

100,000 Gates and Beyond

By BRADLY FAWCETT ♦ Editor

The time is fast approaching when programmable logic devices will exceed 100,000 usable gates of logic and 100 MHz system clock speed. At these performance levels, the device architectures, design tools and development flows that have



been successful for sub-10,000 gate designs will not always suffice. Ignoring for now the issue of how to actually measure "gate capacity," let's examine what will be required of very high-density programmable logic in the near future. (And I'm sure you won't

be surprised to learn that Xilinx is well-positioned to meet those requirements.)

As FPGA-based designs get larger and more complex, FPGA architectures and development tools must support a design methodology that mimics high-density ASIC design flows, while delivering the flexibility and time-to-market benefits of FPGA technology. The FPGA device and its development tools must be "synthesis-friendly" — they must easily yield efficient, cost-effective solutions when starting with a high-level description of the design.

Achieving this goal requires attacking the problem from both ends; the architecture must be symmetrical, regular and provide an "easy" target for both synthesis and "place and route" tools. Furthermore, the tools — especially the synthesis compilers — must be tuned to produce the best results for the particular architecture. This, in turn, implies close cooperation between the designers of the FPGA architecture and developers of the synthesis tools. To this end, Xilinx has a Synthesis Syndicate program dedicated to sharing

information with third-party CAE developers, as well as an ongoing co-development agreement with Synopsys, the leading supplier of synthesis tools.

Some Tools Already In Place

As with today's high-density FPGAs, **timing-driven tools** that take into account the design's performance requirements during placement and routing (such as the XACT-Performance™ feature of PPR) are key to meeting performance goals in large, complex designs. Very large FPGA designs also could benefit from something that is not available today — **timing-driven mapping**. Much of the technology mapping (that is, the mapping of the user's logic into the logic blocks and other resources of the FPGA architecture) must occur during synthesis so the synthesis tools can make the appropriate

XCELL

Please direct all inquiries, comments and submissions to:

Editor: Bradly Fawcett

Xilinx, Inc.

2100 Logic Drive

San Jose, CA 95124

Phone: 408-879-5097

FAX: 408-879-4676

E-Mail: brad.fawcett@xilinx.com

©1995 Xilinx Inc.
All rights reserved.

XCELL is published quarterly for customers of Xilinx, Inc. Xilinx, the Xilinx logo and XACT are registered trademarks; all XC-designated products, UIM, HardWire, XACTstep and XACT-Performance are trademarks; and "The Programmable Logic Company" is a service mark of Xilinx, Inc. All other trademarks are the property of their respective owners.



"As FPGA-based designs get larger and more complex, FPGA architectures and development tools must support a design methodology that mimics high-density ASIC design flows, while delivering the flexibility and time-to-market benefits of FPGA technology."

area/performance trade-offs. This again implies close cooperation between the chip architects and the synthesis providers.

Designs with 100,000-gates are likely to include datapath logic and memory func-

Continued on page 4

Continued from page 2

tions, as well as the traditional “glue logic” found in today’s FPGA designs. **Automatic placement tools** must support the effective placement of these various types of circuit structures, and will need to be augmented by interactive floorplanning tools. XACTstep™ version 6 includes a new version of PPR with improved structured-placement capabilities, as well as the industry’s first FPGA floorplanner. The synthesis tools will need links to the automatic placement and floorplanning tools, allowing for the passing of design constraints and the back-annotation of timing information.

With large designs, design iterations and last-minute changes are even more inevitable than they are today. Thus, the implementation tools should be re-entrant and tolerant of change, so minor logic changes do not cause major alterations in the physical layout. Xilinx pioneered re-entrant FPGA implementation tools with

the Guide option in PPR.

Accurate **simulation** should be available at any point in the design cycle using a common set of simulation vectors. This means that the post-place-and-route timing results need to be back-annotated into the original netlist created by the synthesis program, as opposed to creating a new netlist based on the structure of the FPGA. Once again, this implies a “synthesis-friendly” FPGA architecture, a synthesis compiler capable of efficient technology mapping to that architecture, and a strong link between the synthesis compiler and the “place and route” tools.

