Fall 2000

# Xcell journal

THE AUTHORITATIVE JOURNAL FOR PROGRAMMABLE LOGIC USERS

## PRODUCTS

New Virtex Development Board

New HDLC and ADPCM Cores

## APPLICATIONS

Using Block-level Incremental Synthesis

## SOFTWARE

New WebPack Integrated Synthesis Environment

## NEWS

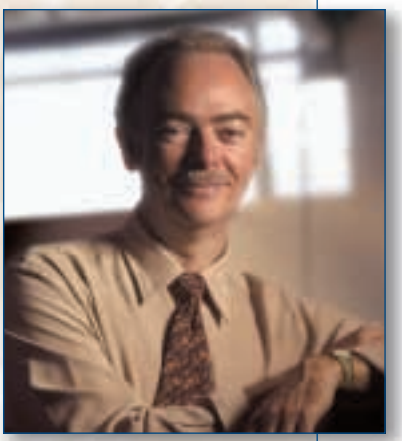
IBM and Xilinx Partnership—PowerPC Processors in Virtex-II FPGAs

## Cover Story

Synplicity Chief Technology Officer Ken McElvain Talks About the Future of Programmable Logic

XILINX®

# The Partnership Model...





Xcell journal

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


Programmable logic technology is advancing at a phenomenal rate; our industry is booming, business has never been better, and Xilinx is growing faster than we ever imagined. Our devices are bigger, faster, easier to use, and less expensive than ever before, and there are many new development tools, support products, and services being offered by a growing number of companies. In addition, as we add new features, increase performance, and lower prices, many new types of applications are opening for our products-and that fuels even more growth and more innovation.

The phenomenal growth of programmable logic technologies is the result of teamwork; we didn't do it alone. As you can read in the article on page 4 from our CEO, Wim Roelandts, Xilinx is based on a partnership model that includes our technology partners, our stock holders, and you, our customers. Clearly, our sustained success is very dependent on the success of those who work with us. That's one reason why we work closely with our technology partners and support their development of new products and services. It's also why we work closely with you, to make sure we are creating the right products for your current and future needs. It's also why we publish this journal.

### Communication is Key

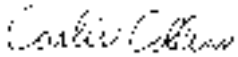
Communication is necessary to sustain a healthy partnership, and effective communication requires a two-way process. This journal intends to bring you the latest information about programmable logic technologies and how to use them. It also brings you the insights of industry insiders about the trends and developments that will affect you in the future, so you can make informed decisions today. We are always striving to include articles that make a positive difference for you, and present those articles in a format that is interesting and easy to read.

Your feedback is important because we want to continue bringing you the types of articles that matter most to you. After all, this is your journal; it exists only for your benefit. So, let me know what you think about our new Xcell Journal; your suggestions for improvement are always welcome.

### Write for Xcell

If you have a novel solution to a common design problem or if you have developed a unique design, why not share it with your peers? Writing technical articles for Xcell is easy, it can gain you a modest amount of fame and fortune, and it's a great way to promote your company. Send me an e-mail with your article ideas, and I'll send you more details.

Thanks for your continued support of this journal. I hope you like the new design.



Carlis Collins
Editor

**Cover Story** Page 8

## Design Technology Advances Unleash Powerful New FPGA Capabilities

The Chief Technology Officer at Synplicity talks about the trends that are shaping the FPGA industry.

**Products**

Page 28

## New HDLC and ADPCM Cores

A strategic partnership with ISS produces advanced communications products.

Page 46

## New Virtex XSV Development Board for Creating Intellectual Property

The XSV Board, designed for the SIP developer community, provides a flexible, low-cost platform that supports a large set of interesting applications.

**Applications** Page 24

## Using Block-Level Incremental Synthesis in FPGA Compiler II and FPGA ExpressBlock-Level

Incremental Synthesis allows you to modify a subset of a design and then re-synthesize just the modified subset.

**Software** Page 34


## New WebPack Integrated Synthesis Environment

Free, comprehensive design environment for CPLDs available over the Web.

**News** Page 13


## IBM and Xilinx Partnership—PowerPC Processors in Virtex-II FPGAs

IBM and Xilinx combine cutting edge technologies to create a revolutionary new product.

