ADDRESSING SRAM VERIFICATION CHALLENGES

Steve Williams, Principal Engineer, sureCore Limited.
Stefan Cosemans, Principal Design Engineer, sureCore Limited.
Dena Burnett, Technical Marketing Lead, Solido Design Automation.
Amit Gupta, CEO, Solido Design Automation.

Introduction

SureCore Limited is an SRAM IP company based in Sheffield, UK, developing low power memories for current and next generation silicon process technologies. Its award-winning, world-leading, low power SRAM designs are process independent and variability tolerant, making them suitable for a wide range of technology nodes. Two major product families have been announced PowerMiser^{TM} and EverOn^{TM}. PowerMiser^{TM} is a general purpose SRAM capable of delivering in excess of 50% dynamic and 20% static power savings compared to industry standard offerings. EverOn^{TM} is a memory developed specifically for the IoT and wearable markets. It delivers near-threshold operating voltages facilitating extremely low power operation. Both product families are based on standard foundry bit cells and no process modifications are needed to deliver these capabilities. Key to achieving market leading low power performance is a comprehensive verification strategy. In this paper, co-written with our partners at Solido Design Automation, the key elements of this strategy will be explored.

Verification is an integral part of any integrated circuit development process. The verification process must establish that the design meets its specified yield and performance criteria over the full range of operating conditions before tape-out sign-off. The process generally involves taking abstractions of the design in appropriate forms, for example post-layout extracted netlists, and running simulations to validate the design performance. The verification process must address many different aspects of yield and performance, so several different types of design abstraction and simulation tooling may be required to complete the process. In the case of SRAM this is particularly true.

Verification of a complete compiler instance space presents several unique challenges. These include, but are not restricted to: (1) the need to maximise the coverage over the entire instance space of the compiler range, and (2) the ability to validate design performance and parametric yield sufficiently over the PVT range. It is essential therefore that SRAM verification is based on a variation-aware strategy.

These challenges also have to be addressed within a viable design timescale. To meet this goal, the overall verification task is split into several unique sub-tasks. These include;

- Behavioural model validation
- Full operating mode functional verification
- Top level variation aware parametric functionality
- Cell level parametric yield validation to 6σ

Each of these tasks involves different levels of design abstraction and employs different simulation strategies and toolsets.

These challenges are made particularly onerous when verifying near-threshold SRAM solutions, such as the sureCore EverOn^{TM} family. In order to realise significant power savings at the system level, this SRAM family operates across a very wide operating voltage range, from nominal supply voltage down to near-threshold operation. For example, in a commercially available 40ULP process node, the EverOn^{TM} SRAM supports supply voltages from 1.21V down to 0.6V across process corners and temperature (from -40°C to +125°C). The memory is built around the foundry's high-density low-leakage bitcell. Simulations have demonstrated that a combination of assist features achieves better than 6σ parametric cell yield in the worst PVT corner. Near-threshold designs that provide such operating ranges demand an extensive approach to verification that relies on a range of validation strategies. These include focussed parametric tests run with Monte Carlo (MC) analysis across the PVT range and high sigma analysis, using Solido Variation Designer.

Within the context of SRAM development, the verification process is complimented by the characterisation process[1], that extracts data for a particular memory in order to facilitate SoC integration flows.

