Massive SoC Designs Open Doors to New Era in Simulation

By Adam Sherer, Cadence

As system-on-chip (SoC) designs have grown in size, simulation technologies have had to evolve dramatically to keep pace. We're now at an inflection point where both speed and capacity are essential and new simulation technologies are needed to meet the demands. In this paper, we'll discuss how simulation has evolved and examine how new technologies such as the Cadence® RocketSim™ Parallel Simulation Engine are eliminating the functional verification bottleneck.

Contents

Introduction

The first generation of commercial simulation technology emerged in the late 1980s and was marked by interpreted-code simulators such as Verilog-XL and RapidSim. Since such simulators compiled to a form of p-code and then interpreted that code, they ran rather slowly, but were suited to the smaller designs of the time. Next came compiled-code simulators in the mid-1990s, providing the speed and capacity for designs that quickly grew larger with the emergence of synthesis. Compiled-code simulators convert source code into machine code before running the simulation. These second-generation technologies were more complex to build, taking about three years to implement simulation for the existing languages and use models at that time. Since then, they have served the industry quite well, implementing a wide range of new standards including e, SystemC®, SystemVerilog, CPF, and UPF. But now, designs are growing even larger and more complex.

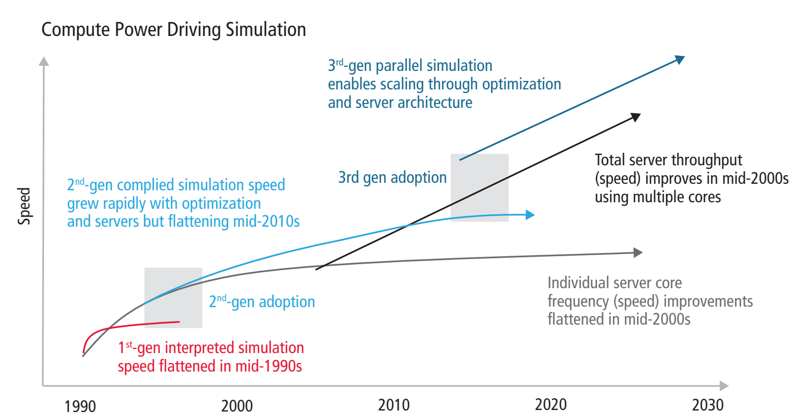

Figure 1: Computer power driving simulation

SoCs now commonly consist of multi-core, multi-million-gate designs, and running tests on these chips can take weeks. Considering the verification team resources required, this effort can easily translate into hundreds of thousands of dollars a day. And if tests need to be rerun, the price tag can quickly reach millions of dollars for a given project.

Fortunately, simulation technologies have evolved to keep pace with the growing size and complexity of SoCs. Parallel simulation capabilities have emerged to enhance verification productivity by accelerating test runtime. In this new era of simulation, engineers no longer need to worry about the verification process being a bottleneck.

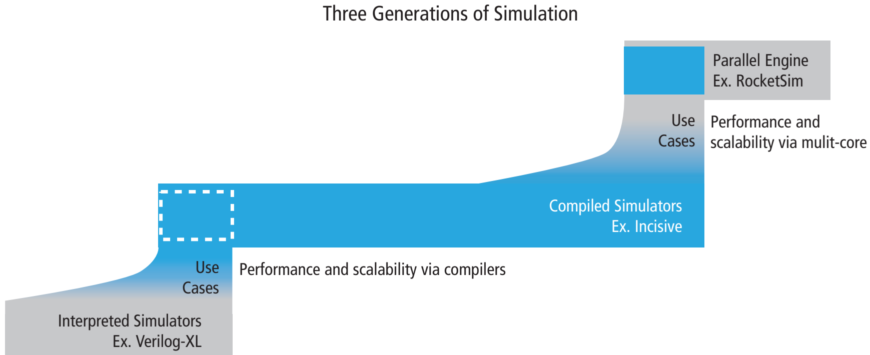

Figure 2: Three generations of simulation

Parallel Server Chip Architecture Ideal for Parallel Simulation

Today, server chip performance is dictated less by increasing clock speed and more by parallel architecture. Meantime, engineers are struggling to create the most efficient verification environment to meet their project schedule requirements.

