

An overview of on-chip compression architectures

DATA-COMPRESSION TECHNIQUES CAN HELP MANAGE THE ESCALATING COST OF TEST IN NANOMETER DESIGNS.

The test time that scan tests require typically dominates manufacturing-test costs for digital designs. The increase in design complexity and the requirements for delay tests have made test time a design parameter that requires active management in nanometer designs. As the number of patterns increases, it takes more tester-buffer space to hold the complete test set, and it takes longer to execute the test set in manufacturing. To address both the data-volume and test-time problems, test engineers and test architects have developed techniques employing on-chip hardware that compresses the test-stimulus and response patterns and then applies them to the chip under test. Luckily, there are many test architectures that engineers can employ for test-data compression.

COMPRESSION-METHODS BACKGROUND

The dominant method of testing digital circuits is the use of an ATPG (automatic test-pattern generator) to target a stuck-at or transition fault model at all of the circuit nodes. In circuits that contain storage elements, engineers can use scan registers to enable control and observation of the storage elements and ensure high fault coverage. When the ATPG generates too many test patterns, the test-application time becomes too long, and engineers must use on-chip-compression techniques to minimize test time and, thus, test costs.

Test compression builds on technology originally developed for LBIST (logic built-in self-test). **Figure 1** shows the general structure of compression logic within a chip or core. The system decompresses a compressed input stream and feeds it into the internal scan chains, some of which may be inside cores within the design. The system optionally feeds the output from the internal scan chains through masking logic and then compresses it into an output stream. Engineers can use several architectures for the input decompressor and the output compressor.

INPUT DECOMPRESSION

Input decompression enables a small number of scan input streams to load stimulus into a much larger set of internal scan chains. Papers have proposed and products have implemented several types of input-decompression architectures. **Table 1** lists the most common.

The simplest input decompressor is broadcast scan. This device simply fans out each scan input to multiple internal chains. The main complaint against broadcast scan is that those chains receiving their values from the same scan-in pin have

directly correlated values, which may impact fault coverage. In most practical implementations, however, this possibility has not been a problem. Engineers obtain the linear-spreader, space-expansion network by XORing combinations of scan inputs to each internal-chain input. The scan correlations are still there, but they are less direct than with broadcast scan.

Another approach to avoiding the scan-chain dependency associated with broadcast scan is the use of multiplexed scan configurations. In its simplest form, this technique uses several broadcast-scan configurations and provides one or more additional scan inputs to switch between them. The scan correlations are still there, but, for some scan cycles, one of the configurations may allow you to attain the desired care bits. A multiplexed linear spreader is similar to the multiplexed broadcast scan. It provides two or more sets of linear equations that you can use to solve and attain the desired care bits from the ATPG.

Alternative approaches for input decompression rely on a sequential linear-feedback and spreading network. The sequential elements are based on LFSRs (linear-feedback-shift registers) or linear automata, but they achieve the same result: Buffer up the input variables from the scan-input streams so that scan cycles not requiring a lot of care bits can defer the use of the variables for later, more demanding cycles. These approaches are almost assuredly the best for dealing with any scan correlation. However, the scan chains still see correlated values, so

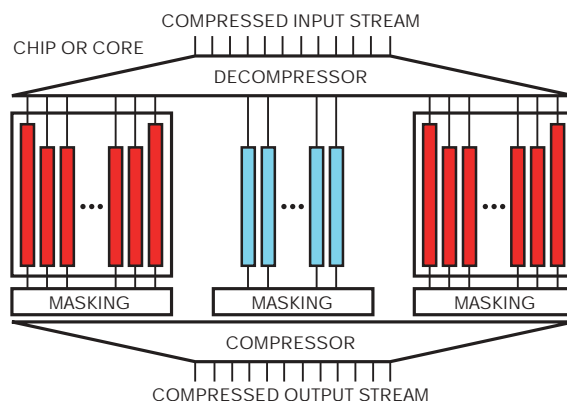


Figure 1 In a general-compression structure, engineers decompress a compressed input stream and feed it into the internal scan chains, some of which may be inside cores within the design.

there are limits to how many care bits one scan cycle or consecutive scan cycles can accommodate.

OUTPUT COMPRESSION

The role of an output compressor is to enable a large number of internal scan-chain output streams to merge to create either a much smaller set of external scan-output streams or to create a signature for each test or set of tests. **Table 2** lists the most common compressors, many of which papers have proposed and products have implemented.

The simplest compressor, an XOR tree, computes simple parity of its inputs. This method has two major limitations: aliasing and X tolerance. If engineers feed each internal-chain output to one scan-output pin through a parity tree and if an equal number of inputs to the parity tree contain errors, the error will be lost due to cancellation. Unknown values (X states) in any chain output mask any other chain feeding the same scan output. Diagnostics of tester failures are possible by assuming that any error bit in the output stream could have come from any of the internal chains that feed to that output. Although this technique causes the simulation of many more faults than would be necessary without compression, it likely reduces diagnostic accuracy over uncompressed analysis.

