Cyclone™: A Low-Cost, High-Performance FPGA

Paul Leventis*, Mark Chan, Michael Chan*, David Lewis*, Behzad Nouban,
Giles Powell, Brad Vest, Myron Wong, Renxin Xia, John Costello

Altera Corporation, 101 Innovation Drive, San Jose, CA 95134, USA
* Altera Corporation, 151 Bloor St. West, Toronto, ON M5S 1S4, CANADA

Abstract

This paper describes the Altera Cyclone™ FPGA, an architecture specifically designed for low-cost, high-volume applications. An optimized routing architecture, simplified I/O structure, and carefully selected features combine to yield a 57% die size reduction relative to a Stratix™ device of similar logic density, with a 7% reduction in performance.

Introduction

FPGAs have traditionally used the density afforded by new process geometries for features such as memories, DSP functionality, processors, and other IP needed in high-end applications. The cost of these devices makes them uncompetitive with ASICs for high-volume designs. However, increasing non-recurring engineering (NRE) costs and design complexity are pushing ASICs out of reach for all but the highest volume designs, leaving a vacuum that lower-cost FPGAs hope to fill.

FPGAs are an attractive solution for those users that can afford the power and speed penalties they incur over full-mask fabrication. In the past, FPGA vendors have targeted the medium- to high-volume market with cost-reduced versions of previous generation products, often disabling features to reduce test time and increase yields. This approach was used with Altera's ACEX™ 1K and Xilinx's Spartan™-IIE families. Because these "low-cost" families are produced on mature processes, cost reductions are limited and leading-edge performance is unattainable.

Cyclone is a new FPGA family designed for low-cost while maintaining good performance. The capabilities, routing architecture, and system-level features have been selected to match the requirements of designs in this space. While borrowing some features from the Stratix™ family [1], the unique goals of Cyclone lead to different trade-offs of area, delay and features, particularly in the I/Os and routing.

This paper gives an overview of the Cyclone architecture, discusses the rationale behind feature selection, and identifies trade-offs made during its design. We will also discuss in detail the area, speed and routability of the routing architecture.

Device Overview

The Cyclone family is manufactured on TSMC's 8-layer 0.13u all-copper FSG process, and consists of five devices [2] as outlined in Table 1. Each Cyclone device comprises a number of logic elements (LEs) grouped into logic array blocks (LABs), M4K memory blocks, IOs on the periphery, one or two PLLs, and some configuration logic connected through a programmable routing fabric. The 1C6 device floorplan is given in Figure 1.

TABLE 1: CYCLONE FAMILY

| Device	LEs	RAM (Kb)	PLLs	User I/Os
1C3	2,910	58.5	1	104
1C4	4,000	76.5	2	301
1C6	5,980	90	2	185
1C12	12,060	234	2	249
1C20	20,060	288	2	301

FIGURE 1: 1C6 DEVICE FLOORPLAN
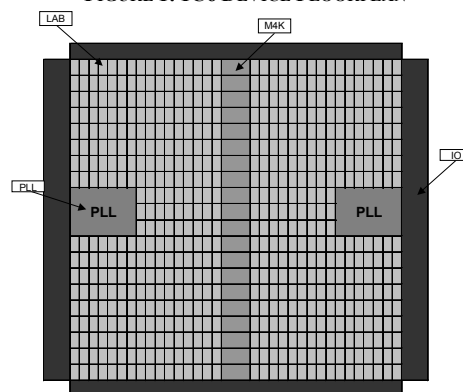
Architecture Details

A. Logic

Cyclone employs the same logic element (LE) as Stratix, as this LE performs well across the range of design sizes targeted by both architectures. The LE comprises a four-input look-up table (LUT) and a register, along with dedicated arithmetic circuitry. The logic elements are arranged in Logic Array Blocks (LABs), groups of 10 LEs that share control signals and a local routing structure.

The LE supports three ways to simultaneously use the LUT and register, as shown in Figure 2. Additionally, flip-flops can cascade between adjacent LEs to form shift registers without consuming general LE inputs or routing.
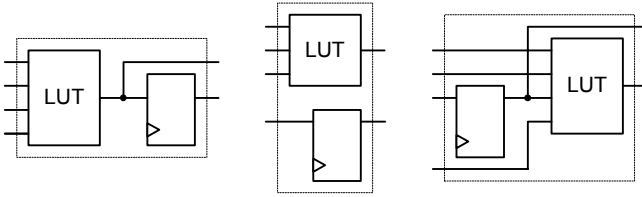
The Quartus® II placement and routing software increases effective logic density by combining a register and LUT into a single LE. Varying levels of effort can be specified, with each successive mode increasing density at the cost of performance and routed wirelength, as shown in Table 2. Aggressive register packing increases effective logic density by 16% while reducing performance by only 3%. Cyclone was architected to support the accompanying 7% increase in wirelength, as the trade-off of performance for logic density is appropriate in a cost-focused family.

