FOCUSREPORT

Published in **June / July 2005** issue of Chip Design Magazine

Navigating the Silicon Jungle: FPGA or ASIC?

FPGA, structured-ASIC, and ASIC design implementations can be differentiated by tradeoff studies and an understanding of the basics behind each target platform.

By John Blyler

As if designing todayâ€™s chips werenâ€™t challenging enough, engineers now have new considerations to balance when choosing the best way to implement their designs. High-end field-programmable gate arrays (FPGAs) are growing in density while handling higher-speed applications and more complex designs. On the other hand, high-end application-specific integrated circuits (ASICs) are considered unpredictable at 90 nm and below. So how can a designer choose the best way to implement a design? Although the line is blurring between some traditional ASIC and FPGA applications, the basic ground rules still apply. Yet designers do need to appreciate the subtle differences between each target platform. This report will sift through the confusion and offer some insights into the best path to silicon implementation.

Traditionally, standard-cell ASICs were considered the best implementation choice for high-volume, price-sensitive applications--especially in the consumer commodity market. At the other end of the spectrum, FPGAs were used mainly to prototype portions of a large ASIC design. FPGAs were limited to low-volume, price-insensitive applications. Structured ASICs, which have some limited design flexibility, fell somewhere in the middle.

Are these generalizations still true? Havenâ€™t recent trends in FPGAs included higher speeds, higher gate counts, and the integration of microprocessor- and controller-based devices? How does the process geometry affect the selection between an ASIC and an FPGA? And where does design complexity fit into the problem? The answers to these and other key questions must be understood before one can select the best path of silicon implementation. As with most engineered solutions, the final answer depends greatly upon a variety of design-tradeoff studies. Even with the results of these studies, however, each decision point carries a host of caveats. But this is just the kind of technical balancing act that most engineers enjoy.

BASIC CRITERIA FOR COMPARISON

Any journal or magazine on chip development will feature some variation of the following four design requirements: power, performance (mainly speed), area or circuit density, and flexibility and cost. The answers to the more general questions of ASICs versus FPGAs are best framed in the context of these constraints. Take power, for example. Typically, cell-based ASICs offer the lowest power consumption. They are followed by structured ASICs and then FPGAs. Chris Tennant, Program Manager for the Custom LSI Strategic Business Unit of NEC Electronics America (www.necelam.com), provides this example: â€œWe compared power consumption between our 150-nm ISSP1 structured ASIC, a 150-nm cell-based ASIC, and a 150-nm FPGA. Operating at full speed, the ISSP1 structured ASIC only consumed 1.7 times more power than the cell-based ASIC. The FPGA used 9.97 times the power of the cell-based ASIC (see Figure 1). Also, keep in mind that to provide the same performance as a cell-based or structured ASIC, a design may need to use multiple FPGAs. This adds extra power consumption, as I/O buffers connecting chip to chip drastically increase power consumption.â€?