One method of reducing the limitations of the XOR-tree space compactor is to combine it with input fan-out to help deal with both the aliasing problem and the issue of unknown (X) values. Here, each internal chain feeds to a unique set of scan outputs, with each scan output fed by an XOR tree of values from some set of chain outputs. Defining a space compactor that can deal with a large number of X values in the same scan cycle requires a large amount of fan-out, making it prohibitively expensive to implement. The more fan-out, the likelier that a larger number of X values in the same cycle will cause every output to be X. Unless you know that there will not be a large number of X values in any scan cycle, this approach will likely require some amount of X-masking capability. Diagnostics become more difficult, because errors now tend to appear on multiple output pins; simply picking all internal chains that feed any of the failing outputs produces too many candidates. Solving simultaneous equations to locate the most likely source or sources of the errors can help to reduce the number of candidates to consider for capturing the defect's effect and thus reduce the number of faults to simulate.

The multiplexed-output-select approach consists of a multiplexed internal-chain-output-select network using inputs to select which set of chain outputs each scan cycle should observe. This approach often causes the system to ignore most of the chain outputs and thus may miss some of the accidental detections that occur. It is unclear how detrimental this approach

TABLE 1 COMMON INPUT DECOMPRESSORS

Input decompressor	Comment
Broadcast scan	The simplest input-decompression scheme
Linear (XOR) spreader	A combinatorial input decompression, requires linear-equation solver
Multiplexed broadcast scan	Alleviates scan correlation from broadcast scan by switching between several broadcast-scan configurations
Multiplexed linear spreader	Alleviates scan correlation from linear spreader by switching between several spreader configurations, requires linear-equation solver
Linear-automata stream	Alleviates scan correlation from linear spreader by circulating input variables within FSM, requires sequential linear-equation solver

might be, but few defects in today's technologies behave exactly like the faults in the ATPG-fault model. Selecting the chain outputs to observe takes care of much of the necessary masking to avoid X values. As long as there is no combining of values from multiple internal chain outputs, there should be neither a need to add X-masking to such a compressor nor an issue with aliasing. Diagnostics are much simpler for this approach, because there is only one source of the error when it is seen at a scan output—the internal chain selected on that scan cycle. However, because the system does not observe many of the additional errors that typically occur when devices fail, it may be difficult for diagnostic-fault simulation to correctly assess the location of the failure with such a reduced response set.

Convolutional compactors add a linear shift register to the output-space-compactor network. Its advantage is that it reduces the aliasing problem associated with a simple XOR tree. The addition of the shift register also may reduce the ability of the compactor to deal with unknown values (X). An X value corrupts the bits in the register and anything they feed, but the X eventually drops out of the register. Diagnostics are more complicated and, again, likely to require the simulation of many more faults than if no compression took place.

Finally, for decades, engineers have used MISRs (multiple input-shift registers) to compress responses, and MISRs are necessary for any form of BIST. The output data continually clocks into the MISR, and, at the end of the test, the signature in the MISR assesses a pass or fail versus the known-good signature. MISRs have some small chance for aliasing in which the defect produces a signature that matches the defect-free circuit, but the likelihood is normally much smaller than the missed faults due to poor fault coverage. The biggest benefit of using a MISR is that response data compression is highest with a MISR,

because there is no need to compare output values on every scan cycle—only the final state. It is even possible to collect a signature across all of the tests and check it only at the end of the test set. MISRs cannot tolerate any X values. Using a MISR requires the use of X-masking to prevent any X values from getting into the MISR unless all X sources have been

TABLE 2 COMMON OUTPUT COMPRESSORS

Output compressor	Combinatorial/sequential	Comment
Space compactor (XOR tree) using only fan-in	Combinatorial	Could alias (lose) errors from even number of chains or when combined with unknown (X) values
Space compactor with fan-out	Combinatorial	Less chance of aliasing, can handle some X
Multiplexed output selects	Combinatorial	May miss accidental detects, can handle X
Convolutional compactor	Sequential	Less chance of aliasing, can handle some X
MISR with space compactor	Sequential	Less chance of aliasing, cannot handle X

blocked. Also, diagnosing failures from the final MISR signatures is highly unlikely. Knowing which of several MISRs caught the error may help isolate the error to a single core or area of the chip but not to a specific failing net. Diagnostics with MISRs typically require either reapplying the test and scan-out (without compression) to collect the diagnostic fail data or to observe the state of the MISR on each shift cycle to catch where the errors enter the MISR on each cycle. One alternative is to add a few uncompressed vectors to the compressed test set and perform diagnostics primarily for failures that occur within that smaller, uncompressed test set.