Design errors are reduced and design cycles are compressed when users don’t need to re-invent common functions. Many leading ASIC providers give users large, pre-defined macro functions, sometimes called cores or megacells. As FPGA densities increase, users should expect similar **large macro functions** to be

GUEST EDITORIAL

Continued from the previous page

families. Xilinx is committed to providing the best programmable logic design system in the industry with an integrated solution capable of meeting all of your PLD design requirements.

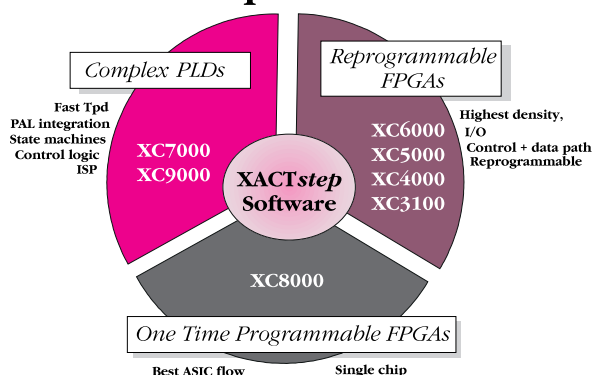
This quarter, we began shipping a major new version of our leading development software. **XACTstep version 6** delivers six new, Windows®-based productivity tools, providing easy-to-use yet powerful design capability. All Xilinx users under warranty are now receiving their updates. Try it — we guarantee you’ll like it.

Service

Providing global, world-class manufacturing, technical support, and sales/distribution support is an essential **foundation** of our product strategy. Many PLD companies severely underestimate the importance of service to the user base. You may have already used one of the several, new, automated technical support facilities that we’ve established this year, such as XDOCS, XFACTS or our home page on the World Wide Web.

In summary, our product strategy is simple but unique: provide leading solutions in all three high-growth segments of the programmable logic industry — complex PLDs, reprogrammable FPGAs, and one-time programmable FPGAs — support them with the industry’s easiest-to-use yet powerful XACTstep integrated software solution, and deliver unquestioned world-class service. Let us know how we’re doing. ♦

Solving the Broad Spectrum of Requirements



available as building blocks for FPGA designs. Similarly, “user-customizable” macro functions, such as those found in the X-BLOX™ library, will ease the designer’s task. Ideally, such macros will be inferred during the synthesis process in order to preserve the portability of the high-level design description, as opposed to having to be specifically instantiated in the HDL code .

Synthesis Friendly Architecture

The features that make an FPGA architecture “synthesis-friendly” include regularity, symmetry and granularity.

In many ways, the choice of logic block **granularity** is a trade-off between utilization and performance. With a smaller block, less of each block is wasted for a given logic function, but performance suffers because more levels of logic are needed for a given large function. Of course, other factors also play a role. For example, configuration information must be supplied to each cell; a very fine-grained cell can increase the number of configuration elements per usable gate. The “ideal” ratio will vary with the size and type of configuration element (SRAM cells versus antifuses, for example). So, for FPGAs, it’s more appropriate to think in terms of a “sea-of-blocks” rather than a “sea-of-NAND-gates”.

For SRAM-based FPGAs, past experience has shown that 4-input lookup-tables are among the most-efficient logic structures; this is not likely to change as FPGAs increase in size. Synthesis algorithms that target lookup-table-based architectures have become more efficient. The lookup-table approach also has the benefit of increasing routing flexibility due to its symmetric nature; that is, signals can be freely swapped among the inputs to the table merely by making the corresponding changes to the contents of the memory cells in that lookup table. Thus, one likely candidate for the “basic logic block” in a

high-capacity, synthesis-friendly, SRAM-based FPGA is the lookup-table/flip-flop pair already common to several Xilinx FPGA families.