Variation- and verification-aware design

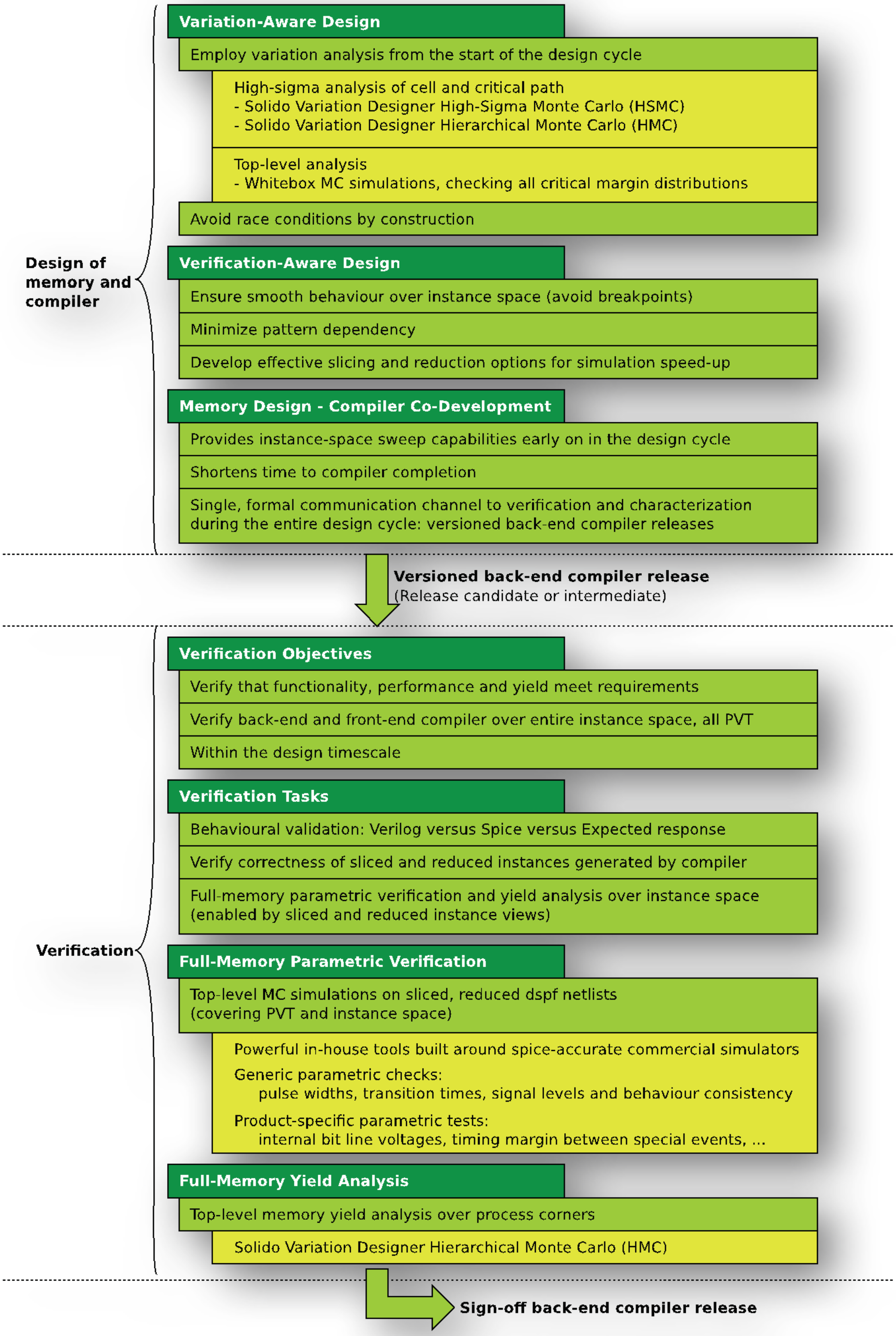
SureCore develops memory compilers that push the boundaries of low power performance. Obtaining high yield is essential, and achieving this while pushing such boundaries can only be achieved if variation considerations are the first step in the design, not an afterthought. Figure 1 shows a simplified depiction of sureCore's design and verification approach.

One of the first steps in the design process is the high-sigma analysis of the cell operation and of the critical bit slice. For this, Solido's High-Sigma Monte Carlo (HSMC) or Hierarchical Monte Carlo (HMC) tool is used. This involves dedicated test benches to test cell read stability, writeability and read correctness (including cell, bit line and sense amplifier), as well as the offset of the sense amplifiers separately. In designs with hierarchical bit lines, additional test-benches are required that include the global sense amplifiers and local write amplifier. For cell-level analysis, HSMC is the right tool, while HMC allows statistical correctness when considering slices where some instances occur more often than others, such as cells and sense amplifiers. In this first phase, ideal excitations are used for the control signals. Later in the design process, these simulations are repeated with the control signals as generated by the actual timing circuit. Although the tools provide a classifier approach that allows the use of non-smooth metrics such as binary outcomes, it is preferential to use well established metrics such as dynamic SNM_{read} and WTP at this stage for the additional insights they provide. As these metrics are smooth and well understood, extrapolation of the distributions from a normal MC run to the tails might seem attractive – this however does not give sufficiently accurate estimates of the actual tail probabilities. When other metrics are used, such as the read current or vdd_{min}, extrapolation will lead to results which can be drastically inaccurate. As such an HSMC approach is mandatory.

