cadence®

Addressing Memory Characterization Capacity and Throughput Requirements with Dynamic Partitioning

By Federico Politi and Ahmed Elzeftawi, Cadence Design Systems

Typical memory characterization techniques using memory compilers and instance-specific memories have a number of tradeoffs—development time, accuracy, performance, and more. Ad-hoc instance-specific characterization methods such as dynamic simulation, transistor-level static timing analysis, and divide-and-conquer suffer from multiple limitations that prohibit usage for 40nm technologies and below. Dynamic partitioning offers an alternative that provides a fast and accurate memory characterization methodology enabling designers to improve design performance, reduce die area, and closely correlate with manufactured silicon. This white paper discusses tradeoffs of these memory characterization techniques and how designers can choose the best solution for their application.

Contents

Introduction

The ubiquitous and ever-increasing use of microprocessors in system-on-chip (SoC) designs has led to a significant proliferation of embedded memories. These chips can have more than 100 embedded memory instances, including ROMs, RAMs, and register files that consume up to half the die area. Some of the most timing-critical paths might start, end, or pass through some of these memory instances.

The models for these memory elements must accurately account for voltage, timing, power, and process variability to enable accurate chip verification. Memory characterization is the process of abstracting a memory design to create an accurate timing and power Liberty model (.lib) that is most commonly used by downstream digital implementation and signoff flows. Ad-hoc approaches to memory characterization do not accurately model the data required for faithful SoC signoff, delaying tapeout and increasing the total cost of the design.

Memory characterization often requires thousands of SPICE simulations. The number of memory instances per chip and the need to support a wide range of process, voltage, and temperature (PVT) corners make these simulations a daunting task. The growing size of memory instances and sensitivity to process variation add more dimensions to this challenge. Further, the need to create library variants for high-speed, low-power, and high-density processes makes automating the memory characterization flow imperative.

This white paper discusses the benefits and challenges of several approaches to memory characterization, and proposes a new dynamic partitioning methodology that addresses the limitations of existing approaches and enables the capacity and throughput requirements for characterization of multimillion-bit memories.

Memory Characterization Methodologies

Broadly speaking, there are two main methodologies for memory characterization: characterizing memory compiler-generated models, and characterizing individual memory instances.

Memory Compiler Model Characterization

Memory compilers construct memory instances by abutted placement of pre-designed leaf cells (e.g., bit columns, word and bit-line drivers, column decoders, multiplexers, and sense amplifiers) and routing cells where direct connection is not feasible. Compilers also generate a power ring, define power pin locations, and create various electrical views, netlists, and any additional files required for downstream verification and integration.

Memory compilers do not explicitly characterize the generated cells, but instead create models by fitting timing data to polynomial equations whose coefficients are derived from characterizing a small sample of memory instances. This approach enables memory compilers to generate hundreds or thousands of unique memory instances with different address size, data width, column/row density, and performance. However, the model accuracy of this approach is poor.

To safeguard against chip failure due to inaccurate models, the memory compiler adds margins, which can lead to more timing closure iterations, increased power, and larger chip area. In addition, the fitting approach doesn't work well for the advanced current-based models, including effective current source model (ECSM) and composite current source (CCS), which are commonly used for timing, power, and noise at 40nm and below.

Instance-Specific Characterization

To overcome the inaccuracies of compiler-generated models, design teams resort to instance-specific characterization over a range of PVTs. This is a time-consuming process but yields more accurate results as compared to memory compiler libraries. However, often due to limitations in the characterization approach and available resources, the accuracy improvement is not as much as it could be, while the cost is high. Instance-specific characterization includes a variety of approaches, such as dynamic simulation, transistor-level static timing analysis, and ad-hoc divide-and-conquer.

Dynamic simulation

Memory can be characterized by treating the entire memory as a single black box and characterizing the whole instance using a FastSPICE simulator, enabling the creation of accurate power and leakage models that truly represent the activity of the entire memory block. FastSPICE simulations can also be distributed across a number of machines to speed simulation time. Unfortunately, this approach has disadvantages, as FastSPICE simulators trade accuracy for performance. Further, the black box approach still requires users to identify probe points for characterizing timing constraints. For a custom memory, the characterization engineer can get this information from the memory designer, but it is not available from memory compilers. In addition, this method doesn't work well for generating some of the newer model formats such as noise models, and cannot be scaled to generate Liberty models that account for process variation such as Liberty Variation Format (LVF).

