VLSI test system grows in pin count and functionality

New pin card supports up to 256 pins and all common logic families; pattern processor mixes stored and algorithmic sources for functional checks

by Alan Whiteside, *Tektronix Inc., Beaverton, Ore.*

□ Very large-scale integrated circuits, with their increased pin counts and improved timing parameters, demand increased versatility and precision from a test system. To ensure high quality, a VLSI test system must be able to accurately simulate a worst-case "real world" situation; to be cost-effective, it must do so for all types of devices. Further, it must reduce test times and program development time in spite of the mushrooming complexity of VLSI parts.

In response to these criteria, the 20/40-megahertz S-3295 VLSI test system (Fig.1) couples an expanded pin count and enhanced functional-test capability with the proven architecture of the S-3200 series. Its 256 data channels, twice as many as previous S-3200 systems, are configured as 128 input and 128 output pins, any or all of which may be connected in pairs as I/O pins.

Functional testing is controlled by a new high-speed pattern processor that can compress and recreate functional patterns at the time of a test run, as well as switch to other pattern sources on a cycle-by-cycle basis. These features are complemented by the voltage-, current-, and time-measurement facilities of the S-3200 series.

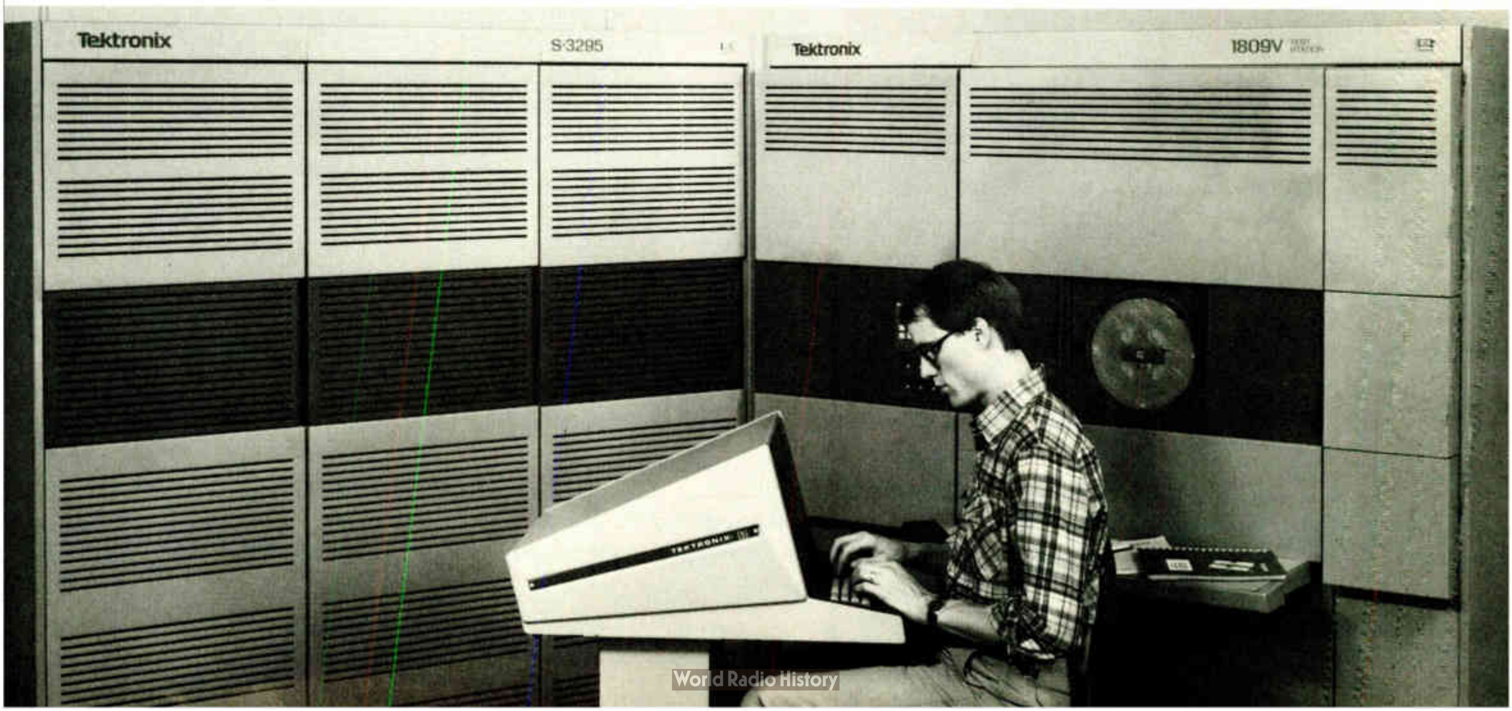
The S-3295's pattern processor is able to compress pattern data by removing repetitive lines and sections of a pattern. This step reduces storage requirements, while preserving the integrity of the pattern. The deleted portions are restored at test time by loop iteration and subroutine calls that are part of the instruction set the pattern processor generates when compressing pattern data. The processor also can store repetitive sections of a test as a single subroutine.

