XCELL

Issue 28
Second Quarter 1998

THE QUARTERLY JOURNAL FOR XILINX PROGRAMMABLE LOGIC USERS

XILINX®
The Programmable Logic Company℠

Inside This Issue:

PRODUCT INFORMATION

SPARTAN™

Spartan The Low Power Leader

The new Spartan Series — a combination of high performance and low power consumption...

See pages 6-8

DEVELOPMENT SYSTEMS

FPGA Express

Synopsys offers high-level design tips...

See page 14

Leonardo

Inferring RAM from VHDL and Verilog using Exemplar Logic's Leonardo...

See page 20

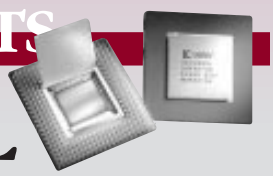
GENERAL FEATURE

Xilinx Student Edition Software Goes Global

Prentice Hall is now selling the Xilinx Student Edition software to students and academic instructors around the world...

See page 4

DESIGN TIPS & HINTS



XC4000XL

Getting the most out of the XC4000XL family...

See pages 25 & 30

EDITORIAL

What Xilinx Values Mean to You

(The Real Competitive Advantage)

by CARLIS COLLINS ◆ Editor ◆ (editor@xilinx.com)

Xilinx is a great place to work, and this is due in no small part to the values that we have chosen for ourselves. Our values clearly define the level of excellence we expect for both our internal operations as well our external relationships with you, our customers. By striving to fulfill our values, we have developed an efficient and challenging, as well as friendly and interesting place to work; therefore you see product innovation and customer service that is second to none.

We enjoy bringing new devices, technologies, and services to you because this is not just our business, it is a big part of our lives as well. We enjoy our work here, and I know that's one of the reasons we continue to be the leader in our industry. Plus, we continuously strive to do better because it makes our personal and professional lives more interesting and rewarding, as it makes your job easier, too.

> ***"Genuine success** does not come from proclaiming our values, but from consistently putting them into daily action."*
>
> **Managing by Values: Ken Blanchard and Michael O'Connor**

Companies need well-defined values, because without them the only measure of success is dollars, and that can be very near-sighted and counterproductive, ultimately leading to everyone's dissatisfaction. So, our values help us focus our energy where it will produce the result we want, which is to be your preferred supplier of FPGAs and CPLDs. And, we believe that if our values remain strong, you will continue to see a significant benefit in both technology and service at every level, and ultimately we both succeed.

The Xilinx values are contained in the acronym CREATIVE, which stands for:

C ustomer Focus - We exist only because our customers are satisfied and want to do business with us… and we never forget it!

R espect - We value all people, treating them with dignity at all times.

E xcellence - We strive for "best in class" in everything we do.

A ccountability - We do what we say we will do and expect the same from others.

T eamwork - We believe that cooperative action produces superior results.

I ntegrity - We are honest with ourselves, each other, our customers, our partners, and our shareholders.

V ery Open Communication - We share information, ask for feedback, acknowledge good work, and encourage diverse ideas.

E njoying Our Work - We work hard, are rewarded for it, and maintain a good sense of perspective, humor, and enthusiasm.

So, do we always succeed in expressing our values in everything we do? No, and I imagine there will always be room for improvement. But, as you can see, it is our intent and our desire to do so. I believe this makes a big difference to you.

We will continue building and strengthening these values because this is the best, and perhaps only way that we can remain the leader in this very competitive industry. Therefore, as always, we welcome your feedback. Do you see our stated values expressed in our interactions with you? Please let us know what you think. You can e-mail your comments to me at editor@xilinx.com. If I get enough responses, I'll publish them, the good and the not-so-good, in our next issue of *XCell*. Now that's "Very Open Communication." ◆

XCell

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124-3450
Phone: 408-559-7778
FAX: 408-879-4780

XILINX®

Xilinx Student Edition Software

by JASON FEINSMITH ◆ University Program Manager ◆ (xup@xilinx.com)

Prentice Hall is now selling the Xilinx Student Edition software to students and academic instructors worldwide. This package (ISBN 0136716296) lists for $87.00 and includes three components which provide a complete design environment for students:

- The Student Edition of the Xilinx Foundation Series Software version 1.3.
- The "Practical Xilinx Designer" Lab Tutorial, by David van den Bout.
- A hardware discount coupon to obtain either the XC95108- or XC4005XL-based development boards (or both) for $129.00 each. A download cable and instructions are included with each board.

The software includes industry-standard schematic entry, HDL entry, state diagram graphical entry, simulation, and design implementation tools for CPLDs and XC4000 Series FPGAs with densities up to 15,000 system gates. Through the Xilinx University Program, this student edition software is also upgradeable to Foundation Express, which includes VHDL and Verilog synthesis based on Synopsys technology.

Now educators can easily incorporate full course and lab materials into their curriculum — and students can access the world's leading programmable logic technology at a very affordable price. This product has achieved tremendous success in the educational community, and according to Prentice Hall the Xilinx Student Edition was their most widely requested new engineering product in January 1998.

The Xilinx Student Edition is licensed for student and educational use only. It is available in academic bookstores, directly through Prentice Hall telesales at 1-800-374-1200, and on the web at www.Amazon.com.

For students and commercial customers alike, the "Practical Xilinx Designer" lab tutorial book (ISBN 0130955027) is also available separately and lists for $38.00.

For more information about the Xilinx Student Edition, please visit WebLINX, our web site: http://www.xilinx.com/programs/phxup1.htm.

For more information on the Xilinx University Program, visit: http://www.xilinx.com/programs/univ.htm. ◆

EDUCATION UPDATE

Courses Updated for New Release

by BILL DETWILER

Our Customer Education Group has recently updated our educational courses to provide timely training that coincides with the latest release of the Alliance Series and Foundation Series version 1.4 software packages.

The 2.5-day FPGA Tools course is focused on how to use the Xilinx tools associated with the 1.4 software release. As an introduction to the Xilinx products and tools, this course is intended for those who have little or no experience with the Xilinx environment.

The new 1-day Foundation Schematic Entry course is also available to expand your knowledge of the Foundation Series software. This course offers both lecture and labs to reinforce the learning experience. The capabilities of the Foundation tools are discussed in detail, including schematic editor, HDL editor, state editor, and a simulator.

Finally, new course development focused on the Foundation Express 1.4 release is underway. This course will provide both lecture and labs. You can find the latest course offerings and dates on the Xilinx web site, WebLINX (www.xilinx.com).

Xilinx distributors worldwide will also be offering these courses at a regional or local level. The exact dates of the distributor courses in your area are listed in the Xilinx Educational Services Brochure. You can also call your distributor or see the educational information on the Xilinx web site. ◆

Designing with the Spartan Series FPGAs

by MARC BAKER

The new Xilinx Spartan Series has already met with great success in high-volume FPGA applications of up to 40,000 system gates. All five 5-V versions are in full production, including a new -4 speed grade that provides the fastest 5-V FPGA in the industry. And, the 3.3-V Spartan-XL devices are on schedule for introduction in the third quarter.

A key benefit of the Spartan Series is the complete software solution. Software support was available even before the devices were introduced, and the same will be true of the 3.3-V versions. You can get started with Spartan designs immediately, knowing that you can take advantage of today's 5-V devices, or tomorrow's even-lower-cost 3.3-V devices.

Design Entry Support

Because the Spartan Series is based on our highly acclaimed XC4000 family architecture, any design entry tool that supports the XC4000E family can be used for the Spartan Series. For schematic capture tools, select the XC4000E library. This library is used automatically when the Spartan family is selected in the Xilinx Foundation Series software. The only XC4000E library components that are not allowed in the Spartan architecture are the edge decoders (DECODE*x*), wired-AND gates (WAND*x*), mode pins, (MD*x*), and asynchronous RAM (RAM16X1, RAM32X1). The Spartan architecture supports both single-port and dual-port synchronous RAM, which have S and D suffixes, respectively. If you accidentally use an unsupported component, the Design Rule Checker will notify you.

FPGA synthesis tool vendors have spent the last few years optimizing their algorithms for our highly successful XC4000 Series architecture. The Spartan Series immediately takes advantage of that development effort, achieving excellent synthesis results from multiple tool vendors. Support is available today from Synopsys, Exemplar, and Synplicity.

Core solutions targeted to the Spartan Series shorten time-to-market by providing a pre-verified solution. The faster speed grade allows Xilinx to offer the 32-bit, 33 MHz LogiCORE PCI interface for the XCS30-4. The upcoming Spartan-XL family will extend that support to 3.3-V solutions. All Xilinx DSP LogiCORE solutions also support the Spartan Series, making FPGAs even more cost-effective for digital signal processing.

Xilinx AllianceCORE partners have been updating and verifying their solutions in the Spartan architecture:

- **CoreEl Microsystems** has introduced Asynchronous Transfer Mode (ATM) products for the XCS30-4 device.
- **Integrated Silicon Systems** has added a Reed-Solomon Decoder for the XCS40-3 device, in addition to their existing Encoder for the XCS10-3.
- **Memec Design Services** offers an industry-standard 8250 core for the XCS05-3.
- **Virtual IP Group** offers five different cores for the XCS40-3, including several UART and 82XX peripheral functions.
- **CAST, Inc**. offers a Viterbi Decoder and a C2910 Microprogram Controller.

Design Implementation Support

The Xilinx Alliance Series and Foundation Series software 1.4 supports implementation of all Spartan 5-V devices. The cost of the complete Spartan software solution is reduced by making all five device densities available in the Base (lower cost) development system.

At the time of the 1.4 software release, the Spartan -4 speed file was not yet available, so it must be downloaded from WebLINX, our web site:

www.xilinx.com
/techdocs/htm_index/sw_speed_files.htm

The -4 speed file on the 1.4 CD is only a placeholder and should not be used to estimate Spartan performance. The complete -3 and -4 speed grade numbers are available in the latest Spartan data sheet on WebLINX:

www.xilinx.com
/partinfo/spartan.pdf

The Spartan Series offers an optimized pinout for the 208-pin Plastic Quad Flat Pack, providing nine more I/O pins and eight more Vcc pins to enhance I/O switching characteristics. The package file must be downloaded from:

www.xilinx.com
/techdocs/htm_index/sw_package.htm

Additional device files available for download include the IBIS models of the I/O structures, useful for board-level simulation:

www.xilinx.com
/techdocs/htm_index/sw_ibis.htm

Boundary Scan Description Language (BSDL) files allow you to take advantage of the built-in IEEE 1149.1 JTAG test capability:

www.xilinx.com
/techdocs/htm_index/sw_bsdl.htm

Designing for the Upcoming Spartan-XL Family

The 3.3-V version of the Spartan architecture will not only bring the Spartan family benefits to 3.3-V applications, but also further reduce the cost for high-volume applications. Prototyping can begin today in anticipation of the silicon introduction in the third quarter of 1998.

If a 5-V power supply is available in the prototype system, use the corresponding 5-V Spartan device as the prototype. The Spartan-XL architecture will be compatible, although timing specifications will differ. You should re-run the design with the Spartan-XL speed files when available.

If the prototype must be a 3.3-V device, the XC4000XL family can be used until the Spartan-XL device is available. Because the XC4000XL architecture is different, you should still implement the design in the target Spartan device first to verify the fit. Once the design fits into the Spartan architecture, lock down the I/O pins and re-implement using the XC4000XL prototype. *See the Spartan/XC4000XL Compatibility chart on page 28.*

Choose an XC4000XL device with an equivalent CLB count to the target Spartan device. The XCS05XL has no equivalent, so use the larger XC4005XL. If the CLB count is sufficient, all other resources will also be sufficient. The design should avoid the PQ208 package, where the XC4000XL prototype would not be able to match the Spartan pinout. The XC4000XL packages also do not offer equivalents to the XCS20/30VQ100. All other Spartan packages have equivalents in the XC4000XL family.

You will need to re-verify the timing once the Spartan-XL speed files are available. To ensure that the Spartan-XL -4 speed will be sufficient, verify the design using the XC4000XL-2 speed grade. The Spartan-XL -4 speed is expected to be at least one speed grade faster than the XC4000XL-2, but using the -2 as a prototype compensates for potential variations in routing and final timing parameters.

The XC4000XL prototype must use one of the two Spartan serial configuration modes, which are selected as all zeroes or all ones on the three mode pins. The Spartan pinout has only one mode pin, equivalent to M0 on the XC4000XL device. To easily adapt the board to the Spartan device, use removable resistors or jumpers on the three XC4000XL mode pins, and simply remove the two jumpers on M1 and M2 when substituting the Spartan-XL device.

Summary

Because we based the Spartan Series architecture on the highly successful XC4000 series, we not only provide excellent devices, but we also provide strong, proven, software support for the family. Designs can be completed today using the production Xilinx development system and a wide variety of third-party tools. You can find the latest information on the Spartan Series and its software support on WebLINX at www.xilinx.com/products/spartan.htm. ◆

Spartan Series Takes the Lead *with Low Power*

by KIM GOLDBLATT ◆ Applications Engineer

> *"The newest improvements in process technology reduce the power required by Spartan devices even further."*

The new Spartan Series, consisting of the 5-V Spartan and 3.3-V Spartan-XL families, offers an exceptional combination of high performance and low power consumption. Benefiting from an architecture based on segmented routing, Xilinx FPGAs have long dominated the field with an indisputable low-power advantage. The newest improvements in process technology reduce the power required by Spartan devices even further. These advancements together with 3.3-V operation will make the Spartan-XL family the low-power leader in the FPGA industry.

The low-power advantage of the Spartan Series comes at a critical time, when today's large, high-performance FPGA designs are using more power than ever before. As the power goes up, the junction temperature increases proportionally. The thermal equation that governs this relationship is:

$$T_J = T_A + P \bullet \Theta_{JA}$$

where

T_J = Junction temperature

Θ_{JA} = Thermal resistance of the package with die

P = Power dissipation

Keeping control over T_J is important because as it increases, device reliability suffers and delays increase. T_J must not exceed the maximum allowable limit for the package in use, and the maximum T_J for the plastic packages used by the Spartan Series is 125°C.

For more information on thermal relationships, see the article entitled "Power, Package and Performance" from *XCell* #22, pp. 28-29 (you can find it on WebLINX, our web site, at www.xilinx.com/xcell/xcell.htm).

Consideration of the thermal equation leads to a fundamental trade-off: For most FPGAs, the density and performance of a design need to be balanced against the cost of keeping T_J under control. FPGAs without segmented routing commonly require heat sinks and fans to keep T_J within limits. Under normal operating conditions, such measures are not necessary for Spartan devices, which, like all other Xilinx FPGAs, are based on a segmented routing architecture that permits dense, high-performance designs virtually unfettered by power concerns.

As an illustration of the power-performance trade-off, the graph shown in **Figure 1** plots dynamic power (watts) vs. clock frequency (MHz) for two devices of comparable density: the Xilinx XCS30 and the Altera 10K20. Both of these devices are available in the PQ208 package, which has a maximum power limit of 2.28 watts. Beyond this limit, T_J goes above 125°C and a heat sink or fan becomes necessary to avoid compromising device reliability. Note that the Spartan device requires no cooling over its entire operating frequency range (as high as 100 MHz for this counter-based design). For the same design, Altera's 10K20 requires additional cooling when running at 40 MHz or higher.

In today's design environment, with ever more critical power budgeting, you need to select the FPGA family that, by virtue of its architecture and process technology, affords the lowest possible power dissipation. By doing so, you will achieve dense, high-performance designs using fewer parts and less board space. This adds value while cutting costs, resulting in a more competitive end-product.

Comparing FPGA Families

Comparing today's FPGA families for low power requires an objective measure that can show the degree to which device architecture and process technology influence power efficiency. This measure should be independent of design particulars (such as design size and

operating frequency), and be able to isolate the effects of different FPGA families on the overall power dissipation of designs. For reasons explained below, a measure known as "K factor" is well suited to the comparison.