**Contents** Fall 2000




Xcell journal


**For a Free Subscription to the Xcell Journal E-mail your request to: literature@xilinx.com.**

Please include:
1. Your full name and mailing address.
2. Your job title.
3. Your e-mail address.
4. Your company name.
5. Is this a new subscription or a renewal?

# The Xilinx Business "Ecosystem"

by Wim Roelandts, CEO, Xilinx

In today's fast paced high technology business climate, the keys to success are complex and they change constantly. It's difficult for any one company to master all of the necessary functions of business management, product design, manufacturing, marketing, sales, and customer support. And, as the breadth of required knowledge and expertise expands, people are becoming much more specialized in their jobs, taking a more narrow focus, because there is so much to master. That's why it has become imperative that high technology companies, such as Xilinx, focus on the key aspects of their business, to become excellent at their core technology and business processes, while partnering with other companies that are also the best in what they do.

Well managed companies must also learn to partner with their customers to develop the insight and the processes that ensure the success of each new generation of technology. And, if the company is successful, its shareholders will profit and will continue to support the company through stock purchases—a key factor in the economic health of the company.

This partnership between companies, specialized partners, customers, and shareholders is like an ecosystem where each part depends on the others, and the boundaries between each are blurred as they work closely together. For each part of this ecosystem to survive and to thrive, all parts must work well together, in balance, and this must become the primary objective of the company management—if any part of the ecosystem fails, all will suffer.

### The Five Habitats

The Xilinx business "ecosystem," as I see it, is composed of five main "habitats," or overlapping systems:

- The Company - People who focus on the primary business goals; the direct employees and managers of the company.
- Customers - People who not only buy products and services but also influence the design of new products.
- Partners - People who support the ecosystem through supplying needed services to the company (such as manufacturing and sales, intellectual property and design support).
- Investors - People who own the company and support it through their continued investment.
- Government - People who have an impact on the overall business environment by making laws and regulations that can affect the company.

Each of these habitats must be healthy and prosperous for the overall ecosystem to be healthy and prosperous.

### The Company Habitat

For a company to prosper and grow, it must create and maintain a balanced ecosystem. Therefore the company must:

- Determine the core competencies required for long term success and become excellent in those key areas.
- Determine which functions can best be handled by other companies (partners), and manage those relationships well.
- Work closely with customers to develop the right products and to provide specialized services.
- Become a learning organization, adapting to the dynamics of the marketplace.
- Be dynamic, adaptive, and free to innovate, with room to make mistakes.
- Take good care of its employees by providing a rewarding, satisfying, and friendly work environment.
- Have a Chief Executive Officer who is a coach, not the "quarterback." Because the business "game" is dynamic, the coach must have competent managers and trust them to fulfill their specific responsibilities.

### The Customer Habitat

For customers to prosper and grow, they need:

- A continuous flow of innovation from the company to help them create leading edge products and remain competitive in their markets.
- Easy access to all of the company's intellectual property.
- Open channels of communication with the company to:
  - help it create the right products and services, and become a better company
  - receive the latest technical and marketing information

### The Partner Habitat

For partners to prosper and grow, they need:

- Advanced information about new products and new technologies coming from the company so they can be prepared.
- Cooperation from the company for product development and marketing.
- Constant communication with the company to share information about trends, and to maintain overall balance.
- A working relationship that helps them manage their business more effectively and be more successful.
- Training.

When partners are prosperous and growing, customers receive better service and a broader range of support products and services.

### The Investor Habitat

For investors to prosper and grow they need:

- The company to be well managed and provide good return on investment.
- To have a good understanding of the company's strategy and business practices.
- To have timely and unbiased information about the company's results.

In many cases, employees, customers, and partners are also company stockholders, and therefore they are doubly motivated to maintain a healthy environment that supports the long term growth of the company. This is important because the company's ability to borrow money and therefore its flexibility and adaptability are directly affected by the company's worth on the stock market.

### The Government Habitat

To ensure that government supports the company and to ensure that the company is a good corporate citizen, the company must:

- Educate the government about the company and its products.
- Give feedback on proposed laws and

policies that can an impact on the company and its industry.