At test time, the pattern processor calls instructions up from a 4-K-by-96-bit emitter-coupled-logic random-access memory. As the test proceeds, the current pattern-processor instruction evaluates loop counters, index registers, and interrupts to derive the address of the next instruction and its associated pattern vector.

Program flow is set by if-then-else statements. For example, the if statement can be programmed to test for an error condition during a microprocessor test. The then statement would initiate a branch to a diagnostic routine if an error were detected, and the else statement would specify to continue standard testing. The addresses of the branches involved are stored in the pattern processor's 16 index registers.

The pattern processor controls and feeds the pin elec-

1. Family growth. The S-3295 VLSI test system above, based on the S-3200 family architecture, uses a new pin card and pattern processor that increases the system's ability to check the functional performance of 20-MHz very large-scale integrated circuits with up to 256 pins.



tronics cards from which each active pin in the device under test receives its test vectors. Each pin card provides four channels of formatted data—two input and two output channels—which can be combined to form I/O channels. For a group of two to four DUT pins, one pin card provides a complete operating environment; not only does it provide the usual formatted input signals, but it also handles I/O switching, output loading, and recording error as well.

Every pin-card data channel is serviced by 64 K of local memory for a total of 256 K per card. Because storage needs vary with differing device types, the local memory has a variety of partitioning options, such as the memory structure in Fig. 2 for one of the two I/O channel pair Ordinarily, the S-3295's pattern processor is used as the pattern-data source—it reconstructs force, inhibit, compare, and mask data that has been compressed and stored in its memory. The pin electronics card supports each of these four data functions with 32 K of serially addressed RAM. In effect, a 32-K-by-4-bit shift register resides behind each channel pair.

Partitioning memory

For long nonrepetitive patterns, though, a deep serial memory is needed to feed the pattern to the DUT without frequent wait states. Therefore the test system allows the user to increase the memory backing up a particular combination of functions or to specify 32-K increments for error storage. The full 256-K of local memory can even be used for a single function (providing either force or compare data, or recording input or error data) if the DUT demands it.

Another important aspect of the pin electronics card is its driver circuitry. The predominant logic technologies—MOS, TTL, and emitter-coupled logic—have widely differing terminal characteristics and input drive parameters. Input currents, amplitudes, offsets, and rise and fall times all play important parts in simulating a real-world stimulus environment for the DUT.

