SYNOPSYS®
Predictable Success

The Benefits of Static Timing Analysis Based Memory Characterization

September 2012

Author
Ken Hsieh
Product Marketing Manager,
Synopsys, Inc.

The Design Challenges of Embedded Memory

The memory content of System-on-Chip (SoC) designs has increased dramatically over the past few years. More and more silicon area consists of memories with different functionality in the forms of embedded SRAM, ROM, and multi-port register files, just to name a few. There are advantages and challenges to using embedded memories in a SoC. The principal benefit is that embedded memories allow special tailored functionalities in a design. They also provide improved performance, lower power consumption, on-demand memory activation with refined standby modes, more compact packaging, and overall cost reduction. On the other hand, embedded memories increase the design complexity due to area and yield optimization challenges. New methodologies for design, analysis and test are needed to overcome these challenges.

Embedded memory characterization is of increasing concern to SoC designers. Accuracy and efficiency of the models at all stages of the design are essential to the success of designing a SoC with embedded memories. The number of memory instances per chip increases rapidly at advanced process nodes. To simulate the full range of process, voltage, and temperature corners (PVTs) and the effects of process variation, the number of memory characterization runs and the data processing time per run grow exponentially.

There are two general approaches to creating memory models. The first approach is based on the characterization of memory compiler generated models, while the second approach is based on the characterization of individual memory instances. Memory compilers are tools that can automate the creation of many different and unique memories very quickly. The characterization is done by fitting timing data to polynomial equations with their coefficients derived from a small sample of memory instances. The advantage of this approach is that the model generation is very fast, but it comes at a cost of less than optimum accuracy.

To overcome the inaccuracies of compiler-generated models, instance-specific characterizations are performed over a range of PVTs. The improved accuracy comes at a cost of long characterization processing time. Although there are several approaches to improve the instance-based characterization throughputs, all of them require the use of SPICE or FastSPICE dynamic simulators to trade-off between performance and accuracy. This still does not guarantee the full verification coverage needed to ensure silicon success.

Typically around 70% of the engineering effort and time performing memory characterization is focused on timing analysis and model generation. Timing verification is the process of validating that a design meets its design specifications by operating at a specific clock frequency without errors caused by a signal arriving too soon or too late.

With the traditional approach, given a specific scenario of input and clock transitions, a dynamic simulator such as SPICE or FastSPICE is commonly used to determine if a particular sensitization leads to a timing violation in the circuit block. By simulating under all possible sequences of transitions, it can be determined whether or not the block under test can operate at the target clock frequency without any timing violations. Unfortunately, the number of vectors required for such an exhaustive validation is exponential to the number of inputs and state elements. Therefore, this simulation approach is impractical for all but very small blocks.

Alternatively, transistor-level static timing analysis has been available for over a decade. Today, the technology has evolved and expanded to cover a range of memory types such as single- or dual-port embedded SRAM, and it offers many advantages to the traditional black-box approach timing analysis.

STA for Memory Characterization

Unlike dynamic simulators, static timing analysis (STA) tools remove the need for simulating the entire memory block under all possible scenarios. Instead, STA tools use fast but accurate approaches to estimate the delay of sub-circuits within the block and use graph analysis techniques to quickly seek out the slowest and fastest paths in the block. The result is that an STA tool can typically find all timing violations in a block in a fraction of the time it would take a dynamic circuit simulator.