- Build a partnership with local government to improve the local community and to reach common goals.

### How the Xilinx Ecosystem Works

Our primary responsibility must be to develop, manage, and balance our business ecosystem if we are to remain the leader in our industry. So, our first task was to decide what key competencies we needed within Xilinx and which we could entrust to outside partners.

We have defined four core competencies that we must keep within Xilinx:

- Product design and technology development.
- Marketing.
- Customer support.
- Partner relationship management.

By focusing on these core functions, we can put our resources to best use.

### Product Design

Product design is our primary focus because innovation is the lifeblood of any high technology company; we spend more money on research and development than any of our competitors. We continuously strive to create leading edge devices and support tools that meet the specific needs of our customers.

### Marketing

It's not enough to create "dream" products with more and more features. Excellent products must consistently meet the needs of our customers and our customers must be fully aware of what we offer. It is the function of marketing to make sure that we are in full communication with our marketplace, in both directions.

### Customer Support

To create superior technology, and make the world aware of it, goes a long way toward the success of any company. However, without effective and consistent customer support, it's not enough. Our technology is complex, and to use it effectively requires an in-depth and constantly evolving expertise. That's why we created our in-house staff of highly-trained support engineers and design services professionals who can quickly help you meet any challenge.

Sometimes the difference between our products and our services blur. For example, our in-house development tools and intellectual property are products, but their sole purpose is to assist you in creating the best possible designs with the least time and effort. So, though most of our revenue comes from the sale of FPGAs and CPLDs, we must also make sure the best tools and services are available too.

We also work with many third-party consultants to provide training, design services, IP development, tool development, and so on.

### Partner Relationship Management

Two of the most important competencies that we decided to entrust to outside partners are manufacturing and sales. So, Xilinx has no fabrication facilities or direct sales force. These are two of the most expensive functions in any company, and the costs are not fixed. In a cyclic industry, such as our own, we gain a significant cost advantage by outsourcing these functions to partners who have made it their business to be the best; plus we gain added flexibility that helps us weather the inevitable down turns without having to lay off employees or suffer heavy losses. Therefore, within Xilinx, we have dedicated people to manage these relationships with our 3rd party manufacturing and sales partners. In addition, we also have dedicated marketing people who manage our relationships with our development tool partners, helping them define and market their products to our customers.

For manufacturing, we have partnered with UMC in Taiwan and Seiko-Epson in Japan. UMC has created one of the world's leading IC fabrication facilities, with the very latest equipment and process technologies. And, because it is their business to manufacture semiconductors, they are always on the cutting edge of process technology.

We work very closely with UMC to develop new manufacturing technologies and to ensure that our designs can make best use of the highest performance, and least expensive, process technologies. Through this partnership, we have moved from 0.5µ technology to 0.15µ in less than five years; and 0.13µ technology is soon to be in use. This fast-paced process migration has helped us to quickly reduce costs and significantly increase density, which means you get faster, less expensive devices, sooner. We could not have progressed this

quickly with an in house fabrication facility—the costs would have been far too high. This partnership has benefited both companies, as well as our customers.

For sales, we use our distributors and independent sales representatives. Sales is obviously a critical function in any company, and is very expensive. We chose to partner with our distributors because Xilinx products are a key source of revenue for them, often providing more revenue than any other product line. Therefore our distributors have a lot of incentive to focus on our products and to use their extensive sales force to focus on Xilinx. This is another win-win partnership that helps both companies adapt to the inevitable ups and downs of the business cycles. Our customers also benefit from the broad range of services provided by our distributors.

At Xilinx, average revenue per employee is $600K; many companies with in-house sales teams are fortunate to have $200,000/year/employee. By keeping our sales costs low, we can put more resources into research and development which keeps us on the leading edge of technology, and makes our customers and our investors very happy.

For software support, we partner with the industry's leading software and development tool suppliers to make sure that you have access to the broadest range of tools and support. We work very closely with the leading software suppliers to make sure that our tools work well together. Though we have a massive in-house effort for software development we also know how important it is to provide you with the new tools and processes that are constantly being created in the marketplace.