Memory compiler verification is a considerable undertaking, so the memory design should aim at making verification as easy as is feasible given the other constraints – verification-aware design. This includes avoiding breakpoints in the instance space and limiting access pattern dependencies. Another crucial aspect is the development of effective slicing and reduction options which provide crucial simulation speed-up. Together, they bring simulation time for a large memory instance down from 2 hours to 2.5 minutes and drastically reduce server memory load. These algorithms are implemented in the back-end compiler[2]. By co-developing the memory design and the compiler, these simulation runtime improvements are not only available for verification and characterisation tasks, but also to the design team.

[1] Additional information can be found in sureCore's white paper "Addressing memory compiler characterisation challenges"

[2] Additional information on slicing and reduction can be found in sureCore's white paper "Efficient memory compilation"



Design of memory and compiler

Variation-Aware Design

- Employ variation analysis from the start of the design cycle
  - High-sigma analysis of cell and critical path
    - Solido Variation Designer High-Sigma Monte Carlo (HSMC)
    - Solido Variation Designer Hierarchical Monte Carlo (HMC)
  - Top-level analysis
    - Whitebox MC simulations, checking all critical margin distributions
- Avoid race conditions by construction

Verification-Aware Design

- Ensure smooth behaviour over instance space (avoid breakpoints)
- Minimize pattern dependency
- Develop effective slicing and reduction options for simulation speed-up

Memory Design - Compiler Co-Development

- Provides instance-space sweep capabilities early on in the design cycle
- Shortens time to compiler completion
- Single, formal communication channel to verification and characterization during the entire design cycle: versioned back-end compiler releases

Versioned back-end compiler release
(Release candidate or intermediate)

Verification

Verification Objectives

- Verify that functionality, performance and yield meet requirements
- Verify back-end and front-end compiler over entire instance space, all PVT
- Within the design timescale

Verification Tasks

- Behavioural validation: Verilog versus Spice versus Expected response
- Verify correctness of sliced and reduced instances generated by compiler
- Full-memory parametric verification and yield analysis over instance space (enabled by sliced and reduced instance views)

Full-Memory Parametric Verification

- Top-level MC simulations on sliced, reduced dspf netlists (covering PVT and instance space)
  - Powerful in-house tools built around spice-accurate commercial simulators
  - Generic parametric checks:
    pulse widths, transition times, signal levels and behaviour consistency
  - Product-specific parametric tests:
    internal bit line voltages, timing margin between special events, ...

Full-Memory Yield Analysis

- Top-level memory yield analysis over process corners
  - Solido Variation Designer Hierarchical Monte Carlo (HMC)

Sign-off back-end compiler release

Figure 1. SureCore's design and verification approach

Verification

Behavioural Validation

The sureCore memory compiler produces several views for system-level validation and integration. Amongst these is a behavioural back-annotatable Verilog model for RTL and gate level simulation. It is imperative that this model accurately reflects the behaviour of the physical design. The sureCore memory compiler comprises of 2 parts: (1) the Front-End compiler (FEC) that creates views for the 'front-end' of the design cycle (such as the behavioural Verilog model), and (2) the Back-End compiler (BEC) that creates all of the physical design views for final integration (GDSII/CDL). Functional accuracy of the Verilog model is validated using the FEC to generate the Verilog model and the BEC to generate an equivalent Spice netlist. Both views are tested against a set of common tests and expected responses derived from the test stimuli, as shown in Figure 2. The suite of test sequences is designed to cover the full range of operating configurations.

