fig. 2, are not resolved in either direction. In Fig. 2, SIMUL will find the clock and data changing simultaneously, which will have the effect of loading an X into the flip-flop, for unknown. Thus, SIMUL refuses to predict a winner of a race for any such circuit even if the path differential is many gates. We consider this approach pessimistic, since if anything can possibly go wrong, SIMUL will flag it with X's. Now, in Fig. 2 we saw that the designer used some slow components to resolve a race, adding delay to the circuit to make it work properly. One of the TESTAID primitives is a delay element of one unit of delay. By inserting this primitive in the data path of Fig. 2, the timing can be specified so the clock is the winner, as the designer intended.

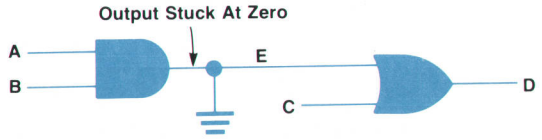
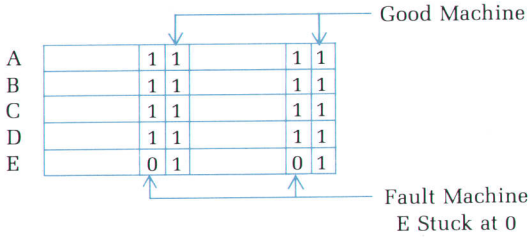


Fig. 3. *SIMUL simulates the properly functioning circuit and all possible faulty circuits. Node E stuck at zero is one possible fault.*

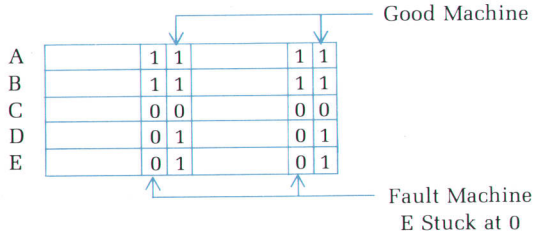
Fault Simulation

It has been mentioned that SIMUL is capable of fault simulation. This is accomplished by making use of the other fourteen bits in a word left over after simulating the properly functioning circuit (termed the "good machine") with two bits in split-bit format. A fault machine is another split-bit pair representing the behavior of the circuit in the presence of a fault.

Consider the circuit of Fig. 3. Signals A through E are 16-bit words of memory. Letting ABC=111, these words appear as follows after fault simulation.



Changing ABC to 110, memory now appears as:



Note that the fault has propagated to signal D (an output pin), where ultimately the tester hardware could see it and flag it. Similarly, other faults can be simulated, using the full word. Thus it is a straightforward task to simulate the good machine and seven faulty machines in parallel. However, many circuits given to TESTAID to simulate contain many thousands of potential faults, and it could take an unacceptable amount of time to simulate these only seven at a time (termed a "simulation pass"). TESTAID takes advantage of a common circuit characteristic to improve simula-

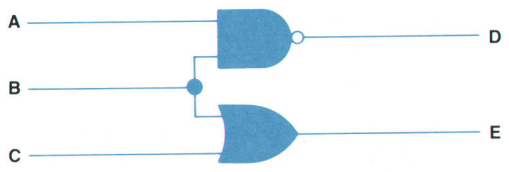


Fig. 4. *SIMUL groups noninteractive faults and equivalent faults. Here faults A stuck at zero and C stuck at zero are noninteractive. Faults E stuck at one and C stuck at one are equivalent.*

tion time. Circuits typically have multiple outputs, and faults may be noninteractive due to their position in the circuit's topology. In Fig. 4, for example, fault A SA0 (stuck at 0) cannot affect signal E, and fault C SA0 cannot affect signal D. But fault B SA0 can appear at either output. We could simulate A SA0 and C SA0 together in the same split-bit column, rather than separately, since by looking at the output behavior of signals D and E, we can later separate their effects. By so grouping, ten faults can be packed into six fault machines for simulation in a single pass, saving one pass in this example. In larger circuits, it often occurs that hundreds of faults are simulated in a single pass of seven fault machines. Additionally, TESTAID can analyze a circuit for fault equivalencies. Two faults are equivalent (such as E SA1 and C SA1 in Fig. 4) if for all possible tests, all tests that detect one also detect the other and vice versa. TESTAID records equivalent groups, and simulates only one fault from each since the circuit will behave identically for the others.

Races and Hazards

One of the most important contributions of SIMUL is its ability to detect the effects of races and hazards in a circuit. This capability is a natural result of the X state and the zero-delay assumption.[1] For the purposes of this article a race is two or more inputs to a circuit element changing at the same time. A hazard is a momentary pulse, or glitch, that may occur as a result of a race. Races are of little interest in purely combinational circuits since the circuit response upon stabilization is independent of which signal actually wins the race. However, in sequential circuits, the winner of the race may determine the resultant stable state of the circuit. The winner is often determined by such imponderables as stray capacitances, propagation delays, temperature, and the ages of components. Races in sequential circuits may not show probabilistic effects until some circuit parameter has had enough time to drift. Thus a circuit containing a race can often escape notice until it is finally tracked down as a chronic field service problem, months or even years after its design. Worse yet, the apparent failing component, such as the D flip-flop in Fig. 2 that latches the wrong state, is really not responsible. It is some upstream component that has had a change in one of its parameters; perhaps a very subtle change. The final horror is that this problem may be intermittent in nature, occurring only occasionally when conditions are just right (usually at the customer's site), and these conditions are difficult to duplicate at the repair center.

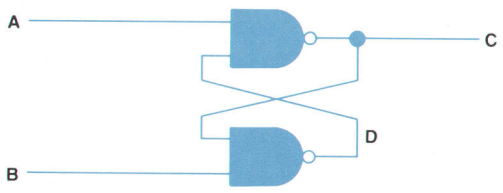


Fig. 5. *An illustration of how SIMUL handles the classic race situation. If AB changes from 00 to 11, SIMUL computes the final value of CD as XX, where X stands for unknown, since these values depend on whether A or B was last zero, and this is impossible to determine.*

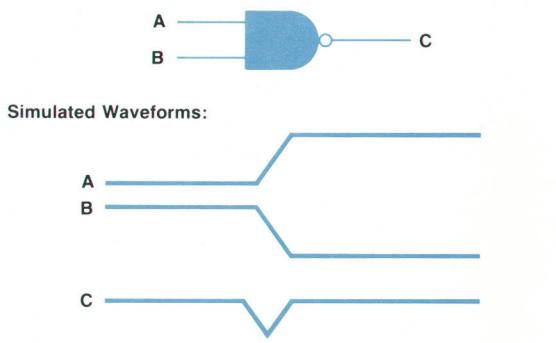


Fig. 6. *In this circuit, if AB changes from 01 to 10, signal C experiences a 1X1 hazard, that is, a pulse may appear at node C. SIMUL predicts this possibility.*

Obviously, it is desirable to avoid such frustrations by making sure that damaging races do not occur in a circuit. Experience with TESTAID has proven that few circuits produced anywhere today are completely free of potential races. They are difficult to spot even in fairly small circuits without some form of design-aid such as TESTAID.

Let us examine how SIMUL handles the classic race situation shown in Fig. 5. Let AB be 00. Then C and D are 11. What happens if we raise A and B to 11 simultaneously? SIMUL carries out two calculations: first for AB equal to XX, representing the period of time during which the states of signals A and B are changing, and second for AB equal to 11, completing their transition.

	A	B	C	D	
One	0	0	1	1	Initial State
Transition ↓	X	X	X	X	First Calculation
Time	1	1	X	X	Second Calculation

Note that signals C and D become unknown. Textbooks on logic cite this example as an illegal input to the circuit because the result is indeterminate. The English-language description of the performance of this circuit is, "The value of signal C (0 or 1) when A and B are both 1 reflects which of the two inputs (B or A) was last 0." The illegal transition from 00 to 11 prevents the circuit from accurately determining "which input was last 0." SIMUL reflects this with an X for "I don't know." Thus SIMUL automatically eliminates false assumptions about memory states in the presence of races.

Hazards (glitches) are another problem. Consider Fig. 6. Let AB be 01. If we then apply 10 to AB, what is the result? As before, SIMUL calculates AB = XX before AB = 10.

	A	B	C	
One	0	1	1	Initial State
Transition ↓	X	X	X	First Calculation
Time	1	0	1	Second Calculation

Signal C experiences what is known as a 1X1 hazard. Fig. 6 shows the simulated waveform corresponding to this hazard. The other hazard, 0X0, can also exist in a circuit. In either case, the signal value is the same before and after the calculations, but a momentary instability might exist during the calculations, that is, the possibility of a pulse exists.

Virtual Memory for TESTAID and FASTRACE

by Douglas L. Baskins

TESTAID can use up to 2.5 M bytes of data array space. This is too much data space for an HP 1000 Computer's memory. Therefore a virtual memory (VM) system was developed to allow the TESTAID and FASTRACE programs to think that all data arrays are in memory, while in fact only some are in memory and the rest are in disc memory. In the VM system, the data array space is broken up into pages of 512 bytes each. Each page of the array space may exist in memory or in disc memory at any given moment. When an access is made to a particular data item, a microcoded instruction is executed to find the page location in memory. If the page is found in memory, the data item's address is calculated and the data item is retrieved. However, if the page is in disc, an old page must be flushed from memory to make room for the new page with the data item. Bringing in the new page always requires a disc access. If the old page has been modified, that is, it does not have an exact counterpart in disc, an additional disc access is required to update the disc copy.

It takes about 21 microseconds to find and retrieve a data item from memory. However, if the data item is in disc, it takes an average of 25 milliseconds or about 1000 times longer to retrieve. The job of the virtual memory (VM) system is to keep the often used data in memory and the seldom used data in disc.

TESTAID was first introduced on a 64K-byte computer. 32K bytes or 64 pages were used by the VM system for data array space. The remaining 32K bytes were used for the RTE operating system and code segments of TESTAID. Therefore a maximum of 64 out of 4848 pages could be accessed at the 21-microsecond rate.

With the introduction of the HP 1000 System and the dramatic reduction in memory prices, large memories have become very practical. For the HP 1000 System, the VM system was extended from 64 pages to 256 pages of possible data array space in memory. In addition, the number of disc accesses (MMA), the memory-to-disc-access ratio (MTD) and the number of primitive evaluations done by the simulator (CPE) are monitored and displayed to the user upon request. With larger memory, simulation times for large boards were dramatically reduced, in some cases by a factor of five or more. Furthermore, the VM system is now able to adapt a "working set" of relevant data array space into memory. When this happens, the total disc access time is reduced to a point where it is insignificant compared to the total simulation time. A remarkable fact is that only 10% or less of the total data array space need be in memory to achieve a working set! This means that an infinite amount of memory would not further improve simulation times significantly.

