Structured ASICs: Opportunities and Challenges

Behrooz Zahiri
Magma Design Automation
behrooz@magma-da.com

Abstract

There is currently a huge gap between the two main technologies used to implement custom digital integrated circuit (IC) designs. At one end of the spectrum are field programmable gate arrays (FPGAs). *These devices have relatively low design costs and short design times, but they also have high per-unit costs and are limited in terms of design size, complexity, and performance. At the other end of the device continuum are* application specific integrated circuits (ASICs). *These components have exceedingly high design costs and take a long time to develop, but they can support extremely large, complex, and high-performance designs, and they have low per-unit costs in large production runs.*

A new category of devices – known as structured ASICs *– is now becoming available. These devices bridge the gap between FPGAs and ASICs in terms of cost and capabilities, but they also pose challenges to device manufacturers and design tool vendors.*

1. Introduction

A large percentage of today's digital *integrated circuit (IC)* designs requiring mid-volume production runs in the order of 10,000 units are not well-served by today's two leading technologies: *field programmable gate arrays (FPGAs)* and *application specific integrated circuits (ASICs)*.

1.1. ASICs

In the case of ASICs – of which the currently dominant form is that of *standard cell (SC)* devices – these are extremely expensive and time-consuming to develop. As IC implementation technologies move into the *ultra-deep submicron (USDM)* realm (specifically the 90 nanometer node and below), power, timing, and signal integrity issues become evermore complex. Reaching closure on these issues takes so much effort that the design team now spends more time addressing these aspects of the design than they spend architecting, capturing, and verifying the logical functionality of the device.

In addition to protracted development times, the photo-masks associated with a new ASIC are becoming prohibitively expensive (in the order of $1 million for a reasonably complex 90 nanometer device). Furthermore, the manufacturing turnaround time to actually fabricate these devices significantly impacts their time-to-market.

The long development and manufacturing times associated with standard cell ASICs pose particular problems with regard to today's short product life cycles and the need to address constantly evolving standards and protocols. However, these devices do have the advantages that they can be used to implement the largest, most complex, high-performance designs. They also have a low per-unit cost when used in large production runs in the order of 50,000 units or more.

1.2 FPGAs

Today's state-of-the-art FPGAs can provide up to 10M system gates, which – depending on the application – equates to somewhere between 1M to 3M ASIC equivalent gates. The addition of features such as embedded RAM, embedded processor cores, and gigabit transceivers means that FPGAs can now be used to implement reasonably large and sophisticated designs (although the largest and most complex designs remain the domain of ASICs).

FPGAs are fully prefabricated by the vendor, which means that there is no manufacturing turnaround time to be accounted for in the design cycle. Furthermore, creating an FPGA design is, in many respects, simpler and faster than would be its ASIC counterpart. This is because considerations such as signal integrity have already been addressed by the device's manufacturer and/or are automatically handled by the FPGA's design tools (in both cases this occurs transparently to the end user).

The fact that many FPGA families can be reconfigured (reprogrammed) means that they are ideal for use with applications whose standards and protocols are constantly evolving. However, FPGAs also have significant disadvantages, in that their designs consume significantly more power and have much lower performance than equivalent ASIC implementations. Furthermore, FPGAs have a high per-unit cost, which makes them an extremely

Proceedings of the 21st International Conference on Computer Design (ICCD'03)
1063-6404/03 $ 17.00 © 2003 IEEE
IEEE COMPUTER SOCIETY

expensive option for anything other than prototyping applications or relatively small production runs.

1.3 The requirement

All of the above points serve to illustrate that there is a huge gap between the two main technologies currently used to implement custom digital IC designs (Figure 1).

