FPGA Challenges and Opportunities at 40 nm and Beyond

Vaughn Betz and Stephen Brown
Altera Corporation, Toronto Technology Centre
151 Bloor St. W, Toronto, ON, CANADA
{vbetz, sbrown}@altera.com

Abstract: FPGA companies are amongst the earliest adopters of next-generation integrated circuit process technology, as the economics of scaling are more favorable for FPGAs than for many other technologies. Moving to advanced processes involves many challenges, including power management, device modeling, increasing I/O performance to match the computational capacity, and enabling very large designs to be completed quickly. In this paper, we highlight the economic and technological advantages of moving FPGAs to more advanced processes, and discuss the techniques Altera is using to overcome the challenges that come with process scaling.

1. Introduction

FPGA companies are now among the most aggressive adopters of next-generation process technology. For example, Altera's Stratix IV FPGA was delivered to customers in 2008, making it not only the first 40 nm FPGA, but also one of the first 40 nm integrated circuits of any type.

1.1. Scaling Benefits

Moving to a more advanced process node has many technological benefits for FPGAs. First, with approximately double the transistor budget, the number of computational elements (logic cells, block RAM bits and DSP blocks) that can be fabricated in a given chip roughly doubles each process generation. Since the cost of a chip is proportional to its area, this scaling leads to a cost saving. In addition, the increased capacity of chips makes it feasible to include new features, such as high-performance I/O protocols (e.g. PCI Express) which require large amounts of logic. At the same time, modern FPGAs must maintain the same overall power budget per chip to meet market requirements. For high-speed FPGAs like the Stratix series [1], for example, the power budget per chip ranges from approximately 2W for the smallest capacity family members to 20W for the largest, regardless of the process node. Consequently, by moving to an FPGA in the latest process technology a system designer can add more functionality for the same cost and power, or cut the silicon area required by a factor of approximately 2, reducing both cost and power.

1.2. Economic Challenges & Opportunities

Scaling favors FPGAs over competing technologies, as fewer and fewer markets justify the cost of an ASIC design in cutting-edge technology. Table 1 summarizes the economic challenge of making an ASIC in a 40 nm process. Mask costs are very high (approximately $4 million for a full mask set), and the average ASIC design requires two full mask sets before it is fully functional. Developing test patterns to determine which chips are functional, and performing product engineering to ensure the product is both electrically and mechanically robust across a range of environmental (e.g. temperature) and electrical (e.g. voltage) conditions is another large cost, and is typically in the 5 to 10 million dollar range. A designer using an FPGA does not bear any of these costs – the FPGA manufacturer is responsible for mask design, test development and product engineering. The other major cost in ASIC development is the design and verification of the ASIC logic and any software that will run on or interact with the ASIC. An FPGA designer also must design and validate logic and any software running on the FPGA (e.g. in a soft processor), so some of this cost also occurs in an FPGA design. It is, however, dramatically reduced vs. that of an ASIC for two reasons:

- Many of the highly demanding physical design and verification steps necessary in ASIC design, such as clock tree synthesis and signal integrity analysis, are not necessary for FPGAs. FPGAs raise the level of design abstraction by providing a pre-verified programmable fabric.
- An FPGA design can be rapidly recompiled and downloaded to hardware, and can rapidly be instrumented with additional debugging logic. This enables a rapid code-verify-debug cycle that resembles the highly efficient flow used in software design. With an ASIC every design change must be extensively verified before going to fabrication, as an error results in several months delay and several million dollars in mask costs. In addition, co-development of software that runs on or interacts with the ASIC is more difficult than it is with an FPGA, as the ASIC hardware is more difficult to modify and takes longer to develop.

Table 1: 40 nm standard-cell ASIC economics.

Mask cost	2 spins x $4M = $8M
Test & Product Engineering	~$7M
Design, verification, software	~$25M
Total Development (R & D) Cost	~$40M
Required revenue to achieve R & D = 20% of revenue	$200M
Required target market size assuming 10% market share	$2B

A healthy company typically invests about 20% of its revenue on research and development. With a $40M development cost, this implies that a 40 nm ASIC requires approximately $200 million dollars in revenue to be economically viable. Assuming a company can capture a 10% market share, this requires a market for the ASIC that has a lifetime revenue of $2 billion or more. There are few markets for fixed-function devices of this size. The net result has been rapidly falling ASIC design starts, and more designs moving to FPGAs. In addition, most ASIC designs still target 130 nm or larger process nodes to keep development costs manageable; however, this means that those designs cannot reap the benefits of the increased integration allowed by more advanced processes.