For research, we partner with universities and research centers around the world to access the latest developments. Plus, the Xilinx University Program is helping train engineering students in the use of programmable logic devices by providing donations, discounted products, and services. Today there are over 1200 universities using Xilinx in class labs; about 15% of all of the engineering universities worldwide.

## The Xilinx Value System

Values are what holds any organization together and define the boundaries within which its employees can efficiently function. Too often a company's values are undefined however and this leads to erratic results and confusion.

To help us manage and balance the Xilinx business ecosystem, we developed a clear and consistent set of values that we live by. In part, these values are the boundaries that allow our managers the room to innovate and to take ownership of their functions. Our values also help to make Xilinx a great place to work which inevitably leads to better products, happier customers, and increased profits.

Our values are contained within the acronym "CREATIVE" which stands for:

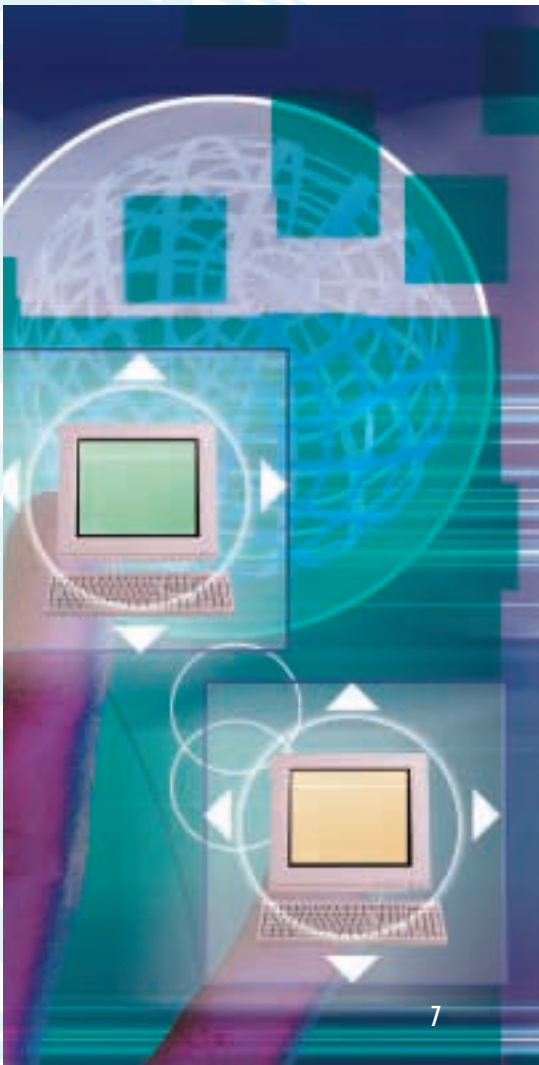
- **C** — **Customer Focused.** We exist only because our customers are satisfied and want to do business with us... and we never forget it!
- **R** — **Respect.** We value all people, treating them with dignity at all times.
- **E** — **Excellence.** We strive for "Best in Class" in everything we do.
- **A** — **Accountability.** We do what we say we will do and expect the same from others.
- **T** — **Teamwork.** We believe that cooperative action produces superior results.
- **I** — **Integrity.** We are honest with ourselves, each other, our customers, our partners, and our shareholders.
- **V** — **"Very"** Open Communication. We share information, ask for feedback, acknowledge good work, and encourage diverse ideas.
- **E** — **Enjoying Our Work.** We work hard, are rewarded for it; and we maintain a good sense of perspective, humor, and enthusiasm.

## Conclusion

Many companies—even successful ones—view themselves as adversaries with their customers and with their marketplace, often using military tactics and focusing on winning at all cost. We think there is a better way.

To become successful, and remain competitive, high technology companies must view themselves as part of a synergistic whole that includes customers, strategic partners, employees, investors, and government. At Xilinx, our role is to maintain the dynamic balance of this system, and to keep it growing and expanding in a way that makes everyone more whole and prosperous.

When you work with Xilinx, you'll see what a positive difference our "ecosystem" attitude can bring to your long term success.



