SYNOPSYS®

WHITE PAPER

DFTMAX Compression Shared I/O

Minimize the Cost of Testing ARM® Processor-based Designs and Other Multicore SoCs

Authors

Chris Allsup
Marketing Manager, Synopsys

Padmashree Takkars
Corporate Applications Engineer, Synopsys

Teresa McLaurin
DFT Architect, ARM

Introductions

A significant design trend in recent years has been the widespread use of ARM® multicore processors in systems-on-chip (SoCs). Designers' ability to easily and cost-effectively employ multiple, high-performance embedded processors to meet the computational demands of the end application has helped fuel the explosive growth in mobile computing, networking infrastructure, and digital infotainment systems.

Teams developing these ARM processor-based designs and other multicore SoCs have had to implement design-for-test (DFT) to achieve their test quality goals. But a key challenge has emerged: as the number of processor cores increases, it becomes more difficult to meet these goals cost-effectively, given the limited number of pins available for testing.

Synopsys' shared I/O capability in DFTMAX™ compression and TetraMAX® ATPG addresses this challenge. The architecture and automation elements of shared I/O work together to lower test cost without compromising test quality and diagnostics accuracy. Shared I/O has become the preferred approach for implementing DFT in multicore processor designs because it reduces test pattern count and ATPG runtime while utilizing the same or fewer test pins than standard techniques.

Non-shared I/O Architecture

Figure 1 illustrates a typical, non-shared I/O DFT architecture that allocates 32 I/O pairs for testing quad-core ARM Cortex®-A15 and Cortex-A7 processors: 6 scan channels per CPU core and 8 scan channels for the non-CPU logic[1]. Both CPU and non-CPU logic encapsulate DFTMAX compressor-decompressors (CODECs) to reduce test application time and test data volume[2] for low-cost manufacturing test. The compression ratios implemented for the CPU and non-CPU CODECs are chosen to ensure a uniform scan chain length across all logic in the design. While higher compression ratios shorten the chain length and achieve greater test time reduction, their values are constrained in practice by a CODEC's minimum number of scan I/O, as well as routing considerations.

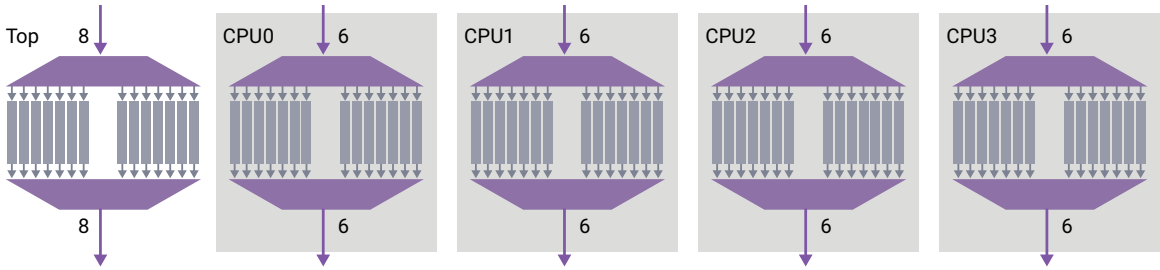

Figure 1: A typical DFT architecture for a quad-core ARM Cortex-A series processor with CODECs embedded in the CPU and non-CPU logic

Mixed-shared I/O Architecture

An example of a shared I/O DFT architecture is shown in Figure 2. This arrangement is often referred to as "mixed-shared I/O" because dedicated pins are still used to test the non-CPU processor logic. 22 of the 24 scan inputs allocated for testing the CPU cores are shared among all four CPU CODECs. In addition, DFTMAX synthesizes integration logic and connects it to the CODEC outputs to enable full observability of the scan chains. This logic is controlled by the remaining two scan inputs that select which of the CODECs are observed from one test cycle to the next. One of the other scan inputs serves as an X-mask enable signal for maintaining high coverage and low pattern count in the presence of unknown logic values (X's).