TABLE 2: IMPACT OF REGISTER PACKING IN CYCLONE

Packing Level	Design Speed	LE Utilization	Wire Use
Simple	-	-	-
Normal	-1%	-10%	-2%
Aggressive	-3%	-16%	+7%
Aggressive w/chains	-5%	-18%	+10%

B. Memory

Each device in the family has one or two columns of M4K dual-port memory blocks. Each block contains 4608 bits and can be organized in widths from 1 to 36 bits. This memory size and organization provides the best area-efficiency across user designs. For designs that require more memory, a 133 MHz (266 Mbps) double data rate (DDR) interface is supported, providing access to large, cheap off-chip memory.

C. Clocking and PLLs

Cyclone features a simple clock network with eight chip-wide global clocks. Placement experiments showed that restricting a design to using only four clocks per row of logic results in no performance or routability impact. However, the device allows six clocks (selectable from eight) per row. The two additional row clocks were added to support I/O registers, which often operate on a different clock domain from logic (LE) registers in the same row.

The global clocks can be driven from four dedicated and eight dual-purpose clock pins, PLL(s), and the core logic. The clock pins can also be used to drive logic (non-clock) signals onto the clock network, allowing high-fanout, low-skew signal distribution. This capability is critical in some IP cores such as PCI. Some pins can optionally apply a programmable delay chain to phase shift their input, assisting the implementation of DDR.

Each Cyclone PLL can synthesize two global clocks and one external clock that is connected to an output pin for use in system-level clock management. The PLLs are located in the core, minimizing disruption to the I/O layout and hence maximizing I/O count. Half of the FPGA routing tracks continue over the PLLs, reducing the routability and speed impact the PLL has on the core.

D. I/Os

The demands of the high-performance FPGA market result in I/Os that support many electrical standards, and the inclusion of dedicated circuitry and sophisticated clock generation to support extremely high speeds (e.g. 3.125 Gbps on Altera's Stratix™ GX [3], Xilinx's Virtex II-Pro™ [4] and Lattice's ORT82G5™ [5]). This circuitry consumes considerable area. For example, in the Stratix 1S20 42% of the die is used by I/O, dedicated serializer/deserializer (SERDES), PLL, and other peripheral circuitry, making this a key area to target for cost-reduced devices.

Each Cyclone I/O consists of a bi-directional pin with three registers – input, output and output enable – and associated logic for clocking and other control signals. Dedicated DDR circuitry is not included in the I/O, and instead must be built in LEs when needed. The I/O buffers are optimized for area while delivering the performance, reliability, and key I/O standards typically required by cost-sensitive applications. Support for LVTTL, LVCMOS, SSTL-2, SSTL-3, LVDS and PCI was designed in, and other standards may be added based on device characterization. Each I/O can operate at voltages between 1.5 and 3.3V and has programmable drive strength between 4 and 24 mA. Most combinations of voltage, drive strength and I/O standard are available.

I/O area was further reduced by limiting the support of some I/O standards to a subset of I/Os. For example, PCI pins are only found on two sides of the device, saving the expense of PCI clamping diodes in half the I/Os. This selection of standards was done carefully to allow critical IP, such as 64-bit 66 Mhz PCI, to be implemented.

High-speed low-voltage differential signaling (LVDS) I/O is not a requirement of the target application space, and dedicated SERDES hardware is costly. Instead, SERDES is implemented in soft logic as required, reducing area needed on average. The resulting LVDS speed of 311 Mbps per channel is adequate for many applications, though considerably less than the 840 to 1000 Mbps supported in high-end offerings from Altera, Xilinx and Lattice.

E. Configuration

Most FPGAs require off-chip configuration devices to store a configuration bit stream and control configuration after power-up. Traditionally, these devices are costly. Two

features were added to Cyclone in order to reduce the total system cost: compression and active serial configuration.

While both Altera and Xilinx offer configuration devices supporting on-the-fly bit stream decompression, Cyclone is the first FPGA to incorporate decompression hardware, enabling cheap, simple memory chips to be used as the configuration device. The compression algorithm takes advantage of the high density and correlation of 0s in the bit stream. This arises from the predominant use of one-hot encoded multiplexors in the routing; 76% of the configuration RAM (CRAM) bits in a LAB are used for routing. Also, CRAM is set to '0' for unused features, and even a highly congested routing will use only 50-60% of the routing multiplexors. A compression rate of 1.7 - 2.2x is achieved on full designs, with a typical rate of 1.9x.

Cyclone supports a new active-serial configuration mode, where the FPGA controls a simple, inexpensive serial configuration device. This device can program multiple Cyclone devices, and unused non-volatile configuration memory can be accessed by user logic.