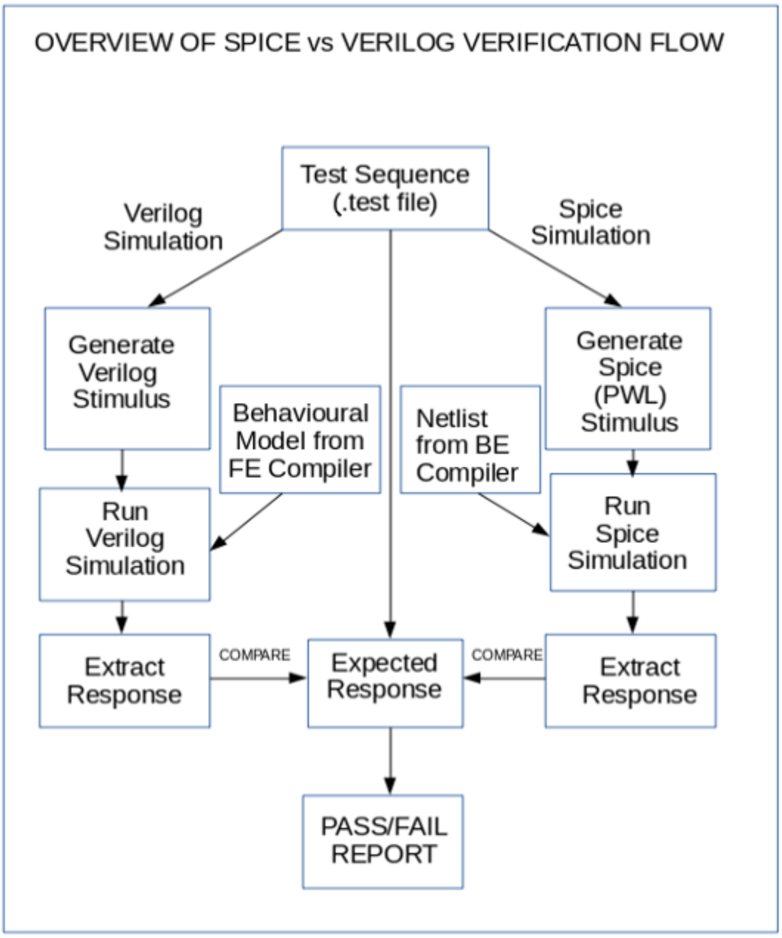


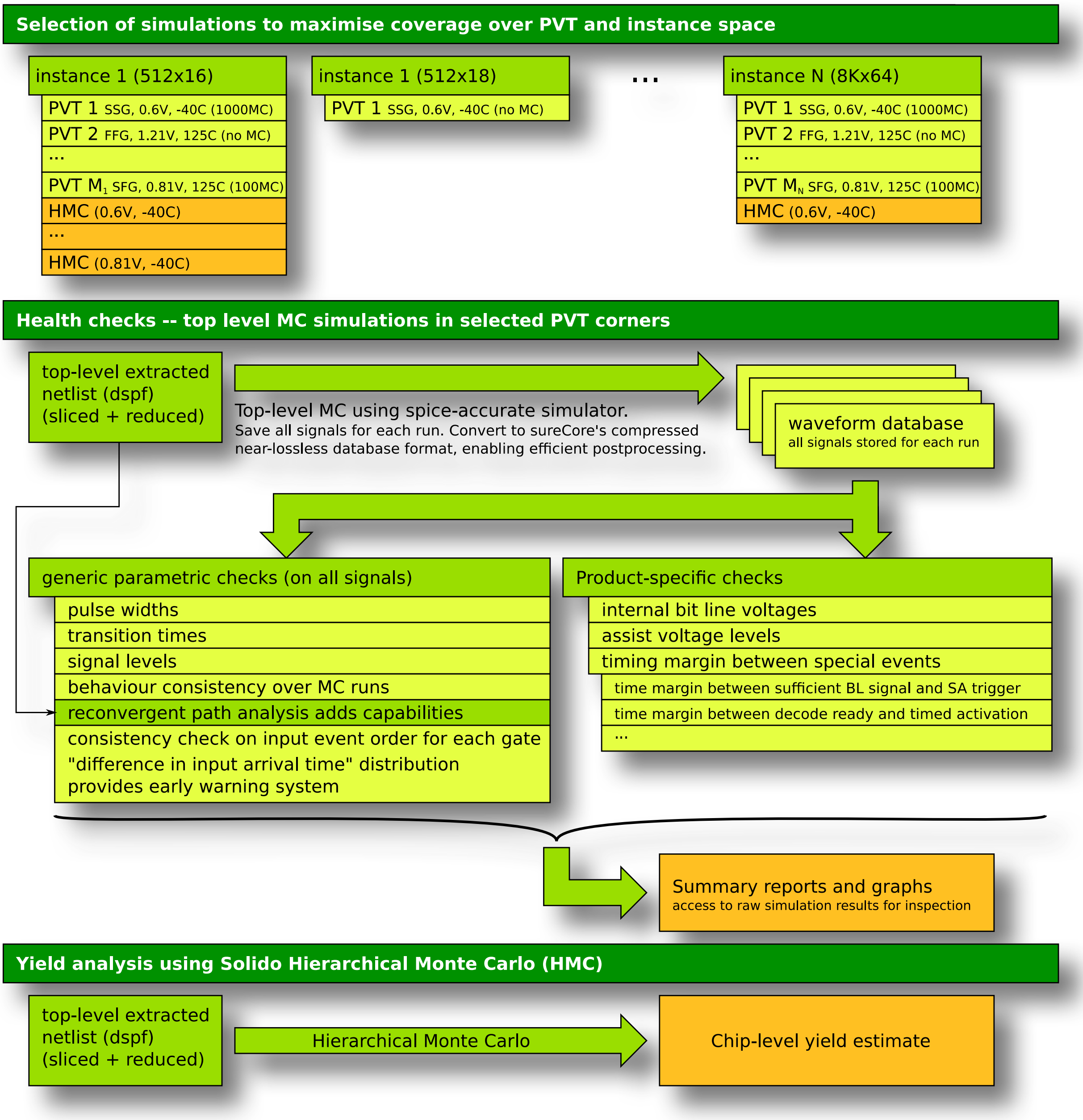
Figure 2: Verilog vs Spice Verification Flow

