sponsored by

Gabe on EDA • EDAmarket

Assembling the Future

A Newsletter About the Design and Production of Electronics

ISSUE 020 • OCTOBER 2012

SUBSCRIBE

PDF and Archives

twitter

In this issue:

A Difficult Book About A Very Difficult Thing

Gabe Moretti

There is a new book out. It is not cheap at $119 a copy but I think the price is justified by the depth of its contents and the criticality of the subject. The book by Trent McConaghy, Kristopher Breen, Jeffrey Dyck and Amit Gupta has the imposing title of: "Variation-Aware Design of Custom Integrated Circuits: A Hands-on Field Guide" published by Springer.

My first reaction to its contents was to think that things cannot continue the way they are going. Complexity is killing productivity and financial returns. Of course one should not be surprised. Asking unnatural things from light is both difficult and expensive.

Reading the book and getting meaningful information from it requires some understanding of statistical analysis, so do not be scared off by the equations. After all our profession requires precision in communications, and there is nothing more precise than a mathematical expression. The goal is to teach engineers the technology of PVT analysis. Process variation (P), power supply

voltage (V), and temperature (T) are fundamental components in determining whether at the end you have soup or hogwash. And if the result is the latter it will be a very expensive one.

The book is not a reference publication, unless you are already very familiar with the techniques covered in the first three chapters. If you are, then chapters 4 and 5 will give you theory and techniques for 3-Sigma and High-Sigma verification and design, and chapter 6 will guide you through building better SPICE models. Otherwise read and digest the first three chapters. If by then you have not had the urge to change profession, go on and find out how to design circuitry that will yield more good die than before.

Just reading the foreword by Jim Hogan and the Table of Contents reminded me of the reason it is taking so long to get to commercial use of 20 nm process. it is clear that we need to change how we do things at the front end, since the problem generated in place and route are too expensive to solve. That will require discipline on the part of both architects and designers, as well as the willingness to accept less than optimal use of real estate.

Talking about real estate gave me an idea. Let's look at the die surface exactly the way local government looks at real estate: through zoning. Some areas are good for commercial development, some for parks and recreation, and some for residential buildings. And of course you can use even finer distinctions: condominiums and apartments, versus single family homes, for example. or heavy industry versus retail stores. You get the point. In this way we can have design rules that are specialized for the type of circuitry you want to create while at the same time you can only place that type of circuitry in zones of the die reserved for it. We know that blocks have different physical, and of course, logical characteristics. Let's not handle them all in the same manner, as if they were just a collection of transistors and wires.

The Benefits of Static Timing Analysis-Based Memory Characterization

Ken Hsieh, product marketing manager, Synopsys

Having fast and accurate models at all stages of a design is essential if SoC designers are to succeed in designing chips with embedded memories. That’s why embedded memory characterization is of increasing concern to design teams. However, the move to new process geometries is exacerbating the challenge – the number of memory instances per chip increases significantly at advanced process nodes. To support the full range of process, voltage, and temperature corners (PVTs) and to cater to the sensitivity of process variation, designers have to perform more and more memory characterization runs. On top of that, the data processing per characterization grows exponentially.

Modeling Memories: Two Approaches

There are two general approaches to creating memory models. The first is based on the characterization of the memory compiler-generated models (memory compilers are tools that automate the creation of many different and unique memories very quickly), and the other is an instance-based memory characterization. The characterization process involves fitting timing data to polynomial equations, where the design team derives the coefficients for the equations from a small sample of memory instances. Although this approach allows design teams to generate models very quickly, the resulting accuracy of the characterization is less than optimal.