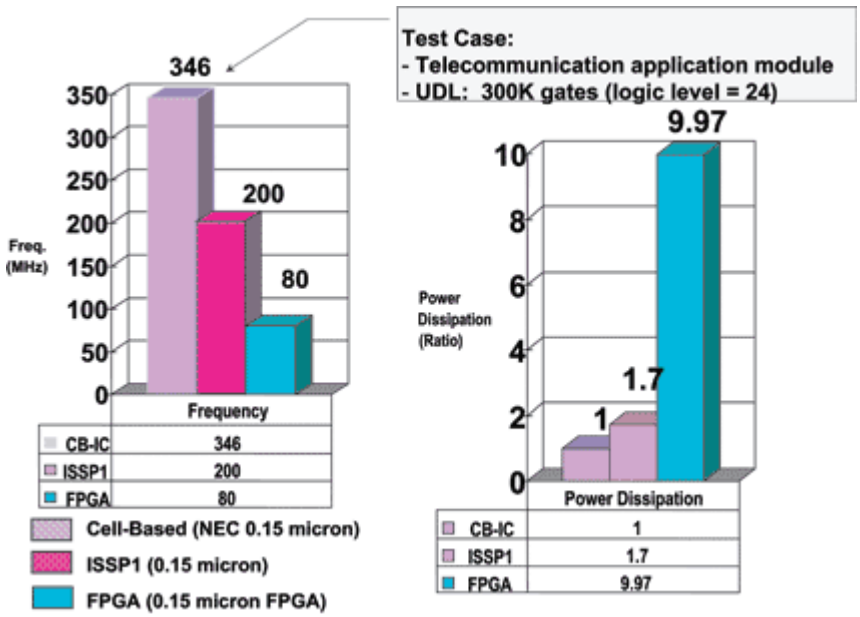

Figure 1: This telecommunications-application study of a 150-nm, 300K-gate design shows the power consumption between FPGAs, structured ASICs, and ASICs.

FPGAs thus consume more power--mainly because they tend to have the lowest gate utilization. They also rely on memory technology and lookup tables, which donâ€™t offer particularly efficient power or speed. But keep in mind the previously mentioned â€œcaveatsâ€? as well as the question about process geometry. Although it is extremely useful, the NEC study was based on 150-nm geometry. Isnâ€™t power consumption much less at the lower nodes like 90 nm and below?

In general, shrinking process technologies do provide the benefits of higher performance and density with lower dynamic power consumption and cost. In other words, FPGAs that are designed at 90 nm will consume less power than those designed at 150 nm. Of course, thereâ€™s no such thing as a free lunch. As the market migrates to 90 nm and below, all of these devices suffer from increased leakage current. This leakage consumption becomes the dominant source of power consumption.

One way to mitigate these effects in FPGAs is to add another oxide layer. As explained by Babak Hedayati, Senior Director of the Product Solutions Marketing & Partnership Programs at Xilinx (www.xilinx.com), â€œMost ICs use two oxide thicknesses in fabrication. A thin oxide provides maximum performance for circuits in the core of the device at the expense of additional leakage current. A thicker oxide provides the ability to handle higher voltages in the I/O pads. Because of the core transistors required for configuration and routing, FPGA manufacturers face an additional challenge in controlling leakage current in comparison to ASICs and structured ASICs. To keep leakage current in check, Xilinx uses a third, middle-thickness oxide when fabricating the transistors that control configuration and routing. Because these functions are not speed critical, the ability to trade off lower performance for lower leakage current helps reduce overall power consumption in the migration from 130 nm to 90 nm.â€?

Performance, which is usually synonymous with system frequency or speed, is another critical factor in the tradeoffs between ASICs and FPGAs. ASICs offer the highest performance possible for most applications. Yet a surprising number of relatively high-speed applications are being done using FPGAs, notes Cadenceâ€™s (www.cadence.com) Brad Griffin, Product Marketing Director for Allegro- SIP. â€œMany of our system (PCB) customers report that their highest-speed devices come from Altera (www.altera.com) and/or Xilinx.â€?

Agreeing that ASICs are still the leaders in overall system performance, the experts at eInfochips design services point out that they have received inquiries from wireless-infrastructure OEMs. Those OEMs want to replace their existing ASIC/ASSP-based boards with FPGA-based design. They aim to take advantage of the higher performance and flexibility offered by the latest FPGAs--particularly in the emerging area of streaming-media technology.

One recent advance in FPGA technology has been the development of high-performance, source-synchronous interfaces in I/O-pin interfaces. Many of Xilinxâ€™s latest FPGAs, for example, use source-synchronous interfacing techniques in which every device transmits a clock signal along with the data. The company claims that this approach enables data rates of up to 1 Gbps per pin.

The next global consideration is silicon size. Assuming a non-pad-limited design, says Elie Massabki, ChipX (www.chipx.com) vice president of marketing, then density is the key factor that affects size. Thus, standard-cell-ASIC designs will normally be the smallest in size followed by structured ASICs and then FPGAs. Keep in mind that FPGAs require a memory lookup table that also consumes valuable silicon area.

Flexibility is considered the one area where FPGAs clearly shine above standard-cell and structured ASICs. FPGAs inherently provide the most design flexibility, as they are capable of implementing software-based design changes in the field. Structured ASICs, which can require as few as two customized metal layers, are the fastest ASICs to change. When reconfiguring a structured-ASIC design, the engineer only needs to reconnect the chipâ€™s available resources. Everything else remains unchanged. Not surprisingly, the customized layers of traditional cell-based ASICs make them the least flexible for design changes.