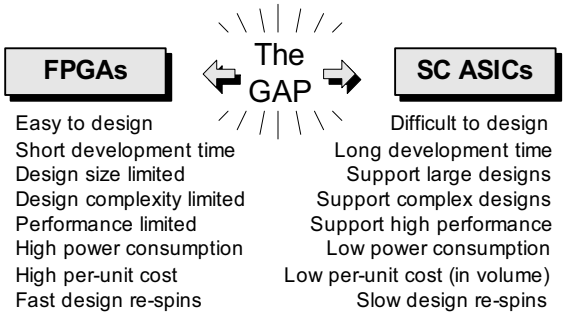

Figure 1. The FPGA-ASIC gap

The rising cost of developing standard cell ASICs means that many companies can no longer afford to use these devices. At the same time, FPGAs aren't appropriate for many of these designs due to capacity and performance issues and/or high per-unit costs. What is required is a new implementation technology that overcomes the design size, complexity, performance, and power consumption limitations of FPGAs, but which also addresses the long development times, high development costs, and long manufacturing lead times associated with standard cell FPGAs. In addition, this new technology should offer a reasonably low per-unit cost, thereby making these components suitable for medium-size production runs.

The solution may well be a new class of devices known as *structured ASICs (SAs)*. This paper introduces the concept of structured ASICs along with some comparisons between standard cell, structured ASIC, and FPGA implementations. Also provided is an overview of some of the alternative structured ASIC architectures that are currently being made available to the market. Finally, the paper discusses the challenges these devices present to vendors of *electronic design automation (EDA)* tools.

2. The structured ASIC concept

The underlying concept behind structured ASICs is actually fairly simple. Although there are a wide variety of alternative architectures, they are all based on a fundamental element called a "tile" by some or a "module" by others (this paper will use the term "tile" henceforth). This tile contains a small amount of generic logic implemented either as gates and/or multiplexers and/or a lookup table. Depending on the particular architecture (see also the discussions below), the tile may contain one or more registers and possibly a very small amount of local RAM.

An array (sea) of these tiles is then prefabricated across the face of the chip. Structured ASICs also typically contain additional prefabricated elements, which may include configurable general-purpose I/O, microprocessor cores, gigabit transceivers, embedded (block) RAM, and so forth (Figure 2).

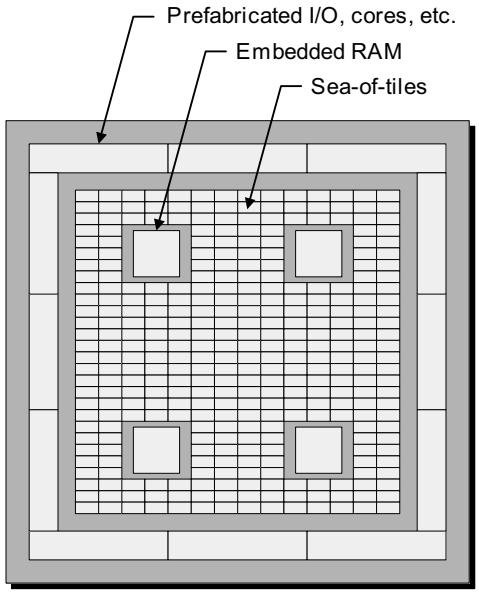

Figure 2. The structured ASIC concept

In many respects these devices are similar to modern, high-end gate array ASICs. The key differentiator with regard to Structured ASICs is that the majority of the metallization layers are also prefabricated. This means that the transistors forming the core logical functions comprising each tile (gates, multiplexers, etc) are already wired together. Also, much of the local and global interconnect has also been implemented. Depending on the architecture, the design engineers need specify only one, two, or very few metallization layers in order to complete the device.

2.1 SA advantages

One by-product of the structured ASIC philosophy is that these devices are much easier and faster to design than are their standard cell cousins. There are a variety of reasons for this, such as the fact that multiple global and local clock domains are typically prefabricated in the master

IEEE COMPUTER SOCIETY

fabric and are implemented in such a way that there are no skew problems that need be addressed by the design engineers.

Similarly, design-for-test considerations are addressed by the fact that functions such as boundary scan (JTAG), full internal scan, and BIST are all typically embedded in the basic fabric. In order to mitigate USDM timing and signal integrity effects, the ASIC vendor works to pessimistic, highly guard-banded specifications. This allows signal integrity issues and timing issues (in the form of setup and hold violation times associated with internal registers) to be automatically addressed by the architecture or the design tools.

Due to the fact that structured ASICs need only a limited number of metallization layers to complete them, the costs associated with generating the photo-masks are dramatically reduced. Furthermore, the fact that the device is largely prefabricated radically shrinks the turnaround time to working silicon. This also means that structured ASICs can undergo faster and cheaper modification cycles in order to accommodate evolving standards and protocols.