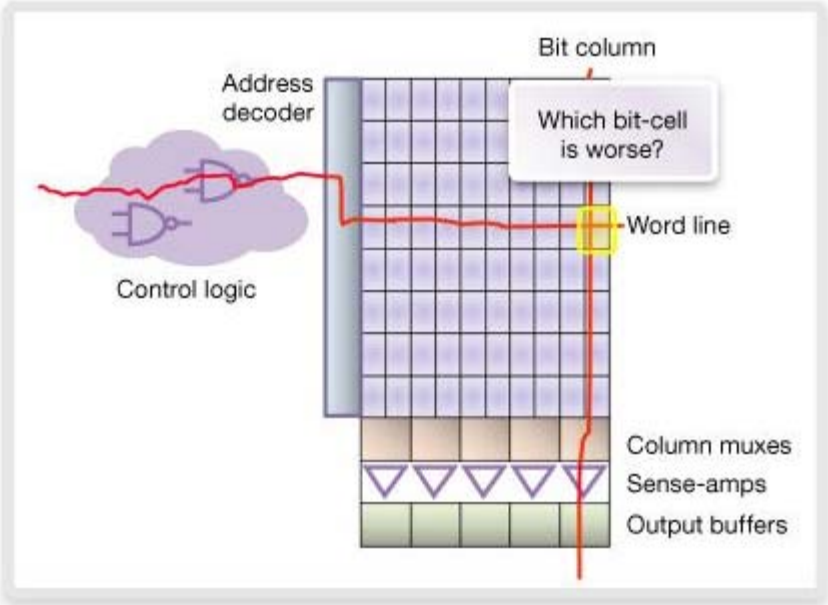
To overcome these inaccuracies, memory designers perform the characterizations specific to each memory instance over a range of PVTs, but this takes time. Although there are several approaches to improving the instance-based characterization throughputs, all of them require the use of SPICE or FastSPICE dynamic simulators to trade off between performance and accuracy. This still does not guarantee the full verification coverage needed to ensure silicon success.

Typically, design teams focus around 70% of the engineering effort and time performing memory characterization on timing analysis and model generation. With the traditional approach, given a specific scenario of input and clock transitions, designers commonly use a dynamic simulator such as SPICE or FastSPICE to determine if a particular sensitization leads to a timing violation in the circuit block. By simulating under all possible sequences of transitions, a designer can determine whether or not the block-under-test can operate at the given 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, so using dynamic simulation is impractical for all but very small blocks.

Static Timing Analysis for Memories

Timing verification is a process of validating that a design meets its specifications by operating at a specific clock frequency without errors caused by a signal arriving too soon or too late. Transistor-level static timing technology has been available for well over a decade. Today, the technology has evolved and expanded to cover a selected range of memory blocks like single/dual-port embedded SRAM, as well as the traditional “black-box” timing approach.

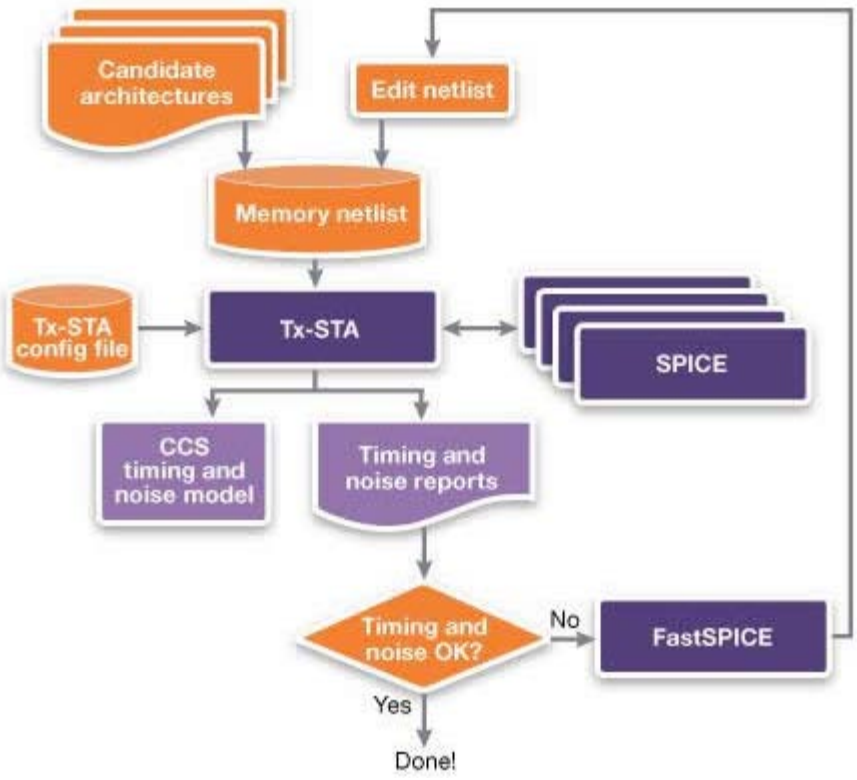
Unlike the dynamic simulation approach, static timing analysis (STA) tools remove the need for simulating the entire block under all possible scenarios. Instead, they use fast but accurate approaches to estimate the delay of sub-circuits within the block and use graph analysis techniques to quickly seek 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]