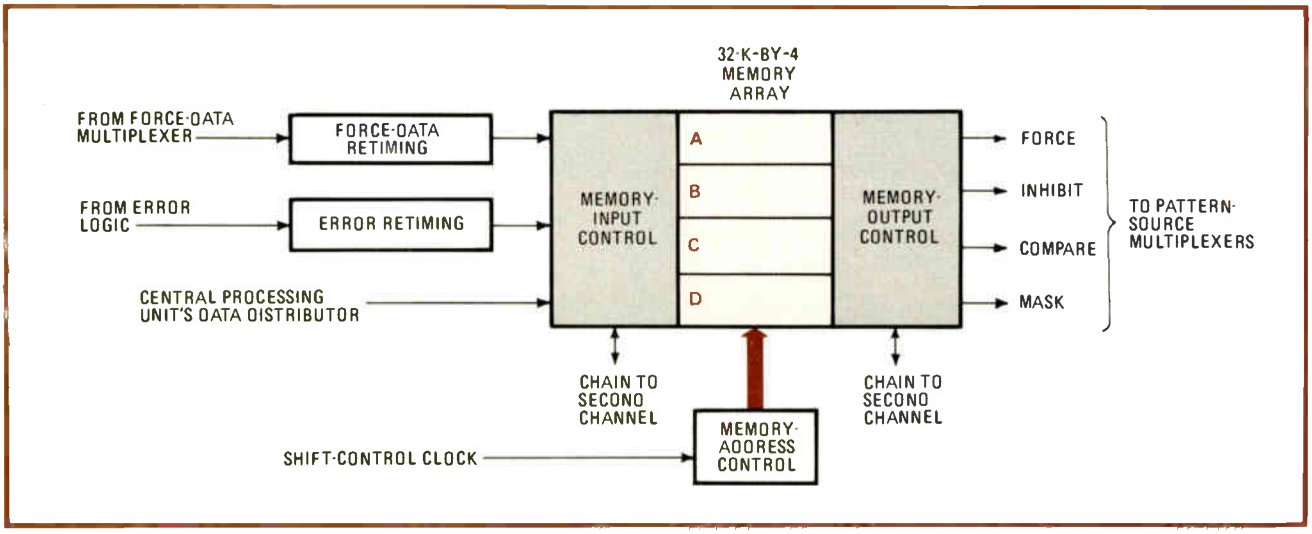
Meeting these divergent test requirements with a single driver design has been an elusive goal challenging designers of automatic test equipment. The S-3295 addresses the problem with a new driver circuit that features a programmable slew rate, low-leakage inhibit, and a wide output-voltage range.

The programmable slew rate allows the test engineer to vary the slope of a signal edge over a 5:1 range. Figure 3 is a multiple-exposure photograph showing the result of setting the driver's slew rate at three different points within its programmable range. At low amplitudes the slew rate is approximately 1 volt per nanosecond, allowing ECL rise times as low as 800 picoseconds. On the other hand, the driver is equally comfortable meeting the much slower rise-time tolerances of the TTL and MOS families. Similarly, the S-3295 driver's amplitude can be accurately programmed, in 366-millivolt increments, over a −5 to +12 V range, which encompasses all of the prevalent logic families.

Drive off and on

To test a multiplexed part where the pins can accept data from bus lines or send data to them depending on the activities being performed, the driver and comparator are connected to the device's I/O pins. When the DUT is running in the output portion (the comparison frame) of its cycle or when it is entering a high-impedance floating state, the driver that forced data during the input portion must be inhibited—in effect, disconnected. Without the inhibition, spurious interaction between driver and test pin could affect the ability of the DUT to dominate the I/O bus.

The S-3295's pin-driver circuit utilizes high-performance Schottky diodes, with their characteristic low capacitance and low storage time, to implement the inhibit function. When the driver is inhibited, the DUT pin sees the extremely high impedance of the reverse-biased Schottky diodes, rather than the 50-ohm impedance of the driver's output transistors. Current leakage through the diodes is minuscule—only 50 nanoamperes at ECL levels—and the capacitance is about 2 picofarads, an order-of-magnitude improvement over existing equipment. Consequently, undesirable loading of the DUT pin



2. Memory flexibility. Local memory on the pin card, shown in schematic form for one pair of input and output channels, is usually formatted into four 32-K-by-1-bit blocks With chaining, however, up to 256 K can be put behind one pin for exceptionally long patterns.

is negligible. The compliance voltage of the inhibited driver is V_H + 1.8 v and V_L – 1.8 v, which well exceeds the requirements of most logic devices.

The user has a choice of three inhibit timing phases per driver. A control bit derived from a pattern source determines the cycles in which inhibit will occur, while the inhibit phase positions the start and stop times within the cycle. A variety of formats can be stipulated under program control to model virtually any real-world force-inhibit sequence.

The test system designer fights a battle on two fronts: while wrestling with the increasing complexity of VLSI circuits, he must also face the concurrent challenge of escalating speeds. Today, a growing demand for test data rates in excess of 20 MHz exists. Fortunately, the fastest devices usually have fewer pins than the slower, more complex ones.

The S-3295 tackles high-speed (40-MHz) driver operation with a multiplexing scheme that joins the formatted data streams of two adjacent drivers on a pin card. The number of fully supported drivers is halved, but two driver pulses can be positioned within a single cycle. Thus the DUT is exercised at twice the usual data rate of the test system.