Transistor-level static timing analysis (STA)

STA techniques that utilize delay calculators can be used to estimate the delay of subcircuits within the memory block to identify the slowest paths. This method yields a fast turnaround time, and does not require vectors to perform timing analysis. However, STA techniques usually identify false timing violations that require further analysis with SPICE/FastSPICE simulators.

The STA approach is also impaired by pattern matching, which needs to be constantly updated as new circuit structures are introduced across different designs and new revisions. If analog structures such as sense amps are in the critical paths of the clock-to-output data, setting up the STA approach becomes more demanding. Further, using transistor-level delay calculators, which are process- and technology-dependent, undermines its claimed SPICE accuracy, as it assumes that delays from active elements and the RC parasitic elements can be separated. However, this assumption is no longer valid due to the presence of parasitic elements between finger devices, which are a typical result of extraction on memory structures in advanced process nodes.

Finally, a static delay calculator is generally severely compromised—either in terms of runtime, accuracy, or both—by the presence of large transistor channel-connected regions. Therefore, a memory array is arguably the worst application for static timing analysis, especially with power-gating methodologies, which are commonly adopted for any memory design at 40nm and below.

Ad-hoc divide-and-conquer

Another approach to memory characterization is to statically divide the memory into a set of critical paths, characterizing each path using an accurate SPICE simulator and then integrating the electrical data from each component back into a complete memory model. This approach improves accuracy by using SPICE simulations with silicon-calibrated models. The simulations can also be distributed across a computer network with each critical path being simulated independently.

This approach has a number of disadvantages. For advanced node memories that have significant coupling or a virtual power supply network, the circuitry making up the critical path grows too large for a SPICE simulator to complete in a reasonable amount of time. The turnaround time to generate a variation model, especially for mismatch parameters, becomes prohibitively expensive with such a large circuit. Additional complications include the need to correctly identify the clock circuitry, memory elements, and critical path tracing through analog circuitry such as sense amps for different memory architectures with varying circuit design styles.

The variation in the memory architectures and usage styles (multiple ports, synchronous or asynchronous, scan, bypass, write-through, power-down, etc.) requires a significant amount of effort by the characterization engineer to manually guide the process, creating stimulus files, simulation decks, and other needed data. This makes the characterization process even slower and more error-prone.

Evolving Technology Requires New Approach

Memories now contain greater functionality, with new structures and techniques to minimize static and dynamic power consumption such as adding header and footer cells, power-off modes, and, in some cases, dynamic voltage and frequency scaling for multi-domain power management. In addition, advanced process node memories require more comprehensive characterization to acquire the complete data needed to capture coupling impact. The abstracted models must faithfully incorporate signal integrity and noise analysis effects, which requires more extracted parasitic data to remain in the netlist, increasing SPICE simulation run-times and reducing productivity.

There's a clear and pressing need for a new memory characterization approach with the following characteristics:

- High productivity, especially for large memory arrays, to accelerate the availability of models to both design and verification teams—plus it must offer high throughput and be easy to configure and control

- Extreme precision—the impact on parametric timing, noise, and power data from the characterization process must be minimized

- Tight integration with a leading-edge SPICE/FastSPICE simulation platform for uncompromised accuracy and scalable performance

- Robust and comprehensive support for all analysis and signoff features required by the downstream tools for corner-based and statistical verification methodologies

- Consistency with other characterized components, such as standard cells, I/O buffers, and other large macros, offering similar methods for the selection of switching points, measurement methodologies, and general look and feel

Dynamic Partitioning—Memory Characterization for 40nm and Below

Existing memory characterization methods, both block-based or static divide and conquer, can be augmented to address current characterization challenges using dynamic partitioning (see Figure 1).
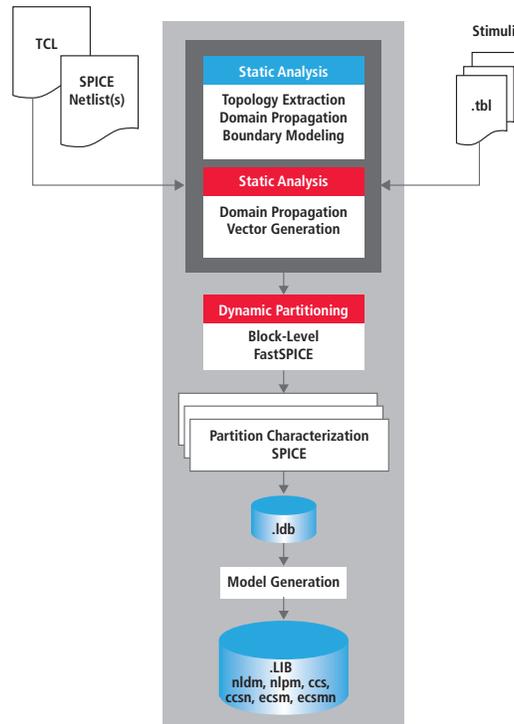

