sponsored by

Gabe on EDA • EDAmarket • EDA Confidential

Assembling the Future

A Newsletter About the Design and Production of Electronics

ISSUE 008 • SEPTEMBER 2011

SUBSCRIBE

PDF and Archives

for Printing

twitter

In this issue:

New Processes Require New Methods

Gabe Moretti

Following the progress of semiconductors manufacturing capability, development costs have risen with every new process node, mostly due to the increased complexity of designs.

But lately, mask costs and manufacturing costs have also shown considerable cost increases, forcing developers to take a harder look at ways to decrease them in order to achieve profitability sooner.

EDA companies have responded to the problem of increased complexity by developing new algorithms and new methods to address it. Although they have kept up with the progress in semiconductor technology they have not done it on a timely basis, with tools always lagging behind manufacturing capability. The result is that early adopters have almost always been forced to use pre-production tools. In addition although it can be proven that the cost of development per transistor has decreased, the cost per die, which directly contributes to profitability, has risen. In fact die cost is now a major barrier to market entry, since it increases the number of units that must be sold to reach the break even point.

EDA vendors are very good at solving problems. Designers as a whole, though, are not very good a preventing problems from occurring in the first place. The reason is in fact quite simple: many prevention activities can appear superfluous, and thus avoidable in order to cut costs. Instead

problem solving is never avoidable: if the problem is not solved, even if at considerable cost, the project is a failure. Going where the money is just makes sense. EDA vendors are not missionaries, they are business people. So as long as the market demands only problem solving, that is what they will deliver.

As we become familiar with the 28 nm process, and begin to experiment with 20 nm and plan for 14 nm, it is time for the entire electronics industry to become serious about problem avoidance. Thus problem avoidance should be the most important part of development. And manufacturing problems should become a stable part of the list of problems to be considered as early as possible in the planning and development phases of a product.

What is a System on Chip (SoC)?

From an architectural point of view a SoC can be divided into three parts: Processing Cores, Memory, and Application Specific Logic.

These three architectural components need to be viewed both as a system and as individual components with specific characteristics and requirements, including during manufacturing. Commercial components must not only be qualified to make sure of their functionality but they must also be evaluated for yield. This evaluation must take two aspects into consideration: n the intrinsic structure of the component and its adaptability to the specific process, and, just as important, the requirement it presents to the adjoining logic. I am talking about the interface protocol, the power requirement, the clock tree architecture, and the noise impacting nearby logic. In other words, does the use of a specific component add restrictions to the way the surrounding logic can be implemented?

Processing Cores

High-performance cores are the building blocks of any application chip today. The entire system performance is often limited by how fast these processing engines can execute. Also, the system's total power consumption depends significantly on how low power these cores can be. Multi-core processors with 2, 4, 8, or 16 cores processors – each with multiple voltage domains – are required to meet the throughput demands of data-intensive advanced applications while still consuming the same power as a single-core design. Designing these low-power multi-core devices is becoming increasingly difficult and implementation mistakes are costly.

Very few companies build proprietary processing cores today. Designers have a variety of cores available to them, both for arithmetic, control, graphics, and communication applications. Today's methodology allows designers to integrate these commercial cores in any way they deem feasible in the proprietary design. Such practice has not, at least not that I can find, created serious manufacturing problems in processes in use today, like 90 and 65 nanometer geometries.

But leading system houses are shipping in volume products manufactured in the 40 nm process, and are getting ready to do the same with the 28 nm process. I suspect that the increase in development cost for these products is in large part due to the problems that the integration of third party components generates, both in the area of power distribution and consumption, data interface, and overall system design. The 20 nm process is at the door. With it come more serious differences between the process implementation of each foundry, to the point that second sourcing is becoming

not only impractical, but downright commercially impossible.

Memory

Modern SoC devices need a very large amount of memory to store large amounts of executable code and data. SRAM, DRAM, Flash and image sensors are all memory types that smartphones, cameras, tablets, notebooks and other portable devices need in abundance. This has created one of the most competitive markets in terms of cost and time to market for memory vendors. The key concerns of engineers are high reliability memory (every bit of memory must work) at the lowest possible cost. Although one can think of a memory as being a digital device, it is either on or it is off, its cells, or bits if you will, are analog circuits.