The driver's data streams are combined at the driver inputs (Fig. 4), as opposed to the less desirable wired-ORing of the driver outputs. By combining data streams at the input, the DUT sees the same signal-path characteristics from the source in either the standard or the double-pulse modes, since it is connected to the same driver output in both instances.

In the double-pulse mode, one driver per pin card becomes the signal driver, and the force formatter of the dormant driver becomes the subordinate—its signals are rerouted to serve the signal driver. Thus the signal driver receives formatted force and inhibit data from its usual source, as well as formatted force data from the other channel on the pin card.

On a cycle-by-cycle basis, the inhibit data bit from the subordinate channel determines when the signal driver switches between data streams. A side benefit of this multiplexing method is that the control driver can gate the two data streams into the signal driver to produce a final output pulse as little as 1 ns wide. The extremely narrow strobes needed to test edge-driven synchronous logic ICs can thus be produced on demand.

DUT loading

A thorough functional test requires that the DUT outputs drive a load approximately that of an operating logic gate's input. Almost any sort of load impedance can be modeled using common passive components, but such schemes are limited in flexibility—a specific passive load circuit is suitable only for one device family. What is needed is a load circuit whose impedance characteristics can be quickly adapted to the immediate test.

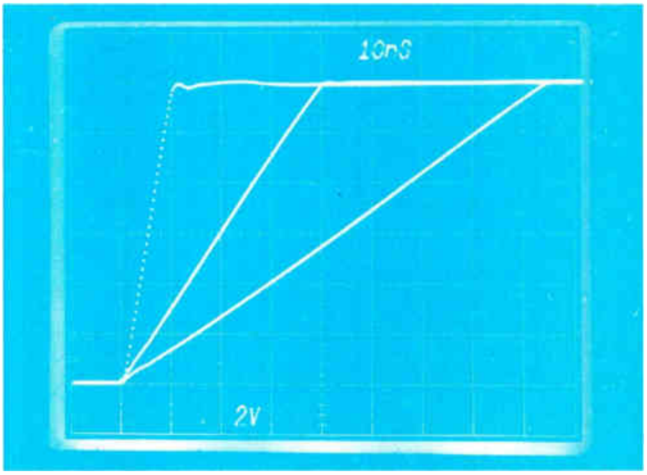
The S-3295 handles this problem with a fully programmable active load appended to each DUT output channel. Whereas passive load circuits operate at fixed source and sink current values, the active load offers independently programmable currents—50 milliamperes in either direction. The turnover threshold voltage, at which the circuit switches from source to sink, is also programmable. When connected, the active load dictates the amount of current that passes through the DUT pin-comparator measurement node. This characteristic permits the user to run device-wide, go/no-go parametric tests under real-world operating conditions.

For ECL devices, however, a passive load is mandated because they usually operate with a terminated transmission line, with the termination implemented by means of a 50-Ω resistor connected to a –2-v supply. This terminating resistor determines the output current drawn from the circuit driving the line. The S-3295's pin card makes available a 50-Ω resistor connected to a programmable power supply, allowing the DUT to experience the same current demands it would encounter in an operational circuit. When the ECL load is selected for a given DUT output pin, the active load for that pin is disconnected.

Checking the test results

The comparator signal path is essentially an analog environment, in that it must carry the DUT output signal to the comparator input without introducing distortions that might alter the outcome of the test. Because comparators do their job by detecting voltage thresholds and because rise-time degradation causes the threshold crossing to shift in time, the comparison path's rise time should be compatible with the rise times produced by the DUT. Therefore the signal path must have low stray capacitance and high impedance, as must any accurate measurement instrument. Furthermore, the comparator should have the versatility to handle the differing voltage ranges of the logic families in use and should be able to detect all modes of DUT operation—logic 1, logic 0, high-impedance state, and error.

For error detection, the S-3295 uses a buffered dual-threshold analog comparator interfaced with an error-logic circuit. The dual-threshold approach allows the user to define an analog "window" that encompasses the failure region for the 1 and 0 states and the pass region for the high-impedance three-state mode. The thresholds (both V_H and V_L) can be set up within either of two



3. Rising to the occasion. To test ECL parts, the pin card's driver electronics can be programmed to produce signals with rise times as fast as 1 V/ns, left in this triple-exposure photograph. The card can provide longer rise times for TTL (center) and MOS (right) parts.

ranges: ±5 V or ±15 V. The system provides for detection of logic-state errors, or three-state errors, or both, and a full-time mask mode (errors disabled) is also available. The actual error detection occurs only during the time specified by one of the comparison phases, of which eight are available to every comparator. The comparison path's rise time is 3 ns or less (1-V steps), easily compatible with most outputs. Special error-detection modes further enhance the tester's capability, making evaluation of device performance much simpler (see "Finding problems, not faults," opposite).