Overall, the capacity, performance, and power consumption of a structured ASIC is much closer to that of a standard cell realization of the design as opposed to an FPGA implementation. Additionally, the faster design time, lower mask costs, and quicker turnaround to final silicon – along with the lower costs resulting from the fact that the majority of the device is pre-fabricated – means that the per-unit cost of structured ASICs is extremely reasonable for medium-low to medium-high production runs.

2.2 SA disadvantages

One problem with structured ASICs is that the current design tools – which are currently predominantly based on traditional ASIC offerings – are both expensive and not well-suited to the task. Another is that the diverse architectures fielded by the various vendors are so new that they have not yet been subject to any form of formal evaluation and comparative analysis (unlike alternative FPGA architectures – such as the tradeoffs between 3-, 4-, and 5-input LUTs – which have undergone extensive research by the industry and academia). Both of these topics are addressed in more detail later in this paper.

3. Alternative SA architectures

This is a somewhat "gray" area, because the majority of vendors with structured ASIC offerings are still working in "stealth mode," which means that detailed descriptions of their internal architectures are not readily available. Thus, the following architectural descriptions are "composites" that have been gleaned from a variety of sources.

In addition to its own unique version of a basic tile, each vendor offers its own selection of hard, firm, and soft IP. Hard IP comes in the form of configurable I/O blocks that can be modified (via the user-definable metallization layers) to handle a variety of standard I/O interfaces. Other hard IP blocks include standard interfaces like PCI, gigabit transceivers, microprocessor cores, embedded RAM, and so forth. Each vendor may offer a family of devices containing different combinations of hard IP blocks combined with various quantities of basic tiles.

Firm IP comes in the form of a library of high-level functions that have been optimally mapped, placed, and routed for this vendors particular architecture, while soft IP is presented as a source-level library of high-level functions that can be included into the users' designs.

In many cases the hard, firm, and soft IP from the various vendors are simply variations on a theme. The real differentiator between devices comes in the contents and architecture of the basic tile.

3.1 Extremely fine-grained

Some vendors are evaluating an extremely fine-grained version of a basic tile that comprises only unconnected components such as transistors and resistors (Figure 3).

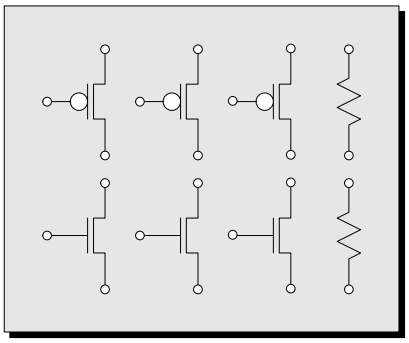

Figure 3. An extremely fine-grained tile

These architectures are extremely close to those of modern high-end gate array devices. The difference being that – in the case of the structured ASIC, metallization has been added so as to almost connect these components in a variety of pre-defined configurations. Thus, the user-definable metallization layers are used to complete the appropriate connections, and to link the tiles into the local and global routing architecture.

IEEE COMPUTER SOCIETY

3.2 Medium-grained tiles

Other vendors have opted for a medium-grained architecture. In this case, the tile might contain some generic logic in the form of gates and/or multiplexers along with one or more flip-flops (Figure 4).

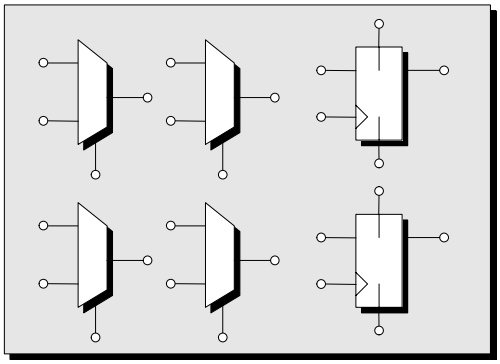

Figure 4. Mux-based medium-grained tile

Alternatively, some medium-grained architectures are based on tiles containing one or more lookup tables (LUTs) along with one or more flip-flops (Figure 5).