The cost of developing an FPGA in an advanced process node is also very high. For example, the development cost of the hardware, CAD software, and intellectual property cores needed for the Stratix IV family of 40 nm FPGAs is approximately $250 million dollars. However, this large development cost can be amortized across a very large target market, since FPGAs are highly programmable and

consequently can target most of the digital design market, which is over $20 billion per year. The very wide applicability of FPGAs is the key to their economic advantage.

1.3. Time-to-Market Advantage

Product lifetimes have been shrinking in recent years. As Figure 1 shows, being slower to market results in lower revenues for a product, as the revenue ramp is delayed, but the revenue decline as the market matures is not. In the example depicted, a product introduced at the beginning of a two-year product lifetime may enjoy revenue of $99M, while a product delayed by six months achieves only $62M in revenue. Due to the need for custom fabrication, the more complex design flow, and the inability to rapidly modify and debug the system, an ASIC design almost always will be significantly later to market than an FPGA design. This is especially true of the most complex designs in the most advanced process nodes.

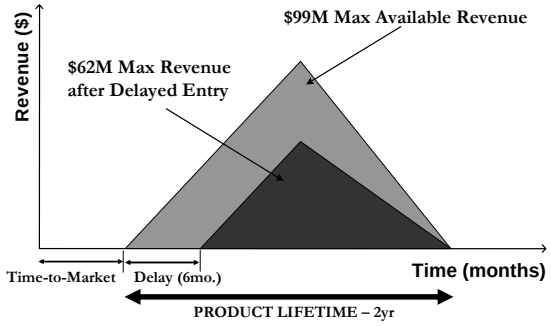

Figure 1: Time to market economics example.

The same time-to-market dynamics that push designers toward FPGAs also push FPGA manufacturers to next-generation processes early. The sooner a larger FPGA with more features is available, the sooner it starts capturing new designs and markets. In addition, semiconductor foundries increasingly use FPGAs as process drivers. FPGAs are large, regular dies with both logic and RAM structures. This makes them ideal to test a new process and to help analyze and improve any defects that affect yield. For both these reasons, FPGAs tend to be among the first devices to a new process node.

1.4. Scaling Challenges

The key challenges presented by advanced process nodes include power management, device modeling, increasing I/O performance, and improvements to designer productivity. In the following sections we will describe these challenges in more detail, and show how Altera has coped with them in its state-of-the-art Stratix IV FPGAs and Quartus II CAD system.

2. Stratix IV Overview

Altera's Stratix IV FPGA was delivered in 2008, making it both the first 40 nm FPGA and one of the first 40 nm devices of any type. The Stratix IV family comprises members with 7 different densities, covering a logic and memory capacity range of more than 10 times. In addition, there are three variants of many of the devices: a GX version with 8.5 Gbps serial transceivers, a GT version with 11.3 Gbps serial transceivers, and an E version without serial transceivers [1].

Table 2 summarizes the increase in device resources achieved in Stratix IV compared to its predecessor, the 65 nm Stratix III family. Logic capacity more than doubles to 820,000 logic elements (LEs), and on-chip memory doubles to 33 Mb of block RAM, plus an additional 8.5 Mb of RAM that can be obtained by using up to half the logic elements as memory. The number of embedded hard multipliers also nearly doubles, to 1360. Overall the computational capacity of Stratix IV is roughly double that of Stratix III. Despite this capacity increase, the power demands of Stratix IV devices are roughly the same as those of Stratix III, ranging from approximately 2W for the smallest (70 kLE) device to approximately 20W for a large, high speed design in the largest (820 kLE) device.

Table 2: Stratix III and IV device resource comparison.