Figure 1. Dynamic partitioning can address characterization challenges by augmenting either block-based or static divide-and-conquer methods

Rather than relying on static path tracing, dynamic partitioning leverages a full-instance transistor-level simulation using a high-performance, large-capacity FastSPICE simulator, and acquires data-specific vectors to derive a record of circuit activity. The critical paths for each timing arc can be derived from the simulation results, such as from the clock-to-output bus or from the data or address buses and clock to each probe point. The probe points where the clock and data paths intersect can be automatically derived from a graph traversal of the circuit without requiring design-dependent transistor-level pattern matching. A "dynamic partition" can then be created for each path, including all the necessary circuitry along with any "active" side-paths such as coupling aggressors.

The dynamic partitioning technique is particularly effective for extracting critical paths through circuitry that contains embedded analog elements, for example, sense amplifiers along the clock-to-output path. Figure 2 shows a dynamic partition for a memory path from clock to output with less than a thousand transistors.
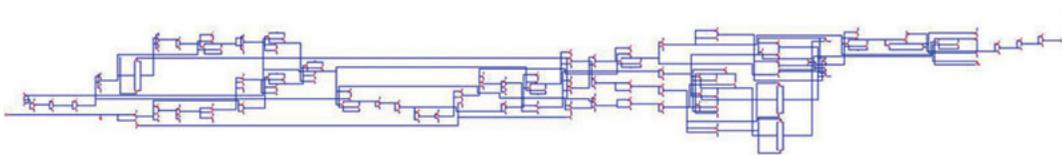

Figure 2. Dynamic partition for a memory path from clock to output

Another benefit of dynamic partitioning is that memories—like many analog blocks—exhibit the characteristic that partition boundaries can adjust during operation. A "read" operation following a "write" to a distinct location exercises different parts of the instance. By offering a more flexible and general-purpose solution, dynamic partitioning provides superior results in these situations. Once this comprehensive partitioning is complete, accurate SPICE simulations are performed independently on decomposed partitions, with the assembled results faithfully representing the timing behavior of the original, larger instance.

The dynamic partitioning approach uses the full-instance FastSPICE simulation to determine large-scale characteristics like power and partition connectivity, while the much smaller individual partitions, each containing typically less than a thousand transistors, are simulated using a SPICE simulator to ensure the highest levels of accuracy. The approach requires tight integration with a high-performance, high-capacity FastSPICE simulator that accommodates multi-million-bit memory cells with RC parasitic elements with superior turnaround performance. Communication between the full-block and partitioned sub-block simulations ensures identical initial conditions and DC solutions, essential for obtaining precise and consistent data. This analysis and partitioning technology can be coupled to sophisticated job control for parallel execution and distribution across the network.

In addition, the partitions can be represented as super-cells to standard cell library characterization platforms, enabling the generation of current source models (CCS and ECSM) for timing, noise, and power analysis, as well as the generation of statistical process variation timing models using the same proven statistical characterization methods for standard cells. The consistent application of a library-wide characterization approach ensures interoperability between all instances in the design, eliminating hard-to-find timing errors due to library element incompatibilities.

As each partition is simulated using a SPICE simulator, the generated timing models are nearly identical to simulating the complete block using SPICE, which is impractical for all but the smallest memory instances. The only source of error is related to tying off inactive gates and wires. Table 1 shows the accuracy of results from characterizing the entire block using the black-box FastSPICE approach and the results from dynamic partitioning comparing both to SPICE golden results. The accuracy difference due to dynamic partitioning is less than 1.5% for delay, transition, and constraints, while using the black-box FastSPICE approach results in up to a 4.5% difference.

Accuracy vs. SPICE	Delay	Transition	Contraints
Black-box FastSPICE	4.50%	3.17%	3.50%
Dynamic partitioning	1.20%	0.90%	0.50%

Table 1. Accuracy difference vs. SPICE for black-box FastSPICE and dynamic partitioning

In addition to delivering accuracy, dynamic partitioning greatly improves the CPU time and total turn-around time for memory characterization by an order of magnitude or more. Table 2 compares two memory arrays showing the number of bits, transistor count of the full memory instance, and transistor count of a typical partition created using the dynamic partitioning approach. The table shows that the partition size is virtually independent of the memory size that keeps SPICE simulation runtime constant.