Estimating the power of an FPGA design is not a simple matter. A number of factors contribute to power dissipation, including the clock switching frequency, the design density, the interconnect structure, the number of interconnects switching, and the supply voltage. No one factor tells the whole story. At best, they can be used collectively as a guideline for calculating the power budget. Along these lines, an equation can be used to model the effect of the factors upon Icc as follows:

$$Icc = K \cdot f \cdot N \cdot S$$

where

Icc = Active Icc in µA

f = Clock switching frequency in MHz

N = Number of interconnect lines (logic elements) in use

S = Percent of interconnect lines toggling at any given point in time

K = Icc scale factor

Here, Icc is shown as being directly proportional to the clock switching frequency (f) and the number of signals switching at any given time (N•S). K serves as a constant of proportionality.

To determine the value of K, first fill the FPGA with as many 16-bit counters as possible and use a common clock for all counters. N is 16 times the number of counters that fit into the chip. For 16-bit counters, S is approximately 12.5%, because on average two of the 16 flip-flops in each counter toggle each clock cycle. With this design, Icc is measured across the operating frequency range of the device. The average value for Icc/f together with values for N and S can be substituted into the equation to calculate K.

The K factor can be used to indicate how efficiently different FPGA families use power, because it reflects the influence of device architecture and process technology on the supply current drawn. The K factors for major

Table 1.

K factors for Xilinx and Altera FPGAs Compared

	Xilinx		Altera	
Power Supply	Family	K	Family	K
3.3 volt	Spartan-XL	<28[1]	6KA	55
	XC4000XL	28	10KV, 10KA[2]	29-45
5 volt	Spartan	33	6K	88
	XC4000E	40	10K	82-95
	XC4000EX	47		

1Engineering estimate
2Preliminary information

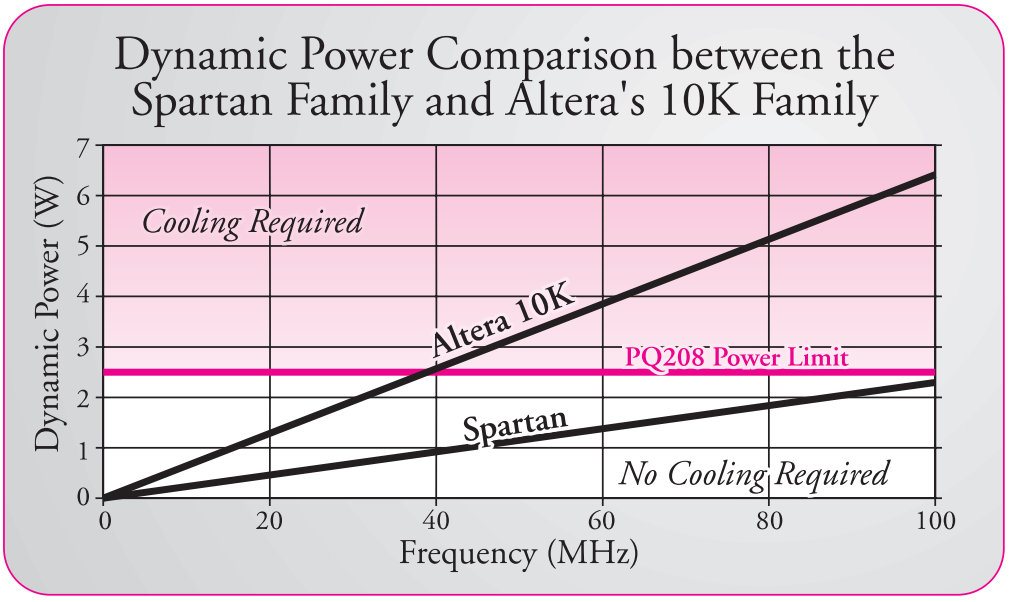


Figure 1.

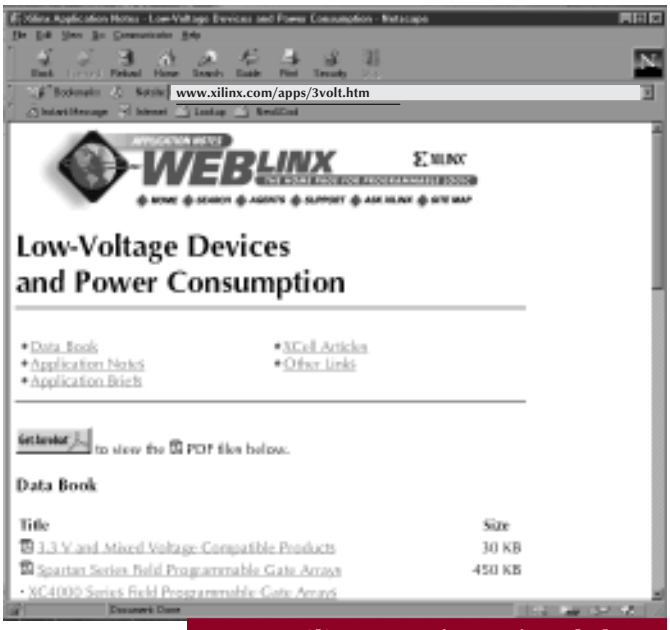
Continued on the next page

Low Power

Continued from previous page

FPGA families from Xilinx and Altera are shown in Table 1. (Altera's data sheets commonly use this model for predicting ICC.)

The K-factors for Xilinx FPGAs are significantly lower than those for similar Altera devices. For example, the Spartan family's K factor (33) is 62% lower than that of Altera's 6K and 10K families (above 82). Thus, for typical designs, the Spartan family uses about 60% less power. As the table shows, for each of the 3.3-V and 5-V operation categories, Xilinx FPGAs exhibit the lowest K values, and thus dissipate less power than their Altera counterparts.



www.xilinx.com/apps/3volt.htm

The Xilinx Low-Power Advantage

The Xilinx low-power leadership can largely be explained by the low capacitance associated with the segmented-routing architecture used on all of our FPGAs. In general, the capacitance of a route is directly proportional to its length. The patented Xilinx architecture uses variable-length routing, which is only as long as needed for a given connection between Configurable Logic Blocks and Input-Output Blocks. As a result, Xilinx FPGAs use significantly shorter routing on average than Altera's FPGAs, which rely heavily on fixed-length long lines. This means Xilinx designs have lower overall capacitance, and thus use less power. The higher levels of power associated with competing FPGAs frequently require not only ceramic packages but also heat sinks and fans.

> *"The most striking observation is that the K-factors for Xilinx FPGAs are significantly lower than those for similar Altera devices."*

Our state-of-the-art process technology makes it possible to reduce power still further. This can be seen in the table, where the Spartan family's K factor of 33 marks an improvement upon the older 4000E family's already low value of 40, resulting in a 17.5% reduction in power.

Finally, 3.3-V operation offers additional power savings over 5-V operation. These savings result not only from the lower supply voltage, but also from the accompanying reduction in ICC. For a 3.3-V supply, ICC and its associated K factor decrease to about 3.3/5 of the value for a 5-V supply. With all else being equal, a 3.3-V Spartan-XL device will use 56% less power than a 5-V Spartan device of comparable density.

Summary

By virtue of segmented routing, an advanced process, and 3.3-V operation, the Spartan Series makes the task of meeting the power budget easier than ever before. As demonstrated by K factor analysis, the Xilinx Spartan Series uses power more efficiently than competing FPGAs, permitting denser, higher-performance designs. For more information on K factor analysis and power in general, consult the "Low-voltage Devices and Power Consumption" section on WebLINX at www.xilinx.com/apps/3volt.htm. Look for application brief XBRF002, entitled "Low-Power Benefits of Spartan and XC4000E/X: An Overview". ◆

Cisco Systems
Using the XC4036XL

by STEPHEN HILLA ◆ Hardware Engineering Manager ◆ Cisco Systems ◆ shilla@cisco.com

Cisco Systems is the worldwide leader in networking for the Internet. We recently developed a new ESCON Channel Port Adapter that allows mainframe computers to connect to our Cisco 7200 Router Family. Xilinx FPGAs helped us create a very successful product.

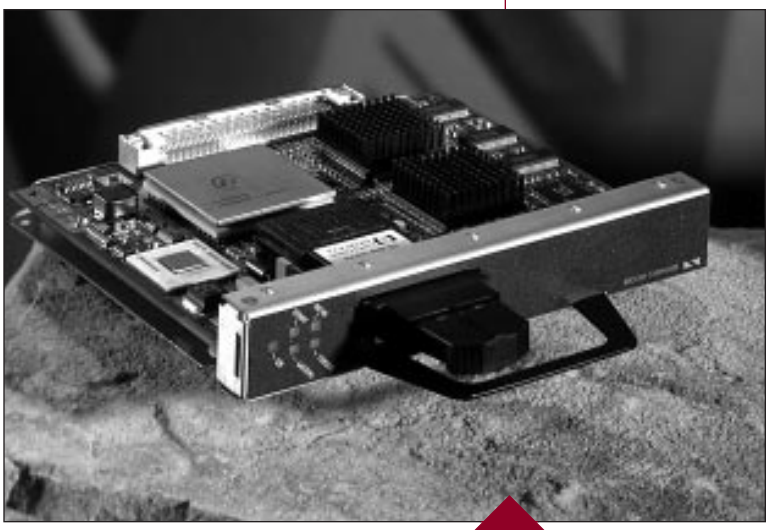
The first prototype of our system did not use Xilinx FPGAs. However, one of the critical elements of our design is a dual-port FIFO, and we could not achieve the speed we needed or keep the power consumption within limits using a non-Xilinx device. So, we had to look for a lower-power, higher-performance alternative.

After a careful evaluation, we chose the Xilinx XC4036XL for five reasons:

1. **3.3-V operation.** We were able to achieve much lower power consumption.
2. **Device performance.** The XC4036XL meets all of our speed requirements.
3. **Dual-port RAM capability**. The Xilinx RAM allowed us to create an efficient, high-performance FIFO that we could not create with the previous FPGAs we tried to use.
4. **Very flexible I/O structure**. The dual IOB flip-flops provide high-speed bi-directional bus capability.
5. **Flexible clock structure**. We used three separate clocks in our design, which were well supported by the XC4036 architecture.

My design team — Joydeep Chowdhury, Marc Edwards, Bill Harris, and Jeff Kidd — needed to pack a 50-MHz system controller, a DRAM controller, and two FIFOs into the device. Although we had never used Xilinx FPGAs before, we were able to complete a working design within about two months.

Using Xilinx SelectRAM, it was easy for us to implement both shallow and wide FIFOs that met our requirements. The XC4036 did a great job for us, and the FIFOs were a "piece of cake." The Xilinx architecture is well suited to creating high performance FIFOs.



We created our design using Verilog with a Synplicity design flow, doing both RTL and gate-level simulation. The local Xilinx FAEs helped us implement the design using the Xilinx Alliance Series software. The design phase went very smoothly. Although we did call the FAEs several times because we were new to Xilinx, we quickly resolved all of the issues. ◆

"The XC4036 did a great job for us, and the FIFOs were a "piece of cake." The Xilinx architecture is well suited to creating high performance FIFOs."

A Marine GPS System and A Desktop Projection System

by RONALD WINDOM ◆ HardWire Product Manager ◆ ronald.windom@xilinx.com

The Xilinx HardWire ASIC family of low-cost, mask-programmed devices makes it very easy for you to replace FPGAs when you need to cost-reduce high-volume applications. The following two success stories show how some of our customers are taking advantage of this capability.

Hand-Held GPS

A major supplier of marine Global Positioning Systems (GPS), uses Xilinx FPGAs for a number of its systems. Their latest hand-held GPS device was in production for about three months (approximately 25K units), using the Xilinx XC3030L-VQ64 FPGA. Because the product was targeted for high-volume production (estimated to be 150K units per year), they also evaluated cost reduction methods.

The designers chose the Xilinx HardWire XC3330L-VQ64 device because of its ability to meet their required 2.7-V operation, and because of its ASIC-like cost structure. In addition, the HardWire device is fully compatible with its FPGA counterpart, so they were able to simply replace the FPGA with the HardWire ASIC, with no change in their PC board.

The company's engineer said this FPGA-to-HardWire ASIC conversion gave them a 30% cost reduction.

Texas Instruments Desktop Projection System

Texas Instruments builds a variety of projection systems, for which Xilinx has supplied a number of FPGAs. Their latest project is the second generation of a desktop projection system, sold to a number of different suppliers who re-sell it under their own brand name (estimated to be 100K units per year). This new system was designed using the Xilinx XC4036XL-PQ240 FPGA, which allowed TI engineers to quickly prototype and debug their design. The first generation of this system sold approximately 25K units, using Xilinx FPGAs. However, cost reduction was a primary consideration for this high volume application.

The TI team chose the Xilinx XC4436XL-PQ240 Hardwire ASIC for their cost-reduction program because of its ASIC-like cost structure, fast conversion time, and quick turn-around. They also chose Xilinx because of our experience in converting full-featured designs using CE, JTAG, and RAM.

TI engineer Philip Bucholz said their FPGA-to-HardWire ASIC conversion gave them a 50% cost reduction.

Xilinx supports a full line of FPGAs with footprint-compatible HardWire ASICs that make it very easy for you to cost-reduce high-volume systems. For more information on our HardWire ASIC capability, visit WebLINX (www.xilinx.com). ◆

The XC4013E
A Military "Bottom Line" Solution

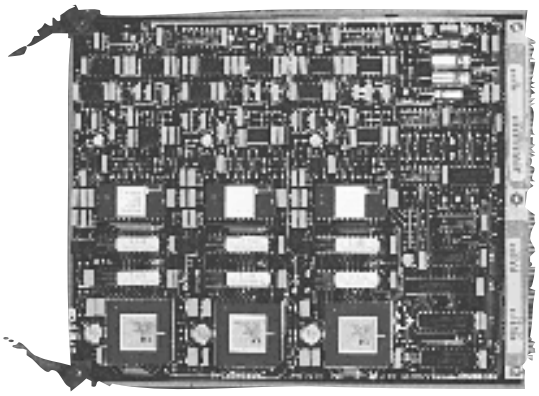
by Ed McCauley ◆ Bottom Line Technologies, Inc. ◆ 908-996-0817 ◆ edmccauley@bltinc.com

Teamwork, ingenuity, and the unique qualities of Xilinx parts combined to create a surprisingly efficient solution in a recent military project. The military customer, the Xilinx FAE, Hamilton Hallmark, and the design firm Bottom Line Technologies, all worked together in upgrading an older product to current technology, and providing a growth path for the future.

The customer had developed a numerically controlled digital sweep oscillator (NCDSO) in the late 1980s, using custom military gate arrays. Using the technology of the time, they designed with 1.5 micron ASICs that were big and expensive compared to today's devices and prices. Each circuit board contained three custom gate arrays that each implemented one NCDSO, and six ROMs that converted phase-angle to sinusoidal waveforms. The system worked, but was no longer practical because of cost and component availability. This customer had a dinosaur system on its hands.

Working with the FAEs from Xilinx and Hamilton Hallmark, this military customer created a solution that replaced the outdated custom parts with Xilinx FPGAs. The change to FPGAs guaranteed that the solution would work for today and tomorrow by taking advantage of the ongoing process and price shrinkage that Xilinx has traditionally passed on to its customers. Not only would they have faster, cheaper parts, they would also never again be stuck with outdated, expensive technology.

The team's first solution replaced the three custom gate arrays with three Xilinx parts and continued using the six associated ROMs. A fourth Xilinx device was used to control the others, serving as a microprocessor interface. The design team was happy with the solution, but they did not have the expertise to complete the conversion; additional engineering resources were needed to design the new Xilinx parts.



The FAEs recommended Bottom Line Technologies (BLT), a New Jersey design firm founded by one of the original Xilinx FAEs. Serving clients from Boston to North Carolina, BLT specializes exclusively in Xilinx design, development, and training. The FAEs felt that the expertise and experience of BLT's engineers would offer the customer a fast and cost effective answer to their dilemma.

BLT proposed an innovative solution. Rather than using the four Xilinx parts and associated ROMs. BLT suggested that they use a unique feature of Xilinx FPGAs to reduce the number of devices needed in the product. By taking advantage of the Dual Port Synchronous RAM, available in Xilinx FPGAs, they could replace the three custom chips with one multi-tasked Xilinx device.