The amount of memory needed to establish a working set is dependent upon the board size to be simulated. Fig. 1 shows the simulation times versus the number of pages in memory for a modest-sized board. This board was given enough patterns to produce 1.5 million primitive evaluations. Excluding the time used for VM disc accesses, 0.15 hour was used for loading code segments, 0.17 hour for the VM microcode instructions and 0.28 hour for actual simulation calculations. The rest of the time was spent for disc accesses in the VM system.

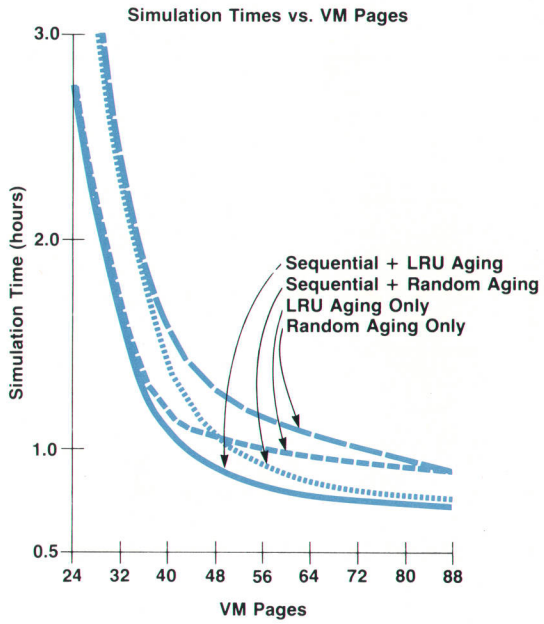


Fig. 1. *TESTAID uses up to 2.5 M bytes of data array space. This exceeds the memory capacity of an HP 1000 Computer, so a virtual memory system keeps most pages of data in disc memory and only the currently needed pages in computer memory. The algorithm that moves pages between the two memories is a combination of least-recently-used and sequential page replacement techniques.*

Algorithms

Two page replacement algorithms were studied to determine their effectiveness. The random algorithm is probably the simplest and the least effective. This method uses a pseudorandom number generator to determine which page should be flushed when a new page must be brought in from the disc. The least-recently-used (LRU) algorithm is probably one of the most effective, but requires many more memory accesses and some computation on every access to the VM system. This method increments a master LRU counter for every access to the VM system and modifies a freshness LRU counter for the page that was accessed. The freshness LRU counter is modified by averaging its contents with those of the master LRU counter, and storing the results back into the freshness LRU counter. Pages that have not been accessed and modified recently will have a count that remains stagnant, while the active pages' freshness LRU counters will keep pace with the master LRU counter. The page with the smallest freshness count will be flushed when a new page needs to be brought in from the disc.

Since memory accesses and computation can be done concurrently in HP 1000 microcode, most of the time used for the VM access instructions is in finding the page location of a particular data item in memory. If a full 256 pages were in memory, an average of 128 pages would have to be searched to find the one that contains the data item. At approximately two microseconds per page, this would make the access time intolerably long. Therefore, a much faster scheme was devised. When a VM access instruction is entered, the page number is calculated from the data item's address. The least five bits of this number are used as an index into a HASH table. Each entry in the HASH table contains a head pointer to a circular list. These circular lists tie a group of page table entries (PTEs) together. Each PTE contains an LRU freshness counter and the location of a particular page in memory. The average length of the 32 circular lists is

256/32 or about eight elements long. It takes about 2.8 microseconds per element in a circular list to see if it matches the page number that contains the data item. If the accesses to the VM system were purely random, one would expect the average search to be about four elements long. However, a digital simulator has order in its data access that is not purely random. Also, the simulator tends to access data in the immediate vicinity of recently accessed data. This "locality of reference" reduced the average search length to a measured 1.2 elements when the head pointer was updated to the newly matched PTE. The reduction of a possible average search length from 128 to about 1.2 was considered a good solution and made a 256-page VM system practical.

The simulator also makes sequential access to the VM system in obvious places. Therefore a sequential access instruction was implemented for use in these places. When a page has contiguous data items with another page it is called an adjacent page. The sequential instruction defeats the LRU page replacement algorithm if there is an adjacent page in memory when the accessed data item must be brought in from the disc.

The sequential instruction has a threefold benefit. Two of these benefits are related to the mechanical properties of a moving-head disc. The DTS-70 System has primarily been used with the HP 7905/7906/7920 discs. Data is recorded on these discs in tracks that are in concentric circles or cylinders, and there are several surfaces, each with a read/write head. The discs revolve at 3600 r/min and thus have a rotational delay of one-half revolution or about 8.3 milliseconds. The time it takes for the heads to go from one cylinder to another depends on the distance the heads have to travel. This seek time varies from 5 milliseconds for an adjacent cylinder to a maximum of 45 milliseconds to travel all the way across the disc. One revolution contains 24 pages of data. Since access time to the disc is a function of how far the heads must move, the sequential instruction tends to (1) eliminate the need for the heads to move because the access is adjacent to the previous access, and (2) synchronize itself such that computation will be done while the next page on the disc is approaching the disc head. This makes the actual average access time less than a revolution. The third benefit is the most important. When a new page is required from the disc by a sequential instruction and an adjacent page is in memory, chances are excellent that it is very fresh. Yet because it is being accessed sequentially, it will not be needed for a very long time. If the LRU page replacement algorithm were used, the fresh adjacent page would be overlooked and one of the pages in the established working set would be flushed. If the sequential access extended through many pages, soon the entire working set would be flushed from memory. The algorithm solves this problem by flushing the adjacent page, thereby preventing the VM system's working set from being flushed when a long sequential search is done.

Fig. 1 shows the simulation times of four algorithms tried in the VM system. The LRU page replacement algorithm plus the sequential page replacement algorithm was the one chosen for the DTS-70 system, because it gave the best performance for both large and small memory systems.

Douglas L. Baskins

Doug Baskins designed microwave sweepers as a non-degreed engineer at another firm (where he obtained three patents) before joining HP in 1970 to get involved in computers. For six years he was a systems engineer in HP's eastern sales region, then joined the lab and developed the VM system. Born in Seattle, Washington, Doug lives in Fort Collins, Colorado, with his wife and son (15). Hobbies include snow and water skiing, scuba diving, racquetball, and photography.

Whether or not it really occurs is a function of the circuit and input waveform parameters. Again, the pessimistic outlook of SIMUL predicts the possibility.

Hazards are quite common in typical designs. For example, Fig. 7 shows a test circuit for a 74LS191 binary four-bit up/down counter. There is an output line on this IC called MAX/MIN which, since it is merely the AND (during count-up) of the four counter bits, is a hazardous signal. By adjusting a circuit parameter (RF trimmer capacitor on output Q_A set to approximately 18 pF) a spike 15 nanoseconds wide and 2.2 volts high appears on MAX/MIN as shown in the oscillograph in Fig. 7. If MAX/MIN is used in a circuit to trigger other clocked circuitry, trouble could occur if the possibility of this pulse is not accounted for.

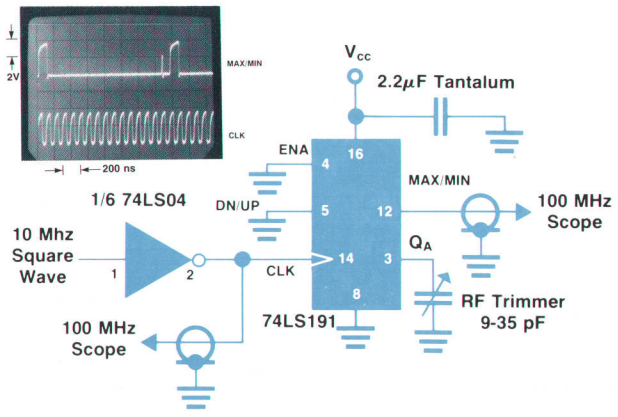


Fig. 7. *Hazards are common in many circuits, as illustrated by the spike appearing on the* MAX/MIN *line in this four-bit binary counter circuit.*

Hazards and races are a function of the design of a circuit. For this reason, TESTAID is now being used to evaluate designs while there is still an opportunity to fix such problems. However, even for a "perfect" design, hazards and races are often still a problem for the test engineer, who is not only concerned with how a circuit works, but also with how it doesn't work. It is possible for a fault to introduce a race or a hazard, resulting in the phenomenon known as the "possible detect." A fault may also prevent the initialization of a circuit. This yields an excellent example of a possible detect. For definition, a possible detect occurs in a fault simulation when the good machine produces a known output (0 or 1) but a fault produces an unknown output X. For example, consider a fault on a flip-flop clear line that prevents it from being cleared. The simulation assumes it is

initially in an unknown state upon power-up. If the first test is a clear, the good machine will clear but the faulty machine will stay unknown, yielding a possible detect. Assuming that a flip-flop has a 50-50 chance of powering-up in the clear state, there is a 50% chance that the tester will detect the clear line malfunction. We would like to get better odds of testing for such faults. SIMUL notes the possibility of detecting the fault, but does not drop it from further simulation as it would on a solid detect. Then it makes a clever assumption: if the tester doesn't see the fault at this point, it's because the faulty machine is the same as the good machine. From this point, SIMUL erases all X's for this fault and sets each signal from the fault site onwards to the same value as the good machine. Future calculations involving this fault machine will be with known states. Now a race or hazard, when enabled by a fault, may again result in a memory element being assigned an X state, leading to a possible detect. This effect too will be noted and erased. After five possible detects, the fault will be dropped. Assuming each one had a 50-50 chance of being seen by the tester and that they are independent occurrences (remember the erasure after each occurrence), the probability of such faults sneaking past five tests is 1/32 or roughly 3%.

Performance

TESTAID has been used to simulate circuit boards containing over 220 MSI chips. Boards of this size contain between 5000 and 8000 potential faults, and have 6000 to 9000 signals to keep track of for each test. SIMUL keeps a tally of the primitive evaluations, and numbers exceeding 100 million evaluations have been observed, as have rates of four million per hour.

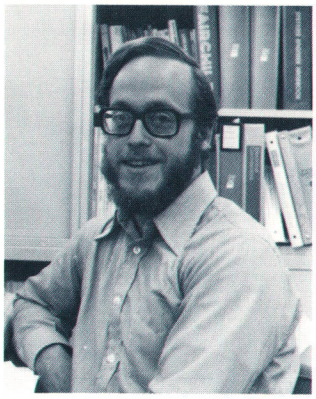
When faults are not being simulated, that is, when just a single pass is being made, SIMUL is effective as an interactive program. A user can type in a pattern and get the result in seconds.

