

Meeting the Challenges of Large, Complex ASICs

New techniques are being used to deal with current and future demands of smaller transistor sizes.

By Graham Inglis, Tality

Every year, consumer demands mean that equipment suppliers demand more functionality and better performance, but with lower power dissipation and a cheaper part price. Fortunately for ASIC designers, Moore's Law continues to apply, and the successive geometry shrinkages provide the underlying technology, which make it possible to meet these demands. This article examines some of the challenges facing designers using the current mainstream technology of 0.18 μm CMOS, and how new techniques are used to deal with these and the future demands of still smaller transistor sizes.

Pressures on ASIC Designs

Producing an ASIC design is expensive — so expensive, in fact, that some observers believe programmable (FPGA) technology will eventually displace ASICs almost entirely in digital applications. However, most of this cost is involved with preparing the design for initial manufacture, and if the device achieves volume production, the unit cost drops far below that of an equivalent multiple component assembly. The other significant advantage of ASIC designs is that they can offer performance, power consumption and physical size characteristics not achievable by other means.

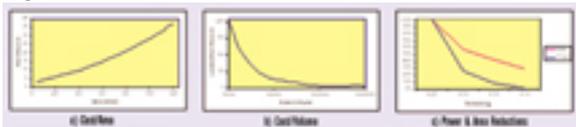


Figure 1. Cost of Silicon and Reductions in Area and Power

Area is the most significant factor in determining the cost of an ASIC. Since processing costs are fixed at the wafer level, the price of each individual device falls as more devices are fit onto a single wafer. Final product cost is also reduced as more functions are integrated on a single chip, thereby lowering component costs. The other main pressures on ASIC designs are to balance opposing requirements for higher performance (more functionality and/or faster clock speeds) and lower power. By implementing a design using a smaller silicon feature size, cost, functionality, speed and power can all be addressed at once. As illustrated in Figure 3, more than three times as much functionality can fit in the same space using 0.13 μm technology as on 0.25 μm technology while using less than 1% of the power.

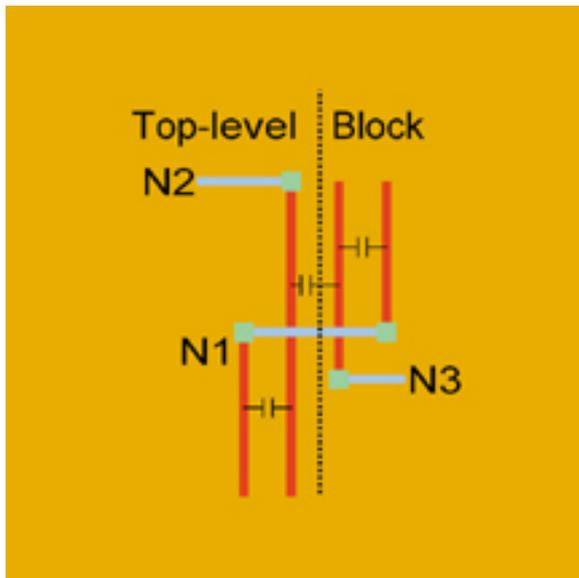


Figure 2. Boundary effects on Parasitic Capacitance

Flow Limitations on Large Designs

As designs grow larger to accommodate more functionality on the chip, older techniques and tools are overwhelmed by the sheer size of the design data. The most common design style for implementing digital designs is based on synchronous registers. A clock signal is distributed to all the flip-flops in the relevant design, such that they all change state simultaneously. Using this design style, it is necessary to check after the circuit is laid out that none of the valid paths through the logic gates exceed the clock period. If any do, this is corrected either by modifying the layout or, occasionally, the circuit architecture.