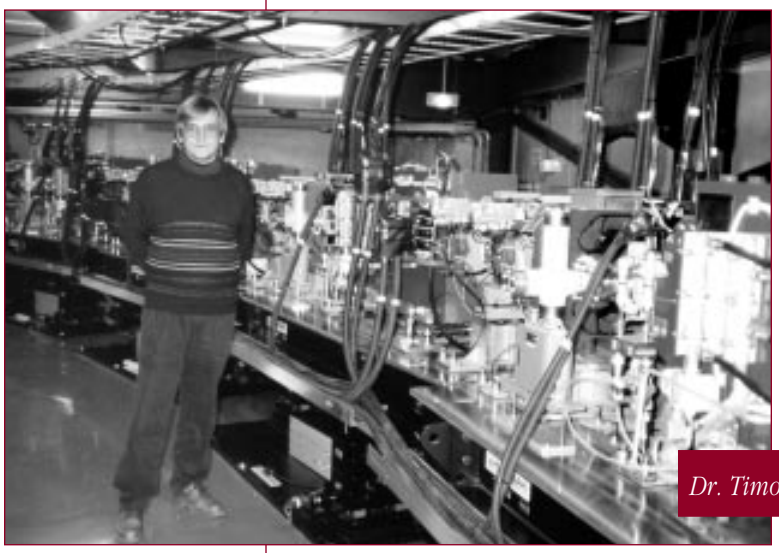
BLT's final solution incorporated all three NCDSOs, plus the microprocessor interface, into a single Xilinx XC4013E FPGA. The pipeline registers for each of the NCDSOs were implemented in the dual-port synchronous RAM of the Xilinx chip. Data written by the microprocessor was also stored in the dual-port SRAM. By changing the addresses to these RAMs periodically, the data and pipeline registers for each of the NCDSOs became accessible during the corresponding time interval.

Continued on next page

XC3195A FPGAs
Getting a "Handel" on *High Energy Physics*

At the KEK High Energy Physics Accelerator Research Organization in Tsukuba, Japan, Dr. Timo Korhonen has been designing and building a large-scale control system for the Accelerator Test Facility. Instead of using microprocessors for the exacting real-time processing tasks, he took a novel approach. Dr. Korhonen is using 40 "accelerator cards" from Embedded Solutions, Ltd. (each containing two Xilinx XC3195A FPGAs), and the Handel programming language to generate the hardware configurations for the FPGAs.

The purpose of the project is to build an electron accelerator to evaluate some of the design considerations for a proposed large linear collider. Due to the need for very low beam emittance, the beam path in the 138-metre circumference damping ring must be accurate to a few tens of microns. Thus it is essential to remove even the minuscule deformations of the hall floor that occur due to traffic, construction work, and natural causes such as temperature variations.



Dr. Timo Korhonen in the accelerator test facility

Continued on next page

The XC4013E

Continued from the previous page

In this manner, all three sweep oscillators (time multiplexed) fit in the single XC4013E. An added benefit of this multi-tasking was that the six external ROMs of the original design were replaced by two. From old design to new, nine custom parts per board were reduced to just three. The icing on the cake was BLT's use of the phase angle ROMs to store the XC4013E's configuration data. As a result, the customer also saved the cost of a separate configuration ROM.

The customer was thrilled by the new Xilinx design. In the conversion, they had reduced the number of parts needed in the product, fully utilized the features of the Xilinx technology, and were guaranteed future price competitiveness provided by Xilinx process migration. The customer also reaped the benefits of the combined knowledge and expertise of the Xilinx and Hamilton Hallmark FAEs working in conjunction with the BLT design team. The customer significantly reduced overall costs and eliminated the threat of another "dinosaur" down the road.

"This was a slam dunk for Xilinx. We were really pleased."

Bottom Line Technologies Inc. is based in Milford, NJ. They can be contacted at 500-447-FPGA, 908-996-0817, or on the Web at http://www.bottomline.com. ◆

Continued from previous page

Here is what Dr. Korhonen had to say about his project:

"My part in the project was to design and build a distributed control system to implement dynamic stabilization of the beam path via 36 movable alignment tables that carry the heavy beam steering electromagnets. Positioning of the tables is accomplished by stepper motors with an accuracy of better than 2.5 microns. Laser position sensing is used between neighboring tables to determine the location in each of the five degrees of freedom. This alignment system has never been used before in an operating accelerator.

"I analyzed alternative hardware solutions including microprocessors, microcontrollers, and DSPs; although they were easy to program, Xilinx FPGAs had the edge in performance and could easily interface to my control equipment. I also had to choose the programming tool, and decided to use a radical new programming language, Handel. I saw that this would make the design process much simpler. Besides, it was a good chance for me to get the experience of performing computations in reconfigurable hardware.

"Handel is a programming language designed at Oxford University and now available as a commercial hardware compiler product from Embedded Solutions Ltd. (ESL). It has been designed so that C programmers can rapidly start to build hardware implementations of their programs without having to learn anything about low-level hardware.

"The control system uses up to 40 of ESL's "accelerator" FPGA modules, each containing two XC3195A FPGAs. I used the design methodology that I use normally when writing software. I summarized the data that had to be handled by the system and then refined it top-down to arrive at a software solution. There were a few (really few) hardware aspects that had to be taken into account, such as the specifications for the circuit interfaces which included how to read the ADC, how to handle the RS232 port, and so on.

"This design phase took just a couple of weeks, and the implementation was done in a couple of days using a PC-based system and XACTs*tep* 5.2/6.0. I also have the new Foundation Series software but haven't yet upgraded to it. After working a few days with Handel, the feeling was just like writing software and at times made me forget that what I was doing was actually hardware development. Recently, I implemented a digital module with more traditional methods and the difference in productivity was really overwhelming.

"The next big challenge will be in applications that need fast real-time I/O; for example, reading the data from beam monitors, processing it, and feeding the results directly back to a control system. This is a task where even the fastest available CPUs find difficulty because the data has to go through the I/O interface. In an FPGA-based system, most of the overhead can be avoided, and thus the response time is shortened. For these tasks, I'll probably use the Xilinx XC4000XV devices."

"With the ever-increasing capacity and speed of Xilinx FPGAs, I expect them to be used more and more in such applications, and Handel will allow applications to be developed more rapidly and with greater maintainability."

Dr. Korhonen

For more information:

Dr. Timo Korhonen:
timo.korhonen@kek.jp

Embedded Solutions Ltd.
(Handel and Accelerator Cards):
sales@embedded-solutions.ltd.uk
http://www.embedded-solutions.ltd.uk/

Oxford Hardware Compilation Group:
http://www.comlab.ox.ac.uk
/oucl/hwcomp.html ◆

High-Level Design Tips for Synopsys FPGA Express

by KIRK A. OWYANG ◆ Manager, Corporate Applications Engineering ◆ Synopsys, Inc.

The advantages of using high-level design languages (HDLs) for FPGA design have become readily apparent as device densities skyrocket to more than 500K gates. The benefits of using VHDL or Verilog HDL for design entry and logic synthesis are especially great if you are using the XC4000E/X, XC5200, Spartan, or Virtex FPGA families. When you design with HDLs, the key to obtaining the best performance using Synopsys FPGA Express technology is to adopt applicable high-level design practices and techniques, some of which are described in this article.

Specifying Design Constraints and Requirements

The most effective way to increase the performance of your design is to thoroughly specify your detailed design constraints and requirements in FPGA Express. Performance requirements include system clock speed, multi-cycle and other sequential path timing delays, and input/output timing. You can also synthesize for best area or speed performance, preserve or flatten design hierarchy, and control operator (resource) sharing. In addition, entering your design constraints improves more than the synthesis results; place and route results also improve because constraints are passed as TIMESPECs in the XNF (or NCF) files.

You can also enter other design specifications. In the FPGA Express interface, you can specify and control port/pad locations, global buffer (BUFG) allocation, slew rate, and I/O register insertion.

All of these constraints and requirements enable FPGA Express and the Xilinx software to extract the best performance from the devices.

Describing Finite State Machines

There are specific ways to describe finite state machines (FSMs) in VHDL and Verilog that result in one-hot or binary encoding. You can create any finite state machine using the following FSM templates, which give optimal synthesis results with FPGA Express synthesis technology.

VHDL Template

In VHDL, you can enumerate states symbolically or you can assign values to states with the ENUM_ENCODING attribute.

FPGA Express supports automatic FSM encoding for enumerated types in VHDL. To take advantage of automatic encoding:

1. Use the VHDL template without the ENUM_ENCODING attribute.
2. From ***Synthesis*** > ***Options*** > ***Project***, choose One Hot or Binary encoding. The default is One Hot.

Use the ENUM_ENCODING attribute when you want to control the state encoding.

The following example of a simple counter shows the VHDL template using an enumerated type for states (shown in bold text). Note that the WHEN OTHERS statement is not needed for this template, though it can be used if required.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity shift_enum is
port (CLK,RST : bit;
 I : std_logic_vector
 (2 downto 0);
 O : out std_logic_vector
 (2 downto 0));
end shift_enum;

architecture beh of shift_enum is

type state_type is (S0, S1, S2);
— Do not use the following 2 lines
for automatic FSM extraction:
attribute ENUM_ENCODING: STRING;
attribute ENUM_ENCODING of
state_type: type is “001 010 100”;

signal machine : state_type;

begin
 process (CLK,RST)
 begin
  if RST= ‘1’ then
   machine <= S0;
  elsif CLK= ‘1’ and CLK’event
then
   case machine is
   when S0 => machine <= S1;
   when S1 => machine <= S2;
   when S2 => machine <= S0;
   end case;
  end if;
 end process;

with machine select
 o <= “001” when S0,
 “010” when S1,
 “100” when S2;
end;
```

Verilog Template

Verilog does not support the enumerated type. However, you can use the Verilog template to design a finite state machine and fully control the state encoding.

In Verilog, state values are defined with the parameter statement. The conventions for this Verilog template are:

- Use the parameter statement to define state values.
- Use a CASE statement and the Synopsys directive //synopsys parallel_case full_case to describe the state machine.

The following example of the same simple counter described in the VHDL template shows the Verilog template using the parameter statement for state values.

```
module shift (clk, rst, in, out);
input clk, rst;
input [2:0] in;
output [2:0] out;
parameter [2:0]
  S0 = 3’d1,
  S1 = 3’d2,
  S2 = 3’d4;
reg [2:0]
  state, next_state ;

always @ (in or state)
begin
  case (state) // synopsys
  parallel_case full_case
  S0: next_state = S1;
  S1: next_state = S2;
  S2: next_state = S0;
  endcase
end

always @ (posedge clk or posedge
rst)
begin
  if (rst) state <= S0;
  else state <= next_state;
end