Acknowledgments

The author would like to acknowledge the significant contributions made to TESTAID by Bill Haydamack, Doug Baskins, and Ravi Apte, one of the original authors.

Reference

1. E.B. Eichelberger, "Hazard Detection in Combinational and Sequential Switching Circuits," IBM Journal, March 1965.



Kenneth P. Parker

An HP employee since 1975, Ken Parker was coauthor of the TESTAID III Circuit Simulator for the DTS-70. Author of several published papers on digital circuit testing, Ken received his BS degree in computer engineering in 1972 from the University of Illinois and his MSEE and PhD degrees (1974 and 1976) from Stanford University. He was born in Lake Forest, Illinois, is single, and lives in Ft. Collins, Colorado. Hiking and skiing keep Ken busy in his leisure hours.

Analog In-Circuit Component Measurements: Problems and Solutions

by David T. Crook

ALTHOUGH THERE ARE A NUMBER of techniques for measuring passive components (resistors, inductors, and capacitors), two techniques are prevalent in automatic test systems: constant current and constant voltage.

In the constant-current technique, a known or measurable current is pumped through the unknown impedance and the resulting voltage drop across the unknown is measured. With the constant-voltage technique, a known or measurable voltage is applied across the unknown impedance and the resulting current flow is measured.

At dc, either technique requires only a straightforward application of Ohm's law to calculate the value of the unknown. For ac measurements, the stimulus (current or voltage) needs to have low distortion and a known or measurable frequency. If just the magnitude of the complex impedance is desired, the measurement device needs only to be capable of simple ac magnitude response, but where the resistive and reactive components have to be identified, the measurement device must be capable of selecting and/or rejecting the in-phase and quadrature components of the detected signal.

The Model 3060A Board Test System has these capabilities and uses them for measuring all three types of passive components on printed-circuit boards.

Guarded Measurements for In-Circuit Testing

In-circuit testing, that is, using a bed-of-nails fixture as a probe for measuring the impedance of components already installed in circuits, complicates the measurement procedure because a component being measured is rarely isolated from other components. These other components

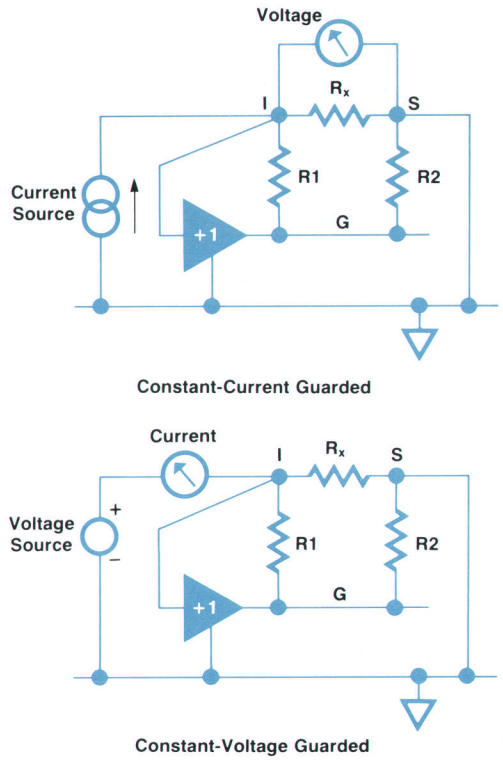


Fig. 1. *A guarded measurement is implemented by using a unity-gain amplifier to force the voltage at node G to be the same as at point I. Hence, none of the drive current flows through R1 so it all goes through the unknown Rx*

could bypass some of the measurement current, causing erroneous results. A guarding technique, similar to that long employed for measurements of low capacitances, can be used to eliminate the effects of the bypassed current and thus enhance measurement accuracy.

A typical situation is shown in Fig. 1. The unknown resistor, R_x, connects to other resistors, R1 and R2, at nodes I and S. To implement a guarded measurement, the other ends of R1 and R2 are connected at node G, which is driven by a unity-gain amplifier. The amplifier forces the voltage at node G to be equal to the voltage at node I so no current flows through R1 and all the current from the source flows through R_x. The voltage drop across R_x is thus an accurate representation of its impedance.

Note that current does flow through R2 but this is in a loop that goes from the grounded side of the source, through R2, through the ground return of the amplifier power supply, back to the source ground point. This current therefore does not flow through R_x so it causes no errors in the measurement.

Errors could be caused, however, by deviations from the amplifier's ideal characteristics that would allow a voltage drop to exist between nodes I and G. Since the output of an operational amplifier seeks to bring the inverting input to the same potential as the noninverting input, the op-amp configuration can be used to eliminate any voltage drop between I and G. Fig. 2 shows how op-amp guarding establishes a virtual ground at node I for both the constant-current and constant-voltage configurations. Both of these implementations, which provide excellent guarding, are used in the 3060A.

Constant Voltage or Constant Current?

The constant-current configuration is used in the 3060A Board Test System to measure low resistances, diodes, and transistor beta, but the constant-voltage technique is used for most other measurements. The constant-current configuration has an advantage in that the voltage reading is proportional to R_x or L_x, but this advantage disappears when the measurement system has a controller that can calculate the reciprocal when the constant-voltage technique is used.

Of more significance, a disadvantage of the constant-current technique is the fact that both the S and I nodes are part of the op-amp feedback circuit. If capacitances are associated with the components to be guarded out, they could cause enough phase shift to make the op-amp circuit unstable. The constant-voltage configuration has only one test node (I) involved with the feedback path so capacitance can contribute a maximum of 90° phase shift to the feedback signal, an improvement of 90° over the maximum 180° phase shift that two capacitive nodes could contribute. In addition, the 3060A provides selectable narrowband or wideband compensation, allowing a tradeoff between exceptionally high stability for dc measurements and good stability with the high gain needed to hold node I at virtual

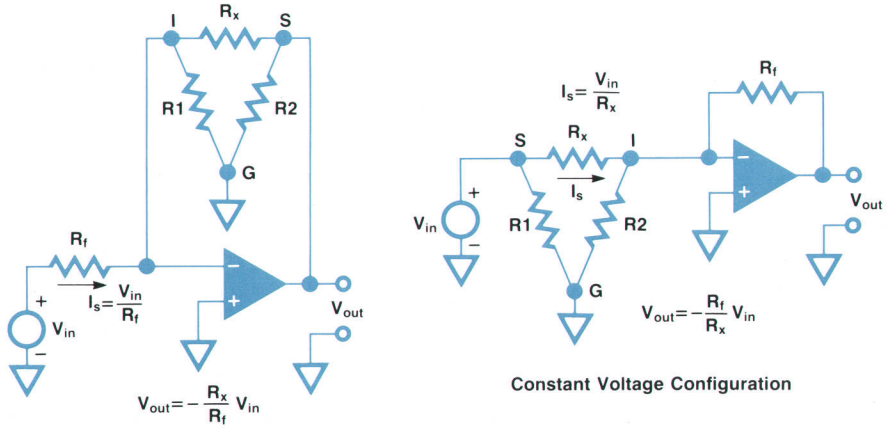


Fig. 2. *Accuracy in guarded measurements is assured by using an operational amplifier to maintain a virtual ground at node I.*

ground during ac measurements.

Another disadvantage of the constant-current configuration is the possibility it creates for test-induced damage. If, for example, a 10-kΩ resistor were misloaded in place of a 1-kΩ resistor, the voltage across the resistor would go to ten times the expected voltage in the constant-current mode. If a semiconductor were connected to the S node, it could be destroyed by the test on the resistor. A subtle ramification occurs if the semiconductor had been tested prior to the resistor and found to be good—it could then be damaged by the resistor test but test results would indicate only that the resistor was bad.

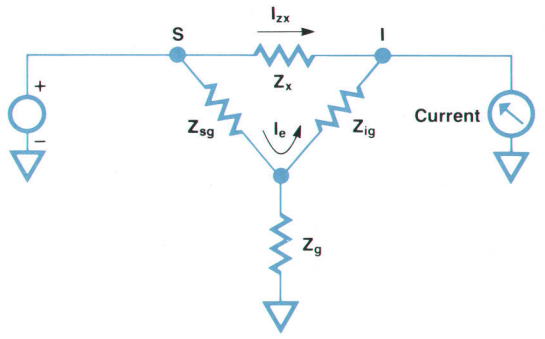
To minimize the possibility of test-induced damage whenever the constant-current configuration is used in the 3060A Board Tester, a voltage-compliance limit, programmable in 0.1-volt increments, provides a hard clamp on the amplifier output. Likewise, there is a programmable current limit in the constant-voltage mode.

The constant-current configuration also slows measurement time when a resistor being measured is paralleled by a large capacitor, since only the constant current is available to charge the capacitor. In the constant-voltage mode, the output of the operational amplifier will far exceed its steady-state value in an attempt to keep node I at virtual ground while the capacitor charges. This results in a much larger charging current so less wait time is required.

Lead Errors

Errors can be caused by voltage drops across the impedances of the connecting leads. Fig. 3 shows the three-terminal guarded circuit with the three lead impedances Z_s, Z_i, and Z_g. The error caused by the guard lead impedance, Z_g, can be determined by evaluating the guard gain error, which is the ratio of the current I_e flowing from node S to node I, through Z_{sg} and Z_{ig}, to the current I_{zx} through Z_x. The assumption is made that this ratio is not strongly affected by lead impedances Z_s and Z_i as long as they are much smaller than Z_x, Z_{sg}, and Z_{ig}. Fig. 4 shows the approximations made.

If Z_g, however, were only 1Ω, which is not unreasonable in an automatic test system where the lead path may involve a length of wire, several connections, and at least one relay, the approximate guard gain error would be given by the ratio of the unknown impedance Z_x to the product of the two shunting impedances, Z_{sg} and Z_{ig}. If Z_x were 10 kΩ and Z_{sg} and Z_{ig} were each 50Ω, the approximate guard gain error would be 4.00, or 400%!



$$\text{Guard Gain Error} = \frac{I_e}{I_{zx}} = \frac{Z_g Z_x}{Z_{sg}(Z_g + Z_{ig}) + Z_g Z_{ig}}$$

If Z_{sg}, $Z_{ig} \gg Z_g$ and if $Z_g = 1\Omega$

$$\text{Guard Gain Error} \approx \frac{Z_x}{Z_{sg}\, Z_{ig}}$$

Fig. 4. *Reduced network equivalent to that of Fig. 3 with the assumption that lead impedances Z_s and Z_i of Fig. 3 have little effect on the guard gain error.*

As shown in Fig. 4, the guard-gain error is directly proportional to Z_g, so any changes in relay contact resistance or bed-of-nails contact resistance would have a noticeable effect on the measurement.

