Top-Down Design

2

1 Mixed-Signal Design Productivity

The mixed-signal design process has changed relatively little over the past two decades, and in comparison to the digital design process, is slow, labor intensive, and error prone. While digital designers have improved their design methodology and adopted design automation, analog and mixed-signal designers by and large have not.

There are two reasons why digital designers are far ahead of analog designers in improving their design processes. First, digital designers confronted the need to design very large and complex systems much earlier than analog designers. Consider that large digital chips today consist of tens of millions of transistors, while complex analog chips contain only tens of thousands of devices. Second, the digital design problem is much more amenable to automation than the analog problem.

Consider a digital design. In most cases digital systems are implemented as finite-state machines (FSM) and constructed from standard cell libraries. Using a FSM formulation acts to unify and homogenize digital design and gives it a well-understood mathematical foundation. This foundation was thoroughly explored in the late ’80s resulting in the commercial logic synthesis tools of the early ’90s. These tools take a register-transfer level description (RTL), a relatively high-level description of a digital system that is created by designers and can be verified with the help of logic simulators, to produce an optimized gate-level description of the system. This transformation is possible because digital systems are constructed from a limited set of relatively simple and well-behaved building blocks. The building blocks of digital systems are gates and registers. The blocks, generally referred to as cells, all share a common I/O model and so are easily interconnected, are derived from a relatively small number of cell types that have very simple and easily described behavior, are easily parameterized in terms of the number of inputs and outputs, and have a simple and easily adjusted performance trade-off that involves only speed and power. Logic synthesizers operate by creating a complete mathematical description upon which it performs transformations in order to create an optimal design in terms of speed, power, and

area. This is a two step process. First, equivalence transformations are applied to the mathematical descriptions in order to reduce the total number of gates, which minimizes the area, and the depth of the logic, which roughly maximizes the speed. This is possible because each block has a simple logical description and a common interface model. Then, the drive ability of each gate is adjusted to provide the lowest power while still meeting speed requirements. This is possible because this speed-power trade-off is easily made in each gate.

Now consider analog design. Analog design has no equivalent to finite-state machines, and so has no unified formulation and no common mathematical foundation. It also has no universal equivalence transformations that allow the topology of a circuit to be easily modified without risk of breaking the circuit. These problems prevent a topological mapping from a behavioral description to hardware. Even if one were mathematically possible, the lack of a common I/O model for analog blocks would prevent the topological modifications that are needed for either mapping or topology optimization.

It might be possible to try to enforce a common I/O model for analog circuits, but doing so would be very expensive. For example, one might simply specify that the signals at the inputs and outputs of analog blocks center around a particular value, have the same maximum swing, and that outputs have zero output impedance and inputs have zero input admittance. The problem is that doing so would necessitate extra circuitry in each analog block that is there simply to assure compliance to the I/O model. That circuitry reduces the overall performance of the circuit by increasing power dissipation, increasing noise, decreasing bandwidth, etc. This differs from the digital world where the common I/O model was achieved naturally and without significant cost. In addition, it is not possible to achieve these ideals at high frequencies. Instead, some common reference impedance would have to be specified, such as the 50Ω used at the system level, but driving such loads greatly increases power dissipation.

Finally, there is no simple way to trade-off the various performance metrics that are important with analog blocks, which makes it difficult to perform a parametric optimization. Sensitivity-based local optimizations can be used, but the improvement provided by these approaches is usually small. Monte Carlo-based global optimizers offer better improvements, but require substantially more in the way of computer resources.

The end result is that analog designers have no equivalent to RTL, a relatively high-level language in which they can describe their design and from which they can synthesize an implementation that is guaranteed to be functionally correct and have near optimal performance. As such they must transform their designs from concept to

implementation by hand, and so the design process is naturally much slower and more error prone that the design process for digital circuits.

The outlook for providing the equivalent to logic synthesis for analog designers is bleak. However, things cannot continue as they are; the current situation is becoming untenable. While a complex digital chip can be designed correctly on the first try in a few months, designing a complex analog chip can require 3-4 re-spins and up to a year and a half to get right. This is problematic for many reasons:

1. The tremendous mismatch in schedule and risk between the analog and digital portions of a mixed-signal design makes it difficult to justify combining analog and digital on the same chip.
2. The high risk makes planning difficult. It is hard to predict when product will be available, and when valuable analog designers will free up.
3. A long time-to-market makes it tough to react to changes in market trends and competitive pressures.
4. Analog and mixed-signal product development demands large investments of time and money. This makes it difficult to justify developing new analog products, especially in tough economic times.
5. Analog and mixed-signal designers are scarce and hard to recruit. Compounding this problem is the inherently low-level of productivity of the current mixed-signal design process, which makes it difficult for small design houses that are not focused on analog to field an analog design capability.
6. Some mixed-signal designs are becoming so large that, with the low productivity of the analog design process, a team of analog designers that is large enough to take on the project and complete it in a timely manner simply cannot be assembled.
7. To compensate for semiconductor processes that are increasingly unfriendly to analog designs results in an increasing use of auto calibration and adaptive filtering. This substantially increases the complexity of both the design and the verification of the design, which magnifies the problems already mentioned.

Clearly a change is needed. It is interesting to note that when digital designers were trying to design systems of a size comparable to today's mixed-signal designs, their design process was not that different from what analog designers are using today. But it was at that point that they began to transition to a more structured and more automated design methodology. For analog designers, substantial automation may not be in the cards in the near future, but the need to transition to a more structured design methodology that is both more efficient and that allows designers to handle the growing size of analog and mixed-signal circuits is clearly needed.

The availability of logic synthesis tools was not the only enabling factor for digital designers to move to more efficient design methodologies. By moving to FSM and RTL, digital designers also gave up considerable performance in terms of speed and power. They made this sacrifice to be able to design the larger and more complex systems quickly. This sacrifice was a critically important enabling factor. Analog and mixed-signal designers have not demonstrated the willingness to make a similar sacrifice. In those cases where performance is not critical, the tendency is to instead convert the circuit to a digital implementation in order to gain flexibility. In the remaining cases sacrificing performance is not an option; however it is also not clear that such a sacrifice is needed. Analog designers do not have the equivalent of logic synthesis, so they will continue to use custom design methodologies. While moving to IP (intellectual property) reuse may entail some sacrifice in overall system performance, changing to a top-down design methodology does not inherently imply lower system performance. In fact, the opposite is usually the case, using top-down design results in higher performance. Rather, the sacrifice that is demanded of analog and mixed-signal designers is that they must learn new skills, such as behavioral modeling, and they must be more disciplined in the way they design.

It is unlikely that analog and mixed-signal designers will ever be allowed on a large scale to trade any substantial amount of performance and power for a reduction in design time. Rather, in those cases where the performance and power requirements are not demanding, a digital implementation is usually preferred.

2 Traditional Approaches to Mixed-Signal Design

At the Design Automation Conference in 1998, Ron Collett of Collett International presented findings from a 1997 productivity study in which his firm analyzed 21 chip designs from 14 leading semiconductor firms. The study revealed a productivity gap of 14× between the most and least productive design teams. The study also revealed that developing analog and mixed-signal circuitry requires three to seven times more effort per transistor than designing digital control logic, though this factor was normalized out of the 14× ratio.

The reason for the poor productivity of those at the bottom end of the scale are increasingly complex designs combined with a continued preference for a bottom-up design methodology and the occurrence of verification late in the design cycle, which leads to errors and re-spins. There's a huge disparity in productivity between those mixed-signal designers who have transitioned to an effective “top-down” design methodology, and those who practice “bottom-up” design and rely solely on SPICE.