The latest development in STA technology has made it possible to time not only the control logic (transistor-level digital circuits) but also paths through the memory core array (i.e., bit-column, word-line, column-muxes, and sense-amps) as illustrated in Figure 1. Using Synopsys' advanced transistor-level static timing analysis solution NanoTime for memory characterization helps to improve design turnaround time and verification coverage, while maintaining accuracy to within ±5% of SPICE. NanoTime supports memory compiler- and instance-based memory characterization. It does not require netlist reduction techniques as commonly practiced in the dynamic simulation approach. Another significant advantage of using the STA approach with NanoTime is that the design team does not have to create vectors to perform the timing analysis. This alone saves tedious verification planning and processing time as well as reduces the potential of human errors when generating the stimulus for the dynamic simulations.

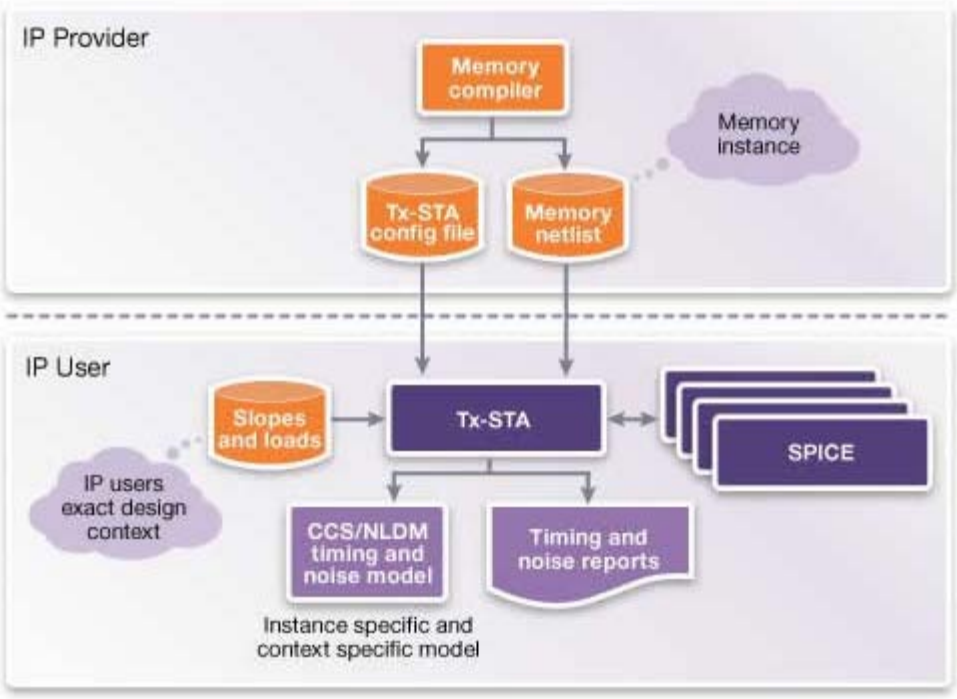
Figure 2 illustrates a typical memory architecture design and characterization flow based on the STA technology. In this flow, the design team uses NanoTime to identify timing violations in the memory designs and forwards the information to SPICE/FastSPICE to further determine what went wrong.



[Figure 2: Memory architecture design and characterization flow]

Modeling teams also use SPICE and FastSPICE simulators 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, it is possible to establish the STA characterization flow to perform the tasks illustrated in Figure 3.



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

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

NanoTime also performs special SRAM setup/hold timing checks as part of the STA process to ensure accuracy. Using traditional dynamic simulation, simple setup and hold checks would take a long time to complete if the designers weren't confident that the simulation tools had done all of the checks exhaustively. But the STA approach gives peace of mind, and it generates timing models quickly for full-chip SoC signoff with gate-level STA tools like PrimeTime.

Summary

The STA-based memory characterization flow using NanoTime has many benefits over traditional dynamic simulation memory characterization approaches. It allows the user to perform memory characterization timing checks without needing simulation vectors and with the peace of mind that the verification coverage is complete. NanoTime's standard features include automatic critical path identification, SPICE netlist extraction, SI crosstalk delay and noise analysis, and CCS timing and noise model generation for memories. It is designed to exploit memory core array regularity and abstraction to support large memories at the expected STA performance. But, most importantly, the accuracy of the analysis is guaranteed to within ±5% of HSPICE.