Feature	Stratix III (65 nm)	Stratix IV (40 nm)
Logic Elements	340k	820k
RAM bits	16 Mb + 4 Mb	33 Mb + 8.5 Mb
18x18 multipliers	768	1360
General I/O	1104	1104
High-speed serial links	0	48 transmit + 48 receive @ 11.3 Gb/s
Hard PCIe blocks	0	4
Clock generation	12 PLL(x10)	12 PLL(x10) + 32 serial recovered + + 24 serial transmit
Clock distribution	16 Global + 88 Quadrant + 132 PCLK	16 Global + 88 Quadrant + 132 PCLK

Stratix IV has not only up to 48 (96 unidirectional) high-speed serial transceivers, it also includes up to 4 PCI Express (PCIe) protocol blocks in "hard" logic. PCIe is a widely used serial standard, and consumes tens of thousands of logic elements for each link if it is implemented using the "soft" approach—that is, using the programmable logic elements. Consequently, it is a net win to include hard logic for this function.

Stratix IV can accommodate very complex systems in which it communicates with many different chips. This often requires a large number of clocks, and Stratix IV has abundant clock generators: 12 phase-locked loops (PLLs) which can each generate 10 clocks, plus up to 56 clocks that can be generated in the serial transceivers. To efficiently distribute these clocks Stratix IV includes 3 distinct types of clocking resources: global clocks that span the entire chip, quadrant clocks that span ¼ of the chip, and PCLKs that span a limited region and are generally used near the device I/Os.

3. I/O Bandwidth

Table 2 highlights one of the challenges of scaling FPGAs. While the computational capacity has doubled, Stratix IV has the same number of general I/O pins as Stratix III. This occurs because neither the large I/O transistors nor the package balls used to connect to a PCB are scaling at the same rate as logic transistors. To bring enough data into the FPGA to keep the computational units busy, we need to increase the data rate per I/O pin.

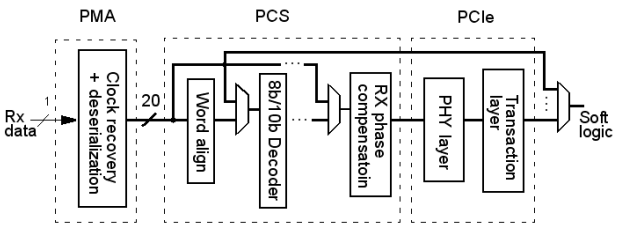

Figure 2: Serial receiver block diagram.

One solution is to build high-speed serial transceivers. The Stratix IV transceivers consist of several configurable layers; some applications may use all of the layers, while others may bypass some to do more customized processing in soft logic. Consider a receive channel, as illustrated by the simplified diagram in Figure 2. First, the Physical Media Attachment (PMA) layer recovers a clock from the data stream, and de-serializes the data from (for example) a 1-bit wide 10 GHz stream to a more manageable 20-bit wide 500 MHz stream. The Physical Coding Sublayer (PCS) can then perform a variety of functions such as word alignment and 8b/10b decoding. Finally, if the PCIe protocol is being used, the hard PCIe block can be programmed to implement the entire, or part of, the protocol stack.

Memory protocols, such as DDR3, are another important high-speed interface. Instead of embedding a clock in the data stream, these standards send a strobe (DQS) signal alongside several data (DQ) signals. The strobe edge is used as a trigger to capture the data at the FPGA and the memory device. Stratix IV supports memory interface speeds of 1067 Mb/s per pin. At those speeds, variation in voltage, temperature and process make it impossible to statically align the DQS signal with DQ signals for reliable data capture. Consequently, Stratix IV *calibrates* memory interfaces by writing and reading training patterns to and from the memory device with different delay offsets between the DQS strobes and the various DQ data bits until it determines the optimal alignment.

The design of high-speed interfaces is extremely challenging, and the challenge is heightened in an FPGA because the circuitry must also be highly programmable so that it can be used in many different systems.

4. Device Modeling

The smaller transistors of advanced process nodes are inherently more variable, since there is less statistical averaging of parameters such as dopant density. In addition, supply voltage is scaling more quickly than transistor threshold voltage, which makes the delay of transistors more sensitive to power supply voltage fluctuations. Finally, the very fast edge rates necessitated by high-speed I/Os increase the inductive coupling between I/Os, making signal integrity a larger concern. These effects are common to FPGAs and ASICs, but the modeling challenge is larger for FPGAs because the CAD tools used by FPGA designers must be fast and easy-to-use to maintain the design cost and design time advantages of FPGAs.