Traditionally this was done by using a modified version of the functional simulation model of the circuit. Since digital simulators essentially work by placing state-change events on a timeline, this technique involves calculating the individual gate and wire delays, and then taking these delays into account when deciding where on the timeline to place the stimulus event to the gates downstream. This approach has two major disadvantages. First, the simulation needs a testbench to provide stimulus signals to the circuit, but as circuits get bigger it becomes increasingly difficult to determine whether any particular testbench actually exercises all the relevant signal paths through the design. The second disadvantage is that as designs get bigger, there are more paths to be tested and increasing depths of flip-flop layers between the primary inputs and outputs mean that the entire approach becomes impractical as the time taken to run the simulation gets exponentially longer.

For those reasons, Static Timing Analysis (STA) has almost completely replaced the timing simulation approach. In order to set up the correct conditions for a particular path to be exercised during simulation, it may be necessary to exercise other paths many thousands of times, each time performing both the actual logical operations to the signals and performing the delay calculations. STA removes this requirement by identifying all the possible paths, and calculating each path delay just once.



Figure 3. Crosstalk

Placement and routing (P&R) is another area where design sizes are beginning to create long run times. P&R has commonly been performed on the whole chip at once, but as designs get bigger, and the optimization algorithms in P&R software get more sophisticated, increased run times have led design teams to take a hierarchical approach to physical design. The basic tools to support this have been available for years, but there are several issues to be overcome. First, it can be difficult to identify the optimal way to divide the logical design for physical implementation. Next, these blocks have to be arranged in the most efficient layout (floorplan), and the interconnect between the blocks has to be routed. Finally, parasitic extraction has to be performed on both the blocks and the top level and the effects of metal features in the other hierarchy level must be taken into account. Figure 4 shows a simple example of this, where the boundary crossing net N1 has cross-coupling capacitance with both the top level net N2, and the block net N3, which share cross-coupling capacitance with each other. A new technique for managing this is to create a simplified view of the other hierarchy level (containing only metal features near the boundary) and include this during the extraction. With this approach, nets that cross the boundary such as N1 will appear in both sets of extraction data, and these can be stitched together to allow delay calculation.

Verification

Typically a test-bench is written to prove the functionality of a design. This is developed in parallel with the behavioral or Register-Transfer-Level (RTL) model and, until recently, was used to verify that the gate-level netlist had equivalent functionality. However, just as it has become impractical to perform gate-level simulation for timing analysis, running it for functional verification invokes too much overhead.

The technique currently displacing gate level simulation is called Equivalence Checking (also loosely referred to as Formal Verification). Like STA, this works more efficiently by checking each functional path once, rather than many thousands of times. Current software can check the functional equivalence of two models of the same design – for example an RTL model and a gate-level netlist, or two different gate level netlists. The tools work by identifying all the flip-flops in each model and matching these up. Functionality in between flip-flops is reduced using Boolean transformations, and the tool compares the two views and checks for equivalence in terms of both flip-flops and functionality.

Although Equivalence Checking removes the need for simulation to verify that the synthesis, test insertion and P&R processes have not altered the original functionality, it is still necessary to verify that the original RTL model performs as intended. Simulation is still the main approach, but just as large designs rendered gate-level models impractical, even larger designs are making RTL model run-times excessive. Moreover, designs increasingly incorporate functional blocks developed

elsewhere, making it more difficult to create a testbench to verify the design as a whole. One solution uses a form of hardware assistance to help the simulation go faster, and a number of competing technologies have been developed in this area. An alternative is rapid prototyping, where the unverified design is realized using some kind of programmable technology which allows the functionality to be tested in a "real-life" environment (although not necessarily at full speed). An increasingly popular approach for System-on-Chip (SOC) designs uses platforms. Here, the basic structure of the processor and standard peripheral blocks is extensively verified once, but derivative designs do not bother to re-verify the complete functionality. Each new block is verified stand-alone, and then the full chip testbench is only used to check that interfaces between the blocks are operating correctly. Whichever approach is taken, the issue of determining whether the testbenches completely verify the design in all possible modes remains. Most designers now use code-coverage tools to check that every path through code has been exercised, but this does not necessarily guarantee the testing of all relevant modes. Software incorporating other formal verification techniques, such as model-checking, are now beginning to be marketed, but this area is still relatively new, and will doubtless see a great deal of change in the future.