Looking back at Fig. 3, it can be seen that current through Z_s and Z_i causes voltage drops that result in the voltage across Z_x being less than the known drive voltage, V_s, and, hence, in error. Also, the presence of Z_i indicates that a virtual ground does not exist at the amplifier input. Instead, there is a virtual ground with an impedance in series with it.

These lead errors may be eliminated in the Model 3060A by providing sense terminals for each signal terminal, as shown in Fig. 5. The A and B leads enable the exact voltage across the unknown to be measured. The B and L leads insure that the potential at node B is brought to the same level as node G so virtually no current flows through Z_{ig} and all the current goes through R_f. The current may then be

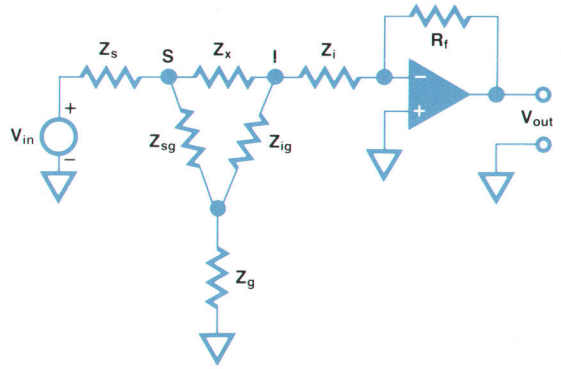


Fig. 3. *Guarded measurement network with the three non-zero lead impedances, Z_s, Z_i, and Z_g, inserted.*

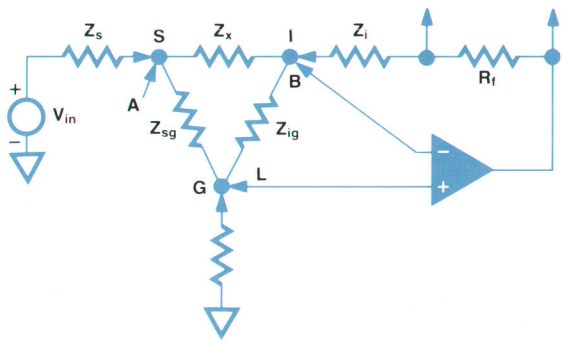


Fig. 5. *Model 3060A Board Test System provides three sense leads, A, B, and L, to eliminate errors caused by lead impedance in the three measurement leads, S, I, and G.*

determined accurately by measuring the voltage drop across R_f which, in the 3060A, is one of seven precision reference resistors selected according to the current level encountered. A resistive network, for example, with an unknown of 10 kΩ and 10Ω shunting resistances on each side to guard can be measured to an accuracy of 2% using the six-terminal connection. The guard gain error for a three-terminal measurement would be 100, or 10,000%. The six-terminal measurement thus reduces the error by a factor of 5000!

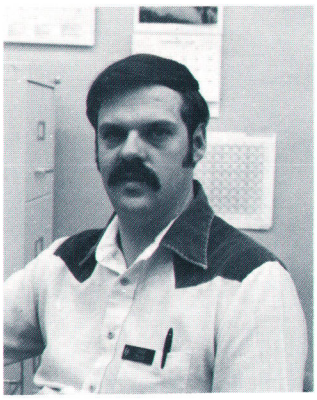
Accuracy Enhancement

Another problem in dc measurements is caused by spurious voltages, primarily from amplifier offsets and thermal voltages in the relay contacts. Although small, these can have a significant effect on tests made on circuit boards that have semiconductor devices, where good practice dictates that the maximum voltage between any two nodes should be less than 0.3 volts. The use of a low voltage prevents semiconductor junctions from forward-biasing and affecting the measurements of R, L, and C.

Suppose that a drive level of 0.1 volts is wanted. A typical thermoelectric junction in either a dry or a mercury-wetted relay may produce up to 40 μV per degree of temperature differential. A five-degree temperature differential is not unusual, so 200 μV may be in series with any lead. This would cause a 0.2% error in the applied 0.1-volt drive, which is not a significant problem when measuring even to 1% accuracy. However, if this thermal voltage appeared in series with the G lead, it would be amplified at the op amp output in proportion to the ratio of the feedback resistor to the resistance from I to G. This can cause errors of several percent where the I to G resistance is less than one-tenth the unknown. This error is not trivial and can have subtle ramifications in that thermal EMFs are a function of the thermal history of the relay in question and of the relays adjacent to it. Thus, boards that tested well last month—or even this morning—may not be running reliably this afternoon because of changes in the ambient temperature.

This is a problem with any high-density relay scanner because the space requirements and cost of truly low-thermal relays are prohibitively high. When spurious voltages are encountered, the 3060A solves this problem by firmware accuracy enhancements. An algorithm measures the signal level caused by thermal EMFs, amplifier offsets, or bias currents. This value is then subtracted from the measured value and a corrected reading is given. Combined with the six-terminal configuration, this is known as extended guarding. It enables measurements ranging from milliohms to gigaohms to be made in various levels of guarding severity. The trade-off, however, is a slower measurement rate.

Extended guarding, plus the use of a synchronous detector, obtains significant improvements in the measurement of reactive components. The basic accuracy and repeatability of measurements made with the Model 3060A Board Tester make the programming of tests for new boards easier and more reliable while making production test more efficient by maximizing circuit-board testability, all of which contributes to the desired goal: less time spent in testing.



David T. Crook

Project manager Dave Crook was responsible for the system design of the 3060A Board Test System and is currently working as 3060A circuit test production engineering manager. Born in Larned, Kansas, Dave received his BSEE degree from the University of Denver in 1969. Dave, his wife, and two-year-old daughter live in Loveland, Colorado. In his spare time, he enjoys photography (including designing electronic darkroom equipment), woodworking, bicycling, hiking, camping, and fishing.

User-Oriented Software for an Automatic Circuit-Board Tester

by Ed O. Schlotzhauer

THE CHARACTER AND PERSONALITY of a test system is determined largely by its software. In general, the software is the main interface between the test programmer and the system, and between the user and the system. A great deal of effort was put into the design of the various software subsystems in the 3060A Board Test System to make this interface "clean," and easy to learn and use.

The Model 3060A Board Test System conducts a series of tests to establish high confidence in the integrity of a printed circuit board before final assembly. The normal order of events is to first test each node for shorts between it and all other nodes to find potentially catastrophic manufacturing defects, such as solder splashes and plating and etching errors, before any power is applied to the board. Next, in-circuit tests are performed on all resistors, capacitors, inductors, transistors, and diodes, using the techniques described in the article beginning on page 19, to

make sure that the analog components are correctly loaded and within specifications. If the board passes these tests, the next step is to power up the board and perform functional tests to examine the specific operation of the board.

Friendly Computer Control

The tests are conducted under control of an HP Model 9825A Desktop Computer. The 9825A was chosen as the controller for this system for a number of reasons. First of all, most of the electronic measurement tools in the board-test system are interfaced through the Hewlett-Packard Interface Bus (HP-IB) and the 9825A is well adapted to working with the HP-IB. Then, our experience with the 9825A has proven it to be a "friendly" machine and very easy for operators to learn and interact with. In addition to that are the general nature of the HPL language it uses,[1] its on-line editing-debugging capability, its reliability, and its relatively low cost. The power of the 9825A, comparable to a 16-bit minicomputer, is further augmented by the microprocessors distributed throughout the 3060A system, which take care of many of the housekeeping tasks.

To simplify the programming and maintenance of the software that the 9825A uses for conducting the board tests, several software packages are included with the board test system. These include the Board Test Language (BTL), the In-Circuit Program Generator (IPG), and confirmation/diagnostic software. BTL is a set of statements executed in the HPL language. IPG is an application program that simplifies generation of the board test programs. The confirmation/diagnostic software verifies the operation of the system and assists in diagnosing system failures.

In-Circuit Program Generation

With many general-purpose test systems, program development costs can easily exceed the cost of the system hardware. Also, since programming is labor-intensive, it often becomes a bottleneck in the test department. Test program maintenance can also become an expensive, time-consuming task.

The In-Circuit Program Generator (IPG), a program to write test programs, was developed to address these problems. Given a description of the circuitry on a board to be tested, it generates a complete in-circuit test program along with a comprehensive set of fixture and program documentation and an analysis of the circuit. By allowing the computer to perform a significant part of the test generation process, it enables faster program and fixture preparation, relieving the user of a great amount of tedium. In addition, this part of the test preparation can be performed by test technicians, IPG enabling them to write good tests even though they may not be familiar with the details and subtleties of in-circuit testing.

The user describes the circuit to IPG by entering each component, the nodes it connects to, its value, and its tolerance. A typical entry is:

R27, 2,17, 10k, 5

In this statement, R specifies the type of component, 27 is the component designator on the schematic of the circuit, 2 and 17 are the nodes to which it connects, 10k is the value of R,and 5 is the tolerance (±5%).

After all the components have been entered, a listing can be obtained for checking correctness of the entries. However, a second type of listing, called a node list, provides a better means of verifying entries. This orders the listing by node number and lists all the components connected to each node. This listing is easy to check against the schematic and, by presenting the data in a form different from the way it was entered, makes it more likely that errors will be detected. Correct data entry is highly important because the model created by IPG for the test program and fixture assignments is derived from this data.

When data entry is complete and checked, simply by telling IPG that it is "done," IPG generates the test program. First, it scans the data for errors such as doubly defined components, bad node assignments, and unusual component values that might be typographical errors. If found, these anomalies are listed for correction or approval.

Next, the components are analyzed using network-analysis methods applied to the measuring network to find the complex impedances from the S and I nodes to the G node for each component. IPG searches the topology file to find all paths around a component and determines the guard connections that should be made to achieve measurement accuracy within the tolerance specified. It uses its knowledge of the 3060A measurement capabilities to make speed/accuracy tradeoffs in selecting the fastest and most efficient test for each component commensurate with the specified tolerance and circuit restraints. At this point, an error analysis is printed for each component indicating the estimated measurement accuracy for the chosen test setup. The calculated complex impedances for reactive components from S to G and from I to G are also listed for the programmer's reference. In some cases, the system cannot test to the desired accuracy, in which case the component is flagged in the error-analysis listing and the listed tolerance is automatically changed to the sum of estimated test accuracy and the specified tolerance.

A final output of the modelling phase is an estimate of the in-circuit test run time.