assign out = state;
endmodule
```

Conclusion

There is no question that the future of the FPGA industry depends upon very large and complex FPGAs such as those in the new Xilinx Virtex family. To reap the maximum benefit from these new devices, FPGA design engineers must adopt design methodologies (such as the use of HDLs) that closely mimic the traditional ASIC design flow. A couple of the key components of a successful HDL-based design flow for Xilinx customers, described above, are detailed design constraint capture and adherence to specific templates for FSM descriptions. Following simple design practices like these ensure the best performance for your Xilinx design.

For more information, see the online help in the FPGA Express software. ◆

“There is no question that the future of the FPGA industry depends upon very large and complex FPGAs such as those in the new Xilinx Virtex family.”

Reduce Compile Times

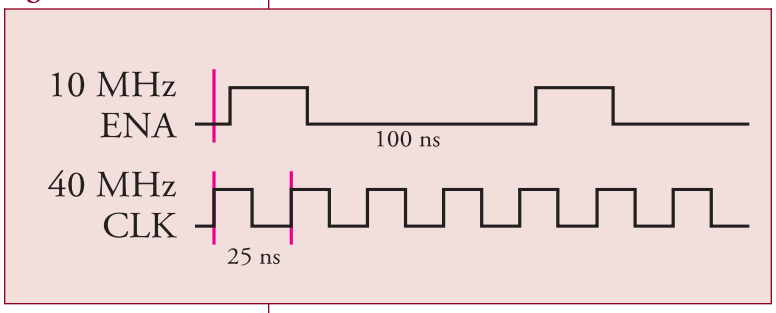
Using Timing Constraints in Foundation *Express*

by LAUREN WENZL ◆ Xilinx Boulder

Using multi-cycle timing constraints for specified paths can decrease place and route run times. Because the place and route tools must work harder to meet aggressive timing requirements, place and route run times can be optimized, if you apply tight timing constraints only to critical logic and apply relaxed timing constraints to your less-critical logic.

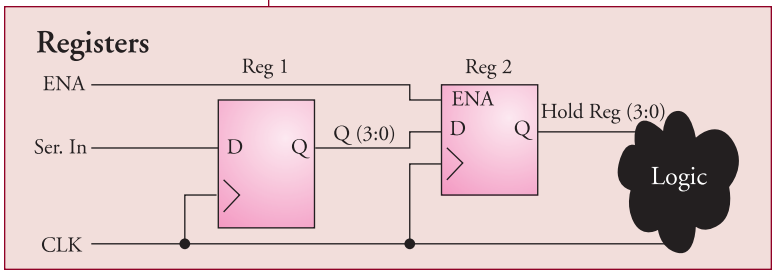
Foundation *Express* provides the ability to create multi-cycle timing constraints quickly and easily by creating timing groups and subsequent subpath timing groups. An example of this is illustrated by the following design, which eliminates clock skew by using derivative clock/enable signals from the system clock for non-critical logic. The technique shown in this example is a popular way to eliminate clock skew problems. Clock skew is avoided by using a single oscillator as apposed to using multiple oscillators. A secondary clock enable is created by dividing down the system clock, as opposed to introducing additional clock signals, which could result in skew between the multiple clocks.

Figure 1.



In this example, assume that an FPGA contains some high-speed interface logic that must run at 40 MHz and some low-speed interface and core logic that must run at 10 MHz. The FPGA has a 40-MHz system clock and uses it to generate a 10-MHz enable signal for internal distribution. The following figure shows how the 10-MHz enable might align with the system clock when the rising edge of the 40-MHz system clock is the active edge. The 40-MHz clock is distributed to the clock input of each FPGA flip-flop, while the enable signal is distributed to each FPGA flip-flop clock enable input. In this case, the primary clock period is 25 ns, but the enabled signal and the subsequent logic will be driven by the slower 10MHz clocked data.

Figure 2.
Simple Shift register



A simple shift register circuit shown in the following logic diagram illustrates how the multi-cycle timing constraint is assigned in Foundation Express.

Register **reg1** is a 4-bit serial-input, parallel-output register. Register **reg2** is a holding register that is loaded with the clock enable **ena**. The paths from the output of **reg1** to the output of **reg2** (net **q**) are multi-cycle paths because the output of **reg2** is based on the combination of the **CLK** and **ENA** signals. The register-to-register timing constraint is 25 ns (1/40 MHz), but the multi-cycle timing constraint which applies to the subsequent logic, is 100 ns (4 x 25 ns).

Once the system clock has been set, **reg1**, **reg2**, and the subsequent logic will automatically be grouped together and displayed in the constraint entry window as having a register-to-register or clock-to-clock timing of 25 ns. This can also be referred to as the rising clock_of_clock to rising clock_of_clock of 25 ns. To create a subpath group of the register-to-register path, click the right mouse button on the register-to-register path groups and select **New Subpath**, in the Paths Constraint table.

The **Create/Edit Timing Subpath** window appears. Use this window to construct

your own path group by selecting specific start-points and end-points. The newly created path group is called a subpath because it is a subset of another path group, in this case, the register-to-register paths in the design.

In this example, the outputs of **reg1** (bus **q**) are the start-points, and the inputs of **reg2** are the end-point for the subpath. A delay of 100 ns is assigned to the subpath. After you create a subpath 'slow_path' and apply the multi-cycle timing constraint, the subpath appears in the Paths constraint table as *slow path-from to slow path-to.*

Notice that an enabled flip-flop can be included in two different path groups: those that include clock-to-clock paths and those that include clock-to-enabled-clock paths. This implies that there are two TIMESPECs with overlapping constraints generated by Foundation *Express.* The constraint for register-to-register timing, which is 25 ns in this case, conflicts with the constraint for the slow path subpath timing, which is 100 ns. The Xilinx implementation software assigns different priorities to these two constraints, placing a higher priority on the more specific one. Because the subpath constraint is more specific than the clock-to-clock constraint, it takes precedence and the corresponding paths can be optimized for the slower speed.

Summary

By prudently specifying aggressive timing constraints you can avoid over-constraining your design, which could result in excessive runtimes in the implementation tools. In this example only the data out of register 1 requires a clocking constraint of 25 ns. The multi-cycle timing of the subpath required a much more relaxed constraint of 100 ns. Because the bulk of the design was constrained to 100 ns rather than 25 ns, the implementation of this design is achieved with a reduced run time. ◆

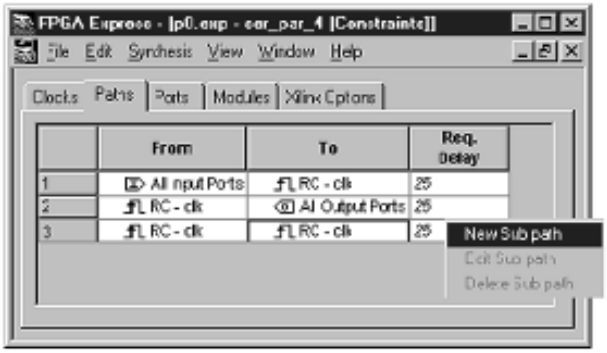


Figure 1:
In the Paths constraint table, to create a subpath group of the register-to-register paths, click the right mouse button on the register-to-register path groups and select New Subpath.

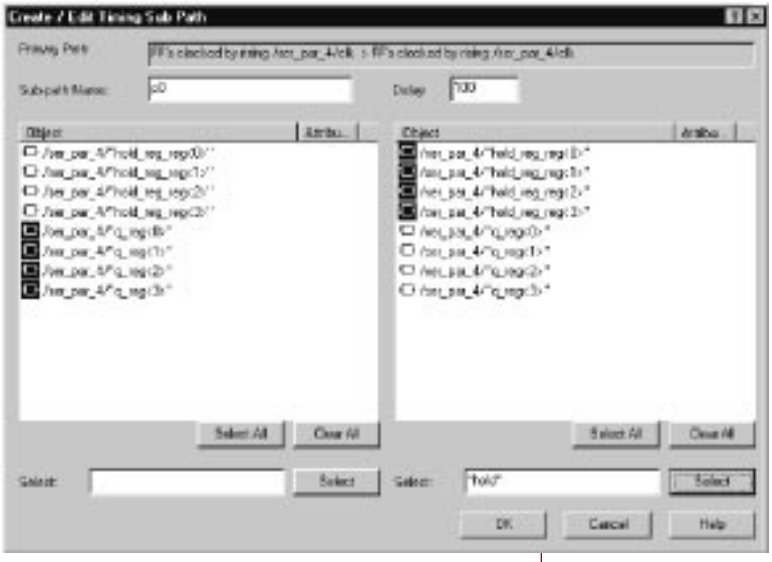


Figure 2:
The Create / Edit Timing Subpath window appears. Use this window to construct your own path group by selecting specific startpoints and endpoints. The newly created path group is called a subpath because it is a subset of another path group, in this case, the register-to-register paths in the design.

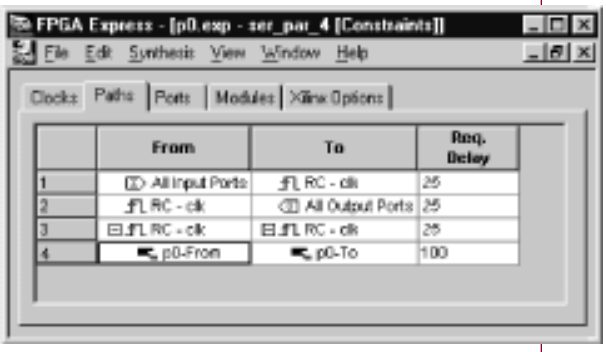


Figure 3:
In this example, the outputs of reg1 are the startpoints and the inputs of reg2 are the endpoint for the subpath. A delay of 100 ns is assigned to the subpath. After you create a subpath and apply the multicycle timing constraint, the subpath appears in the Paths constraint table.

HDL Analyst

A Unique New Tool for Visualizing Synthesis Results

by MARGARET ALBRECHT ◆ Technical Marketing Engineer ◆ Synplicity

Synplicity

HDL Analyst is an optional graphical productivity tool for the Synplify synthesis environment that helps you visualize the results of your FPGA and CPLD designs, improve the quality of your source code, and improve the device performance and area results.

HDL Analyst provides two key benefits:

- It gives you an immediate understanding of the results produced by your HDL code, with links back to the source.
- It is a quick and easy way to learn HDL.

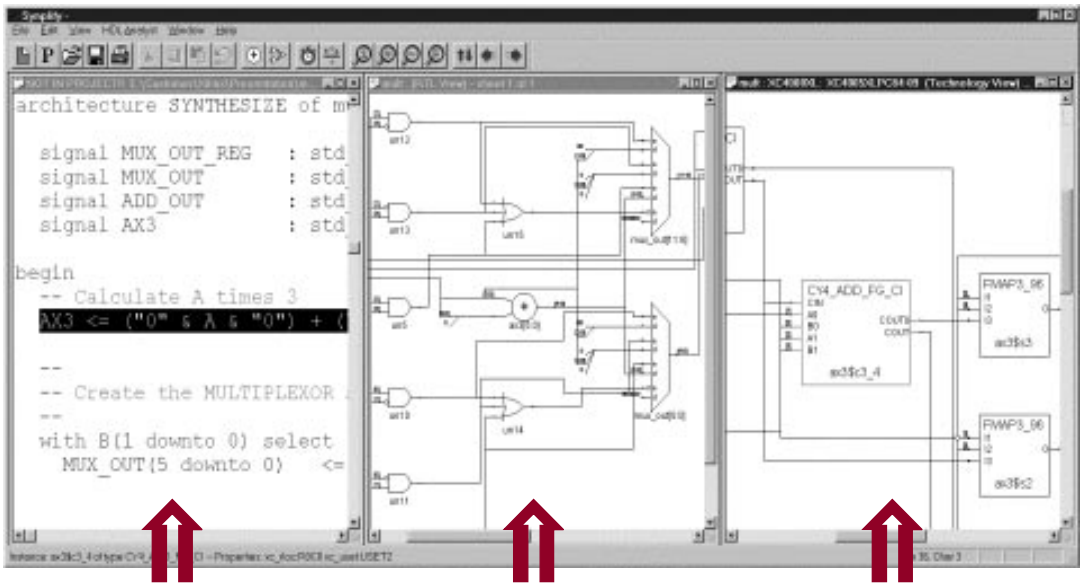
HDL Analyst has the unique ability to generate an RTL block diagram of a design based on your Verilog or VHDL source code. In addition to this high-level graphical representation of your design, HDL Analyst provides a technology-primitive representation of your design (LUTs, CLBs, and so on).

HDL Analyst also provides an environment where you can cross-probe between the RTL block diagram, the technology-primitive schematics, and the HDL source code. For example, highlighting a line of text immediately highlights the corresponding logic in the block diagram and in the detailed schematic views. Conversely, double clicking on logic in one of the graphical views highlights the source code from which the logic was created.

The cross-probing capability of HDL Analyst enables you to visualize where coding changes or the addition of timing constraints might increase performance or reduce the required logic resources. Furthermore, HDL Analyst highlights and isolates critical paths within your design, so problem areas may be identified and corrected by adding timing constraints and re-synthesizing.

Success in designing high-density FPGAs is increasingly dependent upon the availability of high-quality synthesis software. Over the years, Synplicity and Xilinx have established an excellent Alliance relationship which has advanced the state of the art in FPGA design. The primary goal of our relationship is to provide you with high-quality synthesis software and productivity-enhancing tools such as HDL Analyst. ◆

Cross-Probing Simplifies Understanding



Language Sensitive Editor (Describes the functionality)

RTL View (Graphical analysis)

Technology View (Mapping)

Concurrent ISP Operations in JTAGProgrammer 1.4

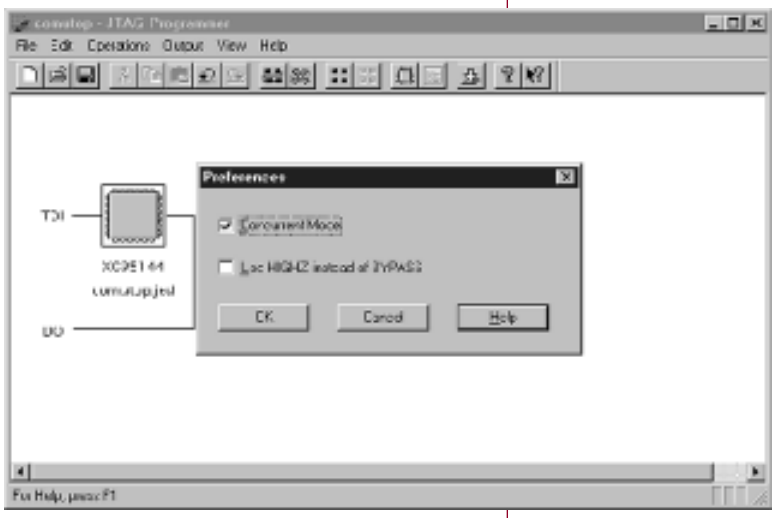
by NEIL JACOBSON ◆ CPLD JTAG Software Development Manager

Our latest version of JTAGProgrammer (included in our Alliance Series and Foundation Series 1.4 software) includes a special option to speed ISP operations in XC9500 multi-part boundary-scan chains. This option is called concurrent ISP mode and is enabled under user control.

When performing ISP operations using the IEEE Std 1149.1 serial protocol (JTAG), a lot of time is spent waiting for the erase and program operations to complete. During this wait time, no other boundary-scan operations are allowed, and no stimulus is driven through the JTAG test access port (TAP). In sequential ISP mode, each device is programmed (or erased) one at a time. The associated programming (or erase) wait time is therefore additive, resulting in a relatively long programming (or erase) time. If a single XC95144 takes eight seconds to program, three identical parts programmed sequentially would take 24 seconds.

In concurrent ISP mode, the erase and programming data are set up for all devices together. Then the associated wait time elapses for all devices in unison. This means that the total execution time for an ISP operation on a boundary-scan chain of several devices will be close to that of the largest (longest to program) device on the boundary-scan chain (plus any overhead associated with applying the programming or erasing stimulus to the TAP). So, if a single XC95144 takes 8 seconds to program, three XC95144's programmed concurrently would take about 12 seconds.

To enable this option in JTAGProgrammer, you should select "Preferences..." from the "File" pull down menu. In the Preferences dialog select "Concurrent Mode" and press the OK button. *(See screen capture.)*



Concurrent mode will then be enabled automatically, when you select more than two devices in an arbitrarily sized boundary-scan chain for erase, program, or verify operations.

Use concurrent mode when you have multiple XC9500 devices in a boundary-scan chain, to optimize the ISP operation execution time. ◆

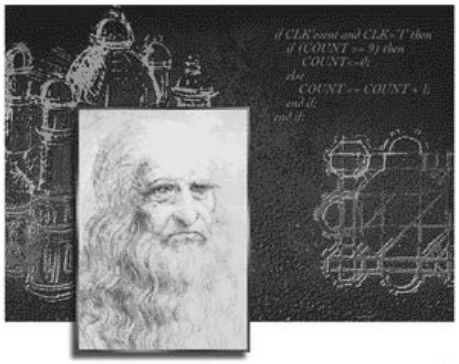
> *"Use concurrent mode when you have multiple XC9500 devices in a boundary-scan chain, to optimize the ISP operation execution time."*

RAM Inference Using Exemplar Logic's Leonardo

By TOM HILL ◆ Manager of Vendor Relations ◆ Exemplar Logic

When using synthesis, component instantiation has been the preferred method for inserting RAM into a design. Although instantiation works, it is cumbersome and adds an unnecessary level of complexity to the HDL coding and simulation steps. Exemplar Logic's Leonardo supports RAM inference from your RTL code. This article describes the technical aspects of inferring RAM from VHDL and Verilog.

Leonardo recognizes RAM elements from a basic two-dimensional array memory model, coded in VHDL or Verilog. When a RAM is detected, Leonardo inserts a generic RAM cell into the netlist along with EDIF properties that tell the Xilinx Alliance Series software how to implement the RAM. Currently, the following two Xilinx RAM types are supported:



LEONARDO™

Advancing the World of Logic Design

RAM_DQ: *Synchronous or asynchronous single-port RAM. Leonardo determines if the RAM is synchronous or asynchronous based on the existence of clock lines.*

RAM_IO: *Synchronous or asynchronous single-port RAM with bi-directional data line.*

Asynchronous, Single-port RAM Example

VHDL

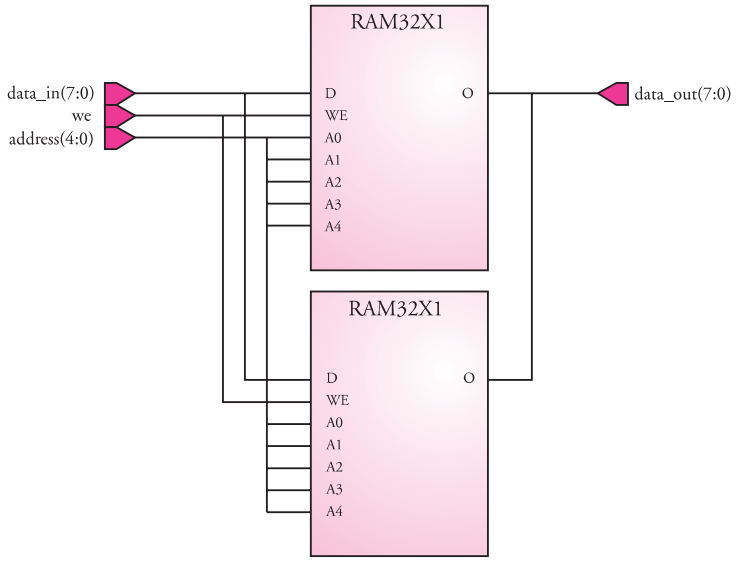
```
     architecture rtl of ram is
type mem_type is array
 (2**address_width downto 0)
UNSIGNED(data_width - 1 downto 0) ;
signal mem : mem_type ;
  begin
    I0 : process
 (we,address,mem,data_in)
     begin
       if (we = '1') then
         mem(conv_integer(address))
 <= data_in ;
       end if ;
       data_out <=
 mem(conv_integer(address)) ;
    end process ;
end RTL ;
```

Verilog