Deskewing all paths

All of the events occurring at the DUT are timed and synchronized by the system clock, which provides 16 phases distributed around the test station. Each driver has available to it eight timing phases, as does each comparator. Each inhibit circuit has three phases to select from. The per-pin phase selection is accomplished in the mainline test program.

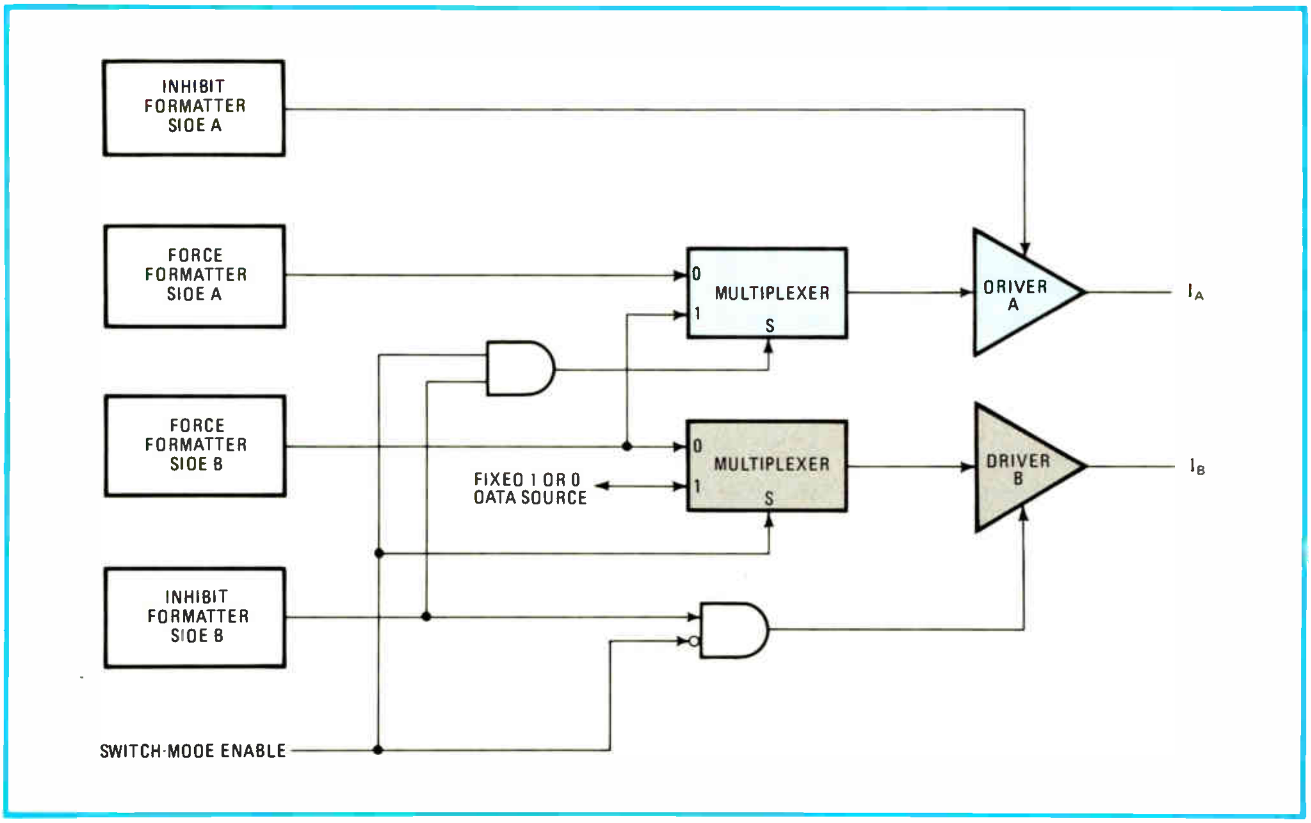
Timing accuracy and consistency are essential to meaningful testing of digital logic devices. A central aspect of the accuracy issue is the shift in time (skewing) of signals that are supposed to be coincident.