In the next step, IPG assigns the circuit board nodes to

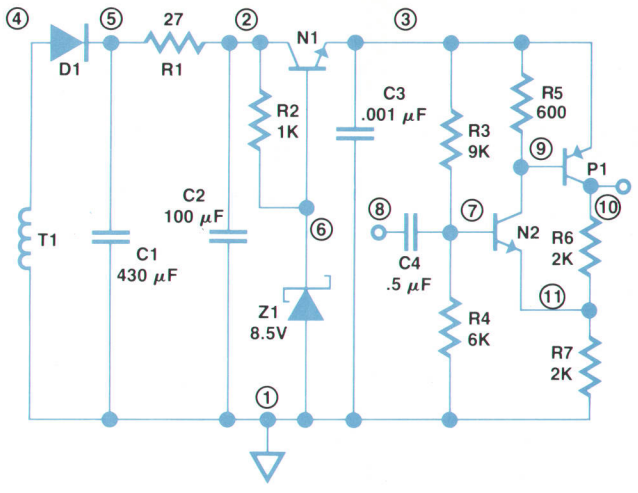


Fig. 1. *Example circuit. Numbers in circles are test nodes. The user describes the circuit to the In-Circuit Program Generator (IPG) by entering each component, its value, and the nodes it connects to.*

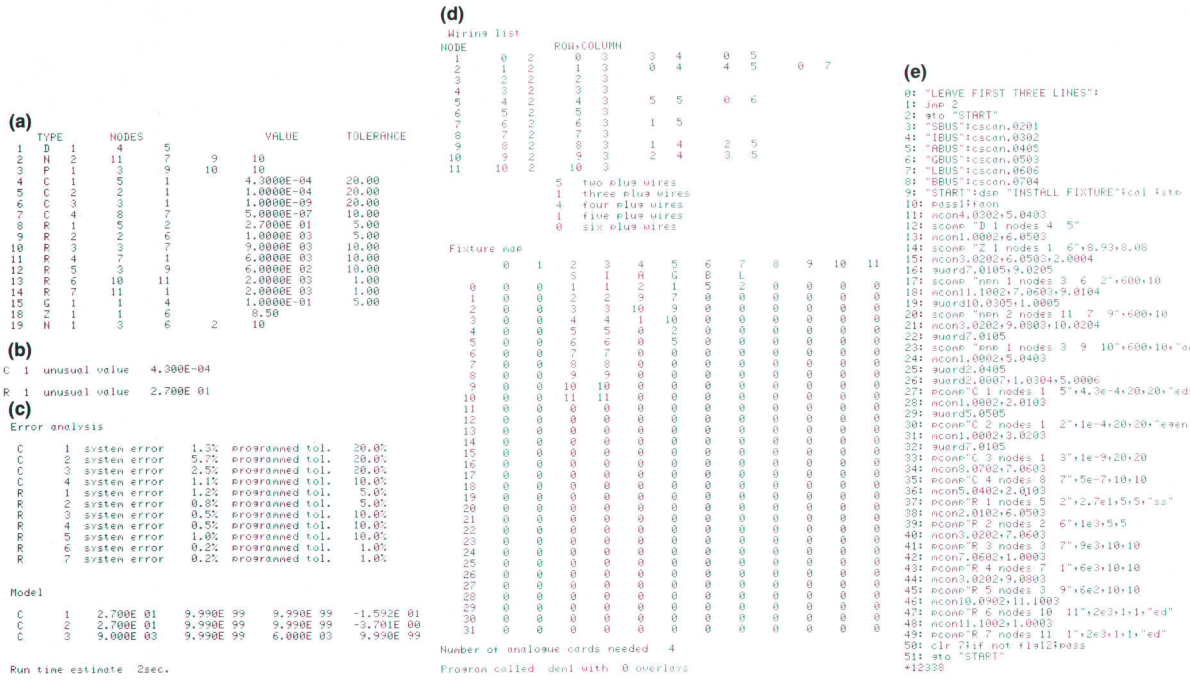


Fig. 2. *Printout of IPG run for the circuit shown in Fig. 1. Components are listed in (a) and unusual values listed in (b). An analysis of the ranges of potential errors is shown in (c). The values listed under "Model" are the complex impedances of reactive components. A wiring list and a map of the patch panel are listed in (d) and the resulting in-circuit test program is shown in (e).*

physical locations in the system's scanner. Statements for performing the scanner column assignments and the statements for connecting the S, I, G, L, A, and B buses are generated and added to the user's program.

Fixture documentation is now generated. First, a wiring list is generated describing the row-column location on the patch board for each node. Then, a map of the scanner indicating the column and node assignments is generated along with a statement of the number of analog cards required to support this test.

Finally, the program lines are generated and stored on flexible disc under a file name previously specified by the user. If the program is too large to be brought into the main memory at one time, it is automatically segmented up to nine times, and all of the overhead statements for program chaining are automatically inserted into the program.

IPG Example

Fig. 2 reproduces a printout of an IPG run for a small circuit. The circuit is shown in Fig. 1. Fig. 2a shows the listing of the component description. Note that the transformer secondary T1 is entered as G1, a G component being simply a general resistive component. The inductance of the transformer winding is not a specified parameter so its dc resistance is entered as a G type for IPG's calculations. The effect of this is that no test will be generated for the component, but its low impedance will be taken into account in the guarding calculations of the other components.

In Fig. 2b, C1 and R1 are flagged as unusual values to be checked. They are correct, so IPG is allowed to proceed. Fig. 2c shows the error analysis of the components, the complex impedance model for the reactive components, and the run-time estimate.

The fixture documentation is in Fig. 2d.

The resulting program is shown in Fig. 2e. This program is a complete and ready-to-run, in-circuit test program for the circuit in Fig. 1.

Board Test Language

Many of the statements shown in Fig. 2e (CSAN, MCON, SCOMP, etc.) are BTL statements. A statement, in this case, is a key word of the programming language that is stored as part of an executable program. BTL (Board Test Language) was developed as an extension of HPL to make it possible to retain all the power and advantages of HPL while adding board-test capabilities that are easy for inexperienced programmers to use. It is also faster, more memory efficient, and more transparent than the use of subroutine calls, and it allows programming at a high level of abstraction. BTL statements are written, edited, listed, and executed just as if they were regular mainframe or ROM-option statements of the 9825A.

Because of the amount of code that would be required for the new statements, it was decided to partition the program so only the code needed to execute a subset of the statements would be in memory at one time. This resides in a fairly small partition of the computer's memory and, in a manner totally transparent to the user, is swapped as re-

quired for other segments of the program stored in flexible disc.

From a thorough study of board-test requirements, tempered by the desire to make the set friendly for the inexperienced programmer while being complete and powerful enough to perform the required tasks, a set of 39 BTL statements was chosen. These range from simple commands that manipulate relays in the scanner to commands like shorts, which executes all the actions needed to conduct an entire shorts/open test on a loaded printed-circuit board.

The shorts test can be performed after a board has been "learned" by the system. To cause the system to learn a board, the shorts table statement stbl is used. The statement includes the number of nodes to test, a string or array variable to hold the shorts table, the impedance threshold that defines a short, selectable between 5 and 125,000, and the time to wait for a node to charge if capacitance is involved, selectable between 0 and 32,767 ms. The threshold and wait terms are optional; the threshold defaults to 10Ω if not specified.

A typical shorts table statement is:

```
stbl 215, A$
```

When a known good board is placed on the bed-of-nails fixture, execution of this statement causes the system to check the impedance between each node pair of the 215 nodes and log into string table A the identity of any node pairs whose impedance is less than 10Ω.

To test a board for shorts, the statement is simply:

```
shorts ⟨string⟩
```

where the string term is the string containing the shorts table. The system then checks all impedance paths on the circuit board for an exact match to the table. If there are shorts or opens that do not match the table, a message, such as the following, is generated.

```
SHORTS
  1 TO 5
  7 TO 36
2 SHORTS
OPENS
  25 TO 26
  18 TO 19
2 OPENS
```

In-Circuit Tests

Once a board is tested for shorts, an in-circuit test is performed on the components. An mcon statement establishes the connections to the desired nodes. This statement specifies the circuit-board node numbers and the patch-panel row and column numbers. The test is then conducted with either a pcomp statement for passive components, or an scomp statement for semiconductors. In general, the only required information given to the pcomp or scomp statement is the type, value, and tolerance of the component to be tested. All other setup data can be defaulted and BTL will choose a consistent set of setup conditions for the test. If necessary because of circuit restraints, any or all of the setup parameters may be overwritten.

Some representative statements are:

Statement	Meaning
pcomp "R17",4700,5,5,"en"	resistor R17, 4.7k, ±5%, use accuracy enhancement;
pcomp "C123", 1e−6,80,20	capacitor C123, 1μF, +80%, −20%;
scomp "N97", 20	NPN transistor N97, minimum beta 20.

If a component fails, a diagnostic message is automatically generated identifying the type of test, the program line number, the measured value, and the test limits. For example, the following two lines were in the test program:

```
119: mcon 12.1102, 25.1224
120: pcomp "R17", 4700,5,5
```

If the component failed, the following message would be generated:

```
BAD COMP
LINE   120
MEAS   4256.84
LMT    4936.00
TO     4465.00
R17
```

Functional Analog Tests

With assurance that all the component values are correct and that no shorts or opens exist on the board, the operator may next request the system to apply power to the board and perform functional tests to find out if the board actually functions as expected. The operator may also make some initial adjustments to speed final assembly and checkout. The following BTL statement is typical of those used to apply power:

```
cps; sps 1,5,2,12.
```

This says, connect the power supplies (at zero volts) and then set power supply 1 to 5 volts and power supply 2 to 12 volts.

The workhorse statement for analog tests is the trans statement, for transfer test. In general, this statement applies one of the analog sources to a selected node and connects one of the detectors to another node. The general form of this statement is:

```
trans⟨failure message⟩,⟨source setup⟩,⟨detector
      setup⟩,⟨result field⟩.
```

The "failure message" defines a diagnostic statement to be included in the message that will be presented if the test fails.

A simple example is shown in Fig. 3. Here, it is desired to test the amplifier circuit by applying a small dc voltage to the input and test for a nominal gain of 10 at the output. The transfer test statement to do this is:

```
trans "Replace IC10", "DC", .1, .01, "3455," 1, "DC", 0, 1.1, .9
```

The parameters following the failure message in the statement give the instructions: select a dc source of 0.1 volt and set a current compliance limit of 0.01A; use the 3455A DVM as the detector on its 1-volt dc range with no waiting; the test limits are 1.1V and 0.9V. Execution of the statement automatically invokes all switching, chooses the right source and detector and sets them to the correct ranges, allows time for settling, then triggers the detector and compares the reading to the test limits. If this test should fail, the following message would be presented:

```
FAULTY TRNS TST
      MEAS   .77
      LIMIT  1.1
      TO     .9
      REPLACE IC10
```

Testing the Tester

Since a circuit-board test system is an integral part of a production pipeline, extended downtime of the test system can bring the entire production process to a halt. Clearly, then, preventive maintenance is a necessity.