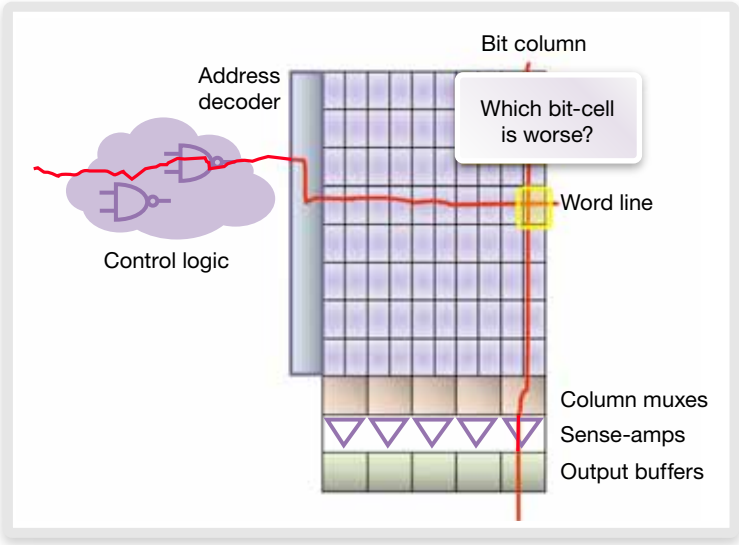

Figure 1: STA-based timing analysis through control logic and memory core array

With the latest developments in STA technology, it is now possible to time not only the control logic (transistor-level digital circuits) of a memory block, but also paths through the memory core array (i.e. bit-column, wordline, column mux, sense amp, and so on) as illustrated in figure 1. The result is an improvement in design turn-around-time, verification coverage, accuracy (within ±5% of SPICE), and productivity for designs with embedded memories. This new capability supports memory compiler and instance-based memory characterization. It does not require netlist reduction techniques as commonly practiced in the dynamic simulation approach. A major benefit of using the STA approach is that there are no vectors needed in performing timing analysis. This alone saves tedious verification planning and processing time and eliminates the potential of human error in generating the stimulus for the dynamic simulations.

Figure 2 illustrates a typical memory architecture design and characterization flow based on the STA technology. In this flow, the STA tool identifies timing violations in the memory designs and forwards the information to SPICE/FastSPICE to further determine what went wrong.

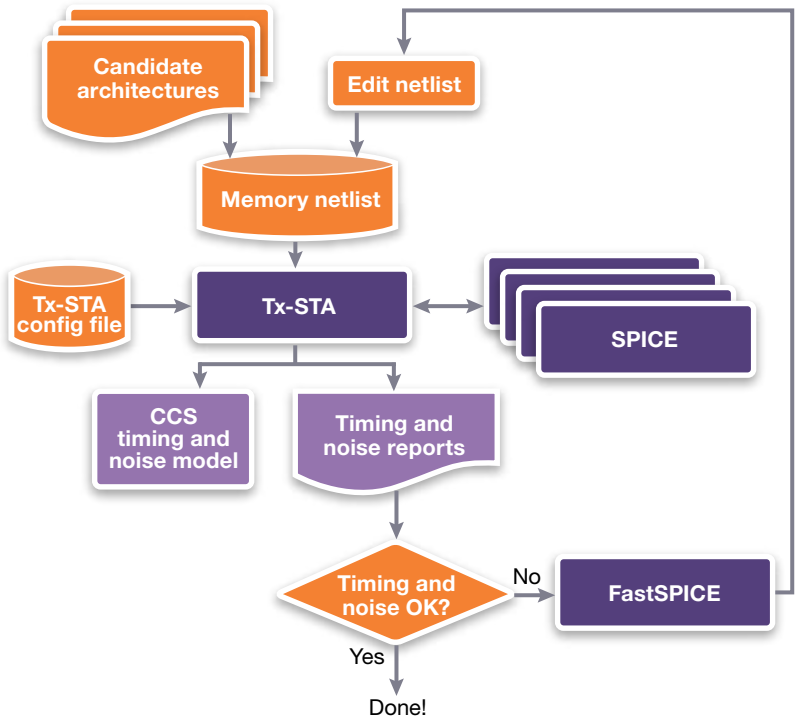

Figure 2: Memory architecture design and characterization flow

Additionally, SPICE and FastSPICE simulators are also used to fine tune the detailed characterization of selected critical timing paths for golden accuracy. The STA memory architecture design and characterization flow also allows users to quickly and accurately generate memory library models within ±5% of SPICE accuracy.

Similarly, for characterization of memory instances generated by a memory compiler, the STA characterization flow can be established to perform the tasks illustrated in figure 3.