F. Packaging

With a low-cost family, the package represents a much larger proportion of total product cost than in high-end FPGAs. Cyclone uses Thin Quad Flat Pack (TQFP) and 1.0mm Fine Line Ball Grid Array (FBGA) packages with wire bond assembly. A modular design supports both logic intensive (e.g. 1C20) and I/O intensive (e.g. 1C4) variants of the basic architecture with minimal design effort. Logic intensive devices use a linear pad ring, placing the I/O cell on a larger pitch, preserving more of the core for logic. I/O intensive devices use staggered pads, with a tighter effective pad pitch to fit more I/O in a given linear dimension. The pad rings use the smallest pad pitch available consistent with high-yield, low-cost manufacturing.

Routing Architecture

The FPGA industry is littered with products that failed to strike the right balance between routing area, speed and routability. Complex trade-offs make experimentation essential to the design of a successful routing architecture.

A. Methodology

Our methodology uses a full CAD flow to obtain place and route data across a suite of benchmark designs for all architectures under consideration. This is made possible through the FPGA Modeling Toolkit (FMT), an extended version of the academic VPR tool [6] from the University of Toronto. This tool was used extensively in the design of Stratix, as described in [7]. The FMT allows a designer to rapidly vary the routing architecture and obtain good estimates of speed, area and routability across a benchmark

set. The FMT area and delay models exhibit high fidelity with actual silicon measurements.

B. Routing Wire Types

Past industry [7] and academic [8] results have shown that reasonably short wires are good for area. Wires that are much longer than a typical connection are rarely used for their entire length, while wires that are too short require multiple buffers and multiplexors to route many connections. Figure 3 shows an FMT sweep where relative area and critical path delay are plotted against routing segment length, measured by the number of LABs spanned. Each data point is the geometric average of results across 65 user designs. This graph indicates that when we limit the routing architecture to one length of wire, wires that span 4 LABs are best for both area and delay.

FIGURE 3: AREA AND DELAY VS. ROUTING SEGMENT LENGTH
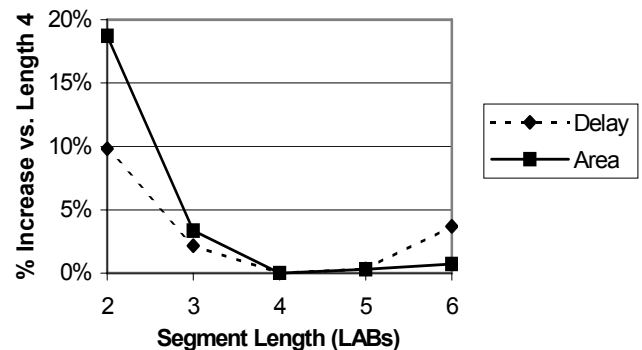
In addition to short wires, many modern FPGAs include a network of longer wires. Examples are the long lines in Virtex-II™ [9] that span the entire chip, and the length 8, 16 and 24 wires in Stratix. These wires confer speed and/or area benefits for large designs, since as circuits increase in size, so too does their average connection length. Our experiments showed that long wires provide little benefit for smaller FPGAs such as Cyclone. Furthermore, by simplifying the routing architecture so that only one length of routing wire is used, a simpler, more compact layout can be achieved.

C. Tracks Per Channel

Channel width, or number of tracks per channel, is selected to achieve good routability while tolerating some no-fits for especially difficult cases. As previously observed [10], this reduces costs for most users while forcing a few designs to use the next larger device.

Figure 4 shows an FMT generated plot of channel width versus circuit size in LEs. Due to the generality of the FMT CAD suite, its channel width predictions may exceed the amount of routing required in a tuned CAD flow. Further detailed experimentation was performed in Quartus II, with a routing model incorporating layout constraints. An

architecture with 80 horizontal length 4 and 80 vertical length 4 wires per channel was found to be acceptable, with only 9% of designs failing to route. After further tuning of connectivity patterns and CAD tools, all routing failures were eliminated. Besides the obvious reduction in metal area, a reduced channel width results in fewer drivers and multiplexors, and fewer inputs per multiplexor, reducing the diffusion area required.

FIGURE 4: CHANNEL WIDTH VS. CIRCUIT SIZE
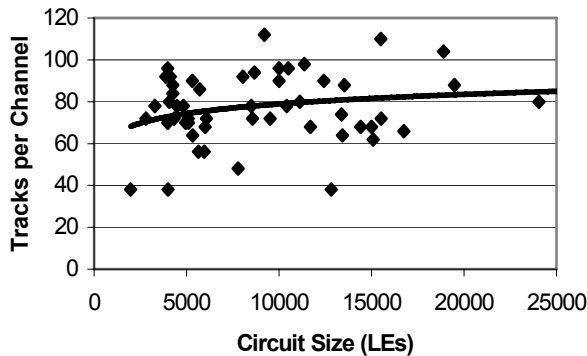
D. Electrical Design and Physical Implementation

Additional area was saved through the re-optimization of transistor sizes to balance area and delay. The electrical design was guided by a delay profile that weights each transistor according to its relative effect on the delay of an average critical path.