In the real world, the progation delays, cable delays, and distributed capacitance of two presumably identical signal paths are rarely equal. Consequently, a signal that is split and sent along two such paths will not arrive at its two destinations simultaneously. The test-system designer must strive to minimize the effect of this phenomenon wherever possible and to counteract it where necessary, since DUT input and compare signals simply must be aligned if valid tests are to be performed.

For each timing phase on every pin card, the S-3295 uses local look-up tables to correct skew differences related to the signal path. In the deskewing circuitry shown in Fig 5, the contents of the registers are delay constants determined at system calibration time under controlled conditions, then stored in a disk file. Upon subsequent system power-up, the constants are automatically transferred to the registers for use by the system. The incremental resolution of the delay data is about 50 ps, and the adjustment range 8 ns.

This look-up–table technique yields pin-to-pin driver and comparator skew figures as low as ±350 ps, and the data is updated whenever calibration is performed. As a result, long-term drift in system performance is counteracted. Furthermore, if a particular application requires even greater precision, the user may construct delay tables based on the parameters of a specific format or timing set used by the DUT.

Though all of the preceding attributes—driver flexibility and precision, loading versatility, and timing accuracy—are necessary requisites for a VLSI test system, they are ultimately of no benefit to the user if the system is not able to provide the types of patterns he or she needs to check out a device. Typically, the functional pattern data going to the DUT derives from one of two types of sources: stored-response sources such as a serial memory or a pattern processor or an algorithmic processor that generates pattern data at test time. Each pattern source is mated to a specific set of applications; no source can be



4. Doubling up. For 40-MHz operation, the inputs to two drivers can be tied to one, using the switch mode enable to inhibit one driver (driver B). Tying inputs together, rather than outputs, lets the device under test see the same back impedance, regardless of its speed.

sacrificed without a corresponding loss in the utility of the system.

The S-3295 integrates stored-response and algorithmic modes into its pattern processor. The pattern processor can itself store up to 4,096 words of force-compare pattern (as well as 1,024 words of inhibit-mask pattern), and it has control over the two other possible sources of functional data: the pin card's local memory and the algorithmic pattern input.

Choosing sources

The serial-memory approach is primarily useful in testing devices requiring long, nonrepetitive data streams. Logic ICs produced by computer-aided design techniques are ideal candidates for serially sourced patterns. CAD patterns—the code lines used in modeling and debugging logic circuitry on a computer—usually consist of densely packed unique pattern lines. They cannot be condensed any more, nor can they be characterized by an algorithm.

The S-3295's pattern processor is well suited to testing such parts: it can select the pattern source at test time on a cycle-by-cycle basis. During every cycle, one of eight programmed combinations of force, inhibit, compare, and mask data sources can be supplied to the DUT.

This programmable processor enhances the local-memory partitioning of the pin card. If the pattern processor provides inhibit and mask data, the 32-K local-memory segments usually allocated to those functions can be used for force-compare pattern storage instead, effectively doubling the force-compare memory. As needed, the source combinations can be changed in successive cycles.

For functional testing of serial-scan-path design devices (CAD-designed parts with auxiliary serial I/O pins that allow the test engineer to monitor otherwise invisible internal processes), the serial stored-response approach typically is used. However, scan testing has unique requirements that may not be met by simple read-write patterns. For example, the stored pattern may have to be subdivided into segments, each of them constituting a module of a full test. Furthermore, pattern-storage needs can exceed the practical capacity of local memory. Finally, specialized circuitry is needed to store and trace serial-scan-testing errors.

For these parts, the S-3295 offers a dedicated pattern source, the scan buffer, which is an optional 3-megabit serial memory (1 megabit each for force, compare, and mask) that can be segmented into chains 64 K to 1 Mb in length. This buffer operates under control of the pattern processor and feeds the external signal input on the pin cards. It includes a 4,096-word error memory that keeps a record of every location of a failing chain bit, as well as additional circuitry to flag failing chains.