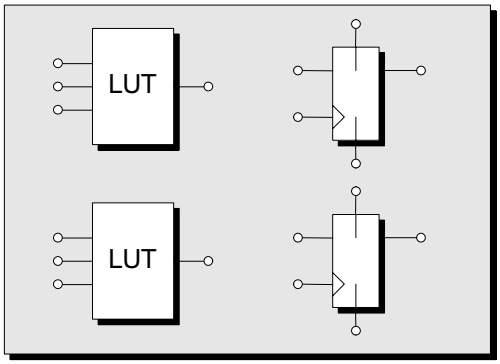

Figure 5. LUT-based medium-grained tile

In both of these cases the polarity of the flip-flops' clock inputs (i.e., whether each register should be positive- or negative-edge-triggered) and the polarity of their set and reset inputs can be determined by the customized metallization layers.

3.3 Hierarchical tiles

As yet another alternative, some architectures commence with a *base tile* containing only generic logic in the form of prefabricated gates and/or multiplexers and/or lookup tables (Figure 6).

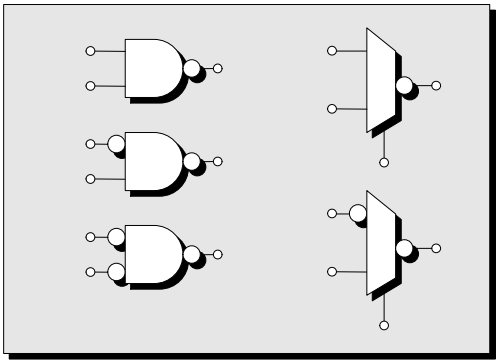

Figure 6. An example "base tile"

An array of these base tiles (say 4 x 4, or 8 x 8, or 16 x 16) are combined with special tiles containing registers, memory elements, and other logic to form a *master tile*, then an array (sea) of these master tiles is prefabricated across the face of the chip.

3.4 Fine versus medium versus coarse

One consideration with regard to the granularity of the architecture is that fine-grained implementations require a lot of connections into and out of each tile compared to the amount of functionality that can be supported by the time. By comparison, as the granularity of the tile increases to medium-grained and higher, the amount of connections into the tile compared to the functionality it can support decreases.

3.5 Tracks versus vias

One final twist that can potentially be applied to all of the above architectures is that some devices require the customization of a number of metallization layers (this might be two tracking layers, or it could be two tracking layers and one or more via layers).

By comparison, at least one vendor is fielding an architecture that requires the customization of only a single via layer. In addition to cutting photo-mask and production costs to a minimum (and further reducing back-end production times), this scheme means that the prefabricated track segments are extremely well characterized in terms of parasitics, delays, and signal integrity issues. The disadvantage is that you have less flexibility with regard to routing, but this may be mitigated to some extent by the granularity of the architecture as discussed in the previous point.

4. SA versus SC versus FPGA

IEEE COMPUTER SOCIETY

Once again, our ability to provide definitive comparisons between structured ASICs and other technologies is somewhat limited due to the lack of hard data supplied by the device vendors.

One important metric is the density of usable equivalent gates per square millimeter (mm^2). This can be confusing even when it comes to comparing standard cell devices to FPGAs, because the former user the concept of an equivalent gate (typically a 2-input NAND), while the latter often base things on the concept of a "system gate." The problem is that the mapping of FPGA system gates to ASIC equivalent gates is design-dependent and is a function of the mix between combinatorial and sequential logic. Keeping this in mind, it is generally accepted that standard cell architectures can support an equivalent gate density of approximately 100,000 gates/mm^2, while FPGAs can only offer around 1,000 gates/mm^2, which is a factor of 100:1. By comparison, some structured ASIC architectures are rumored to support around 33,000 gates/mm2, which is a factor of 3:1 compared to standard cells. That is, a structured ASIC can support 0.33x the number of gates as a standard cell device and 33x the number of gates in an FPGA component in the same area.

With regard to performance, if the same design is implemented in standard cell and FPGA devices, it is typically the case that the FPGA can only achieve 10% to 20% of the performance if the standard cell implementation (in terms of clock frequency). By comparison, early results on structured ASICs suggest that these devices can achieve 70% to 80% of the performance of a standard cell implementation.