Results

Because Cyclone is the first low-cost product produced in 0.13u, there are no low-cost products to which we can fairly compare. To highlight the achievements of the Cyclone architecture, we are providing a comparison with Stratix, a full-featured FPGA produced on the same process. The Stratix 1S20 has a similar number of LEs as the 1C20, but has far more memory, I/Os and other high-end features.

FIGURE 5: STRATIX 1S20 VS. CYCLONE 1C20 AREA
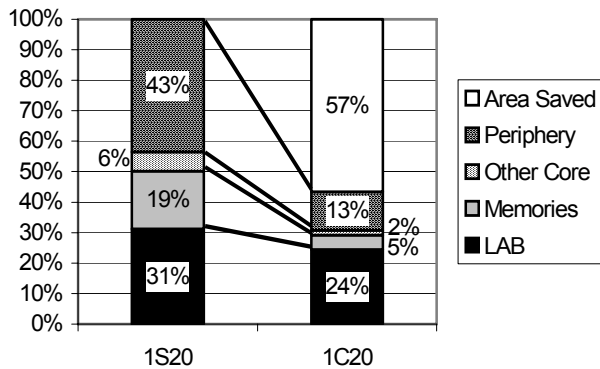
A. Area

By targeting the lower density design space with a new routing architecture and electrical redesign, LAB area

(including the associated routing) was reduced by 27% relative to the Stratix LAB. Coupled with improved I/O area and a reduced number of memories and I/Os, a 57% overall die size reduction was achieved between the Stratix 1S20 to the Cyclone 1C20. Figure 5 shows a break-down of this area reduction.

B. Performance and Routability

Figure 6 shows a speed comparison over 58 user designs implemented in Cyclone versus the same designs implemented in Stratix. On average, Cyclone is 7% slower, and all designs route, regardless of logic utilization.

FIGURE 6: CYCLONE SPEED VS. STRATIX SPEED
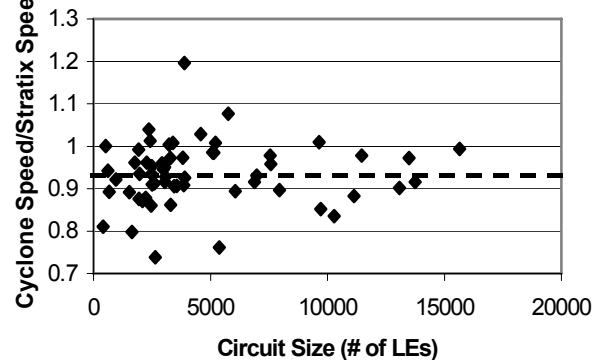
Speed was reduced relative to Stratix through area-targeted transistor resizing, exclusion of the longer wires, and removal of DSP blocks. The speed reduction was mitigated by a smaller LAB area, which reduces the physical length of routing resources, and the removal of many non-LAB blocks, allowing LABs to be packed closer together.

Conclusion

Carefully designed for low-cost, Cyclone provides a considerable die size reduction while delivering performance approaching that of high-end FPGAs. This effectively narrows the ASIC-FPGA cost and performance gap, enabling FPGAs to further penetrate the high-volume ASIC business.

Acknowledgements

References

[1] Stratix Data Sheet, *www.altera.com*
[2] Cyclone Data Sheet, *www.altera.com*
[3] Stratix GX Data Sheet, *www.altera.com*
[4] Virtex II-Pro Data Sheet, *www.xilinx.com*
[5] Lattice ORT82G5 Data Sheet, *www.latticesemi.com*
[6] V. Betz and J. Rose, "Automatic Generation of FPGA Routing Architectures from High-Level Descriptions", *ACM/IEEE Symposium On FPGAs (FPGA) 2000*, pp. 175-184.
[7] D. Lewis, et. al., "The Stratix PLD Routing and Logic Architecture", *FPGA 2003*, pp. 12-20.

[8] V. Betz and J. Rose, "FPGA Routing Architecture: Segmentation and Buffering to Optimize Speed and Density", *FPGA 1999*, pp. 59-68

[9] Virtex II Data Sheet, www.xilinx.com

[10] A. DeHon, "Balancing Interconnect and Computation in a Reconfigurable Computing Array (or, why you don't really want 100% LUT utilization)", *FPGA 1998*, pp. 69-75.