Real-time pattern generation

In contrast to the serial storage and pattern-compression schemes where both input and output states are fully described, algorithmic patterns contain no stored-response information. At test time, an algorithmic pattern processor generates the regular, predictable patterns used to test semiconductor memories. The algorithmic approach yields the most condensed pattern storage of all—billions of pattern lines can be issued from the half dozen instructions that define a "galloping" pattern, for instance. The pattern processor may be switched between

Finding problems, not faults

While basic go/no-go device testing is all that is required in most production situations, data from the cycles preceding an error is indispensable if device designers or evaluation engineers are to make a meaningful assessment of device performance. The design or evaluation engineer must be able to determine where in the pattern a device failed in order to spot address-related failures in the case of a memory or instruction-related failures in the case of a microprocessor. If the failing test cycle is to be correctly localized, the test must, in effect, stop immediately upon detection of the error in the device under test—that is, during the same cycle.

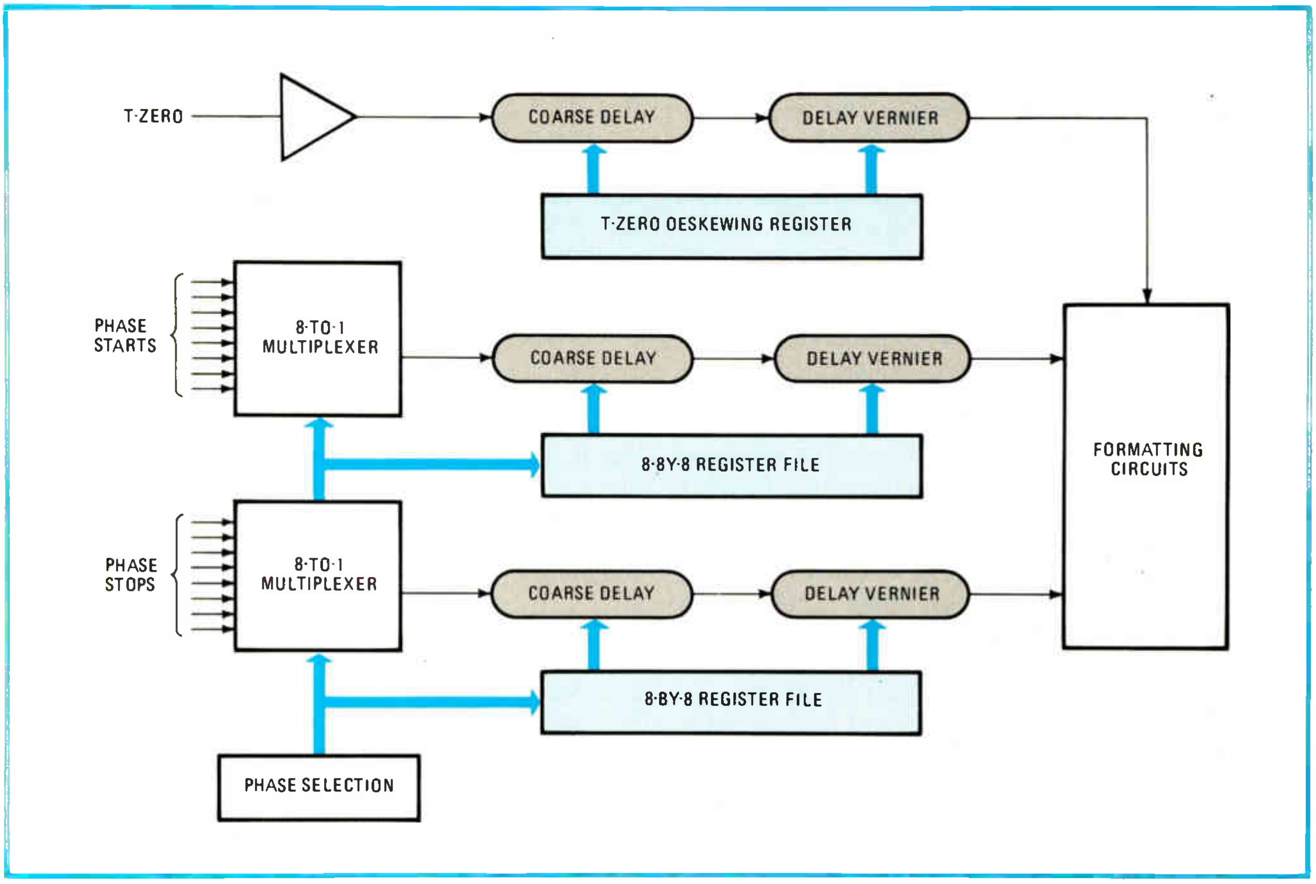
However, inherent delays between pattern sourcing and DUT response make it very difficult to stop during the actual error cycle, particularly at maximim test rates. To counteract these delays, a test system must provide a means of accounting for any cycles that occur between error detection and test shutdown. Once the error cycle is determined, the pattern information from that cycle can be retrieved and digested during the postmortem.