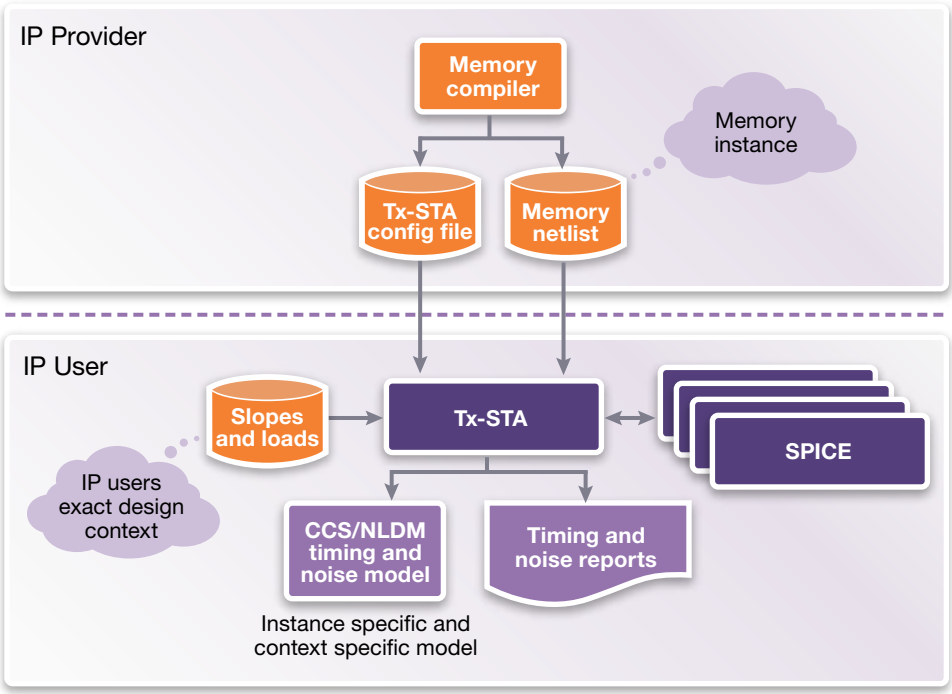

Figure 3: Instance-specific memory characterization flow for the IP users

This flow is ideal for IP users and allows them to carry out the in-context instance-specific memory characterization for various PVTs without the need for simulation vectors or pre-selected critical timing path for analysis. The generated timing models can be in the form of CCS models or the standard NLDM models. The generated models can then be used by the implementation and analysis tools for golden signoff.

Furthermore, special SRAM setup/hold timing checks are also performed as part of the static timing analysis process to ensure accuracy. Below are some of the setup checks that are made possible using NanoTime, the transistor-level STA tool:

- wordline off before precharge on
- wordline on before read mux on
- write enable on before wordline on
- data valid before wordline off
- read mux on before sense amp enable on
- sense amp enable on before read mux off
- precharge off before read mux on
- sense amp precharge off before read mux on
- sense amp out precharge off before read mux on
- sense amp enable off before precharge on
- sense amp enable off before sense amp precharge on
- sense amp enable off before sense amp out precharge on
- precharge off before write enable on
- input valid before write enable on

Customized hold checks are also available:

- wordline on after precharge off
- write enable off after wordline off
- data valid after wordline off
- precharge on after read mux off
- sense amp precharge on after read mux off
- sense amp out precharge on after read mux off
- sense amp enable on after precharge off
- sense amp enable on after sense amp precharge off
- sense amp enable on after sense amp out precharge off
- precharge on after write enable off
- input valid after write enable off

Using traditional dynamic simulation, simple checks like those listed above would take a long time to execute without the confidence of completeness. However, with the STA approach, all checks are done exhaustively and timing models are generated quickly for full-chip SoC signoff.

Conclusion

In summary, the STA-based memory characterization flow demonstrates many benefits over traditional dynamic simulation memory characterization approaches. It allows the user to perform memory characterization timing checks without the need for simulation vectors and with the confidence that the verification coverage is complete. Other standard features for memory analysis with the transistor-level STA tool include automatic critical path identification, SPICE netlist extraction, signal integrity crosstalk delay and noise analysis, and CCS timing and noise model generation. The tool is designed to exploit memory core array regularity and abstraction to support large memories at the expected STA performance. Most importantly, the accuracy of the analysis is guaranteed to within ±5% of HSPICE.

Synopsys, Inc. • 700 East Middlefield Road • Mountain View, CA 94043 • www.synopsys.com