Variation-aware Full-Memory Parametric Verification

The sureCore verification flow includes a range of targeted parametric tests. In addition to validating basic functional write-read operations, these tests also validate parametric performance over the full range of specified PVT corner points. In the case of the sureCore EverOn^{TM} family implemented on a 40ULP process node, these corners cover an operating voltage range from 0.6V to 1.21V and a temperature range from -40˚C to 125˚C. This is in addition to all the process corners.

The tests are run using Monte Carlo simulations that are executed at the top level, full memory instance view using sliced and reduced netlists. The netlist slice and reduction algorithms are separately validated for accuracy. The verification checks are structured to maximise test coverage across the compiler instance space.

Figure 3 shows a simplified depiction of the scripted parametric verification flow. It works by analysing the saved waveform databases containing all signals at the full memory level from every Monte Carlo run performed on every selected PVT and instance space corner. Analysing such complete waveform databases and comparing behaviour across the different MC and corner runs allows a wealth of information to be extracted regarding the parametric health of the design, leading to the establishment of confidence in projected performance and yield capabilities.



Selection of simulations to maximise coverage over PVT and instance space
instance 1 (512x16)
PVT 1 SSG, 0.6V, -40C (1000MC)
PVT 2 FFG, 1.21V, 125C (no MC)
...
PVT M1 SFG, 0.81V, 125C (100MC)
HMC (0.6V, -40C)
...
HMC (0.81V, -40C)
instance 1 (512x18)
PVT 1 SSG, 0.6V, -40C (no MC)
...
instance N (8Kx64)
PVT 1 SSG, 0.6V, -40C (1000MC)
PVT 2 FFG, 1.21V, 125C (no MC)
...
PVT MN SFG, 0.81V, 125C (100MC)
HMC (0.6V, -40C)
Health checks -- top level MC simulations in selected PVT corners
top-level extracted netlist (dspf) (sliced + reduced)
Top-level MC using spice-accurate simulator.
Save all signals for each run. Convert to sureCore's compressed near-lossless database format, enabling efficient postprocessing.
waveform database
all signals stored for each run
generic parametric checks (on all signals)
pulse widths
transition times
signal levels
behaviour consistency over MC runs
reconvergent path analysis adds capabilities
consistency check on input event order for each gate
"difference in input arrival time" distribution provides early warning system
Product-specific checks
internal bit line voltages
assist voltage levels
timing margin between special events
time margin between sufficient BL signal and SA trigger
time margin between decode ready and timed activation
...
Summary reports and graphs
access to raw simulation results for inspection
Yield analysis using Solido Hierarchical Monte Carlo (HMC)
top-level extracted netlist (dspf) (sliced + reduced)
Hierarchical Monte Carlo
Chip-level yield estimate

Figure 3. Parametric verification and yield analysis.

The parametric checks analysed on every internal node include measurements on the signal transition times, pulse widths, signal levels and signal behaviour consistency. The capabilities of the automatic checks are further enhanced by sureCore's in-house Reconvergent Path Analysis tool, described in the next section. This tool determines all gates in the design with multiple active inputs, and checks the relative order of these events. In addition to these generic checks, a set of targeted product family specific parametric tests are included in the standard flow. These product specific tests will differ between sureCore EverOn^{TM} and PowerMiser^{TM} families for example, and will include checks on identified critical parametrics such as the measured internal bit line voltages at the associated sampling trigger point (Figure 4).