To do this, the S-3295 uses a last-in, first-out history stack and an error-event counter. The stack, located in the pattern processor, keeps a running record of the 255 most recently executed program-counter locations. Immediately upon detection of an error, the counter begins tracking the number of cycles until the stop-on-error process has run its course. Thus the stack can point out the precise location and sequence of data that cause the failure.

The S-3295 also provides another error-storage mechanism particularly useful in the algorithmic pattern mode. Under mainline program control, a shift-on-error condition can be set up before the functional test begins. For every error the pin cards' comparators detect, the force pattern and the error data from all pin cards operating during the failing cycle are shifted into local memory by a clock pulse.

Proper alignment of stimulus data and the related errors is guaranteed by a set of retiming registers. This shift-on-error function enables the test system to decipher a device under test—a read-only memory, for instance—by comparing the DUT output against a fixed bit (usually 0) and storing errors as described.

A third error-storage variation involves the use of the stop-on-error mode in conjunction with continuous local memory storage of force and error data during a test. The test stops upon detection of an error, and the error event counter records the number of post-error events. Thus, the failing pattern line is pinpointed, accompanied by 32-K lines of pattern preceding the error (or more, depending on memory partitioning). In this application, the system acts as a very powerful logic analyzer.



5. Digital deskew. The time it takes a signal to reach the output pins of the tester is measured at calibration time and digitized. At test time, the T-0 and 8-by-8 registers use these digitized constants to adjust the signal delay in accordance with the phase selected by the system user.

stored-response and algorithmic modes on a cycle-by-cycle basis. A common application of this feature is in testing microprocessors with on-board memory; the stored-response vectors are used to test the processor's registers, arithmetic and logic unit, and so forth, while the memory is exercised with patterns.

Functionally, a memory is easy to conceptualize as a simple matrix of storage cells. Addressing the cell locations and storing and retrieving data are usually seen as straightforward processes, but these functions represent only the tip of the iceberg.

Beneath the surface lie unforeseen threats to smooth testing—page-mode addressing, regional-data inversions, pattern sensitivity, topological encoding: the list goes on indefinitely. A truly versatile algorithmic pattern generator must deal with these memory-test issues, as well as expediting the creation of simple, widely used and accepted test patterns.

In its algorithmic mode, the S-3295 pattern processor's 4,096-word memory is loaded with algorithmic instructions that compute pattern data in real time. As with the stored-response modes, the pattern output per cycle is dictated by loops, branches, and subroutines.

However, each algorithmic instruction is also linked to a group of dedicated registers: two 12-bit address generators, a 12-bit Z-axis generator; and a 32-bit force-compare data register. The Z-axis and the data registers may be expanded to twice their standard width.

The full power of this pattern processor is needed in testing today's dynamic RAMs. These memories are usually partitioned into regions, with each region's data inverted relative to that of a neighboring region: in one a charge stored in a cell might represent a logic 1, while in its neighbor the absence of charge represents logic 1. Further complicating testing is the fact that dynamic RAMs usually scramble their addresses; consecutive addresses do not access adjacent cells. If it is necessary to address adjacent cells sequentially, as in testing with a surround-disturb pattern, the address scheme must be unscrambled. Before using such a pattern, a background of all 0s typically is written into the memory to ensure a known starting state, and this also requires regional and addressing knowledge.

In the algorithmic mode, the S-3295 pattern processor generates address vectors by arithmetically manipulating base registers, minimum- and maximum-address registers, the parity-mask register, a group of general-purpose registers, and a 4,096-word topology RAM. The DUT's data-inversion regions can be determined by address-parity states and base-cell equality.

In the test situation outlined above, the basic background-fill/surround-disturb pattern is written using the maximum address, base, and general-purpose registers; the address output is channelled through the topology RAM to propagate the incrementing address count into corresponding physical locations in the memory matrix. Finally, an address-parity test condition is established to invert the data word based on region. □