

Technical White Paper

The Importance of Memory in EDA Tools

This document describes how the design of increasingly complex chips, that can easily occupy 12 to 14 GB of memory for a several million-gate digital design, requires both powerful Electronic Design Automation (EDA) tools and significant computing resources with the installation of a substantial amount of RAM memory.

The writer, through extensive interviews with senior design engineers at several of the industries leading design firms explains the high level of importance that RAM maintains when running memory intensive EDA applications.



The Importance of Memory in EDA Tools

In the last decade the density of integrated circuits has skyrocketed. Following “Moore’s Law,” chip lithography has shrunk from 2.5 μ m to 0.13 μ m, resulting in a 400-fold increase in the number of gates per unit area. Designs of a million gates are now commonplace, and Moore’s Law shows no signs of failing.

Designing these increasingly complex chips requires powerful EDA tools and significant computing resources. Simulating a million-gate digital design, for instance, requires that a system keep track of three or more transistors per gate, three or more connections per transistor, and the state of each node in the system. This design database, along with the memory needed to contain the EDA tool, can easily occupy 10 Gbytes for a million-gate digital design. Analog designs, using SPICE to model the system behavior, require several orders of magnitude more memory to keep track of a design’s behavior.

The EDA tool vendors have long recognized the ever-increasing demand for memory in design, and have been trying to accommodate. In 2001 EDA vendors started releasing tools that took advantage of the 64-bit addressing space of high-end workstations. Cadence, for instance, released a complete 64-bit synthesis, place, and route tool set: BuildGates, PKS and SE-PKS. The previous generation of 32-bit tools could address no more than 2 Gbytes, which Cadence says gave customers a maximum capacity of 1.5 million gates. The additional memory available to the 64-bit tools, which need a minimum of 4 Gbytes to operate, extends that design capacity as much as 100-fold.

Unfortunately, workstation vendors have yet to catch up to the increased capacity of EDA tools. Mid-sized systems such as the Sunblade family, for instance, use a 64-bit architecture but have a maximum system memory of 8 Gbytes. That’s 4 Gbytes each for the dual processors in the workstation, barely twice what the 32-bit tools can offer. This restricted memory capacity of mid-range workstations comes partly from financial considerations. In large configurations, memory can comprise 50%-70% of the workstation’s cost.

8GB is Not Enough

But there is also an assumption that an 8-Gbyte memory capacity is adequate for many current designs. That’s only partly true, according to NEC’s ASIC EDA engineering manager. “We’re using Cadence’s PKS and internally developed tools that need a lot of memory. For many steps in the design process, 8 Gbytes is enough. You don’t need more than that 80% of the time. But the other 20%, those with several million gates, need more like 12-14 Gbytes. If you don’t have it, the tools tell you you’re out of memory and just stop. Sometimes that happens only after days of running the calculations.”

A senior IC designer from Altera, has had similar experiences. “We use a Synopsis software tool and Mentor Calibre for physical verification. When we run our Sunblade machines in their optimal state, they only have 4 Gbytes per processor. Our designs have

a couple of hundred million transistors, which easily use more than 4 Gbytes in physical verification. If there is enough memory a job can run in 8 hours. Without enough memory it can run for more than two days.”

As design sizes continue to increase, more and more designs will exceed the workstation’s capacity. This leaves designers struggling to find a way around the workstation’s memory limits. Approaches such as hierarchical design, design fragmentation, and virtual memory are all being used (Figure 1), but only with limited success. Each workaround has its drawbacks.

In the hierarchical design approach, the designer doesn’t have access to the details of the entire design at one time. Instead, the design description comes in layers of increasing detail, ranging from functional blocks down to individual transistors. Designers can view the entire design at the block level of detail, then “drill down” to see the gate and transistor level of individual blocks but not all the blocks together. This approach reduces the amount of memory the tool utilizes because only the details of the block currently being viewed need to be available in memory. The approach only works for design capture and simulation, however. Layout and timing tools need the design to be “flat,” with all the details available simultaneously, in order to function.

Designing in Sections Carries Risks

The need to work with flat designs prompts the second memory workaround: design fragmentation. In this approach the designers break the project into more-or-less independent blocks that can be designed, simulated, and laid-out individually. Final layout is then a matter of stitching together the individual blocks. The designers need only enough workstation memory to handle one block and a description of its interface to the other blocks.