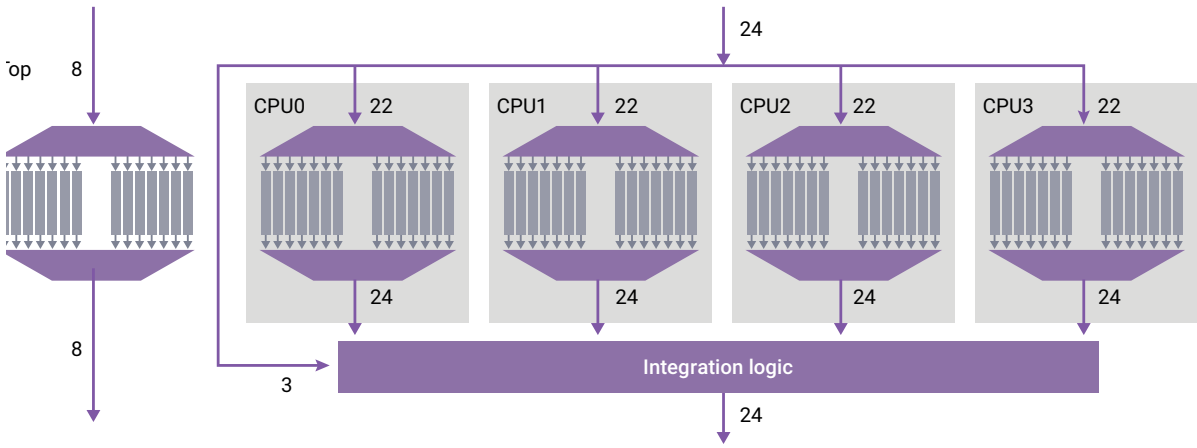

Figure 2: The DFTMAX mixed-shared I/O capability shares scan inputs among the CPU CODECs and synthesizes integration logic to observe their outputs

DFTMAX was used to implement the non-shared I/O architecture of Figure 1 and, with only slight modifications to the DFT scripts, the shared I/O architecture of Figure 2 for two 28-nm quad-core Cortex-A series processors. TetraMAX ATPG power-aware patterns[2] were then generated based on stuck-at and transition delay models. For each design, equivalent chain length and high fault coverage were used to compare results across architectures. Figure 3 summarizes the improvement in ATPG pattern counts and runtimes using shared I/O for the (a) quad-core ARM Cortex-A7 and (b) quad-core ARM Cortex-A15 processors. The results demonstrate that shared I/O requires substantially fewer patterns and less ATPG runtime, which lower test costs.

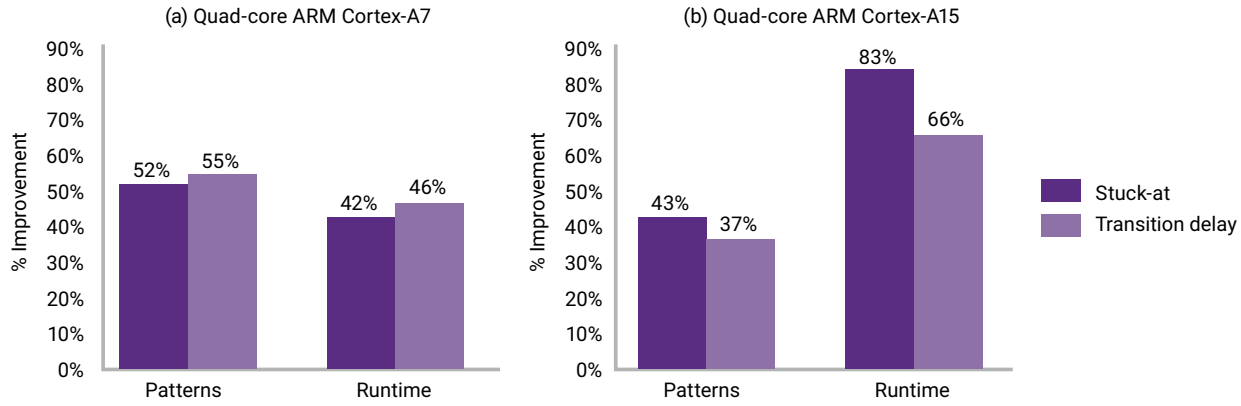

Figure 3: Improvement in ATPG results for DFTMAX shared I/O compared with the non-shared approach for (a) quad-core ARM Cortex-A7 and (b) quad-core ARM Cortex-A15 processors

All-shared I/O Architecture

A variant of shared I/O known as "all-shared I/O" can be advantageous in situations where dedicating the typical number of pins for testing conflicts with packaging constraints or potentially undermines other cost-saving techniques such as multisite testing[3] that rely on utilizing fewer pins. As shown in Figure 4, all-shared I/O extends sharing of the scan inputs and outputs to the non-CPU logic.