In the case of power, FPGAs typically dissipate 10x to 15x that of an equivalent standard cell implementation. Once again, early results on structured ASICs suggest that these devices consume only 2x to 3x the power of their standard cell counterparts.

Some additional metrics that are being quoted (although not referenced) is that the development costs of a structured ASIC design are only 25% those of a standard cell equivalent. Furthermore, the production unit price of a structured ASIC is said to be only 10% of an equivalent FPGA (assuming that an FPGA can meet the design's size, complexity, and performance requirements).

5. The EDA challenge

In order to support early releases of their technology, the majority of structured ASIC vendors are currently using existing standard cell-based design and implementation tools, specifically synthesis, place, and route. This is a major problem, because while them may be designed in a hierarchical manner, at the bottom level standard cell designs are essentially fine-grained and "flat" consisting of gates and registers.

By comparison, structured ASICs are more akin to FPGAs in terms of their block-level (tile-based) structure. Even in the case of a simple tile-based architecture, a tile-aware synthesis engine should be able to offer a minimum of a 25% increase in quality of results (QoR) over a traditional standard cell-based engine. Similarly for the placement and routing engines, and the problems are only exacerbated in the case of hierarchical structured ASIC architectures featuring arrays of master tiles, which themselves comprise arrays of base tiles. In this case, traditional mapping, placement, and routing engines are simply not capable of addressing these architectures in any meaningful way.

Of all of the engines, routing probably offers the most significant challenge. Current design flows based on traditional standard cell routing technology are employing a variety of ad hoc solutions, such as disabling certain metal layers or writing scripts to focus the router on specific areas. However, the majority of these solutions require a significant amount of user intervention.

For all of these reasons, it is now generally accepted that synthesis, placement, and routing engines that were specifically crafted for use with block-based FPGA architectures offer the optimal starting point for a new generation of structured ASIC equivalents.

5.1 Hybrid ASIC-FPGA devices

For a variety of reasons, there is currently a significant amount of interest in embedding one or more FPGA cores into ASIC fabric. Thus far, these cores have been embedded in standard cell ASICs. This has lead to problems due to the lack of commonality between the design tools, flows, and methodologies for these two implementation technologies.

However, if block-based structured ASIC synthesis, placement, and routing engines are developed as discussed in the previous point, then embedding FPGA cores in structured ASIC fabric would offer significant advantages. This is because design tools, flows, and

IEEE COMPUTER SOCIETY

methodologies would share a lot of commonality between these two implementation technologies.

5.2 Architectural evaluation

One concern with regard to the structured ASIC architectures currently being released to the market is the lack tools that are available to the vendors to evaluate these architectures. Thus, another challenge to the EDA community is to provide structured ASIC architects with appropriate architectural evaluation and analysis tools.

6. Summary

It is not the purpose of this paper to suggest that structured ASICs will displace any of the existing implementation technologies. FPGAs will always be the most appropriate way to realize some designs, while standard cell ASICs will always dominate the largest, most complex, high-performance, large production run designs.

However, structured ASICs are closer to FPGAs in terms of the low costs and fast turnaround associated with creating a design. At the same time, they are much closer to standard cell implementations in terms of capacity, performance, low power consumption, and low per-unit costs for medium-low to medium-high production runs.

Thus, structured ASICs offer the possibility of dramatically reducing the time-to-market and also dramatically reducing the engineering, EDA tool, photo-mask, and production costs that are currently associated large, complex, high-performance custom digital ICs.

These devices are now starting to become widely available while at the same time it is becoming expensive for structured ASIC vendors to build and maintain tools for customers targeting them. The challenge for EDA vendors is to create powerful cost-effective industry common tools that can accommodate the unique requirements of these devices. In this, some EDA vendors such as Magma Design Automation have an inherent advantage, because they already dominate the RTL-to-Silicon flow for standard cell ASICs, and they have acquired the best in class FPGA synthesis, place, and route tools. Furthermore, Magma is in an unique position to help structured ASIC vendors develop and optimize their architectures, because it provides a quantitative analysis and a full understanding of the impact of each architectural decision on performance, area, density, and routability.

IEEE COMPUTER SOCIETY