Memory Timing Analysis and Characterization using NanoTime

NanoTime | HSPICE | PrimeTime

About the Author

Ken Hsieh is a Product Marketing Manager for NanoTime and ESP-CV at Synopsys, Inc. He has 23 years of EDA experience with the past 15 years focusing on EDA marketing. Ken had worked for EDA companies like Nassda, IKOS, Mentor Graphics and Cadence in various application engineering and sales roles before joining Synopsys in 1998. Ken received his Bachelor of Science in Electrical Engineering from Texas A&M University and his Master of Engineering in Electrical Engineering from California State Polytechnic University, Pomona.

Low Power, High Verification Complexity

Cary Chin Director Technical Marketing, Synopsys, Inc.

When I started in the industry back in 1982, our “state-of-the-art” designs contained a whopping 2K equivalent logic gates, quickly progressing to 4K, and then a monstrous 8K gates over the next few years. Back then, the task of verifying a digital design was pretty simple: look at the schematics, write out some vectors to simulate, and verify that the correct data came out. A good verification engineer could get through a 2K design with pretty good coverage in a couple of weeks.

Thirty years later, in a remarkable testament to Moore’s law, we are indeed nearing one million (220) times the device density as we contemplate devices with hundreds of millions to billions of gates. Assuming that we engineers haven’t gotten that much smarter (I know I haven’t), the productivity of one engineer-week to verify one thousand gates leads us to an initial estimate of one million engineer-weeks to verify our billion gate design. That’s about 20 years of elapsed time even with a team of 1000 engineers. But that’s not all. We all know that because of interactions and dependencies, the verification problem grows hyper-linearly. Suddenly even our 20 year, 1000 person verification cycle estimate seems optimistic.

Of course, the entire industry would have imploded long ago if the above were the only solution. The one thing about engineers is that we are a clever bunch. Faced with problems that are spiraling out of control, we always manage to develop just the right new algorithm or heuristic, the latest methodology, or simply expand the compute farm. Engineering is all about identifying problems, and coming up with solutions to those problems that enable us to come up to the next problem.

In the case of functional verification, methodologies that are scalable, tools to implement those methodologies, and parallelized machine resources have enabled us to deliver devices that function correctly with an unimaginable level of complexity. But can we keep it up?

One of the main focuses of processor and SoC design in recent years has been reducing power consumption. With chip densities exploding along the path predicted by Moore’s law, and technology nodes shrinking from 90nm through 65nm, 40nm, 28nm and beyond, problems associated with power consumption (both static “leakage” power and dynamic “switching” power) threaten to derail our ability to produce functional devices. And, as usual, clever engineering has produced solutions from silicon technology improvements through tools and methodologies that have skirted the inevitable collision between Moore’s Law and physical reality.

But what is the cost in terms of verification complexity? It turns out the cost is substantial. Advanced low power design techniques revolve around two main goals:

1. Shut down regions that are not being used to minimize the static (leakage) power wasted in the system.

2. Lower the operating voltage as much as possible to reduce dynamic power consumption, because switching power is proportional to voltage squared.

While these goals are simple, their implementation on today’s advanced devices is extremely complex. Unpowered devices on a chip can create floating nodes, which can impact device performance and functionality, and ironically, increase power consumption. These intermediate points must be isolated so that that any impacted circuits take on known values when their drivers are turned off. And in the case of voltage reduction, the interfaces between voltage areas must be

carefully checked in all operating modes. Further, where voltages can be dynamically changed during operation, the entire range of voltages must be verified so that intermediate voltages do not create issues with functionality or reliability.

Traditional functional verification takes place with an implicit assumption that all devices are powered on with stable power supplies. With these new advanced low power design technologies, the functional verification problem that was already exploding due to device complexity becomes an order more complex. All voltages and power states must be carefully checked, and proper functionality must be guaranteed not only in each state, but also in every possible transition between states. This complexity is the semiconductor industry's biggest challenge. In order to avoid 20 (or 200!) years of verification, we must once again look to well-designed methodologies that are power-aware and scalable, and integrated power-aware design and verification tools to maximize the benefit of our compute resources. After all, those trillion gate chips are just around the corner!