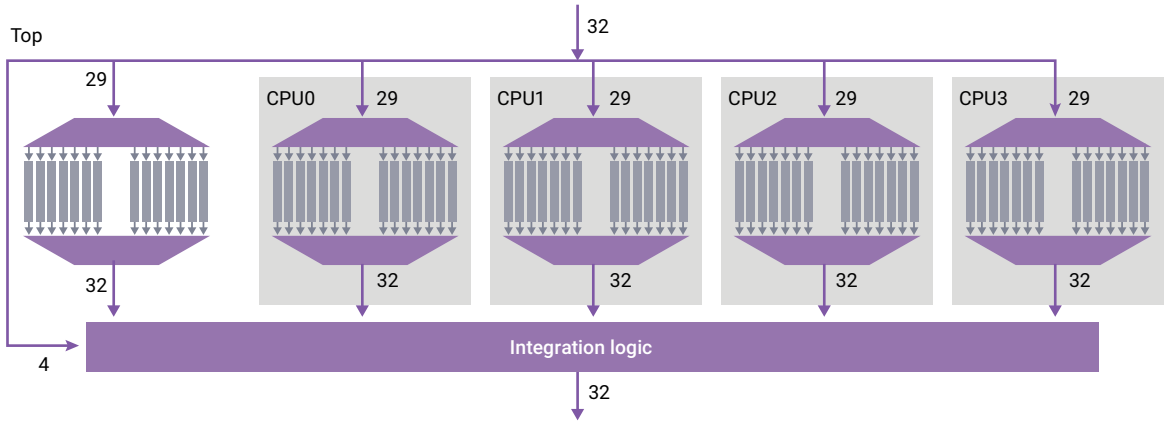

Figure 4: The all-shared I/O capability in DFTMAX shares scan inputs among both the CPU and non-CPU CODECs and synthesizes integration logic to observe all the CODEC outputs

With only slight modifications to the DFT scripts and using the same scan chain length as the previous runs, DFTMAX was used to implement all-shared I/O on the 28-nm quad-core ARM Cortex-A15 processor for various numbers of scan I/O pairs. Figure 5 summarizes the improvement in TetraMAX ATPG power-aware pattern counts compared with the non-shared, 32 scan channel configuration of Figure 1 at the same high fault coverage.

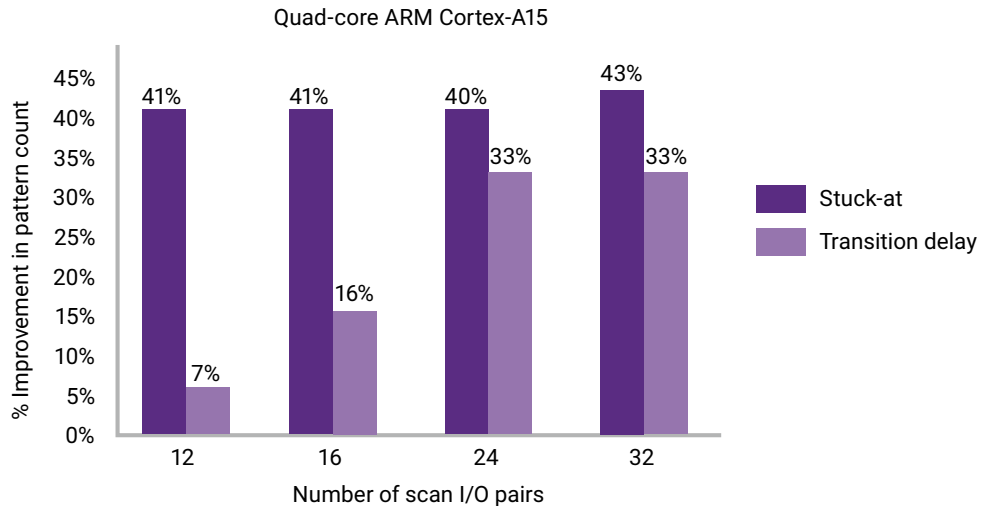

Figure 5: Improvement in pattern count vs. number of scan I/O pairs for DFTMAX all-shared I/O compared with the 32-channel non-shared I/O architecture (Figure 1)

The ATPG results reveal the all-shared I/O pattern counts for 24 and 32 scan I/O pairs are on par with the mixed-shared I/O results (Figure 3b). Moreover, all-shared I/O requires fewer patterns than non-shared I/O even when substantially fewer scan channels—60% fewer for 12 channels—are used. These results indicate that all-shared I/O is a cost-effective strategy for achieving high test quality and low pattern count when pin resources are limited.