```
      module ram (data_in, address,
we, data_out);
 parameter data_width=8,
address_width=8, mem_elements=256;

 input  [data_width-1:0]    data_in;
 input  [address_width-1:0] address;
 input                      we;
 output [data_width-1:0]
data_out;

 reg    [data_width-1:0]    mem
[mem_elements:0];

 always @(we or address or data_in)
begin
   if (we) mem[address] = data_in;
 end

 assign data_out = mem[address];
endmodule
```

There is no limit to the size of the RAM that can be inferred; Leonardo will build a RAM array out of available elements for a particular technology. In the **Figure 1** example, two 32x4 RAMs are required.

Figure 1. In this example, two 32x4 RAMs are required.



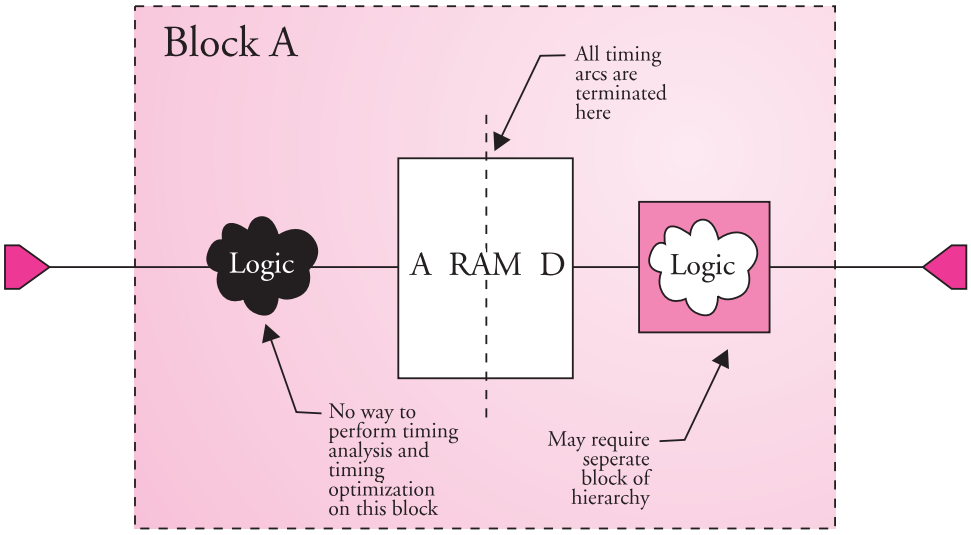
Timing Analysis and Optimization of RAM Control Logic

Because Xilinx uses a black-box approach to RAM instantiation, no timing information is available for the inferred RAM cell. This prevents Leonardo from performing timing analysis and timing optimization on logic directly connected to the RAM, which often includes the address-to-data-out path of the RAM. Therefore, you should perform a timing analysis in the Alliance Series environment to detect any timing problems through paths that contain RAM. *(See Figure2.)*

If further optimization is required on a path, use the following procedure:

1. Re-optimize the circuit with the *"-delay"* and the "*-effort standard"* options.
2. Place critical logic into a separate hierarchical block. This could be done with the *group* command or may require a code re-write. Leonardo supports the VHDL block statements for hierarchy which may provide a simple method of achieving this.
3. Set timing constraints and perform optimizations.

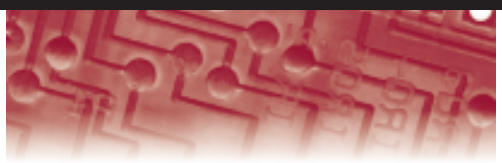
Figure 2. Leonardo's black-box approach.



Conclusion

If you're using RAM, Exemplar Logic's Leonardo can decrease your development time and make your life a little easier. For more information on Leonardo, contact Exemplar at 1-800-632-3742, or visit www.exemplar.com. ◆

Printed Circuit Board Design Considerations



by PETER ALFKE ◆ (peter@xilinx.com)

Steady advances in IC technology have fueled 30 years of rapid progress in digital system speed and complexity. In the past, system speed was determined by gate and register performance, and you could easily take advantage of ever faster, bigger, and cheaper integrated circuits.

The printed circuit board (PCB) was just a means to hold ICs in place; PCB layout was an exercise in topology and economics. Analog issues such as crosstalk, phase and amplitude distortion, reflections, ground bounce and so on could thus safely be ignored, or treated as minor irritants, because synchronous digital logic is forgiving in amplitude and timing variations.

Times have changed. At today's circuit speed, the PCB and its analog characteristics play a strong, if not dominating, role in determining digital system performance.

CMOS ICs are no longer the slow and forgiving circuits of the past. They are now as fast as (if not faster than) the fastest TTL circuits; outputs ramp between 0 and 5 volts in 1 ns, clock rates approach 150 MHz, and ICs have up to 500 signal connections to accommodate multiple 32-bit wide busses. The trend is moving toward much higher speed and far more I/Os.

However, signals still travel along PCB traces at only half the speed of light, and sharp signal edges get reflected at every trace discontinuity. You must now not only control the source-to-destination path, but you must also pay attention to the complete circuit loop and its inductance, from the positive power supply terminal back to the negative terminal, then through the decoupling capacitors back to the positive supply terminal.

This means that the PCB plays an important role in controlling the integrity of interconnect signals. Trace width and trace spacing as low as 4 mils (0.1 mm), controlled line impedance, and multi-layer boards with clean ground and Vcc planes, are all required to minimize reflections, ground bounce, and crosstalk. Maintaining clock integrity is a big problem.

Year	*Max.System Clock*	*PC-Board Complexity*	*PC-Board Min.Trace Width*	*IC Design Rules*
1965	1 MHz	1-2 layers	100 mil = 2.5 mm	10µ
1980	10 MHz	2-4 layers	20 mil = 0.5 mm	3µ
1995	100 MHz	4-8 layers	4 mil = 0.1 mm	0.5µ
2010 ?	1000 MHz	8-16 layers	1 mil = 25 µ	0.1µ

Time Domain vs Frequency Domain

Digital designers usually express delays and rise times in the time domain, while analog designers often use the frequency domain to describe circuit and component performance. The frequency domain is more meaningful for analyzing many analog phenomena. For rough estimates, we can easily convert rise and fall times into a frequency spectrum as follows:

There is a knee frequency

$F = 0.5 \bullet (1/ (T_{rise} \text{ or } T_{fall}))$

For analyzing circuit behavior, it is sufficient to evaluate the frequency response up to the knee frequency, but there is no need to go higher.

For a typical rise time of 2 ns, the knee frequency is 250 MHz. When low-pass circuits are cascaded, the resulting transition time is the square root of the sum of the squared transition times. For example:

- A 250 MHz scope with a 250 MHz probe displays a 2 ns rise time as a 3.5 ns rise time ($\sqrt{2^2 + 2^2 + 2^2}$). This is a +75% error.

- A 500 MHz scope with a 500 MHz scope probe displays a 2 ns rise time as a 2.4 ns rise time ($\sqrt{2^2 + 1^2 + 1^2}$). This is only a +20% error.

Beware of slow scopes and slow scope probes.

PCB Characteristics

- Min. trace width: typically 10 mil (0.25 mm), down to 4 mil (0.1 mm)
- Min. trace pitch: twice trace width
- Hole diameter: 20 mil, down to 8 mil (0.5 mm to 0.2 mm). Thin traces and small holes make the board more expensive.
- Layer thickness: typically 8 mil (0.2 mm), down to 4 mil (0.1 mm). The dielectric constant ε is 4.5 for typical FR4 PCB material.

Beware of Trace Inductance

A 0.5-inch long, 10-mil wide trace, over an 8-mil thick PCB layer, connected to the underlying ground plane through a 14-mil via at the end, has an inductance of 9 nH. To a 2-ns rise time, this is an impedance of 15 Ω.

Typical Lumped Parameters

- **Capacitance** - A narrow trace has a capacitance of 2 pF per inch (0.8 pF per cm). Copper area has 130 pF per square inch (20 pF per cm^2).
  - Ceramic decoupling capacitors have a series resistance of several ohms, and an inductance of 3 to 30 nH. Use several capacitors in parallel to lower the high-frequency impedance.
  - Via capacitance is 0.5 pF (insignificant) but the inductance of >1 nH can be significant for very fast transitions (3 Ω for a 1-ns transition).
- **Inductance** - A 1-inch long wire (25 mm) has an inductance of 80 nH, which at 100 MHz means 50 Ω (definitely *not* a short circuit).
- **Resistance** - 1/4 W axial resistors have a series inductance of 2.5 nH and a parallel capacitance of ~2 pF. 1/8 W axial resistors have 1 nH and <2 pF.

Transmission Lines

Short connections can be treated as lumped capacitors, but any line with a propagation delay of more than 30% of the signal transition time must be analyzed as a transmission line. With a 3-ns transition time, any line longer than six inches (15 cm) is a transmission line. With a 1-ns transition time, any line longer than two inches (5 cm) is a transmission line.

Characteristic Impedance of a PCB Trace

- 50 Ω for outer trace, width = 2x PCB layer thickness (microstrip).
- 75 Ω for outer trace, width = 1x PCB layer thickness (microstrip).
- 50 Ω for inner trace, width = 0.6x PCB layer thickness (stripline).
- 75 Ω for inner trace, width = 0.25x PCB layer thickness (stripline).

Higher characteristic impedances are unrealistic on modern PCBs.

How to Prevent Signal Degradation Due to Reflections (Ringing)

- Keep the trace short (best solution, but often impossible).
- Terminate at the destination (end termination). *(See Figure1.)*
  Problem: High sink or source current, or both. High power consumption.
  Solution: Capacitive coupling to the terminating resistor. Time constant should be longer than the transition, but much shorter than the flat time. For example: 200 pF x 50 Ω = 10 ns.
  See ***Figure 2***.
- Terminate at the source (series termination).
  Advantage: No static power consumption.

Figure 1.

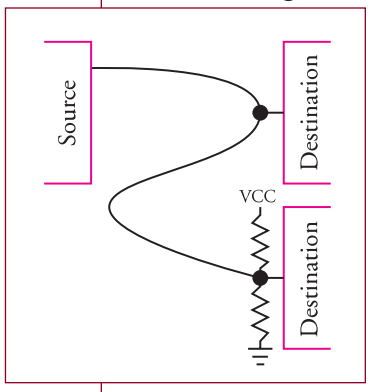


Figure 2.

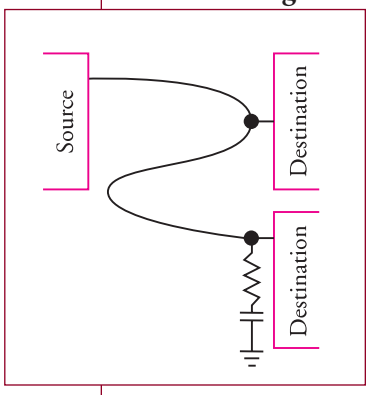
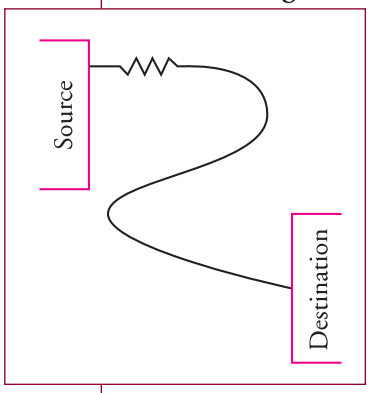


Figure 3.



Continued on next page

Continued from the previous page

Problem: Use only for single-destination.
50% amplitude along the trace!
See ***Figure 3***.

Signals propagate at the speed of light divided by the square root of the effective dielectric constant. The speed is therefore higher on the outer traces that have air on one side, than on the inner traces that are completely surrounded by epoxy.

- Propagation delay on an outer trace: 140 ps per inch (55 ps per cm).
- Propagation delay on an inner trace: 180 ps per inch (70 ps per cm).

Ground Plane

Always analyze the whole current loop: Vcc terminal to Vcc distribution, to Vcc-pin, to device output, to PCB trace, to ground distribution, to ground terminal, through the decoupling capacitors, and back to Vcc. Keep this loop inductance as small as possible by reducing the area of the loop. The return current tries to follow the signal trace; it follows the path of least inductance (least energy).

- Make it easy for the signal and return paths to stay close together.
- Avoid obstacles in the ground plane.
- Investigate the reason for any detour of the current.
- Watch out for slots in the ground plane causing detours in the return path, leading to crosstalk and ringing.

Ground Bounce

Ground bounce is caused by a voltage drop on the inductance between chip internal ground (bonding wire + leadframe, especially in ceramic PGAs) and PCB ground. The pulse width increases with capacitive loading, but the pulse amplitude is independent of capacitive loading. Ground bounce can cause wrong output levels on adjacent outputs and can cause inputs to be misinterpreted. For a High-to-Low transition, the internal ground jumps first to a positive voltage, then undershoots. *(See the 1998 Xilinx Data Book, pages 13-16)*

Synchronous designs with one common clock are surprisingly resilient. They tolerate ground bounce, because it occurs directly after the triggering clock edge, whereas input levels only matter at setup time before the next active clock edge. However, beware of all asynchronous inputs and clocks — they are susceptible to misinterpretation due to ground bounce. Make sure these signals are parked far away from the input thresholds.

Crosstalk

Crosstalk is especially strong when many lines run closely parallel, as in data busses. Inductive crosstalk is usually bigger than capacitive. Crosstalk can be minimized by using an unobstructed ground plane. Synchronous busses can tolerate a lot of crosstalk if it occurs only after the synchronous data transitions.

Problem: Asynchronous control lines, strobes, interrupts that run parallel to the data or address busses.
Solution: Increase the spacing.

Vcc Decoupling Capacitors

Supply decoupling is not a luxury. For fast internal and external transitions, these capacitors are the only instant source of current. The power supply with its big electrolytic capacitor is too far away and has too much inductance. (See *XCell #20*, pages 42-43). Low-impedance ceramic decoupling capacitors are required to supply dynamically changing Icc inside the chip, and to provide a return path for external current changes. In CMOS systems, all power is dynamic. The instantaneous current peaks are much higher than the average dc current, which is between 100 mA and 2 A for the larger Xilinx devices.

Decoupling capacitors must have low inductance and low series resistance. The capacitance value is irrelevant, as long as it is sufficient. Use 0.01 to 0.1 µF capacitors, mounted very close to each Vcc pin and directly connected to the ground plane. Keep the lines very short. A narrow, 0.25-inch (6 mm) long trace represents 20 nH; a current change of 100 mA/ns causes a voltage drop of 2 volts across this inductance.

Credits

For an excellent, in-depth treatment of these subjects, read “High-Speed Digital Design” by Johnson and Graham, Prentice Hall, 1993, or attend a class given by the author, Howard W. Johnson, Redmond WA. (howiej@olympic-tech.com). ◆

XC4000XL FPGAs Interface to SDRAMs at 100MHz

by BRAD TAYLOR

Xilinx XC4000XL FPGAs can easily interface to modern systems running at 80 MHz. However, some applications require even higher I/O speed. Devices such as SDRAMs, SSRAMs, and GigaBit Ethernet ICs require I/O speeds of up to 133 MHz using 3.3-V TTL signaling. This article describes how the unique I/O system of the XC4000XL FPGA enables you to build a full-speed SDRAM controller.

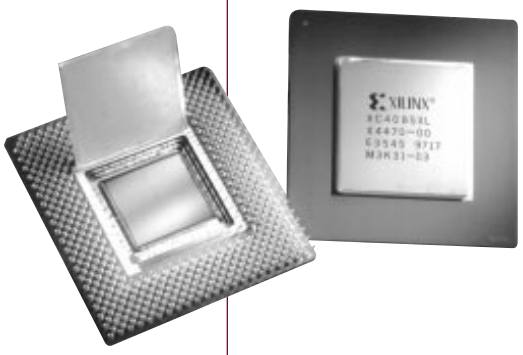
SDRAMs are becoming the new standard for large memory devices. This trend follows the introduction of EDO DRAM several years ago, which replaced page-mode DRAMs. EDO DRAMs run at 33 MHz (roughly double the speed of typical page-mode DRAMs). The new SDRAMs run at 66-125 MHz, and are now being used for main memory storage in PCs. They are quickly becoming low-cost devices, selling for less than $3/MB.

The Synchronous SDRAM interface

SDRAMs are clocked and fully synchronous, referencing all I/O transactions to the positive clock edge. The timing model is very simple, with all pins behaving the same. For a *write* operation, the data, address, or control information to a 100 MHz SDRAM must be present on the pin 3.0 ns before the clock edge (T_{SU}) and held valid until 1 ns past that clock edge (T_{HOLD}). For a *read* operation, a read-request is clocked into the device. Three clock periods later, data will appear on the data pins. This data is guaranteed to be valid 7.5 ns after the third clock edge (T_{OH}) and will be held valid for 3.0 ns past that clock edge (T_{DV}). **Figure 1** shows these timing relationships.

"No-delay" Input Modes

All XC4000 FPGAs contain the ability to capture input data using input flip-flops (IFFs). By default, the inputs are configured to include an additional delay that balances the clock delay. The purpose of this delay is to eliminate the need to hold data valid after a clock edge. However, when interfacing to SDRAMs, this additional delay may be unnecessary because the SDRAM holds the data valid after a clock transition. Xilinx XC4000 FPGAs have a special input mode known as "no-delay" to support this requirement. The advantage of this no-delay mode is that it significantly reduces the input set-up time, and thus allows much faster operation.