Information about each measured parametric test from every Monte Carlo batch run on each PVT/instance corner is collated into a summary report for ease of interpretation. The summary collates information about the maximum and minimum bounds observed on each parameter and measured against a specified test limit. The summary log is supported by the generation of a complete results database that allows examination of distributions and statistical analysis to be carried out where further investigation may be required.

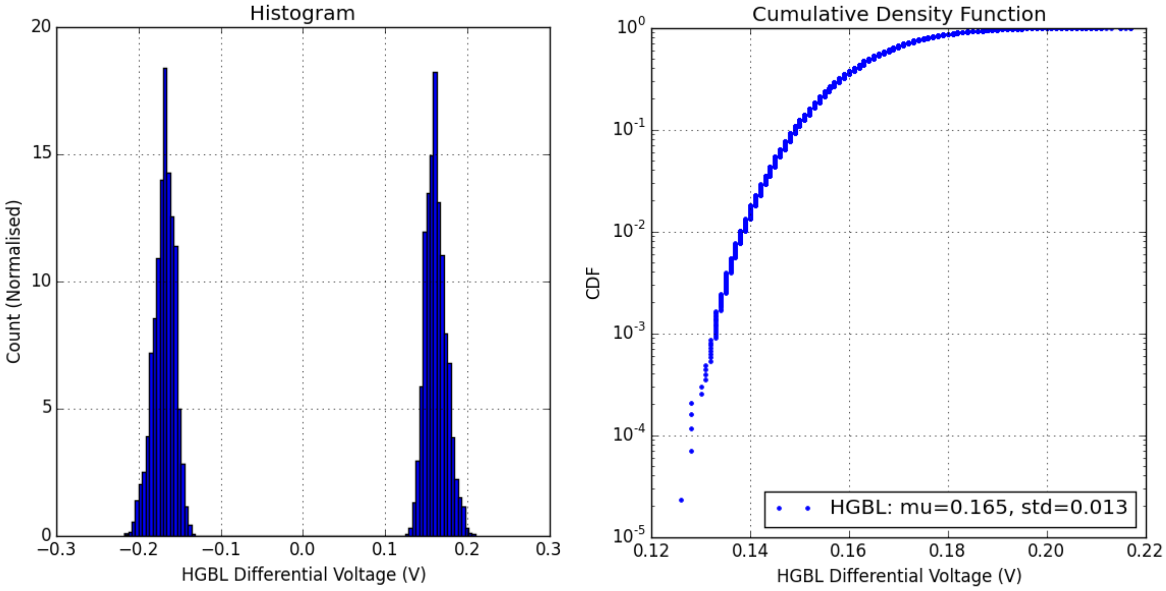


Figure 4: Example parametric distribution, one of many captured by the automatic parametric health checks. Even in the worst PVT corner (0.6V, SSG, -40˚C), sufficient global bit line signal is available.

Reconvergent Path Analysis

To further strengthen the verification effort, sureCore developed a Reconvergent Path Analysis tool. This tool extracts all gates and their connectivity from the dspf netlist. A very limited amount of configuration has to be provided to properly deal with virtual supplies, pass gate logic and special constructs such as the local bit lines. This information is then combined with the simulation waveform database. One way to use this is to visualise the activity in the memory, as shown in Figure 5. The triangles indicate rising and falling edges at the output of a gate, the lines between triangles indicate that one output signal is the input for another gate. Some special events are also highlighted. This interactive graph provides a wealth of information to the memory designer.

The same gate information can be used to extend the automatic verification flow. For gates that have multiple active inputs, the input events should always arrive in the same order for all MC runs. For example, for a WL driver, the output of the predecoders should be ready before the timed activation signal arrives, otherwise timing control is lost. When this happens in the tail of the distribution, decoder delay variation causes word line pulse shrinking, which creates an unexpected heavy tail towards short word line pulse widths. If no explicit check on the order of these signal is in place, then it would be very easy to overlook this problem when running only a few thousand MC simulations since the issue doesn't immediately manifest in the WL pulse width, let alone in the behaviour at the IO ports. Reconvergent Path Analysis finds all gates with multiple active inputs and checks the distribution of the relative timing between the signals. If input order violations are likely to happen within the yield target, Reconvergent Path Analysis can flag these, even if the actual situation did not occur in the performed MC simulations.