Optimizations for Identical Cores

The improved ATPG efficiency from shared I/O is derived in part from sharing the scan inputs among multiple cores. However, Synopsys' synthesis-based test solution provides additional optimizations, enabled when processor cores are identical, that further improve pattern count and runtime, as well as the ability to isolate defective parts. Let's now look at these optimizations.

DFTMAX shares the scan inputs ***uniformly*** across the identical cores, as shown in Figure 6. Uniform sharing of scan inputs is especially relevant in scenarios where they are also shared with non-CPU CODECs that may have differing numbers of inputs. This capability also ensures that TetraMAX can perform its own optimizations to improve ATPG efficiency. For example, TetraMAX effectively divides the fault list into two sets: a 'unique' set corresponding to just one of the CPUs plus the non-CPU logic, and a 'non-unique' set corresponding to the remaining identical CPUs. Assigning a higher priority, in terms of ATPG effort, to the unique fault set decreases pattern count and runtime.

The select lines going into the integration logic block in Figure 2 control logic used to isolate defective scan chains, making it possible to determine which of several faulty values belong to which scan chains in which cores. The integration logic also contains XOR trees that provide virtually the same high observability as the non-shared I/O architecture that relies on dedicated connections to output pins. When the processor cores are identical, it is possible to improve diagnostics resolution using a technique borrowed from image processing. Rotating the order of a CODEC's outputs with respect to the order used in its neighboring CODEC, depicted in Figure 6, makes it more likely that a fault can be detected and isolated to a particular core. Synopsys experiments have shown that bit rotation improves diagnostics resolution by approximately 2x.

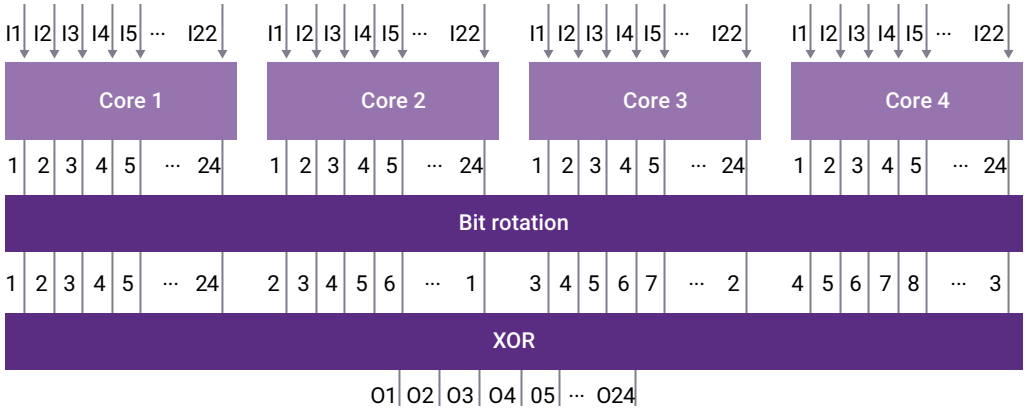

Figure 6: DFTMAX optimizations for identical cores include sharing scan inputs uniformly to increase ATPG efficiency and rotating bits at the cores' outputs to improve diagnostics resolution

Scan chain isolation and bit rotation also enhance TetraMAX's ability to manage unknown logic values captured across multiple scan chains and cores, which further improves fault coverage, pattern count, and runtime.

Conclusion

A solution optimized for testing complex multicore processor designs is needed to achieve both high-quality and cost-effective manufacturing test. DFTMAX shared I/O lowers the cost of testing ARM processor-based designs and other multicore SoCs in two fundamental ways: First, it reduces ATPG pattern count, which decreases test application time and test data volume. Second, it reduces the number of test pins needed to generate high-coverage test patterns. DFTMAX optimizations such as scan chain isolation and bit rotation improve diagnostics resolution and, combined with TetraMAX optimizations, enable greater ATPG efficiency for designs that utilize identical cores.

References

[1] McLaurin, T., Frederick, F., Slobodnik, R., "The DFT Challenges and Solutions for the ARM Cortex-A15 *Microprocessor," Proc. International Test Conf*., 2012.

[2] Hay, C., "White Paper: Testing Low Power Designs with Power-Aware Test," April 2010, Synopsys, Inc.

[3] Kuntzsch, C.; Shah, M.; Mittermaier, N., "Massive Test Cost Reduction by Advanced SCAN Testing," SNUG Germany 2010 Proceedings.

01/09/18.CS12009_DFTMAX_ARM_WP.

Pub: Dec. 2017