At the 40 nm node, Altera addressed these challenges with a series of major upgrades to its Quartus II CAD suite. Each delay used in static timing analysis is now a min-max range of possible delays, which is used to model process and other variations across the die. Global variations are modeled by performing timing analysis at multiple "corners", or global process, voltage and temperature values. For Stratix IV, three corners, plus suitable on-die variation models, are sufficient to capture all of the timing variability. The placement-and-routing engine within Quartus analyzes the timing at multiple corners as it optimizes, to ensure it is producing an implementation that robustly meets timing at all corners. The calibrated memory interfaces described in Section 2 necessitated novel upgrades to the timing engine, as the timing of these circuits is adaptive rather than truly static.

The Signal Integrity Analyzer is a new tool within Quartus II that analyzes I/O signal integrity to flag unsafe designs and to guide signal integrity optimization. It performs a circuit simulation of the I/O drivers within the FPGA, the signal paths through the package and to the board, and the power distribution network created by the board, package and die, as all these elements contribute significantly to simultaneous switching noise. Using HSPICE to simulate this system takes approximately one week for a full FPGA design—this is unacceptable for most FPGA designers. Through novel simulation techniques the Quartus II Signal Integrity Analyzer reduces CPU time to about 30 minutes on a typical workstation, while still producing results within a few percent of silicon measurements.

5. Power Management

As Section 1.1 described, the power budget per FPGA is fixed by system design constraints, regardless of the process node. Since moving to a next-generation process roughly doubles the number of transistors in the device and makes them inherently leakier, power management is a major concern. Sub-threshold leakage is the dominant component of leakage. While it can be controlled by using a larger transistor threshold voltage (Vt), increasing the Vt of all transistors comes at a major cost in performance.

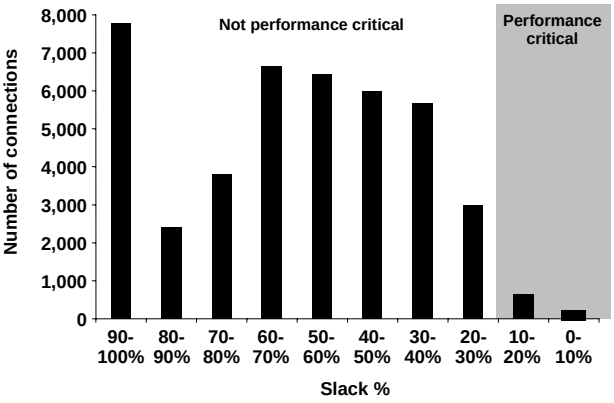

Figure 3: Slack histogram for an example FPGA design.

Figure 3 motivates a different solution. This figure comes from a 300 MHz FPGA design. The horizontal axis lists the slack of a connection as a percentage of the clock period. The vertical axis shows how many connections in the design fall within each range of slacks. Notice that the vast majority of connections have more than 20% slack – they could slow down by at least 20% of the clock period without impacting the critical path, and therefore without slowing down the design. This suggests that only a minority of transistors need to be fast

(and therefore) leaky to maintain the speed a design. Different designs have different critical paths however, so we need a programmable way to change some transistors from "slow and not leaky" to "fast and leaky" to accommodate the needs of different circuits.

MOS Transistors are often thought of as 3-terminal devices, where a gate voltage controls the conductivity between the source and drain. However, as Figure 4 shows, there is a fourth terminal: the body. Connecting the body of an nMOS device to a negative voltage, instead of the conventional 0 V, increases its threshold voltage, making it slower and less leaky. By using a pass transistor controlled by an SRAM cell to select the body voltage for a transistor, we can modify its threshold voltage and hence its speed vs. power trade-off. In Stratix IV, all the transistors in a tile (a group of logic elements) share the same programmable body voltage, making the amortized cost of the SRAM cells and pass gates to control the body voltage very small. The Quartus II CAD system automatically reclaims power by setting all tiles that do not contain critical paths to the low power state; this increases their delay by 20% and reduces their leakage by 2.5X. On average, 85% to 90% of the tiles in a Stratix IV device can be set to the low power state without impacting any critical paths, reducing the overall static power by more than 2X with no performance impact [2].