Preventive maintenance of the Model 3060A Board Test System is implemented by a confirmation program that verifies that all system functions are operating and within the confirmation specifications. This takes only a few minutes to run, and it was designed to be run by operators that have limited knowledge of the system, so it may be used frequently.

If the confirmation program determines that there is a fault in the system, diagnostic programs can be called to quickly isolate the fault to a particular sub-unit of the system, and sometimes to a particular circuit board or a relay.

The overall system design of the Model 3060A placed high emphasis on serviceability. Functional partitioning into sub-units enhances serviceability by allowing individual diagnoses of the sub-units, most of which have their own microprocessor-controlled diagnostic routines. These may be initiated by the system controller, or manually with the aid of the front-panel keyboard and display. For example, the user can check the performance of the Analog Stimulus/Response Unit (ASRU) without involving the scanner by connecting a component to the ASRU's front terminals and entering the test to be performed on the front-panel keyboard. The result is then read on the ASRU's digital displays. Also, system cabling enables individual sub-units to be isolated from the rest of the system, thereby simplifying fault isolation by removing system interactions.

Certain stages of the performance tests, however, require interconnections between the various sources and detectors. These are provided by special test fixtures that plug on to the scanner in place of the dedicated patch-panel/bed-of-nails fixtures. These route the test signals into appropriate paths and also provide stimuli that normally may not be available. Two test fixtures are provided with the system, each working with particular sections of the confirmation/diagnostic programs.

The confirmation/diagnostic software is recorded on a single flexible disc. It consists of three main programs: (1) a configuration program; (2) the confirmation program; and (3) a diagnostic program.

The configuration program generates a directory of system hardware that is subsequently used by the confirmation program. It interrogates the hardware by way of the HP Interface Bus and calls for certain operator responses. In most cases, the only keyboard response required of the operator is either a yes or no followed by the CONTINUE key. Once it has been done, this program does not need to be run again unless hardware configuration changes are made, such as increasing the number of test nodes.

The confirmation program starts by allowing the 3455A DVM to test itself using its own internal verification routines. This is the only instrument needed for checking the scanner, which is tested in the next step. The scanner test is then followed by tests of the other sub-units in a sequence that is designed to eliminate most system measurement interactions.

The confirmation program was designed to execute as completely as possible without termination. When the program finds a fault, an error message is printed and if the fault will not interfere with other tests, the program continues. Otherwise, messages are printed indicating either program termination or those portions of the program that were skipped as it went on.

An abbreviated version of the confirmation program that requires no operator interaction is also provided to encourage frequent use.

If a fault cannot be corrected with the information provided by the confirmation messages, then the diagnostic program is called. Each of the program's stand-alone test modules is selected from a menu by execution of a single key, and may be selected in any order. This program enables a fault to be isolated to a removable subassembly.

With these tools for maintaining and verifying system performance, the considerable advantages of automatic testing need not be lost because of excessive downtime.

-Roland H. Burger
-John J. Ketchum
-Scott E. Woodard
-James M. Brown

Functional Digital Test

The BTL software supports static digital pattern tests by providing statements to make it easy to apply patterns, detect the results, and do the associated support functions.

The digital drivers require two levels to be set: one for the low level and one for the high level (within ±16V). The receivers are threshold devices that need only one level programmed. As an example, to set driver reference 1 to TTL levels of 0 and 5 volts and receiver reference 1 to 1.2 volts, the statement is:

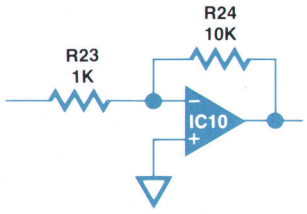


Fig. 3. *Circuit used as an example for the transfer test described in the text.*

```
dvrref 1,0,5; rcvref 1,1.2.
```

Two independent sets of driver and receiver references may be programmed, enabling the testing of boards that have mixed logic families.

An arbitrary set of driver or receiver nodes that the programmer chooses to manipulate collectively may be organized into groups, allocated by group statements. For example, to establish a driver group of three bits named "input" connected to reference 1, the statement would be:

```
group "input", "D", 3,1
```

Next, the desired nodes are mapped into the group by an assign statement. With the assignment done, digital patterns may be applied to the circuit by using the apply statement. To apply the pattern 110 to the group above, the statement is:

```
apply "input", "110"
```

The BTL statement that looks at the response of the circuit being tested is receive. Except for the addition of a failure message, the receive statement is similar to the apply state-

ment.

To simplify programming, the decimal equivalent of a bit pattern may be entered. For example, an 8-bit receiver group named "trigger" may be tested for the desired binary pattern equivalent to decimal 240 as follows:

```
receive "REPLACE U20", "trigger", 240
```

If the test fails, a message like the following would appear:

```
FAULTY PATTERN
LINE 245
11 ! !  00**
REPLACE U20
```

The short-hand notation for the failed bit pattern is interpreted as follows:

1—1 expected and read correctly;
!—1 expected but 0 is read—failing 1;
0—0 expected and read correctly;
*—0 expected but 1 is read—failing 0.

Functional Tests with Signature Analysis

An option to the 3060A Board Test System gives the system the hardware capability for taking "signatures" from the device under test. These signatures are identical to those that would be obtained by the HP Model 5004A Signature Analyzer.[2] Especially useful for testing microprocessor-based systems, this provides confirmation that the unit under test is functioning correctly.

The option provides 40 low-capacitance inputs for probing the board under test. In addition, all of the analog nodes may be used for taking signatures. Like the rest of the digital hardware, the threshold levels are programmable between −16V and +16V. In addition, the test circuitry can run at clock rates up to 10 MHz.

There are two primary statements associated with the signature option, saset and sig. Saset selects the nodes for the start, stop, and clock inputs and the edge that each is to use. Sig generates and tests the signature. A sig statement has the form:

```
sig "U68 pin 9", "9HC1".
```

Execution of this statement initiates a start-stop cycle during which a digital pattern generated by the board under test is used as a stimulus and the digital pattern appearing at the tested node is condensed to a 4-digit hexadecimal number in the signature analysis circuitry by a well-defined polynomial relationship. The resultant signature is compared to the expected signature supplied by the sig statement, in this case, 9HC1. If the test fails, an error message indicating the location of the faulty node is printed.

Acknowledgments

IPG was implemented entirely by Roger Khanna. Thanks are also due Roger for his work on BTL, especially the transfer and component test statements.

References

1. D.E. Morris, C.J. Christopher, G.W. Chance, and D.B. Barney, "Third Generation Programmable Calculator has Computer-Like Capabilities," Hewlett-Packard Journal, June 1976.
2. R.A. Frohwerk, "Signature Analysis: A New Digital Field Service Method," Hewlett-Packard Journal, May 1977.

Ed O. Schlotzhauer

Born in Tyler, Texas, Ed Schlotzhauer is a 1971 BSEE graduate of the University of Houston. Ed was project manager for the 3060A Board Test System software and designed the firmware for the HP-IB Interface Card. Before joining HP in 1973, he worked as R&D engineer for an aerospace firm. A resident of Loveland, Colorado, Ed is married, has one daughter, and spends much of his leisure time doing church work, running, camping, hiking, and skiing.

Hardware Design of an Automatic Circuit Board Tester

by David T. Crook, Brian M. Wood, Francis L. Fiedler, Kamran Firooz, and Roland H. Burger

THE MODEL 3060A BOARD TEST SYSTEM was designed and packaged as a complete measurement system. Yet, it was partitioned functionally into various subunits to allow manual operation for experimenting with test parameters and to assist in troubleshooting of individual functions. Even so, during normal operations all testing operations are handled by a few pushbuttons on the control panel mounted on the scanner (Fig. 1).

The interface between the circuit board under test and the board test system is a bed-of-nails fixture. Kits supplied with the system enable preparation of a bed-of-nails fixture for each type of circuit board. Flexible leads connected to the contact pins in the bed-of-nails panel are inserted into appropriate holes in a patch panel according to the fixture map generated by the in-circuit program generator described in the preceding article. When the assembled fixture is installed on the scanner, pins protruding from the

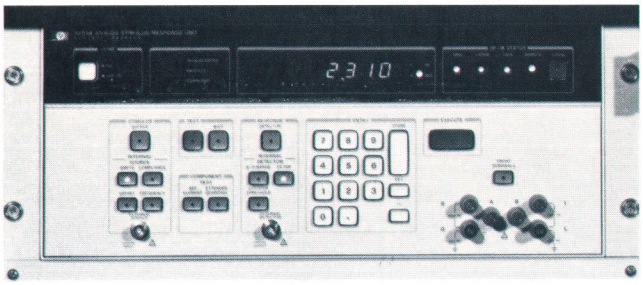


Fig. 1. *Front panel of analog stimulus/response unit enables manual control of test procedures. The digital displays show instrument status and also provide diagnostic messages during troubleshooting.*

patch panel make contact to paddle contacts within the scanner. The fixture is attached with a cam latching operation that generates the necessary wiping action and contact pressure to assure low-resistance connections.

The Scanner

The scanner provides the switching between the measurement points on the board under test and the stimulus/response measurement functions. The switching is performed by mercury-wetted reed relays that are mounted on plug-in circuit-board assemblies.

At present, three types of relay board assemblies are available for the stimulus/response interconnections. The first type is the general-purpose analog board assembly. Each of these has 64 relays arranged in two columns of 32 each. All the 32 relays in each column connect to a common bus and the bus connects through other relays to any of the seven system buses (S, I, G, A, B, L, and D). All the buses, except the D bus which is used to transfer data internally between cards, are routed to triaxial connectors on the rear of the scanner, where cables carry the bus signals to the ASRU (analog stimulus/response unit).

The second type of board is the digital board, a dedicated board designed for use with the DSRU (digital stimulus/response unit). It also has 64 relays arranged in two columns of 32 relays each. However, these relays do not connect to buses but connect directly to the DSRU through a multiconductor cable; the relays in one column are dedicated to the DSRU driver outputs and those in the other column, which also have buffers to assure less than 10-pF load capacitance at each paddle contact, are dedicated to the DSRU receivers. The driver output relays may also be used as general-purpose relays.

The third type of board is dedicated to signature analysis. Known as the Scanner S.A. board, it contains the high-speed portions of the signature-analysis circuitry with the remaining circuitry located in the DSRU. The Scanner S.A. board has the logic and relays for switching any of 40 pins in the corresponding patch-panel column to the signature analyzer input circuit, and for selecting any one of four pins for each of the start, stop, and clock inputs. The D bus is included on this board so signals for signature analysis can be routed through from any analog board output.