Cost is the final high-level consideration in the choice of silicon implementation. Lately, much has been made of the high mask costs for ASIC design. Those prices certainly are high--especially at 90 nm and below. But FPGAs cost money too. High-performance (90-nm) FPGAs can cost hundreds or thousands of dollars per unit, notes NECâ€™s Chris Tennant. Most ASIC NREs, for example, include 10 free engineering samples. Say the FPGA implementation requires three devices. The 10 ASIC prototypes would instead require 30 FPGA devices, which can cost more than $30,000. Structured-ASIC NREs start in the $50,000 range and rise with the designâ€™s size and process technology. In addition, the design tools tend to cost more for an ASIC. Tennant concludes by saying that cell-based ASICs are the most expensive design platform. Design-tool costs run up to $300K. Overall NRE costs range anywhere from $1 to $3 million.

COMPLEXITY IS THE KEY

It may seem like ASICs are the most risky of the available silicon implementation choices. Traditional cell-based ASICs can be expensive to design, manufacture, and change. Many view this inflexibility to change as the biggest reason to avoid ASICs. But Naveed Sherwani, President and CEO of Open Silicon (www.open-silicon.com), believes that the ASIC problem has been greatly exaggerated and misunderstood. â€œI believe that ASICs have been projected in the market as being expensive, unreliable, and unpredictable,â€? says Sherwani. While this statement may be true for very high-complexity ASICs (e.g., 10 million gates at 90 nm), it is not true for most ASIC projects. When people talk about the overall ASIC business, Sherwani notes, they take the reputation of these high-end, risky designs and paint a picture of all ASICs.

The real crux of the ASIC confusion lies in the mistaken belief that all ASICs are the same. Sherwani suggests that designers consider the volume-complexity curve for ASICs. From this curve, three distinct types of categories emerge (see Figure 2). High-complexity ASICs offer the greatest risk. By far, the greatest volume of ASICs falls in the medium-complexity area. Finally, low-volume and lower-complexity ASICs are the least risky. In terms of complexity, this area is where most FPGA and structured-ASIC designs would fit.

Figure 2: The infamous bell-shaped curve helps explain the three categories of low-, medium-, and high-complexity ASICs.

But how can complexity be measured? Sherwani puts forth a 22-point criteria that includes factors like the number of gates; process technology; package type; heat dissipation; IP; number of memory types; distribution of memories; and clocking schemes. From the analysis of this criteria plus chip-volume projections, a determination of the complexity domain--low, medium, or high--can be established. If the design project falls in the high-complexity domain, the design will be less predictable and reliable. For medium-complexity projects, success can be more certain. Sherwani explains that medium-complexity ASICs are much more likely to work the first time. They also have a very small chance of re-spin and very high predictability. If a design fits into the low-volume, low-complexity category, it may be best implemented using an FPGA.

This report has made no mention of the vast array of EDA tools that is needed to design FGPA, structured-ASIC, or ASIC designs. Rather, it has focused on the technical criteria and tradeoffs that must be considered when implementing a design in silicon. It is not always clear if a design will be implemented as an FPGA or ASIC. It therefore makes sense to use EDA tools that do not lock the developer into any one design flow, architecture, or vendor. Yet as Mentorâ€™s (www.mentor.com) Juergen Jaeger, Director of Marketing, Design Creation and Synthesis, notes, a few basic differences exist between the two silicon types: â€œThe pre-built nature of FPGAs drives a â€˜use or loseâ€™ approach to features and capabilities. FPGA design--more often than ASIC design--must match functional requirements with the device architecture. Thus, common steps like synthesis or place and route (P&R) have all proved to differ subtly in the FPGA domain. But steps such as RTL simulation remain basically unchanged.â€?

In the end, nothing can replace good, sound engineering analysis. Choosing the best way to implement a design depends upon a host of tradeoff analysis. One also must understand the subtle differences between target platforms--from FPGAs to structured ASICs or ASICs.