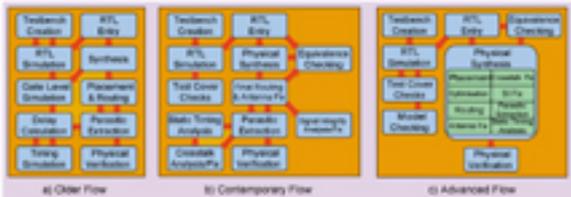


Figure 4. Flow Development

New Physical Effects

Above 0.25 μm , it was sufficient to use generic synthesis constraints such as wire load models. Since these work by using an average routing length for all the nets within a block, there would usually be a few cells driving longer nets which were found to be overloaded after P&R. These could be upsized using an in-place optimization (IPO) synthesis run, and the developers of the P&R software soon started integrating cell-resizing algorithms into their toolsets. However, as the device geometries get smaller, the intrinsic delay of the cell reduces, and the proportion of total cell-to-cell delay $\#151$ due to interconnect $\#151$ gets increasingly significant. This means that more of the nets, now longer than the wire load model prediction, end up causing cell overloads and timing problems. At 0.18 μm , even custom (specific to the block) wire load models break down, so physical synthesis, whereby the synthesis tool also performs the placement and approximate routing, is used to minimize the number of cells which end up overloaded after final routing.

With smaller technologies, a number of physical effects that can cause manufacturing or reliability problems have increased in significance.

$\#149$ **Antenna effect** occurs when a long length of wire picks up static charge during manufacturing and causes damage to the device. Current software can fix this either by routing adjustments or by inserting a protection diode.

$\#149$ **Crosstalk** increasingly occurs in smaller technologies when a weak "victim" waveform is corrupted by a neighboring "aggressor" waveform. It can be avoided by shielding or wider spacing between nets, by

Meeting the Challenges of Large, Complex ASICs

Published on Wireless Design & Development (<http://www.wirelessdesignmag.com>)

adjusting the size of buffers, or by inserting extra repeater buffers to reduce the susceptibility of victim nets. Currently designers usually perform analysis using special standalone software and then use another pass of the P&R software to fix the problems, but most tool suppliers are working on integrating this analysis so that fixing occurs automatically.

•**Power Mesh IR Drop** is where the resistance of the power routing is high enough to cause a significant voltage drop to be seen by some of the circuits, which then function incorrectly. It is avoided by analyzing the power requirements of the device and ensuring adequate power metal to distribute the necessary current.

•**Electromigration/Wire Self-Heat** are both also caused by excessive current density on power or signal tracks, which causes the wire to become open-circuit. On signal tracks are avoided by matching the width of the track to the driver cell output current; on power tracks the Power Mesh IR Drop analysis will highlight problem areas.

•**Hot Electron Injection/Impact Ionization** happens on new technologies due to a very short semiconductor channel that allows high-energy electron collisions, which damage the transistor. This is avoided by ensuring that loads on cells are appropriate to the drive strength.

Flow Evolution

To summarize the impact of these pressures on the overall design flow, each individual change is reasonably straightforward, but as can be seen from Figure 4b, much of the software currently available has been developed standalone, leading to a somewhat cumbersome flow. Worse, any change for (say) crosstalk will also have an impact on the circuit timing, potentially creating new violations. There is therefore a significant risk that on large designs it will be extremely difficult to achieve a layout that is completely satisfactory in all respects. The software must become more integrated to combat this, as illustrated in Figure 4c, so that when fixing one set of problems, it recalculates and re-fixes the issues already resolved. Of course, by the time this is the standard flow, designers will be working on even newer technologies, which will no doubt create yet more challenges!

Graham Inglis is currently a Chief Consulting Engineer at Tality's SOC Design Centre in Livingston, Scotland where he manages digital chip design and implementation projects. Mr. Inglis can be reached at inglis@tality.com.

Source URL (retrieved on 04/28/2015 - 8:36am):

<http://www.wirelessdesignmag.com/product-releases/2001/10/meeting-challenges-large-complex-asics>