Designing and verifying memory is difficult, but it is also difficult to integrate a memory block in a design. Luckily, in part because of this difficulty, standards for memory interface are available, as is a significant inventory of verification IP that designers can use to assure themselves that the integration meets their requirements and does not produce undesirable side effects. This latter issue will grow in importance as the industry adopts more advanced processing nodes. As memory blocks are positioned closer to other logic the interference between the two different areas of the die will become more important.

Application Specific Logic

Network processing, data storage chips, and consumer multimedia chips such as those in digital televisions are among the biggest silicon chips in the world. These chips also contain both analog and digital circuitry that must operate as a system, not as a collection of separately designed parts interfaced in a follow-on step. The massive sizes of these devices have made the cost and time associated with developing them nearly impractical. Project teams need to simulate these designs flat. The need to break the designs into smaller pieces and run them hierarchically destroys the ability to evaluate the physical effects of the integration. Furthermore, with designs of this size comes a lack of predictability among different designers and different design teams. Front-end designers lack predictability to back-end physical design resulting in numerous time-consuming iterations with suboptimal results. It is imperative to avoid such predicament. EDA vendors have solutions that allow simulation of very large flat circuits, but the volume of output may be overwhelming. More progress must be made in formal verification and in design methods that avoid errors by construction, in order to simplify verification. Mixing analog circuitry with digital also requires a careful choice of process. Engineers must use geometries that support the physics of analog circuitry while optimizing speed and power of the digital portion as well as of the overall system.

A New Methodology

Design rules, now the sole requirement of place and route engineers and tools, must be elevated to the system level. How a chip is architected will make a difference, not only on the cost of the downstream development, but on the manufacturing cost as well. The proper isolation and grouping of blocks with similar physical operating requirements, such as power and timing, the interface will be required. Thus there will be a need to have a number of levels of design rules, spanning system level, RT level, and gate level.

Correct by construction will not just be a desirable target, it will become a requirement. Formal verification methods will continue to gain in importance, but the tools used today for hardware/software tradeoff will need to be strengthened and expanded. The availability of much greater number of transistors makes exploring tradeoff implementations not only attractive, but a requirement in areas like power consumption and execution speed.

The industry has made significant advances in virtual prototyping, allowing software development and verification to take place in parallel with hardware development, but much more must be done in the area of hardware/software tradeoffs. One major problem is that software engineers are limited when choosing a processor core. In almost all cases a system house first enters into a business agreement with a IP processor provider and then selects the core. This is backward to a software engineer who would like to evaluate a number of different cores before choosing a provider. What is happening today is what use to happen at the beginning of commercial computing. Companies would choose a mainframe and software programmers were stuck with it.

It would be ideal if there was a service that allowed a project team to evaluate processors from more than one provider. This would then support a better evaluation process of potential implementation strategies.

In a purely architectural space, a 3D packaging that separates the three functional parts of a SoC is ideal. Its major advantages are a simplified power distribution and a possible use of different process nodes depending on the circuit characteristics. Advanced processing nodes, like 20 and 14 nm, could be reserved for the high speed requirements of processing cores and memories, while less dense processes can be used for peripheral interfaces, analog circuitry, and human interface protocols.

A close partnership must be established among IP suppliers, foundries, EDA vendors, and probably even semiconductor equipment vendors to establish a standard for the integration of specific IP at a given process node. In order to maintain users' flexibility in the choice of IP, foundry, and EDA tools, the standard should be independent of the choice of design tools and the chip manufacturer. This may not be possible, given the proprietary characteristics of the implementation of a given process by each foundry.

It is more feasible to develop a standard that specifies the type of requirements, rules, methods, and the format or formats to be used, while allowing the foundry the freedom to implement them.

Today's method of having the design flow of each major EDA vendor approved by a foundry may be adequate, but it has an inherent fault with respect to the EDA industry. It makes the use of point tools provided by startups much more risky and difficult to use. Startups have been the lifeblood of innovation in the EDA industry. If their commercial success is limited by the new requirements, the industry will need a new business model: the change will be revolutionary and the cost will be staggering.