Memory/Logic Integration

As stated before, large designs typically need to integrate both memory and logic functions. The **on-chip memory** capability pioneered in the XC4000 family will continue to be a desirable feature for very high-density FPGAs. There are three ways of implementing memory in an FPGA:

- small, distributed blocks that can be used as memory or lookup-table-based logic (as in the XC4000 architecture),
- larger embedded blocks of dedicated memory, or
- configuration memory cells that can optionally be used as contiguous memory in the end application (as in the XC6200 architecture).

Each approach has its advantages and disadvantages, and future FPGAs are likely to offer at least one if not some combination of these options.

Connections

The performance of programmable routing resources is more dependent on the number of programmable switches that must be traversed along a signal path than the length of the metal lines. However, it is wasteful both in terms of resource allocation and overall performance to use a long metal segment for a short point-to-point connection. Thus, the programmable routing resources in a large FPGA should be **segmented and hierarchical** in nature, with a mix of local (short), mid-range, and long metal lines. While the amount of local interconnect surrounding a block does not need to change as the logic array grows, the number of longer interconnect lines should increase. A good analogy can be made to the layout of streets in a growing city; the size and relative density of local streets

*Continued on
the next page*

Continued from the
previous page

“Xilinx is poised to deliver new FPGA solutions that fulfill all the requirements for very high density designs.”

does not need to change dramatically as the city gets larger, but the size and number of freeways and major highways must grow with the city.

General-purpose routing resources need to be complemented by dedicated, global, **low-skew nets** for distributing clocks and similar high fan-out control signals. As a rule of thumb, the clock distribution scheme should provide for a worst-case clock skew that is less than 5 percent of the clock period (for example, less than 0.5 ns for a 100 MHz clock).

Internal three-state buffers that provide the capability of implementing bi-directional and multiplexed busses within the FPGA, as pioneered in the XC3000 and XC4000 architectures, will remain a critical architectural feature for larger FPGAs. The

need for efficient, on-chip bus-sing will become more acute as 32-bit and 64-bit busses become more common in microprocessor-based systems.

As FPGA gate capacity increases, the number of **input/output pads** also needs to increase. However, the ability to implement larger designs in a single device will eliminate the need to partition designs among multiple FPGA devices, which will reduce I/O needs to a large degree. Input and output buffers will need to be compatible with a wide variety of signaling standards (for example, JEDEC 3.3V, 5V TTL, 5V CMOS and GTL) to facilitate interfacing to most available IC technologies. Time-to-market demands may cause the user to freeze the I/O placement early in the design cycle, allowing the FPGA design and PCB layout to proceed in parallel. Thus, abundant routing between the I/O blocks and logic array will be required to ensure flexibility in pin placement during design iterations. Furthermore, as with current Xilinx families, maintaining the same package “footprint” across several family members

will give system designers the option of moving to higher or lower density devices without modifications to the PCB layout.

Power

High gate capacities and clock rates mean **greatly increased power consumption**. The next generation of large FPGAs probably will be based on the 3.3 V power standard. The transition to 3.3 V logic decreases dynamic power dissipation by 56 percent as compared to a 5 V device running the same design at the same speed. Circuit design techniques that minimize the turn-on overlaps of the P- and N-channel transistors in a CMOS device will be employed to further reduce power consumption. Of course, advanced packaging with good thermal conduction characteristics also will be required.

Both the architecture and underlying transistor-level implementation of a high-density FPGA family should be “**scaleable**” to take advantage of future improvements in IC fabrication technology. Again, Xilinx has a strong track record in this regard, as process advancements have led to faster, larger, and less expensive devices in established FPGA families, such as those based on the XC3000 and XC4000 architectures.

On Toward One Million

While there is plenty of life left in the evolution of Complex PLD architectures (as evidenced by continuing advancements in the Xilinx CPLD product lines), for the near term, PAL-like architectures cannot deliver the flexibility, density, or ASIC-like design flow needed for designs exceeding 100,000 gates.

Meeting the requirements of designs over 100,000 gates will require more than just larger versions of today’s devices and tools. Using the highly-successful XC4000 and XC5000 architectures as a stepping stone, Xilinx is poised to deliver new FPGA solutions that fulfill all the requirements for very high density designs.

And we’re already thinking about one million gates... ♦