Early attempts at parallel simulation technology for designs focused on separating hierarchical structures. Each sub-hierarchy was then assigned to a processor and the simulation was connected through interprocess communication. These attempts uniformly failed to fulfill the promise of parallel simulation. Scalability was limited to two to four cores yielding less than twice the speed-up and dependent on a manual compile process for typical designs.

Successful parallel simulation must account for multiple cores, clocking domains, complex interconnect fabrics, hundreds of IP cores, and many other components of a design. Developing this capability takes a lot of hard work and ingenuity to analyze the billions of computing elements and the dependencies among them in order to identify the short/simple calculations that can be parallelized and distributed across multiple cores in a server.

Focusing on the language level and identifying dependencies among independent threads of execution has led to the creation of a parallel simulation engine that delivers, on average, 3X to 10X faster simulation depending on the design:

- 3X, on average, for Verilog/SystemVerilog RTL
- 5X, on average, for gate-level functional simulation
- 10X, on average, for gate-level design for test (DFT)

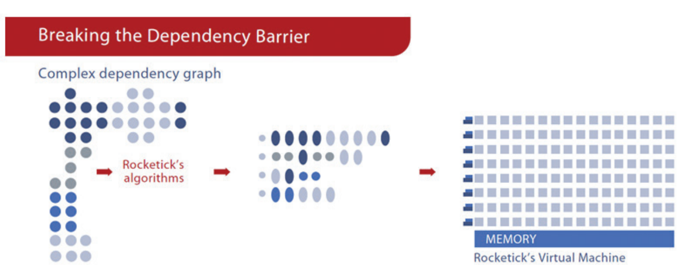

Figure 3: Revolutionary parallel simulation algorithms

Running on standard multi-core servers (up to 64 cores) and featuring patented fine-grain multi-processing technology, the RocketSim engine separates the simulation that runs on compiled-code simulators into portions that can and cannot be accelerated. The parts that can be accelerated, such as the gate-level or SystemVerilog RTL portion, are placed on the parallel engine. Verification engineers don't have to change the testbench, the design, or the assertions using this engine. Since the tool works at the language level, it isn't dependent on the process node or whether the design is a CPU, processing, or datapath design. Design hierarchy isn't a factor, either.

Each phase of the first two phases of simulation has lasted about 20 years. It stands to reason that this new era of parallel simulation will also be the foundation for simulation over the next two decades.

Parallel Simulation and Simulation Acceleration Are Complementary

SoC tests run for hours to days on compiled code simulators, in part because simulator performance is dependent on both the size of the design and the number of events to be executed at every time tick (event density). As a result, verification engineers reduce event density with more focused SoC tests and turn to simulation acceleration using hardware engines like Cadence Palladium® platforms to get 100X to 1000X faster runtime because the performance of hardware engines is not affected by event density. But this comes at a price. The thinner simulator tests are less characteristic of the function of modern SoCs, while hardware engines can't directly verify Z and X states typical of modern designs because they execute 2-state logic.

The RocketSim engine complements acceleration because it can execute 4-state logic while having a lower performance dependency on event density compared to second-generation simulators. The evidence of this effect is most apparent in the speed-up that the RocketSim engine has with DFT parallel scan tests. This combination of 4-state logic and performance allows verification engineers to create more event-dense tests with multiple concurrent scenarios that are more characteristic of modern SoCs. The complementary nature of parallel simulators and hardware accelerators now enables verification of SoC reset and power mode change in simulation and then replays longer versions of these tests for SoC functional tests on hardware.

Conclusion

While parallel simulation could be the architecture for the next 20 years, who knows what the future will hold in terms of the designs themselves? The characteristics of quantum computing and biological system computing could bring new challenges in modeling. Will fine-grained parallel simulation be the answer? Will we need more parallel structures? Will fault injection and error correction need to be integrated into future simulation tools? It goes without saying that design size and complexity will continue to drive demands for greater capacity and speed in simulation engines. Simulation technologies, in turn, will need to continue evolving to stay up to some big tasks ahead. The RocketSim Parallel Simulation Engine is clearly the first step into the future of simulation.

Further Information

Learn more about the RocketSim Parallel Simulation Engine at https://www.cadence.com/content/cadence-www/global/en_US/home/tools/system-design-and-verification/simulation-and-testbench-verification/rocketsim-simulation-engine.html

Sources

[1] http://accellera.org/activities/working-groups/portable-stimulus