The problem with the block design approach lies in these interfaces. The problem is that the design team may inadvertently break feedback paths between adjacent blocks, so that the whole design does not get accurately simulated by combining the individual block simulations. As an IT manager at Cypress Semiconductor notes, “We’ve tried breaking our designs into chunks so that tools such as Pspice, which will suck up all the memory you can give it, would run on the workstations. When we’ve had failures in the first silicon, the failures occurred at the interfaces of the chunks.” The block approach thus saves tool memory, but risks first-silicon failure.

To be able to keep the entire design available to the tool and designer, without having it all stored in active memory, design tools have employed the technique of virtual memory. With virtual memory, the workstation keeps a portion of the design in mass storage, such as a hard disk drive. When the tool needs a design element that is not in active memory, the workstation automatically saves to disk a section of the design that is not in current use, and loads the section that is being called for. This allows the workstation to have an apparent memory capacity as large as the hard drive.

Virtual Memory Slows Tasks

One trouble with virtual memory, however, is that retrieving circuit elements from disk takes as much as 300 times longer than accessing data from workstation memory. The result is a slowdown of tool performance. The time needed to complete a simulation run, for instance, normally scales with design size. Double the design size and the simulation run takes twice as long, so long as the design can be kept entirely in workstation memory.

If the larger design requires virtual memory, however, the time to completion grows faster than the design size. “When using virtual memory the run time will vary by the amount of swapping needed,” says a senior ASIC designer at Analog Devices. “This manifests itself in a number of ways. It can take as much as four times longer to run using virtual memory instead of real memory. Or the tool might simply say it can’t do the job, and quit.”

A second problem with virtual memory comes if the tool run finds itself repeatedly calling for information that is on the hard disk. This can occur, for instance, if it is working at the boundary between two blocks that the virtual memory handler is alternating on disk. In this case, the tool repeatedly requests information from blocks A and B, but the virtual memory handler only keeps one block in workstation memory and leaves the other on disk, swapping the two as needed. The result is “thrashing,” a condition in which the system spends most of its time in disk access, exchanging data blocks.

When a thrashing condition develops, the tool run time can increase dramatically. Data memory access time effectively drops from nanoseconds to milliseconds. Worse, the tool may simply stop functioning because an error occurred in the repeated transfers to disk. If this occurs 100 hours into a 110-hour run, the results are disastrous for the design team.

Memory Expansion Alternative

The drawbacks of virtual memory and the other memory-limit workarounds seems to leave designers with a choice between poor and risky performance, or the expense of moving to an expensive workstation network to handle larger designs. There is a third alternative, however. It is possible to extend workstation memory beyond its current limits.

The Sunblade 2000, for instance, has slots for two banks of four memory modules in its design. The largest module that Sun offers is 1 Gbyte, for a total workstation capacity of 8 Gbytes. The slots can handle modules of more than 1 Gbyte. They simply haven’t been available.

Dataram’s unique memory and packaging technologies change the situation. It can double the amount of memory an OEM module contains, extending the workstation’s memory capacity to 16 Gbytes. This, in turn, eliminates the need for virtual memory,

design fragmentation, hierarchical approaches for most of today's designs. For the largest designs, those workarounds remain available but become much less restrictive, requiring fewer swaps and allowing larger blocks and flatter designs.

For future designs, Moore's Law will continue to hold. Designs will get larger, increasing the demand for workstation performance that will force today's standard workstations to give way to more powerful systems. Memory will always be a critical factor for EDA tools, however, and always a cost consideration in workstations. The double-density memory that Dataram offers can lower memory cost as much as 50% relative to OEM offering for a full memory complement. And as succeeding generations of chip designs tax new workstations to their limits, the Dataram density advantage ensures that design teams can get the extra memory they will need.

About the Author: Richard Quinnell is a technology journalist and former design engineer with more than 30 years experience in system design and EDA. As a journalist he has watched the development of design automation from its earliest stages, when chips were being designed with 5 and 10 um design rules and workstations had only a few megabytes of memory. Richard is a frequent contributor to Electronics Design Chain, EDN, Electronic Business, TechOnLine and CommVerge.

Notice

BuildGates is a trademark of Cadence Design Systems Inc.

Calibre is a trademark of Mentor Graphics Inc.

Sun Blade is a trademark of Sun Microsystems

All other product names mentioned herein may be trademarks of their respective companies.

The information contained in this document is subject to change without notice.