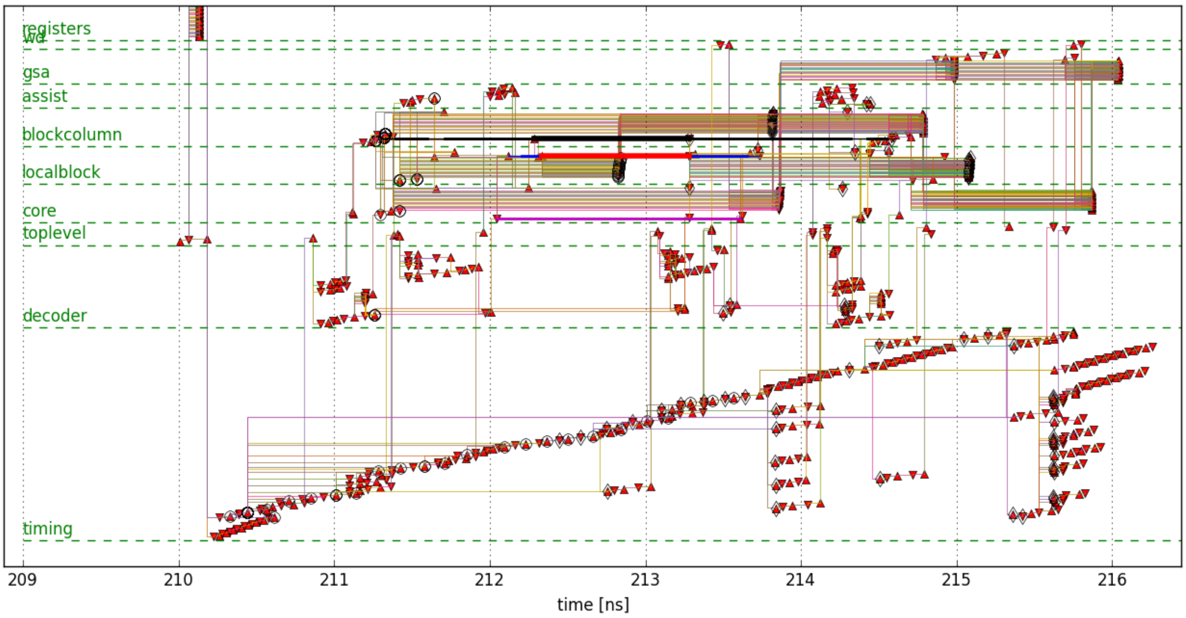


Figure 5: Visualisation of the activity inside the memory (read at reduced voltage with several assists enabled). The same information is used in Reconvergent Path Analysis.

Variation-aware Yield Analysis Validation

As chips become more complex, the chance of failure also increases, creating difficulty in measuring the effects of variation on designs quickly and accurately. Often, extra margin is added to compensate for this uncertainty, sacrificing power, performance, and area. Two available tools for SRAM yield analysis verification and validation are Solido Design Automation's High-Sigma Monte Carlo (HSMC) and Hierarchical Monte Carlo (HMC). Both of these variation-aware techniques meet requirements for fast, accurate, scalable, and verifiable techniques for reducing margins in near-threshold designs. These tools are an intrinsic part of the sureCore verification process.

Solido's HSMC approach[3] produces an accurate high-sigma (greater than 3σ) analysis of a distribution by optimizing the specific statistical sampling, reducing the number of required SPICE simulations to accurately realize a particular yield assessment. The HSMC approach prioritizes the cases that are most likely to fail, focusing on the worst-case scenarios, therefore streamlining the number of SPICE simulations required. This technique targets analysis on the extreme tail of a distribution, providing a lean process using fewer resources and simulations to analyse cases where verifiable analysis is most needed. Instead of running all simulations, HSMC provides accurate information in orders of magnitude fewer simulations, reducing over- or under-design in near-threshold situations.

HSMC can provide accurate information about the behaviour of a design at the extreme tail of a distribution, making it an ideal tool for fast and accurate high-sigma Monte Carlo analysis. In bitcell analysis for example, HSMC is typically able to find the first 100 failures within the first 5000 simulated samples. In traditional Monte Carlo analysis, finding the same number of failures would typically require up to 1.5 million samples, often without finding a single failure in the first 5000 samples[4]. Including HSMC accurately accelerates the design loop by reducing potential design iterations and the need for over-margining in worst-case situations, which is of crucial importance for near-threshold designs. Similar behaviour is observed in sense amp power consumption but all 100 failures can typically be found within the first 1000 Monte Carlo samples.