XC4000XL No-delay Setup and Hold Requirements

The no-delay setup requirement (the time data must be stable before the FPGA clock-pin edge) is less than 1.7 ns for XC4000XL-09 FPGAs.

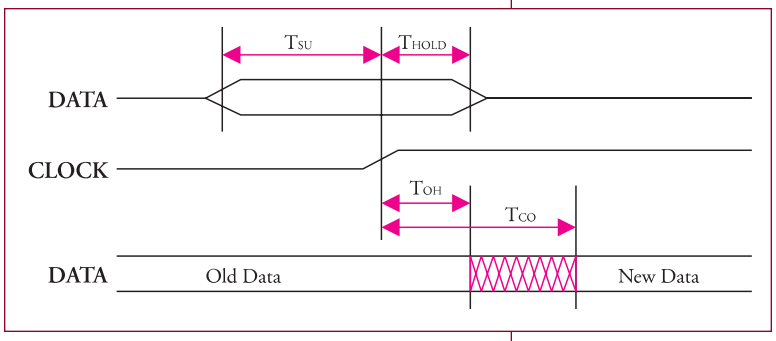


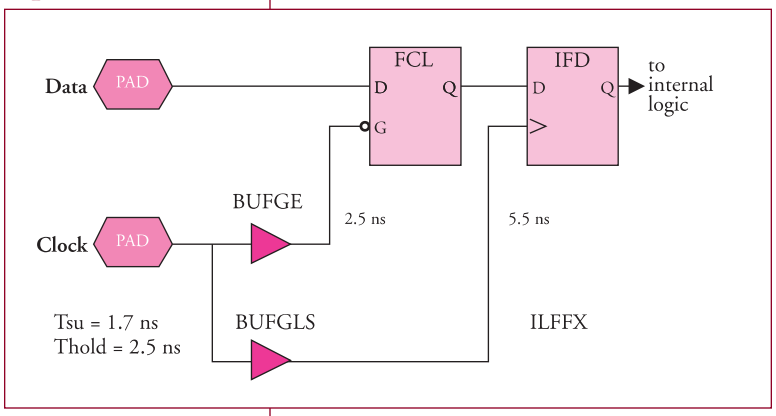
Figure 1. The XC4000XL family's unique I/O system enables the SDRAM controller to operate at full speed.

The no-delay hold requirement (the time data must be held stable after the FPGA clock-pin edge) is no longer than the clock delay from the clock pad to the IOB clock input node. For the XC4020XL-09 (and all smaller devices) the normal clock delay from the global low-skew clock distribution network (BUFGLS) will always be less than 3 ns. Larger FPGAs such as the XC4085XL can have

Continued on the next page

clock delays of less than 3 ns for hundreds of IOBs when you use special I/O clock buffers. In both cases, the clock delay to the IFF (and thus the FPGA input hold requirement) can be kept below the 3 ns value for which the SDRAM is guaranteed to hold data valid after a clock edge.

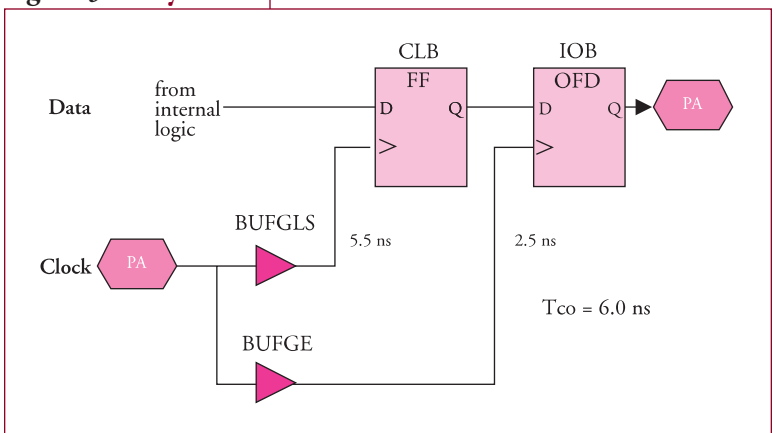
Figure 2. Fast Capture Latches



XC4000XL High-speed I/O Clock Distribution Features

XC4000XL FPGAs contain special internal clock buffers known as global early buffers (BUFGEs). These buffers can distribute an early clock to I/O pins. For even the largest XC4000XL FPGAs, the BUFGEs can be used to distrubute a clock to up to 64 IOBs in less than 2.5 ns (-09 speed grade). There are BUFGE clock buffers in each corner of the FPGA, and each of these buffers can distribute a clock to the IOBs in the quadrant it occupies.

Figure 3. Early Clock



More than one BUFGE may be used in parallel if it is necessary to distribute an early I/O clock to the IOBs in more than one quadrant.

Fast Capture Latches Enable the Use of Early I/O Clock

The data that has been captured by the early clock (BUFGE) must be transferred to the logic inside the FPGA, which is clocked by the slower global low-skew clock (BUFGLS). XC4000XL IOBs contain a special fast capture latch (FCL) option which can hold data until it is transferred to the normal IFF in the IOB, which is clocked by the slower global clock. Once the IOBs are configured in this special early capture mode, the operation is transparent.

As a result, data is captured with minimal set-up time and a limited hold time with respect to the clock pin. Yet, it is available at the IFF output, synchronous with the global low-skew clock (BUFGLS) used for all internal logic. *See **Figure 2.***

Early Clocks get the Data out of the FPGA On-time

In addition to reducing input hold time, the Early I/O clock buffers also speed up FPGA output times. The FPGA's clock-pad-to-output-valid pin-to-pin delay (T_{CO}) is the sum of the clock delay to the IOB clock node plus the clock-to-pad delay. If the I/O clock delay is less than 2.5 ns, T_{CO} will be less than 6 ns for XC4000XL-09 FPGAs. This is a respectable performance for an FPGA and is equivalent to that of the fastest TTL devices. Inside the FPGA, data must be transferred from GLS-clocked registers to the BUFGE-clocked IOB output registers. *See **Figure 3.***

A race condition between the data and the early I/O clock could exist if the BUFGLS delay were significantly less than the BUFGE delay. The XC4000XL architecture prevents this possibility as long as the timing tools indicate that the BUFGE delay is less than the BUFGLS delay.

Time Constraints Ensure Correct Data Transfer

Because Xilinx software supports static timing constraints, it is easy to ensure the proper transfer of data from the BUFGLS-clocked registers to the BUFGE-clocked registers. You can do this by reducing the timing constraint for the selected paths by the GLS clock delay.

Clock Loading Adjustment can Reduce T_{CO} to 5.5 ns

The 1998 Xilinx Data Book contains a "capacitive load factor" table (page 4-77), that lists output delays in the presence of high capacitive loads such as those presented by SDRAM modules (up to 100 pF). For example a 100 pF load increases the output delay by 1.8 ns. Conversely, a 35 pF load reduces the clock-to-output delay by 0.5 ns. With a clock delay of 2.5 ns, the output delay is a maximum of 5.5 ns.

Output Hold Time Required by High Speed Devices

FPGA and PLD vendors have traditionally not published minimum output hold times. Because 0.5 ns to 1 ns of input data hold is required by many high-performance devices, Xilinx will soon publish a minimum output hold time. This parameter (T_{OH}) is expected to be in the range of 1 ns for BUFGE-clocked outputs and 2 ns for BUFGLS-clocked outputs.

Board Delay, Clock Skew, and Clock Jitter

In addition to the on-chip delays, there is always a certain delay between the devices on the pc-board (typically about 150 ps/inch). Many systems rely on reflective switching for data to reach its final value. This requires a round trip and increases the value to 300 ps/inch. Fixed clock skew and random clock jitter between devices must also be taken into account to ensure valid data transfer between devices.

Putting it all Together

By taking advantage of these high-performance I/O features, even the largest XC4000XL FPGAs can meet aggressive time specifications. These parameters are "pin-pin" specifications in that they are referenced to the clock and I/O pins only. T_{SU}, T_{HOLD}, T_{CO}, and T_{OH} are defined in **Table 1**.

FPGA to SDRAM Transfer at 100MHz

T_{CO}(fpga) + T_{SU}(sdram) = 6.0 ns + 3.0 ns = 9.0 ns; (This allows 1.0 ns slack for board delay and clock jitter.)

T_{DV}(fpga) ³ T_{HOLD}(sdram) = 1.0 ns ³ 1.0 ns; (Requires the board delay to compensate for clock jitter.)

Check the SDRAM to FPGA Transfer at 100MHz

T_{CO}(sdram) + T_{SU}(fpga) = 7.5 ns + 1.7 ns = 9.2 ns; (This allows 0.8 ns slack for board delay and clock jitter.)

T_{OH}(sdram) ³ T_{HOLD}(fpga) = 3.0 ns ³ 2.5 ns; (This allows 0.5 ns slack.)

> *"The I/O performance that was formally obtainable only with custom devices or high-performance ASICs is now available to*

Table 1: Pin-to-Pin I/O Parameters

Device	T_{OH}	T_{CO} (35pf)	T_{SU}	T_{HOLD}
SDRAM (10 ns)	3.0 ns	7.5 ns	3.0 ns	1.0 ns
XC4085XL-09*	1.0 ns	5.5 ns	1.7 ns	2.5 ns

** The FPGA specifications assume that the XC4000XL FPGA uses up to four early clock buffers for I/O with up to 64 IOB clock loads each, is configured with FCL No-delay inputs and fast outputs, and has its outputs loaded with 50pF each.*

Beyond 100MHZ

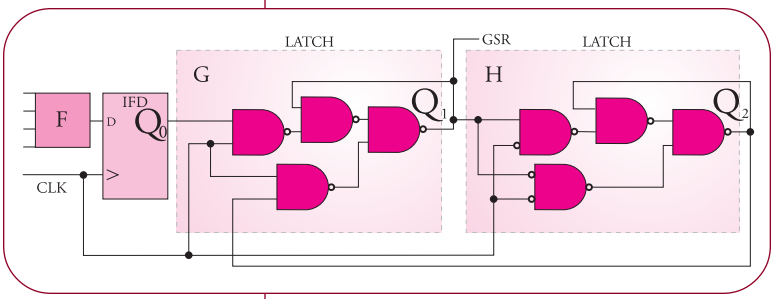
Other features, such as PLLs, or known clock skew between the various clocks, can be used to further increase I/O performance. For smaller FPGAs, such as the newly introduced XC4002XL, 133 MHz operation is easily achieved. The I/O parameters obtained by using these techniques demonstrate that FPGAs are compatible with high-speed devices such as SDRAMs. The I/O performance that was formally obtainable only with custom devices or high-performance ASICs is now available to Xilinx FPGA users. ◆

Self-Initiated Global Reset

by PETER ALFKE ◆ (peter@xilinx.com)

For some designs, it may be necessary to drive the device Global Set/Rest (GSR) from on-chip logic. This poses a problem, because all flip-flops on the chip are affected by GSR, which will cause the operation to end prematurely, resulting in unpredictable behavior. Described here is a reliable solution that fits into a single CLB. The circuit consists of two latches built into the G and H function generators, using direct feedback. These latches are unaffected by GSR.

Figure 1. One CLB



Circuit Description:

The GSR input condition is decoded in look-up table F, and the next rising clock edge sets flip-flop Q0. Q0 in turn sets latch Q1, which drives the GSR signal that affects all flip-flops, even resetting Q0, but leaving Q1 unaffected. The subsequent clock Low signal sets Q2, the following clock High signal resets Q1, and the following clock Low signal resets Q2.

As shown, this circuit generates a GSR signal that lasts one clock period. If necessary, this period can be extended throughout the following clock Low time by driving GSR with the logic OR of Q1 and Q2. This circuit is hazard-free and reliable, but may cause problems with simulators that cannot cope with combinatorial feedback loops. ◆

***Send your circuit ideas to* EDITOR@xilinx.com**

Using the XC4000XL for prototypes, and the lower cost Spartan-XL for production.

SPARTAN-XL/XC4000XL COMPATIBILITY CHART

Spartan-XL (Production) Speed Grades -3,-4			**XC4000XL (Prototype)** Speed Grades -3,-2			
Part No.	**Total CLBs**	**Usable I/O**	**Part No.**	**Total CLBs**	**Usable I/O**	**Common Package**
XCS05XL	100	61	**XC4005XL**	196	61	PC84
		77			77	VQ100
XCS10XL	196	61	**XC4005XL**	196	61	PC84
		77			77	VQ100
		112			112	TQ144
XCS20XL	400	113	**XC4010XL**	400	113	TQ144
XCS30XL	576	192	**XC4013XL**	576	192	PQ240
		192			192	BG256
		113			113	TQ144 (HT144[2])
XCS40XL	784	193	**XC4020XL**	784	193	PQ240
		205			205	BG256
Speed	-3		**-3**			
Grade	-4		-2			

See article on page 5

CMOS I/O Characteristics

by PETER ALFKE ◆ (peter@xilinx.com)

This article will give you an overview of our device I/O characteristics, to help you create better, more reliable designs.

All Xilinx devices use CMOS technology, which means there are two types of transistors:

- N-channel transistors, turned on by a positive gate voltage.
- P-channel transistors, turned on by a negative gate voltage.

For either transistor, the turn-on voltage must exceed the ~1-V threshold voltage.

Figure 1 shows a complementary inverter, consisting of a p-channel pull-up transistor and an n-channel pull-down transistor with both gates driven in common.

Outputs

All Xilinx devices, except for the XC9500 family and the original XC4000 family, have complementary outputs. However, for XC4000E, XC4000EX, and Spartan families, you must specify this option explicitly. The default on these devices is "TTL output" as described below.

Complementary outputs *(See **Figure 1.**)* are pulled "rail-to-rail," maximizing the output swing, especially desirable when driving other CMOS logic. With no DC load, the output voltage swings precisely between ground and Vcc with no voltage drop (the device output specifications of 0.4 and 3.86-V refer to particular dc loading conditions).

"TTL outputs" *(See **Figure 2.**)* have a reduced voltage swing, which achieves faster performance, especially for the High-to-Low transition when measured at the usual 1.5-V level. The term "TTL output" is actually a misnomer, derived from the similarity with the "totem-pole" structure of bipolar TTL outputs that use only npn transistors for pull-down and pull-up. Similarly, the "TTL output" structure in CMOS uses only n-channel transistors for pull-down and pull-up. This reduces the output High voltage (V_{oh}) by one threshold voltage, (1 to 1.5-V) below Vcc.

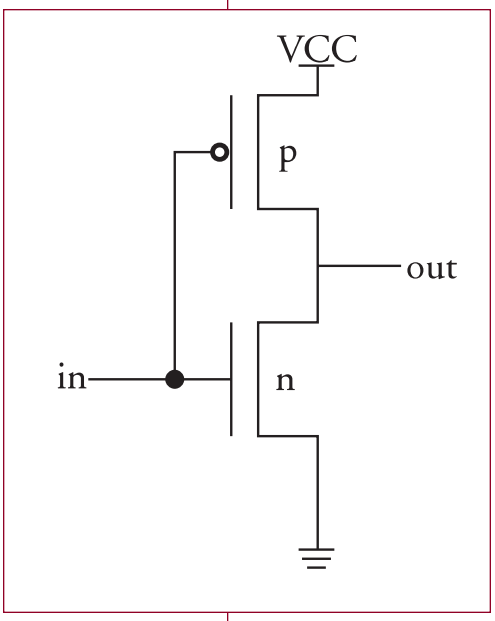
At 3.3-V supply voltages (and lower), complementary "rail-to-rail" or CMOS is the only available (and meaningful) output option.

The output impedance for our FPGAs is 15 to 30 Ω in the Low state, and 30 to 50 Ω in the High state. The output impedance for our XC9500 CPLDs is 10 to 12 Ω in the Low state, and 70 to 120 Ω in the High state

The XC9500 and the original XC4000 devices have TTL-level outputs only. On XC4000E, XC4000EX, and Spartan devices, TTL-output is the default, but can be changed to complementary output. If any data sheet specifies V_{OH} as >3.5-V, it is a complementary output. If V_{OH} is specified as 2.4-V, it is a TTL-level output.

Note that an output driving a long interconnect line or PC board trace can see reflections that drive the output well above Vcc and well below ground. Such reflections usually last for just a few nanoseconds (<10 ns) and are usually suppressed by the ESD protection diodes.

Figure 1.
Complimentary Inverter or Buffer.



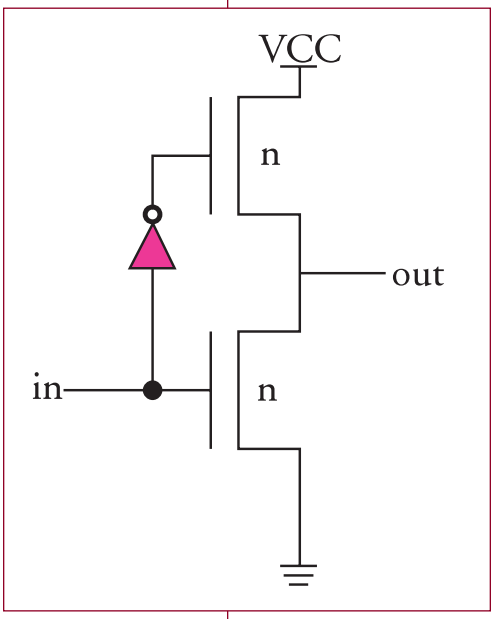


Figure 2.
Totem-Pole "TTL Output" Buffer

Continued on next page

How to Evaluate the XC4000XL for Your Next Application

by PETER ALFKE ◆ (peter@xilinx.com)

A lot of data and applications information is available on our XC4000 FPGA families. This article will help you find what you need, focusing specifically on the XC4000XL. All page numbers listed here refer to the Xilinx 1998 Data Book, available in print, on the AppLINX #5 CD, and at www.xilinx.com.

If you have never used Xilinx FPGAs, read **XC3000, XC4000, and XC5200: A Technical Overview for the First Time User** (page13-5). This will give you a broad overview of the basic features common to all Xilinx FPGAs.

If you want to get a more detailed comparison between the different Xilinx FPGA families, read **Choosing a Xilinx Product Family** (page 13-7) and see the table on page 13-12. This will give you a feel for the relative advantages of the different families.

If you have narrowed your choice to the 3.3-V XC4000XL family, then see page 4-5, the first page of the data sheet, for a list of important features. The ten XC4000XL devices represent the most advanced evolution of the industry's most popular FPGAs — the XC4000 series. The XC4000XL family uses 0.35 micron technology to achieve the highest speed and the lowest power. This cutting-edge technology requires a 3.3-V supply, but all XC4000XL inputs and outputs are fully compatible with 5-V devices. You can thus mix devices using old and new-technology on the same PC board.

The basic advantages of all XC4000 families lie in their architectural features:

- SelectRAM offers thousands of very fast 16 x 1 bit RAMs with synchronous or asynchronous write; also configurable as dual-port RAMs.
- Dedicated carry logic speeds up arithmetic and counters
- Segmented routing supports high clock rates

Continued on next page

CMOS I/O

Continued from previous page

Inputs

TTL input thresholds are compatible with older systems, and are popular in bus-oriented systems. TTL is therefore the default and the most popular input option. It means that a voltage below 1.2-V is interpreted as Low, while a voltage above 1.4-V is interpreted as High. This mimics the behavior of bipolar TTL circuits where this threshold is determined by two forward-biased silicon diodes. In Xilinx FPGAs, this "TTL" threshold is achieved by a reduced supply voltage on the input buffers, controlled by a global option that affects all device inputs.