DEALING WITH UNKNOWN VALUES

A typical design with tests generated by an ATPG has unknown logic values that propagate in the scan chains to the output streams. Except for the multiplexed-output selects, compression logic loses efficiency when it has to contend with a large number of unknown or unpredictable values in the output streams. This situation is a concern for any compression scheme that relies on MISRs to compute signatures, but it also can cause non-MISR-based results to suffer. All compression schemes except the multiplexed-output approach linearly merge the output streams and thus are affected when an X merges with non-X values, causing the ATPG to ignore and effectively mask out some of the non-X internal-response values. As more unknown values become part of the output stream, engineers may have a tough time seeing defective responses, possibly resulting in a loss of fault coverage or an increase in test-vector count to make

up for the fact that such tests detect fewer faults, either those that the ATPG targets or those that emerge by accident.

One way to alleviate the problem of unpredictable values in the output stream is to mask these values before they enter the compressor. In the case of MISRs, this step may be necessary to avoid the creation of corrupted and unpredictable MISR signatures. In the case of the space-compactor approaches, it may be necessary just to allow detection of certain faults; there is no absolute requirement to mask out all X values. However, because X values that do propagate through mask out known values from some chains when too many X values do appear on the same scan cycle, only a few chains may be observable on that cycle. This situation reduces the ability to see where additional errors may have been and reduces the ability of diagnostics to locate the true failure location.

Masking out some values while allowing others to pass requires you to pass some information into the device. Thus, adding an X-masking capability consumes more of the input bandwidth, possibly taking that bandwidth away from care bits that target faults. Remember, however, that the observation points for faults are also effectively care bits in this compression environment. So, it is not unreasonable to allocate some of the input-side bandwidth to the X-masking.

INTEROPERABILITY REQUIREMENTS

Typically, vendors insert compression architectures at the RTL or gate level, and the ATPG and diagnostics engines need to understand the architecture of the compression to generate

and diagnose tests with these structures on-chip. The supplier that inserts the compression then also by default becomes the ATPG and diagnostics supplier. Semiconductor companies, however, often demand choices for ATPG and diagnostics from multiple vendors for a given application. As a result, the industry is moving toward enabling interoperability of ATPG and diagnostics for different compression architectures.

MORE AT EDN.COM ▶

+ Go to www.edn.com/ms4205 and click on Feedback Loop to post a comment on this article.

The Accellera Consortium (www.accellera.org) of EDA vendors and users launched an OCI (Open Compression Interface) technical committee within the last year. The OCI technical committee is defining a

standard that you can use to describe the required information about on-chip compression structures to ATPG and diagnostics tools. More information is available at the Accellera Web site (**Reference 1**).

Test-data-compression techniques manage the escalating cost of test in nanometer designs. As designers adopt delay test to detect small delay defects in nanometer designs, they will also adopt on-chip compression methods. Luckily, the test industry has developed several popular compression structures over the past few years and will continue development as new challenges arise. **EDN**

REFERENCES

1 Accellera Open Compression Initiative, www.accellera.org/activities/oci-tc/.

- 2 Barnhart C, et al, "Extending OPMISR beyond 10X Scan Test Efficiency," IEEE Design & Test of Computers, September to October 2002.
- 3 Rajski, J, et al, "Embedded Deterministic Test for Low Cost Manufacturing Test," Procedures of the International Test Conference, pg 301, 2002.
- 4 Samaranyake, S, E Gizdarski, N Sitchinava, F Neuveux, R Kapur, and T Williams, "A Reconfigurable Shared Scan-in Architecture," Procedures of the VLSI Test Symposium, pg 9, 2003.
- 5 Koenemann, B, C Barnhart, B Keller, T Sneath, O Farnsworth, D Wheeler, "A SmartBIST Variant with Guaranteed Encoding," Procedures of the Asian Test Symposium, pg 325, 2001.
- 6 Mitra, S, KS Kim, "X-Compact: An Efficient Response Compaction Technique for Test Cost Reduction," Procedures of the International Test Conference, pg 311, 2002.
- 7 Wohl, P, J Waicucauski, and S Patel, "Scalable Selector Architecture for X-Tolerant Deterministic BIST," Procedures of the Design Automation Conference 2004, pg 934.

AUTHOR'S BIOGRAPHY

Brion Keller is a senior architect at Cadence Design Systems. He has 27 years of experience in ATPG, fault modeling and simulation, logic BIST, test-vector compression, diagnostics, and general design for test, including more than 23 years at IBM. Keller has a bachelor's degree in computer science and chemical engineering from Pennsylvania State University (University Park).