Memory Array Size	16x8	2048x256	Ratio
Bits	128	524,288	4,096.0
Transistor count	13,455	3,328,009	247.3
Dynamic partition transistor count	1,023	1,420	1.4

Table 2. Dynamic partitioning transistor count independent of memory array size

In contrast, Table 3 shows the total CPU time improvement for memory instance characterization using a dynamic partitioning approach compared to a black-box FastSPICE approach.

Memory Type	Speedup
288Kb SRAM	10X
1Mb Custom memory	31X
36Kb SRAM	67X

Table 3. Performance improvement from dynamic partitioning vs. black-box FastSPICE approach

The dynamic partitioning approach can be quickly deployed either for instance-based characterization or integrated into a memory compiler. The additional information required is minor—either stimulus or functional truth-tables derived from the memory datasheet. It is applicable to all flavors of embedded memory such as SRAM, ROM, CAM, and register files, as well as to custom macro blocks such as SERDES and PHY.

Library Characterization Suite Utilizes Dynamic Partitioning

Cadence offers a comprehensive Liberty library characterization suite for memory and mixed-signal blocks utilizing the dynamic partitioning technology in addition to nominal characterization, statistical characterization and variation modeling, and library validation for standard cells and I/Os. The dynamic partitioning technology enables macro characterization to be within 2% accuracy compared to reference SPICE simulations, yet orders of magnitude faster.

Cadence® Liberate™ Characterization Suite includes the following solutions that support dynamic partitioning. Both use the same modeling technology as the foundry-trusted Virtuoso® Liberate Characterization Solution to create Liberty models with timing, power, leakage, noise, and advanced timing and noise models.

- **Virtuoso Liberate MX Solution** enables the characterization of large memory and custom blocks utilizing dynamic partitioning technology to optimize runtime, yet maintain the accuracy needed for Liberty models. Further, the Liberate MX Solution can run the Virtuoso Variety Statistical Characterization Solution to create LVF models for memory instances.

- **Virtuoso Liberate AMS Solution** provides mixed-signal characterization utilizing hybrid partitioning technology comprised of static analysis and partitioning methods in addition to dynamic partitioning technology. The Liberate AMS Solution enables a one-step .lib generation for timing, power, leakage, noise, and advanced timing and noise models.

The Liberate Characterization Suite also includes the following tools for standard cells, I/Os, and variation modeling to ensure consistent library characterization and modeling methodology across all design IP.

- **Virtuoso Liberate Characterization Solution,** the foundry-trusted foundation of the characterization suite, provides ultra-fast library characterization of standard cells and complex I/Os, with advanced timing, power, and noise models.

- **Virtuoso Liberate LV Library Validation Solution,** a comprehensive library validation system, provides library function equivalence and data consistency checking, revision analysis, and timing and power correlation.

- **Virtuoso Variety Statistical Characterization Solution** enables modeling of random and systematic process variation, AOCV/SOCV table generation, and LVF. The tool calculates delay sensitivity to global and local variation, is fully distributed and multi-threaded, and avoids costly Monte Carlo analysis runs.

All products in the Liberate Characterization Suite share a common infrastructure covering netlist processing, static analysis, stimulus generation, and simulation automation to improve characterization productivity. Further, all products are tightly integrated and validated with with Cadence Spectre® Accelerated Parallel Simulator for accurate SPICE simulations. For memory characterization, the Liberate MX Solution is tightly integrated with Cadence Spectre eXtensive Partitioning Simulator (XPS) for block-level FastSPICE and with the accurate partition-level SPICE simulations performed by Spectre Accelerated Parallel Simulator.

Conclusion

Applying dynamic partitioning to large memory instances eliminates the errors seen in solutions that rely on transistor-level static timing analysis, use the black box FastSPICE approach, or statically divide the memory. This new methodology provides the most precise and comprehensive models for design, verification, and analysis available today, while enabling very fast runtimes, often an order of magnitude faster for large macros.

With the superior level of precision offered by dynamic partitioning, designers can be much more selective in the application of timing margins during design and verification. As a result, they can achieve improved design performance, optimal die area, and closer correlation with manufactured silicon. When coupled with a proven standard cell library characterization platform, this characterization methodology provides a more consistent solution for cells and macros, which are critical for electrical signoff and analysis of large and complex SoC designs. Utilizing dynamic partitioning, designers benefit from comprehensive, accurate, and efficient models, and achieve a significant speedup in total turnaround time.

cādence®

Cadence Design Systems enables global electronic design innovation and plays an essential role in the creation of today's electronics. Customers use Cadence software, hardware, IP, and expertise to design and verify today's mobile, cloud and connectivity applications. www.cadence.com