The signal paths from the bed-of-nails contacts to the relays are on flying leads with Teflon insulation, and the relays are specially designed to maintain high circuit-to-ground resistance. Short lead lengths and small loading capacitances assure repeatable results over a wide range of operating conditions. On the analog board assembly, the S and I buses, which carry the measurement signals, are Teflon-insulated throughout to maintain a circuit-to-ground resistance of greater than 10^{12} ohms. The other buses go to the edge connector on conventional pc board traces and have a circuit-to-ground resistance of greater than 10^{7} ohms.

User convenience rated a high priority in the design of the scanner. The bed-of-nails fixture and the patch panel are easily removed as a unit to be replaced by another unit for testing a different board. The scanner tilts up to allow quick removal and insertion of the relay boards. Air-damped shock absorbers support the scanner in the tilted-up position, and let it fall gently when it is lowered.

The system control panel, although included as part of the scanner, functions independently of the scanner. Control panel and scanner share power supplies, a microprocessor, firmware, and the HP-IB connector, but have separate HP-IB addresses.

The Analog-Stimulus/Response Unit

The ASRU is organized into two major sub-sections: source and detector, as shown in Fig. 3. Each section is electrically isolated from the other and from ground, and is independently guarded. Optoisolators couple measurement and control signals into and out of the guards.

The source section supplies both dc and ac stimuli. The selected stimulus drives an output amplifier that may be configured as either a constant-voltage or a voltage-controlled, constant-current source with programmable current or voltage compliance limits.

Dc test voltages originate in a reference supply controlled by a low-temperature-coefficient reference diode. The selected dc reference (±5V) passes through a 14-bit multiplying digital-to-analog converter (MDAC), that functions as a programmable attenuator, and then through the output amplifier, which has the necessary gain to give the ±14.2V output range.

The 100-Hz, 1-kHz, and 10-kHz sine waves originate in three Wien-bridge oscillators. These operate at fixed frequencies using low-temperature-coefficient elements to obtain 0.1% frequency accuracy and stable 5V rms output amplitude. When the output of one of these is selected, it

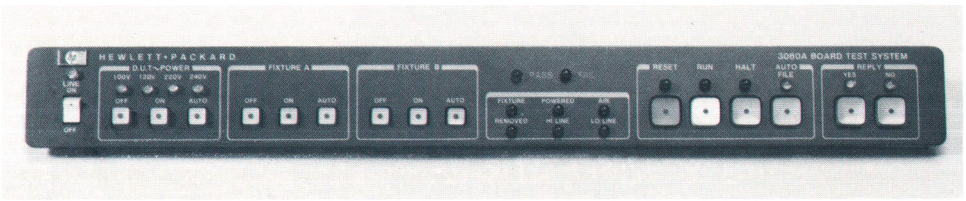


Fig. 2. *Board test system control panel, adjacent to the test fixture, has pushbuttons that handle all the testing operations, and indicators that show system status.*

passes through the same MDAC and amplifier chain as the dc reference.

Sine, square, and triangle waves, tunable over a range of 10 Hz to 100 kHz, are supplied by a function generator. It is a straightforward current-integration type using current sources to charge and discharge a capacitor between fixed voltage limits. Square and triangle waves are an inherent product of this circuit, while the sine waves are formed by clipping the triangle at different levels in two parallel paths and summing the resulting trapezoids with the triangle wave. This technique has an advantage over other shaping methods in that the clipping levels do not require critical adjustment to achieve low distortion levels.

The function generator's output frequency is determined within each of the four decade ranges by a control voltage supplied by the same voltage reference and MDAC that are used for supplying the dc stimulus. The output amplitude is set by an 8-bit MDAC that attenuates the function generator output.

All the ac waveforms may be programmed with dc offsets between −12.8V and +12.7V. The offsets originate in the +5V reference, are processed through an 8-bit DAC, and are applied to the summing node at the input of the output amplifier.

Up to four external sources may also be used. These connect to connectors on the ASRU, one on the front panel, three on the rear. Relays within the ASRU connect them to the device under test as required by the test program.

Analog Detectors

For making dc measurements, the detector section has a dual-slope voltage-to-time converter and a time-interval counter. A preamplifier with gains of ×1, ×10, and ×100 gives a range of 1 μV to 15 V. The dual-slope converter normally operates with 4½-digit resolution and a nominal accuracy of 0.2%, but it also has a 5½-digit, high-resolution mode with nominal 0.02% accuracy and better than 40-dB normal-mode rejection, though with slower response.

The ac detector is essentially a switch that reverses the polarity of the input signal every half cycle before feeding it to the dual-slope converter, thereby enabling the average value of ac waveforms to be measured with the same resolution and measurement range as dc voltages. To measure the in-phase component of the signal, the dual-slope converter starts run-up in synchronism with the zero-crossing of the source. To measure the quadrature component, it starts with a precise, digitally-derived delay equivalent to 90°. Because of the precision desired for the delay, this detection technique can be used only with the three fixed-frequency, sine-wave sources.

Also included as part of the detection system is a reciprocal-taking counter that measures frequencies up to 5 MHz. It can operate with either ac coupling, or with dc coupling that allows use of a threshold level programmable over a range of −12.8V to +12.7V. It can also measure pulse widths ranging from 300 ns to 1 second.

The HP Model 3455A Digital Voltmeter, which gives 6½-digit dc resolution or 5½-digit ac resolution with true rms response, and the 3437A Digital Voltmeter, which gives sampled voltage measurements at rates up to 4000/s, connect to the device under test through programmable relays within the ASRU. One or the other of these voltmeters normally serves as the detector for measurements in which the function generator serves as the source. If needed, up to four additional detectors may connect to the system through program-controlled ASRU relays.

The detector section also houses the measuring operational amplifier that is at the heart of the in-circuit component measurements (see article beginning on page 19).

Microprocessor Control

All of the ASRU functions except the HP-IB interface are controlled by a 6800 microprocessor using 20K bytes of ROM. The HP-IB interface is controlled by a fast, HP-developed, 8-bit microprocessor that is optimized for control functions. It communicates with the 6800 via a direct-

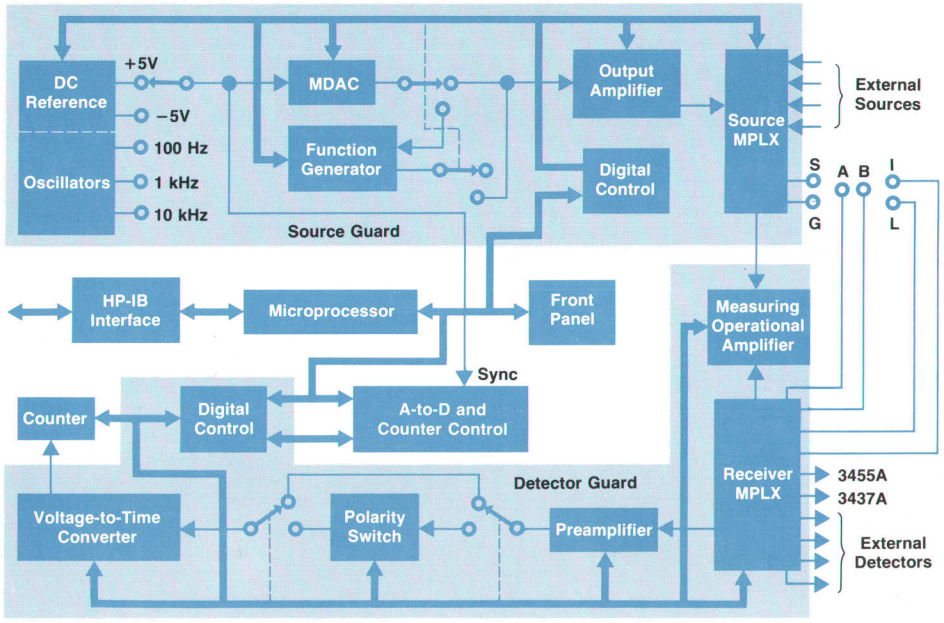


Fig. 3. *Block diagram of analog stimulus/response unit. Dc and ac stimuli are generated within one guarded section and measurement results are detected in an independently guarded section. Both sections are under control of a microprocessor.*

memory access (DMA) channel to speed data transfer between the HP-IB and the 6800.

The front-panel has its own hardware for scanning a local RAM and updating the displays and annunciators at a flicker-free rate, relieving the 6800 of this overhead. Whenever a change in the displayed information is necessary, the 6800 merely writes data to a block of 16 RAM locations.

The microprocessor sets up the instrument internal configurations for the various measurements, controls the dual-slope converter and the counters that obtain measurement results, and converts the measurement quantities into the desired units, such as converting waveform period to frequency. It also averages readings to improve accuracy.

The microprocessor firmware includes extensive diagnostic routines. It also includes an Autocal routine that is executed automatically at turn-on and whenever the test program calls for it. Major sections of the Autocal routine include:

- Measurement of the fixed oscillators' frequencies, necessary not only for the accuracy enhancements but also for adjusting the run-up time of the dual-slope converter so it equals an integral number of cycles of the waveform during ac measurements.
- Measurement of the gains and offsets for each range of the detector systems. These are stored in RAM and applied to the raw readings.
- Measurement of the source and measuring op amp offsets, which are applied to DACs to minimize the offsets.
- Measurement of the reference resistor values using a precision wirewound 10kΩ resistor as a standard, to derive correction factors that are stored in RAM.

Each section builds upon the previous one. For example, the reference resistors are measured after the measuring op amp offset has been zeroed and the detectors that make the measurement have been calibrated. The basic accuracy and repeatability of measurements made with the Model 3060A Board Tester are thus a function of two basic dimensions:

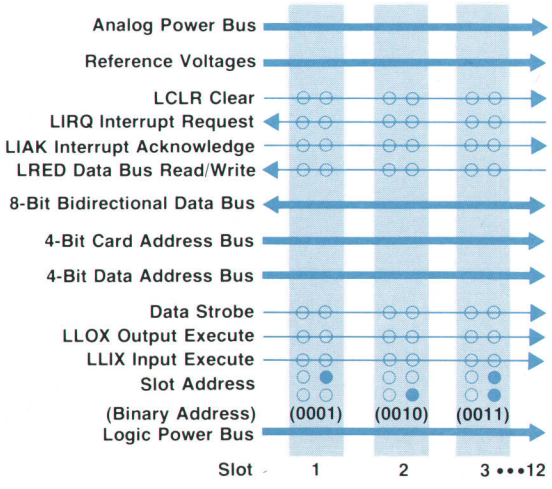


Fig. 4. *Back-panel wiring in the digital stimulus/response unit is identical for the 12 available slots, enabling any digital test card to be plugged into any slot.*

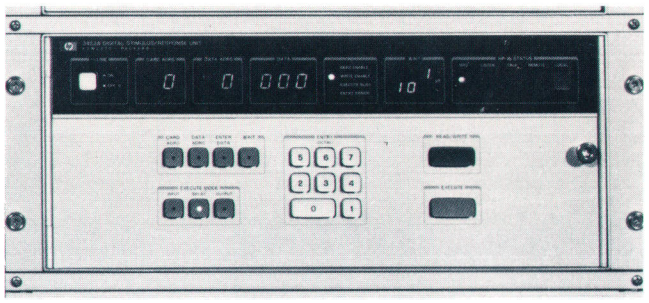


Fig. 5. *Front panel of the digital stimulus/response unit.*

time and resistance. A quartz-crystal oscillator provides the reference for measuring frequency, and a wire-wound resistor provides a fundamental impedance reference for the dc measurements, which are basically ratiometric.