CMOS inputs are specified such that a voltage below half the supply voltage is interpreted as Low, and a voltage above half of Vcc is interpreted as High. The actual threshold is usually somewhat lower than 50% of Vcc. In XC4000XL devices it is 38-40% of Vcc. For 5-V devices, the input threshold is globally selectable as either "TTL" or "CMOS"; the CMOS input option offers additional noise immunity and reduces static power consumption.

Inputs have a small amount of hysteresis, which makes the threshold for the rising edge a little higher than for the falling edge. Slow transitions will, therefore, switch cleanly, as long as there is no system noise greater than 100 mV. In a real system, this hysteresis does not make much difference. Slow transitions on data, control, and other combinatorial inputs just cause extra unpredictable delay. Slow transitions with more than 10 ns rise- or fall-time are very dangerous, since they invite any noise or ground-bounce to cause double triggering. ◆

Continued from the previous page

➤ Powerful I/O structures allow fast inter-chip operation.

The architecture of all XC4000-series devices is described on page 4-9. The functionality of the dedicated pins is described on page 4-41. The methods of configuring (programming or customizing) the devices are covered on page 4-46.

Capacity

The table on page 4-6 gives you a feel for the capacity of the 10 different XC4000XL family members. Logic capacity can be expressed in many different ways. The trailing digits of the part name describe the capacity in kilogates (e.g., the XC4010XL holds roughly 10,000 gates). However, gate-count is not really a meaningful metric — Xilinx FPGAs implement their logic in look-up tables, not in gate-array-like 2-input NAND gates.

Capacity is better described by Logic Cells, where each Logic Cell is one 4-input look-up table plus one flip-flop. Most designers have a good feel for the number of flip-flops required. It is reasonable to assume that less than two Logic Cells are needed for every flip-flop in a typical design. This allocates, on average, up to two look-up tables in front of each flip-flop. Highly structured and pipelined designs might use only one look-up table per flip-flop, and thus use only one Logic Cell per user flip-flop.

Speed

XC4000XL devices come in four speed grades: -09, -1, -2, and -3. The lower the number, the faster the device (read 09 as 0.9). Page 4-71 lists the delays of a large number of parameters inside the chip, and for the inputs and outputs. All parameters are guaranteed over the operating range of 0-to-85° C junction temperature and 3.0 to 3.6-V supply voltage. Xilinx never publishes “typical” numbers, because they are of little use and can be misleading.

I/O Pin-to-Pin Parameters

➤ Input set-up time (with zero hold time guaranteed) on page 4-78.

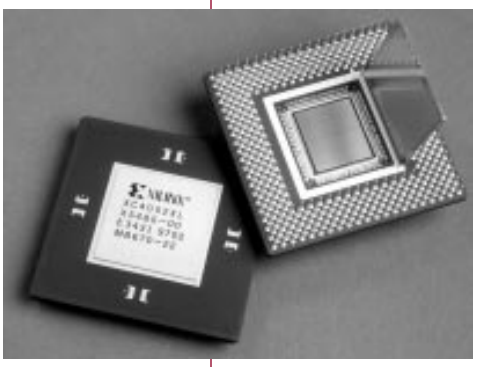
➤ Clock input to output data delay on page 4-77.

The sum of these two parameters defines the clock period for inter-chip communication.

I/O electrical characteristics are described in *I/O Characteristics of the ‘XL FPGAs*, on page 13-13, covering 5-V tolerance, PCI compliance, sink and source I/V curves, and the effect of capacitive loading.

On-chip Performance

The delay through a 4-input look-up table (page 4-74) is the shortest logic delay (1.2 to 2.7 ns). It is independent of the complexity of the equation. (For example, three cascaded XORS are as fast as a single inverter.) Other parameters describe the flip-flop set-up time of 0.6 to 1.1 ns (either through a 4-input look-up table or bypassing it) and the internal clock-to-out delay of 1.5 to 2.1 ns.



RAM

Any 4-input look-up table can be used as a 16x1 RAM, and two tables can be combined to be a 16x1 dual-port RAM. The write operation can be made synchronous; the 16 locations can be written into as if they were flip-flops. Note that the XC4000XL devices do not have any pulse-width constraints in their synchronous clocking. (The restrictive note on page 4-15 does not apply to any XC4000XL device.)

Dedicated Carry

A dedicated carry structure simplifies adders, subtractors, comparators, accumulators, and counters, and makes them faster; a 32-bit accumulator can run at 60 MHz.

Interconnect Resources

The XC4000XL family has considerably more interconnect resources than the 5-V XC4000E family, although both families use identical logic structures. Interconnect delays can be a short as 0.1 ns (between neighbors) or more than 20 ns when the signal has to

Continued on next page

Continued from previous page

pass through many intermediate switching points. Interconnect delays are not listed in the data sheet, but they are reported with 0.1 ns accuracy by the design software. You can constrain the software to achieve a specified maximum delay for any path in your design. This method can make interconnect delays as predictable as logic delays.

Available Package Options

Available package/device type options are listed on page 4-147. Note that Xilinx offers many chips in the same package with identical footprints. This makes it easy to migrate to a larger or smaller device when features must be added or deleted.

Page 4-111 lists the specific pin-outs per device type and package.

Configuration Modes

There are six ways to load configuration data into the device, as described in **FPGA Configuration Guidelines** on page 13-31. The most popular mode is master serial mode, described in detail on page 4-62. The device is configured from data stored in a Xilinx serial PROM as described in section 5. Page 4-50 lists the number of configuration bits required, while pages 5-2 and 5-12 list the number of bits available in the different SPROM devices.

Power Consumption

Static power consumption is negligible, just a few milliwatts. Dynamic power consumption is proportional to the clock frequency, and is thus design dependent.

The typical dynamic power ranges from milliwatts for small devices at low clock rates, to several watts for large devices clocked at 50 to 100 MHz. When compared to competing devices, the XC4000XL family uses less power because of its 3.3-V supply voltage and because of its segmented routing structure that minimizes interconnect capacitance.

Xilinx guarantees the performance parameters up to a junction temperature of 85°C (commercial grade), and provides a de-rating factor of 0.35% per degree C for junction temperatures up to 125°C in plastic packages.

The junction temperature can be calculated from the power consumption and the ambient temperature by using the package-dependent values for thermal impedance, with and without airflow, as printed on page 10-5. ◆

Application Notes

The XC4000XL architecture is a functional superset of the XC4000E architecture, which in turn is a superset of the original XC4000 architecture. All application notes written for any of these families are applicable to the XC4000XL. The following examples are all available at http://www.xilinx.com/apps/4000.htm:

XAPP080: Supply-voltage Migration, 5V to 3.3V

XAPP088: I/O Characteristics of the XL FPGAs

XAPP065: XC4000 Series Edge-Triggered and Dual-Port RAM Capability

XAPP057: Using Select-RAM Memory in XC4000 Series FPGAs

XAPP056: System Design with New XC4000X I/O Features

XAPP051: Synchronous and Async. FIFO Designs

XAPP052: Efficient Shift Registers, LFSR Counters, and Long Pseudo-Random Sequence Generators

XAPP013: Using the Dedicated Carry Logic in XC4000E

XAPP023: Accelerating Loadable Counters in XC4000

XAPP018: Estimating the Performance of XC4000E Adders and Counters

XAPP014: Ultra-Fast Synchronous Counters

XAPP010: Bus-Structured Serial Input/Output Device

XAPP009: Harmonic Frequency Synthesizer and FSK Modulator

XAPP008: Complex Digital Waveform Generator 16-Tap, 8-Bit FIR Filter

XAPP054: Constant Coefficient Multipliers for the XC4000E

XAPP055: Block Adaptive Filter

XAPP062: Design Migration from XC4000 to XC4000E

XAPP015: Using the XC4000 Readback Capability

XAPP079: 4MBit Virtual SPROM

XAPP017: Boundary Scan in XC4000 and XC5200 Series Devices

XAPP011: LCA Speed Estimation: Asking the Right Question

Using Programmable Logic to Accelerate DSP Functions. A Guide to Using Programmable Gate Arrays for Application-Specific DSP Performance

Building High Performance FIR Filters Using KCMs

The Fastest FFT in the West

The Fastest Filter in the West

Plug and Play ISA in Xilinx FPGAs

Dynamic Microcontroller in XC4000

Pulse-Width Modulation in Xilinx

Configuring Mixed FPGA Daisy Chains

XC4000/XC5200 PC84 Footprint Compatibility

Low-Power FPGA Achieves 400 MHz Performance

by PETER ALFKE ◆ (peter@xilinx.com)

This article discusses several techniques for creating low-power designs demonstrated by a 6-digit frequency counter with a maximum frequency input of 400 MHz. When the input frequency is below 10 MHz, current consumption is below 2 mA, and the operating current at 100 MHz input frequency is below 10 mA. This low-power design, which is implemented in an XC4002XL FPGA, has the following characteristics.

- The time base is derived from a 32 kHz oscillator, controlled by a typical watch crystal.
- The 400 MHz input frequency resolution (250 picosecond period) is made possible by pre-scaling the display counter with a binary ripple counter. This means that only one flip-flop in the FPGA must toggle at the incoming frequency. This flip-flop is carefully placed near the input pin and near the clock input. The use of a global clock, which we usually suggest for synchronous designs, is avoided here in order to save power and increase performance.
- The 6-digit BCD counter is synchronous within each digit, but ripples between digits. This takes advantage of the 4-input look-up-table architecture, while also minimizing total power consumption.
- The non-multiplexed LCD display is driven directly from the complementary CMOS outputs. Because the 48 segments and five decimal points each require AC drive voltage without a DC component, the display backplane is driven with a 0 to 3.6-V square wave of 128 Hz, and the individual segments are driven either in phase or out of phase with their backplane. The necessary XOR gating is easily performed in the new XC4000XL output structure; no additional logic is required.
- The control structure is simple and efficient, operating from a 32 kHz oscillator. The basic assumption is that the display must be updated twice a second. A higher or lower update rate would be irritating to the observer. This 500 ms measuring time and 6-digit display means that input frequencies above 100 MHz must be prescaled by a factor 500 before being counted in the BCD counter. Auto-ranging reduces the prescaler for lower frequencies. Below 10 MHz the display has a fixed resolution of 10 Hz, but with leading-zero suppression in the display.

  The whole control structure uses fewer than eight CLBs (12% of the design).
- The low power consumption allows operation on a rechargeable battery. This makes the instrument portable and avoids real problems and certification issues relating to high-voltage line operation.

> *"By paying close attention to the layout and design of your system, you can achieve both high performance and low power consumption, using standard Xilinx FPGAs."*

Conclusion

By paying close attention to the requirements and the structure of your system, you can achieve both high performance and low power consumption, using standard Xilinx FPGAs. ◆

by Kamal Karaitem

Foundation Series Software

What is the new library format being used by Foundation 1.4, and are there any issues involved?

The Foundation 1.4 Library Manager allows you to use long names (up to 255 characters) for library objects. Foundation 1.3 and earlier versions allowed names up to 24 characters only. Thus, the library format has been changed from version 4.0 in Foundation 1.3 to 5.0 in Foundation 1.4

When you open a project which was created in pre-F1.4 software, the Project Manager will ask you if you wish to convert the library to the new 5.0 format. If you answer **yes**, the library will be converted and you will be able to make changes to it in F1.4. If you answer **no**, the 4.0 format will be preserved, and the library will be read-only in F1.4.

Libraries in the 4.0 format can be converted to the 5.0 format to enable using them in access modes other than **R/O**. To convert a 4.0 library to the new 5.0 format:

1. Go to the Library Manager. (This is done by selecting **File > Project Libraries...** and then double clicking on the **Lib Manager** button, or by selecting **Tools > Library Manager**.)

2. Select the library you want to convert.

3. Select **Convert to version 5.0** from the Library menu.

The original 4.0 format library will be saved in the **BAK_24** folder, which will be created under the **<project_name>\Lib** directory. The conversion affects two library files (DIR and SYN). If it is necessary to revert back to the 4.0 format library, simply copy the contents of the **BAK_24** directory into the **LIB** directory.

NOTE: During the start of the Foundation Series suite, the system checks the format of all libraries. If it finds the 4.0 library, then the library access mode is automatically changed to **R/O**. Conversion to the 5.0 format is run automatically when you change the library access mode to **R/W** or **R/A**. The **R/S** access mode of system libraries is not changed.

5.0 format libraries cannot be used with older tool sets (F1.3 and earlier). F1.4 does not contain a conversion tool that converts from 5.0 to 4.0 format. To revert back to 4.0 format, you must copy the backed-up library, as described above.

How do I simulate bi-directional signals in the Foundation Logic Simulator?

Use the **Chip Controlled** (CC) mode when simulating bi-directional signals. This will allow a chip (symbol) output to override the test vector.

Suppose you are simulating a bi-directional pin **BIDIR**, and have applied a stimulator to **BIDIR**. You want this stimulus to appear on **BIDIR** when the 3-state buffer is disabled. Now, when the 3-state buffer is enabled, you want to view its output on **BIDIR**, but the input test vector is overriding the output. Using **CC** mode on the **BIDIR** stimulator will allow the output of the 3-state buffer to override the stimulus.

To apply the **CC** mode to a stimulator, select the signal in the Waveform Viewer, and either click on the **CC** button in the **Stimulator Selection** Window, or select **Stimulator > Chip Controlled Mode**. Stimulators that are in **Override** (**OV**) mode are displayed in red in the **Waveform Viewer**. Stimulators that are in **CC** mode are displayed in black.

If you are using a **CMD** (script) file, use the **release (r)** command to release the bi-directional pin, as shown below:

```
h DATA   #assigns input stimulus
 to bidi pin DATA
l EN     #tri-states output enable
sim
h EN     #enables output so that
 data is driven OUT onto DATA
r DATA   #releases bidi pin DATA
 so that data may be driven out
sim
```

I use Workview Office, but I don't like having to go to DOS or the EDIF Interfaces GUI to run the required programs for implementation and simulation. Is there any way to simplify these flows?

Yes. The Tools menu in ViewDraw can be customized to include Xilinx-specific functionality. For example, when this command is run from a DOS session:

```
custmenu
 %XILINX%\viewlog\data\viewblox.txt
```

Two new commands appear in the Tools menu:

✔ Add LogiBLOX

✔ Change LogiBLOX

Three more Xilinx-specific commands can be added to the ViewDraw Tools menu. A file containing these commands, **xvdrawm1.zip**, can be found at: ftp://ftp.xilinx.com/pub/swhelp/viewlogic

or at:
http://www.xilinx.com/techdocs/htm_index/sw_viewlogic.htm.

Within this ZIP file are two script (.txt) files, an executable (**vfuncsim.exe**), and a **readme.txt**. After following the instructions in the **readme** file, open ViewDraw and check the **Tools** menu. In the **User** section, you will see three new commands:

✔ **Write Xilinx EDIF** – This command allows you to run **EDIFNETO** without going to the EDIF GUI or to DOS. It includes the **Level=Xilinx** property required for Xilinx designs.

✔ **Xilinx Functional Simulation** – This command prepares a **FUNC_SIM.VSM** file for simulation. This is necessary for designs with RAM, ROM, or instantiated netlists.

✔ **Read Xilinx Timing EDIF** – This command imports **TIME_SIM.EDN**, produced by the Design Manager in preparation for a timing simulation.

Reference solution # 1985: http://www.xilinx.com/techdocs/1985.htm

I've upgraded to Workview Office 7.4, and now I see errors when reading a timing simulation EDIF file back into Viewlogic for simulation. What's going on?

EDIFNETI is the Viewlogic program that will read an EDIF netlist back into the Viewlogic realm as part of the simulation flow. A change was made with the 7.4 version of this tool, and the following error may been seen as a result:

```
viewbase: Error 413: Error -
 Schematic Pin Z on XBA1.1 is
 not on its symbol...
```

This message repeats for each pin mismatch it finds. This error occurs due to an increased level of error checking within **EDIFNETI**.

Solution — Viewlogic has built a patch to return to the functionality of version 7.31. This patch is located at:

ftp://ftp.viewlogic.com/pub/support/pc/Workview_Office/update/edfupdt1.exe

Viewlogic's readme for this patch is located at:

ftp://ftp.viewlogic.com/pub/support/pc/Workview_Office/update/edifneti.html

Also, when reading hierarchical EDIF into Viewlogic, do not name any port on a symbol the same as the symbol itself. To avoid any conflict, **EDIFNETI** changes the name of that port, potentially causing problems with annotation to the original schematic.

Reference solution # 3081: http://www.xilinx.com/techdocs/3081.htm

Timing

Timing Analyzer will not open large report files. How can I use this tool to view large reports?

The Alliance Series 1.4 release resolves the problem on Windows NT and Workstation platforms by allowing report sizes of up to 32MB. However, Windows 95 will still continue to have this problem until an upcoming release. You may still view the large reports created by Timing Analyzer in Windows 95 by going to the directory referenced in your **TEMP** environment variable (usually **C:\WINDOWS\TEMP**) and opening the most recently modified non-empty file (prefixed by "xil_") in a text editor.

Another option is to save the report file directly to disk without relying on Timing Analyzer to display it. To do this, select **View > Console** in Timing Analyzer. In the console window that pops up, enter the equivalent command in the command field. For example, to do an **All Paths** type report, enter **AnalyzeAllPaths s c:\mydir\allpaths.twr**. Timing Analyzer will not attempt to display the file, but will save it directly to disk. The other reporting commands are **AnalyzeTimingConstraints** and **AnalyzeDesignPerformance**.

You may abbreviate these with "aap", "atc", and "adp" respectively (example: atc s c:\mydir\timing_constraints.twr).

My timing report will not show PAD to IFD or OFD to PAD paths. How can I get these timing numbers?

Currently, Timing Analyzer (or TRACE) will not report on a path from a pad to a IOB input flip-flop. This problem will be fixed in an upcoming release. It will report on the path from an IOB output flip-flop to a pad if you use the **OFFSET** constraint; however, the delay value reported may be more conservative than the "Guaranteed Input and Output Parameters" reported in the 1998 Xilinx Data Book. The pad-to-input-register and output-register-to-pad delays are fixed, so the data book's "Guaranteed Input and Output Parameters" numbers can be used.

Useful WebLINX Links to our Expert Journals

Implementation tools	http://www.xilinx.com/support/techsup/journals/implmnt/index.htm
Timing & Constraints	http://www.xilinx.com/support/techsup/journals/timing/index.htm
Foundation	http://www.xilinx.com/support/techsup/journals/foundatn/index.htm
Viewlogic	http://www.xilinx.com/support/techsup/journals/viewlogc/index.htm
Synopsys FPGA Compiler	http://www.xilinx.com/support/techsup/journals/synopsys/index.htm
Synopsys FPGA Express Users	http://www.xilinx.com/support/techsup/journals/fpga_exp/index.htm
Mentor Graphics	http://www.xilinx.com/support/techsup/journals/mentor/index.htm
Cadence	http://www.xilinx.com/support/techsup/journals/cadence/index.htm
Boundary Scan / JTAG	http://www.xilinx.com/support/techsup/journals/jtag/index.htm
CPLD Core tools	http://www.xilinx.com/support/techsup/journals/cpld/index.htm

EPIC

How do I probe a net in EPIC?

There currently is no "probe" command in EPIC, but there is a simple PERL script contained in **$XILINX/userware/utilities** named **add_probe**. You may run this file to generate a script that can be executed in EPIC. You may also add the probe directly in EPIC.

Reference solution #2979: http://www.xilinx.com/techdocs/2979.htm

JTAG Programmer

JTAGProgrammer issues a "Boundary Scan Integrity" error while executing Boundary Scan instructions on a chain of Boundary-Scan-capable devices. What does this error mean?

Chain integrity must be assured before the results of the Boundary Scan testing can be relied upon. A chain may have many faults that can interfere with the integrity of the chain, such as:

- ✔ A component in the chain may be missing, dead or incorrectly loaded.
- ✔ A component in the chain may have a broken connection on one of its TAP pins.
- ✔ A TDO to TDI connection between two components could be shorted to another node.
- ✔ A component in the chain may not have sufficient power supply.
- ✔ System or clock noise could cause a component in the chain to jump TAP states.
- ✔ A combination of all the problems described above may exist in a chain of components making it very difficult to diagnose.

JTAGProgrammer performs the integrity test by relying on the device's ability to set a fixed data pattern in its Instruction Register in the **Capture-IR** state of the TAP Controller. This concatenated data can then be shifted out on the TDO pin of the last device. The value of **attribute INSTRUCTION_CAPTURE** in the BSDL (Boundary Scan Description Language) file for each device in the chain shows what this data will be.

Additionally, the IEEE standard 1149.1 mandates that the two LSBs of this data should be "01".

For a simple test case, assume that there are three devices in the chain and the Instruction Registers are only two bits wide. In this case, for a good chain, we will see "010101" coming out on the TDO pin of the last device.

As an example of a bad chain, assume there is something wrong with the connection of the first device in the chain. This device then will not output "01" on its TDO pin. Instead, it may push a value such as "11". We will see "110101" coming out on the TDO pin of the last device. By examining the data stream closely, we can see that the last two devices in the chain are connected properly and there is a problem with the first device in the chain.

JTAGProgrammer will issue a **Boundary Scan Integrity** error if receives unexpected data at the TDO pin of the last device. The integrity failure can occur for a single device as well as for multiple devices in the chain. ◆

COMPONENT AVAILABILITY CHART

PINS	TYPE	CODE	XC3020A	XC3030A	XC3042A	XC3064A	XC3090A	XC3020L	XC3030L	XC3042L	XC3064L	XC3090L	XC3142L	XC3190L	XC3120A	XC3130A	XC3142A	XC3164A	XC3190A	XC3195A	XC4003E	XC4005E	XC4006E	XC4008E	XC4010E	XC4013E	XC4020E	XC4025E	XC4028EX	XC4036EX
44	PLASTIC LCC	PC44		◆												◆														
	PLASTIC QFP	PQ44																												
	PLASTIC VQFP	VQ44																												
	CERAMIC LCC	WC44																												
64	PLASTIC VQFP	VQ64		◆					◆							◆														
68	PLASTIC LCC	PC68	◆	◆											◆	◆														
	CERAMIC LCC	WC68																												
84	PLASTIC LCC	PC84	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆					
	CERAMIC LCC	WC84																												
	CERAMIC PGA	PG84	◆	◆	◆										◆	◆	◆													
100	PLASTIC PQFP	PQ100	◆	◆	◆										◆	◆	◆				◆	◆								
	PLASTIC TQFP	TQ100														◆	◆													
	PLASTIC VQFP	VQ100		◆	◆				◆	◆			◆			◆	◆				◆									
	TOP BRZ. CQFP	CB100	◆		◆										◆	◆	◆													
120	CERAMIC PGA	PG120																			◆									
132	PLASTIC PGA	PP132			◆	◆											◆	◆												
	CERAMIC PGA	PG132			◆	◆											◆	◆												
144	PLASTIC TQFP	TQ144			◆	◆	◆			◆	◆	◆	◆	◆			◆	◆	◆			◆	◆							
	CERAMIC PGA	PG144																												
	HI-PERF TQFP	HT144																												
156	CERAMIC PGA	PG156																				◆	◆							
160	HI-PERF QFP	HQ160																												
	PLASTIC PQFP	PQ160				◆	◆											◆	◆	◆		◆	◆	◆	◆	◆				
164	TOP BRZ. CQFP	CB164					◆												◆	◆		◆								
175	PLASTIC PGA	PP175					◆												◆	◆										
	CERAMIC PGA	PG175					◆												◆	◆										
176	PLASTIC TQFP	TQ176					◆					◆		◆					◆											
	HI-PERF TQFP	HT176																												
191	CERAMIC PGA	PG191																						◆	◆					
196	TOP BRZ. CQFP	CB196																							◆					
208	PLASTIC PQFP	PQ208					◆												◆	◆		◆	◆	◆	◆	◆				
	HI-PERF QFP	HQ208																							◆	◆	◆		❖	
223	CERAMIC PGA	PG223																		◆						◆	◆	◆		
225	PLASTIC BGA	BG225																							◆	◆				
228	TOP BRZ. CQFP	CB228																								◆		◆	❖	
240	PLASTIC PQFP	PQ240																								◆				
	HI-PERF QFP	HQ240																								◆	◆	◆	❖	❖
256	PLASTIC BGA	BG256																												
299	CERAMIC PGA	PG299																										◆	❖	
304	HI-PERF. QFP	HQ304																										◆	❖	❖
352	PLASTIC BGA	BG352																											❖	❖
411	CERAMIC PGA	PG411																												❖
432	PLASTIC BGA	BG432																												❖
475	CERAMIC PGA	PG475																												
559	CERAMIC PGA	PG559																												
560	PLASTIC BGA	BG560																												

APRIL 1998

PINS	TYPE	CODE	XC4002XL	XC4005XL	XC4010XL	XC4013XL	XC4020XL	XC4028XL	XC4036XL	XC4044X	XC4052XL	XC4062XL	XC4085XL	XC40125XV	XC5202	XC5204	XC5206	XC5210	XC5215	XC9536	XC9572	XC95108	XC95144	XC95216	XC95288	XCS05	XCS10	XCS20	XCS30	XCS40
44	PLASTIC LCC	PC44																		◆	◆								
	PLASTIC QFP	PQ44																											
	PLASTIC VQFP	VQ44																		◆									
	CERAMIC LCC	WC44																											
64	PLASTIC VQFP	VQ64													◆														
68	PLASTIC LCC	PC68																											
	CERAMIC LCC	WC68																											
84	PLASTIC LCC	PC84	◆	◆	◆										◆	◆	◆	◆			◆	◆				◆	◆			
	CERAMIC LCC	WC84																											
	CERAMIC PGA	PG84																											
100	PLASTIC PQFP	PQ100	◆	◆	◆										◆	◆	◆				◆	◆	◆						
	PLASTIC TQFP	TQ100																			◆	◆	◆						
	PLASTIC VQFP	VQ100	◆	◆											◆	◆	◆									◆	◆	◆	◆	
	TOP BRZ. CQFP	CB100																											
120	CERAMIC PGA	PG120																											
132	PLASTIC PGA	PP132																											
	CERAMIC PGA	PG132																											
144	PLASTIC TQFP	TQ144		◆	◆										◆	◆	◆	◆									◆	◆	◆	
	CERAMIC PGA	PG144																											
	HI-PERF TQFP	HT144				◆	◆																						
156	CERAMIC PGA	PG156													◆	◆													
160	HI-PERF QFP	HQ160						◆	◆	◆																			
	PLASTIC PQFP	PQ160		◆	◆	◆	◆									◆	◆	◆	◆			◆	◆	◆					
164	TOP BRZ. CQFP	CB164																											
175	PLASTIC PGA	PP175																											
	CERAMIC PGA	PG175																											
176	PLASTIC TQFP	TQ176			◆												◆	◆											
	HI-PERF TQFP	HT176				◆	◆																						
191	CERAMIC PGA	PG191															◆												
196	TOP BRZ. CQFP	CB196																											
208	PLASTIC PQFP	PQ208		◆	◆	◆	◆										◆	◆										◆	◆	◆
	HI-PERF QFP	HQ208						◆	◆	◆									◆					◆	◆					
223	CERAMIC PGA	PG223																◆											
225	PLASTIC BGA	BG225																◆	◆										
228	TOP BRZ. CQFP	CB228							❖			❖																	
240	PLASTIC PQFP	PQ240				◆	◆											◆										◆	◆
	HI-PERF QFP	HQ240						◆	◆	◆	◆	◆							◆										
256	PLASTIC BGA	BG256			◆	◆	◆	◆																				◆	◆
299	CERAMIC PGA	PG299						◆											◆										
304	HI-PERF. QFP	HQ304						◆	◆	◆	◆	◆							◆										
352	PLASTIC BGA	BG352						◆	◆	◆									◆					◆	◆					
411	CERAMIC PGA	PG411							◆	◆	◆																		
432	PLASTIC BGA	BG432							◆	◆	◆	◆	❖	❖															
475	CERAMIC PGA	PG475										◆																	
559	CERAMIC PGA	PG559											◆	◆															
560	PLASTIC BGA	BG560									◆	◆	◆	◆															

◆ = Product currently shipping or planned

❖ = New since last issue of *XCell*

Corporate Headquarters
Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Tel: 408-559-7778
Fax: 408-559-7114

Europe
Xilinx, Ltd.
Benchmark House
203 Brooklands Road
Weybridge
Surrey KT14 0RH
United Kingdom
Tel: 44-1-932-349401
Fax: 44-1-932-349499

Japan
Xilinx, KK
Daini-Nagaoka Bldg. 2F
2-8-5, Hatchobori,
Chuo-ku, Tokyo 104
Japan
Tel: 81-3-3297-9191
Fax: 81-3-3297-9189

Hong Kong
Xilinx Asia Pacific
Unit 4312, Tower II
Metroplaza
Hing Fong Road
Kwai Fong, N.T.
Hong Kong
Tel: 852-2424-5200
Fax: 852-2494-7159

Rome wasn't built in a day. Your ASIC can be.

SPARTAN™

FPGA

www.xilinx.com

PN:XLQ298

XILINX®

2100 Logic Drive
San Jose, CA 95124-3450

First Class Presort
U.S. Postage
PAID
Permit No. 2196
San Jose, CA