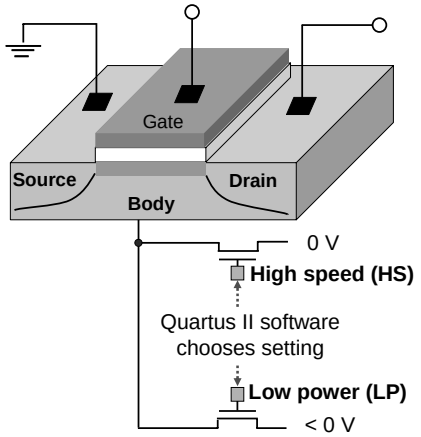

Figure 4: Programmable body bias.

6. Designer Productivity

Figure 5 shows that FPGA capacity has increased by nearly 70X over the past decade, while CPU speed has increased by only 16X. This means that we must innovate to keep the runtime of FPGA CAD systems reasonable.

Over the last several years, Altera has reduced the runtime of Quartus II by more than 5X through a combination of finding new CAD algorithms with better run-time and by creating parallel CAD algorithms to exploit multi-core CPUs [3]. This more than bridges the gap between the growth of FPGA capacity and CPU speeds. In addition, we have created new "incremental" compilation flows so that small design changes do not require a full recompilation; this greatly reduces run-time for such changes.

As ever-larger systems become feasible in FPGAs, it is important to raise the level of design abstraction to allow the specification of these systems in a short time. There has been significant progress in raising the abstraction level in some domains. For example, Altera's DSP Builder Advanced Blockset [4] automatically converts a Simulink specification into a very highly optimized FPGA implementation of the desired signal processing. SOPC Builder [5] allows complex systems to be created without HDL coding by automatically connecting IP cores via a switch fabric and arbitration logic. Several systems that implement programs in higher-level languages such as C [6] or Brook [7] in hardware have also been developed in recent years. While recent progress is encouraging, much more remains to be done in this area.

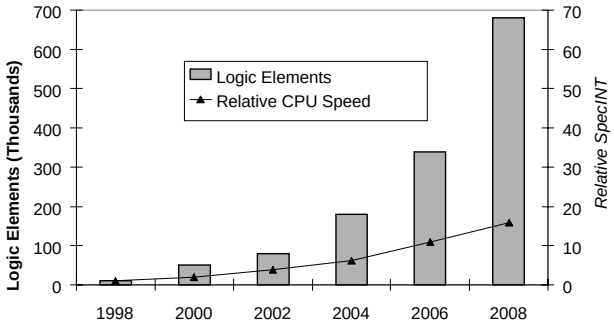

Figure 5: FPGA capacity and CPU speed vs. time.

7. Conclusion

Moving to the next-generation process roughly doubles FPGA capacity and allows the integration of new system-level features. Because they are highly programmable, the target market for FPGAs is very large, allowing FPGA companies to recoup their investments in advanced-process FPGAs. For these two reasons, process scaling is highly favorable for FPGAs, and FPGAs are among the earliest adopters of next-generation process technology.

There are many challenges to overcome to realize the full benefit of advanced processes. Chief among them are achieving sufficient I/O bandwidth, modeling the complexity of modern devices with tools that are still easy-to-use, managing power, and improving designer productivity. Great progress has been made on these challenges at the 40 nm node, but more innovation will be needed as the march to ever smaller transistors continues.

8. References

[1] Stratix IV Device Handbook, available at *www.altera.com*.

[2] D. Lewis, et al, "Architectural enhancements in Stratix-III and Strativ-IV, *FPGA*, 2009, pp. 33 – 42.

[3] A. Ludwin, V. Betz and K. Padalia, "High-quality, deterministic parallel placement for FPGAs on commodity hardware," *FPGA*, 2008, pp. 14 – 23.

[4] DSP Builder Advanced Blockset User Guide, available at *www.altera.com*.

[5] Quartus II Handbook, Volume 4: SOPC Builder, available at *www.altera.com*.

[6] D. Pellerin and S. Thibault, *Practical FPGA Programming in C*, Prentice Hall, 2005.

[7] F. Plavec, Z. Vranesic, and S. Brown, "Enhancements to FPGA Design Methodology Using Streaming," *FPL*, 2009, pp. 294 – 301.