The Digital Stimulus/Response Unit

The DSRU supplies high and low logic levels to specified nodes on the circuit board under test and senses the responses at other nodes. To provide the flexibility needed for a versatile circuit-board test system, it is constructed as a general-purpose card cage with the front panel hinged to allow access to the fifteen card slots.

Three of the card slots are dedicated to mainframe operations—one for the microprocessor and its 2K bytes of firmware, one for the logic-level reference voltages, and one for a board containing optoisolators that enable the digital-circuit ground to be isolated from the chassis and HP-IB grounds. The remaining 12 slots are available for installation of various cards that may be required by the test program. These include cards that have digital drivers and receivers, a signature-analysis card, and a clock generator card. Also included in this unit is a card for programming the four system power supplies that power the circuit board under test.

To allow any of these cards to be inserted in any slot, the motherboard wiring is identical for the 12 slots, as shown in Fig. 4. Four hardwired pins at each slot position determine a slot address. The hardwired code is compared on each card to codes that are sent on the four card address lines to determine whether that card is to respond to information on the 4-line data address bus. The data address bus allows selection of addresses on the selected card for storage or retrieval of data.

The six reference voltages that establish two sets of high and low logic levels and detector threshold levels originate in one of the mainframe cards. These are individually programmable over a range of ±5V and are amplified on the cards where they are used to give a range of ±16V.

Digital Source

Data representing the desired states of the output drivers is sent over the data bus in 8-bit bytes and stored in flip-flops at the indicated addresses on the selected card. When an execute command is given, all the stored data is transferred to a second rank of storage, the output of which sets the position of CMOS SPDT switches. One position of these switches connects the corresponding output stage to the logic-high reference level and the other connects it to logic

Board Testing with Signature Analysis

Circuit boards that have microprocessors and other algorithmic state machines are presenting severe challenges to test engineers because of the extreme difficulty of generating effective test patterns and isolating and identifying faults. This difficulty is primarily the result of the dynamic nature of many LSI devices and the presence of several components communicating on a common bus.

Signature analysis,[1] when properly designed into a circuit board, provides a way of finding the causes of errors in processor-based circuits relatively quickly. It has a further advantage in that it allows the system under test to run at its normal clock rate (up to 10 MHz), thus assuring that dynamic memory elements are kept "alive." Operation at normal clock rates also enables timing errors to be detected.

To use signature analysis for testing a circuit board, provision should be made for opening feedback paths and for supplying signals for starting and stopping the signature-analysis circuitry precisely with respect to the test pattern. During a test, the signal analysis circuitry accepts the long string of 1's and 0's appearing at a test node, processes it, and displays the result as a four-digit hexadecimal number. This number, unique for the digital pattern appearing at that node, is the node's signature.

In applying signature analysis to a production test system, the test engineer uses a known good board to gather and document signatures for all the pertinent nodes on the board. During production test of a board, the test system compares the signature at each node to the known good value and indicates a fault if there is any difference. Backtracing to a node that has a correct signature then identifies the location of the fault.

The signatures obtained by the Model 3060A Board Test System equipped with the signature analysis option are identical to those obtained with the portable Model 5004A Signature Analyzer.[2] The signatures may thus be used for troubleshooting in the field (provided that the test patterns have been stored in a small section of memory on the board). Signature analysis thus contributes to rapid field service as well as production test.

The signature analysis option of the Model 3060A has a multiplexer that can select any of 40 nodes for signature analysis. In addition, it can accept inputs supplied through the analog channels, giving a maximum of 1064 nodes that can be tested by signature analysis. It also has multiplexers for selecting one each of four clock signals, four start signals, and four stop signals. In addition, it has a count-down mode that stops the measurement a selected number of clock pulses following the start pulse, useful where a stop signal is not readily available.

Signature analysis is applicable to those portions of the circuit board that operate synchronously with the board's clock, such as the processor, memory, peripheral chips, latches, and so on. The asynchronous parts, such as the interrupt system, may be checked by the static digital tests that are standard on the Model 3060A.

–Kamran Firooz

References

1. R.A. Frohwerk, "Signature Analysis: A New Digital Field Service Method," Hewlett-Packard Journal, May 1977.
2. A.Y. Chan, "Easy-to-Use Signature Analyzer Accurately Troubleshootls Complex Logic Circuits," Hewlett-Packard Journal, May 1977.

low. The selected states of all the active output stages are then held until a new pattern is applied or until the program terminates the digital portion of the test.

The receiver circuits consist of comparators that compare the incoming signals to the threshold level. The resulting 1's and 0's are clocked into 3-state registers where the data can be read out in 8-bit bytes.

Driver outputs are protected against excessive externally supplied voltages. Should an excessive voltage occur, one of two zener diodes will turn on and current through the diode will clamp the output and develop a voltage across a series resistor. If this voltage is greater than a reference voltage, a signal notifies the scanner to open all relays while sending a service-request signal to the DSRU's microprocessor. Back-up protection is provided by slow-blow fuses.

The receiver inputs are protected by a series current-limiting resistor and ±17V clamp diodes. A resistor to ground assures that unused inputs will be determinate.

Test Pattern Execution

Pulses that latch the output and receiver data are sent on the LLOX and LLIX lines, LLOX for output execute and LLIX for input execute. The programmer specifies a delay between the output of driver data and the latching of receiver data ranging from 1 μs to 1s in decade steps. Alternatively, the LLOX and LLIX signals may be individually programmed, allowing the user to program software delays in 1-ms steps.

When two DSRUs are installed in the system, the LLOX and LLIX lines may be synchronized through a separate sync cable.

Any card can request a priority interrupt of the DSRU's microprocessor—in case of a driver over-voltage, for example—by pulling the IRQ line low. The microprocessor then institutes a poll, starting with card address 0. As it checks each card, it reads bit 0 of data address 5, which is the on-card status register. When it finds a 1 there, it sends out an IAK pulse, to clear the IRQ flip-flop on that card, then places the address of that card in the lower four bits of the serial poll status byte and sends it to the system controller.

Like the ASRU, the DSRU may be operated manually from the front panel (Fig. 5).

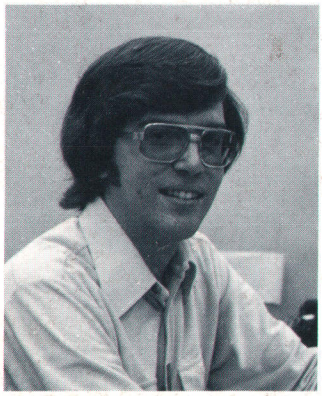
Primary Power Control

The system power module distributes ac power to the subunits through pre-assigned receptacles and it earth-grounds the subunits at a common node to prevent ground loops. At turn on, it applies power to the system only when the cooling air flow and primary power line conditions are correct. It continuously monitors the air flow and the primary line, for high and low voltage levels and cycle drop-outs, and it shuts down the system whenever improper conditions occur. However, it continues to power the indicators that show what went wrong.

Acknowledgments

The Model 3060A Board Test System was largely defined by Art Minich, who was the original project manager. In addition to contributions made by those mentioned in other articles in this issue, contributions were made by Harlan

Talley to the scanner hardware and software, by Dave Bender to the mechanical designs of the scanner and the power module, by Joe Gorin to the analog detector hardware, by Bob Illick to the ASRU diagnostic firmware, by Barry Taylor to the systems and ASRU product designs and also to a system thermal design that limits temperature rise to a low level regardless of system configuration, by Jim Jones to the product design of the ASRU and DSRU, and by Sandy Charney to the fixturing. Jerry Nelson, as section leader, provided the leadership necessary to bring the project to completion.



Brian M. Wood

Brian Wood joined HP in 1973 upon receiving his BSEE degree from the University of Arizona. He designed the logic sections of the digital and analog stimulus/response units of the 3060A Board Test System and is now production engineer for the 3060A. Born in Washington, D.C., Brian is married and lives in Loveland, Colorado. He spends much of his spare time downhill skiing, playing handball, and making long distance amateur radio communications.

Francis L. Fiedler

Born in Carroll, Iowa, Francis Fiedler is a 1968 BSEE graduate of the University of Denver and a 1972 MSEE graduate of Colorado State University. Francis designed the source section of the ASRU and contributed to the design of the 3455A and 3465A DVMs, the 3581A Wave Analyzer, and the 3320A/B/C Frequency Synthesizers. A resident of Loveland, Colorado, Francis is married, has three children, and enjoys flying single engine planes, woodworking, hiking, skiing, and fishing.

Roland H. Burger

Roland Burger joined HP Palo Alto as a test technician in 1967 and transferred to the Loveland Instrument Division in 1969 where he worked on various ATE. With the 3060A from the beginning, he designed the power module and control panel and contributed to the diagnostics software. He has done course work at Heald Engineering and Foothill Colleges in California. Roland lives with his two sons (16 and 17), keeps in shape weightlifting, and relaxes with various crafts, including woodworking.

Kamran Firooz

A native of Iran, Kamran Firooz is a 1972 BSEE graduate of Louisiana State University and a 1974 MSEE graduate of Ohio State University. With HP since 1974, Kamran contributed to the design of the digital section of the 3060A Board Test System and worked as R&D project manager for the 3060A. A resident of Loveland, Colorado, Kamran is married, has two children, and enjoys woodworking and skiing in his spare time.

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

HEWLETT-PACKARD JOURNAL

MARCH 1979 Volume 30 • Number 3

Technical Information from the Laboratories of Hewlett-Packard Company

Hewlett-Packard Company, 1501 Page Mill Road
Palo Alto, California 94304 U.S.A.
Hewlett-Packard Central Mailing Department
Van Heuven Goedhartlaan 121
Amstelveen-1134 The Netherlands
Yokogawa-Hewlett-Packard Ltd., Suginami-Ku
Tokyo 168 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

HP Archive

On-line curator: John Miles, KE5FX

jmiles@pop.net