As an extension of HSMC, Solido Hierarchical Monte Carlo (HMC) provides variation-aware statistical verification on critical paths, providing a lean process for fast, scalable, verifiable, and accurate full memory Monte Carlo analysis. This is especially important when determining yield for the entire chip, including control logic, sense amps, and bit cells, where a simulation for a single instance can be time- and resource-intensive.

For example, in a case to achieve desired overall yield of 3σ on a typical memory chip, required yield at the control logic-, sense amp-, and bitcell-level are 4.25σ, 5.1σ, and 5.95σ, resulting in up to billions of Monte Carlo simulations to achieve full coverage (Table 1). Current techniques to ensure full design coverage include potentially running all components to 6σ, running local variation at FF and SS corners, and combining required yield from each sub-block assuming that all worst-cases occur simultaneously. Each of these strategies results in over-design, and is very complex to implement.

[3] See Solido White Paper "HSMC for High Yield and Performance Memory Design"

[4] *ibid*

Table 1. To achieve desired yield of 99.865% (3σ, or 1 failure per 741) on a typical memory chip, required yield of each individual element ranges from 4.25σ to 5.95σ, requiring billions of Monte Carlo simulations for full chip coverage.

Component	# of Replications	# of Monte Carlo Simulations	Required Yield (σ)
Control logic	128 (per chip)	1.81 million	4.25σ
Sense amp	64 x 128 ≈ 8000	80.7 million	5.1σ
Bitcell	128 x 64 x 128 ≈ 1 million	7.56 billion	5.95σ

Solido HMC provides accurate statistical reconstruction of the entire on-chip memory structure through building a statistical hierarchical reconstruction. It applies a similar sampling approach as HSMC, but carries-out multiple parallel high-sigma analysis across each memory component (control logic, sense amp, bitcell) to meet the desired chip yield. This fast, verifiable technique optimizes chip yield and reduces over-design while still maintaining full variation coverage with Monte Carlo accuracy.

Near-threshold SRAM Verification

Near-threshold designs such as the sureCore EverOn^{TM} family demand an especially rigorous approach to verification. When operating at 0.6V in a 40ULP node (near-threshold), the delay of a regular logic gate can increase by more than a factor of 10 due to mismatch. As the delay dependency on ΔV_T is exponential as weak devices enter sub-threshold, the distribution of delay is strongly non-Gaussian, so extrapolations should be treated with extreme caution. Even when considering two paths consisting of larger devices, or of a long chain of gates, delay difference between the paths can vary dramatically between e.g. SFG and FSG if the paths are not identically exposed to NMOS and PMOS transistors. Incorrect internal timing sequences can be catastrophic, especially under low voltage operation. Because of this increased sensitivity to variation, care must be taken to cover an extended PVT corner range during verification and internal glitch conditions must be adequately examined.

Near-threshold operation poses a challenge for bitcell operation as standard foundry bit cells will not operate at lower level supply voltages. Alternative cells are much larger and have higher leakage and are hence not an attractive option. This necessitates the use of assist circuitry to deliver bit cell functionality and performance at low voltage. This does increase the number of critical events that need to be monitored and validated during the verification process, along with a need to verify the acceptability of the assist levels across the process and temperature corners, and across the instance space. To ensure high yield, sureCore performs HSMC simulations on the cell and bitslice in the worst-case PVT corners, using excitations corresponding to the worst instance size. Even in these worst conditions and process corner, sureCore's EverOn^{TM} memories achieve a HSMC cell failure rate below 1e-9 (6σ).

Summary

Verification is the most critically important part of SRAM compiler development. Delivering low power SRAM solutions further exacerbates the challenges as near-threshold operation compounds multiple issues and increases the effects of process variation. This paper has highlighted some of the methodologies and tools sureCore uses in order to meet the challenges in a robust, practical, and timely manner. Of course, this must be complemented by a similarly extensive silicon evaluation programme including cross PVT testing as well as HTOL validation to demonstrate long term reliability. By combining these two elements SureCore has demonstrated robust world beating low power memory for power critical applications.

sureCore : When Power is Paramount

June 2017