# Design Technology Advances Unleash Powerful New FPGA Capabilities

**The Chief Technology Officer at Synplicity talks about the trends that are shaping the FPGA industry.**

by Ken McElvain
Chief Technology Officer, Synplicity, Inc.

In today's marketplace there is enormous pressure to create increasingly complex systems and get those systems to market as quickly as possible. To challenge these efforts, there is a lack of qualified engineers, industry standards are constantly changing, ASIC development costs are skyrocketing, and new technologies are quickly making yesterday's methods obsolete. These problems are intensified with the fierce competition for lucrative emerging markets. The risks are significant; one equipment vendor estimates the cost of missing a market window to be more than $1 million a day.

However, there is also good news. Today's multi-million gate FPGAs offer great promise for designers confounded by the limitations of traditional ASIC implementation. Indeed, the increased

time-to-market demands and lower development cost of FPGAs, combined with today's FPGA capacities well in excess of a million gates, are yielding a profound increase in the number of applications being realized in programmable form. From networking and telecommunications designers grappling with narrow market windows and evolving standards, to a broad scope of designers seeking low-risk rapid prototyping, the number of people turning to FPGA solutions is quickly expanding.

### Engineers Designing both FPGAs and ASICs

Only six years ago, engineers were divided into two distinct groups: those that designed ASICs and those that designed with FPGAs. The FPGA designers were primarily schematic based while the ASIC designers had adopted RTL technologies. These two groups didn't work together.

Today, things have changed. Many designers developing FPGAs are also designing ASICs, often at the same time. In addition, designers are now less concerned about the underlying device technology because the gap between ASIC and FPGA performance is narrowing. However, ASICs will remain the technology of choice for certain types of applications. If a design must operate at the low end of power consumption or at the extreme upper end of performance, or if you are designing a very high-volume, cost-sensitive system, an ASIC is likely the most cost-effective solution. If design flexibility or remote design upgradability are needed or if engineering time and risk must be minimized, an FPGA is a logical choice.

### The SoC Opportunity for FPGAs

The advent of systems on chips (SoCs) is more than a simple extension of ASIC technology to higher density. With SoCs, ASIC designers have become system designers, and the event-driven simulation technology traditionally used is no longer sufficient for verifying functionality. Instead, what designers urgently need is a faster, higher-level, hardware-oriented verification path that accepts actual system inputs to yield actual system outputs. Fortunately, just such an approach exists in RTL prototyping, a methodology being adopted widely by both ASIC vendors and system houses.

In contrast to conventional ASIC designers who tend to focus on the functions within their chips, SoC designers must give much more consideration to the system-level nature of the device's inputs and outputs.

TODAY'S MULTI-MILLION GATE FPGAS OFFER GREAT PROMISE FOR DESIGNERS CONFOUNDED BY THE LIMITATIONS OF TRADITIONAL ASIC IMPLEMENTATION.

-KEN MCELVAIN, SYNPLICITY

Software simulation, which checks circuit functionality against a blast of test vectors, falls short of allowing system I/O to be tested. Other drawbacks of software simulation are well known; it is too slow and can't reproduce electromechanical interactions such as fetching data from a drive. Simulation can take as much as 60 percent of the design cycle time and require too many iterations between RTL and gate-level implementation. Also, it does not allow for co-verification of hardware and software.

These roadblocks stand in the way of what SoCs require. For example, consumer and communication designs (areas of great promise for SoCs) compete within tight time-to-market windows that suffer when debugging takes many iterations through long cycles of simulation and synthesis. In addition, many chips for these markets must run at high clock rates to verify their correct operation. Inadequate verification of these designs can mean one or more fab re-spins, which can cost over $1 million per re-spin and take up to six months.

To take fuller advantage of the SoC phenomenon, designers can instead turn to RTL prototyping, a new methodology that is possible through the emergence of fast, dense FPGAs such as the Xilinx Virtex FPGAs and synthesis technologies that can translate ASIC RTL into multiple FPGAs. By building a version of their design in hardware using multiple FPGAs, designers can use RTL prototyping to leapfrog all of the limitations of software simulation.

### FPGAs Offer Design Flexibility

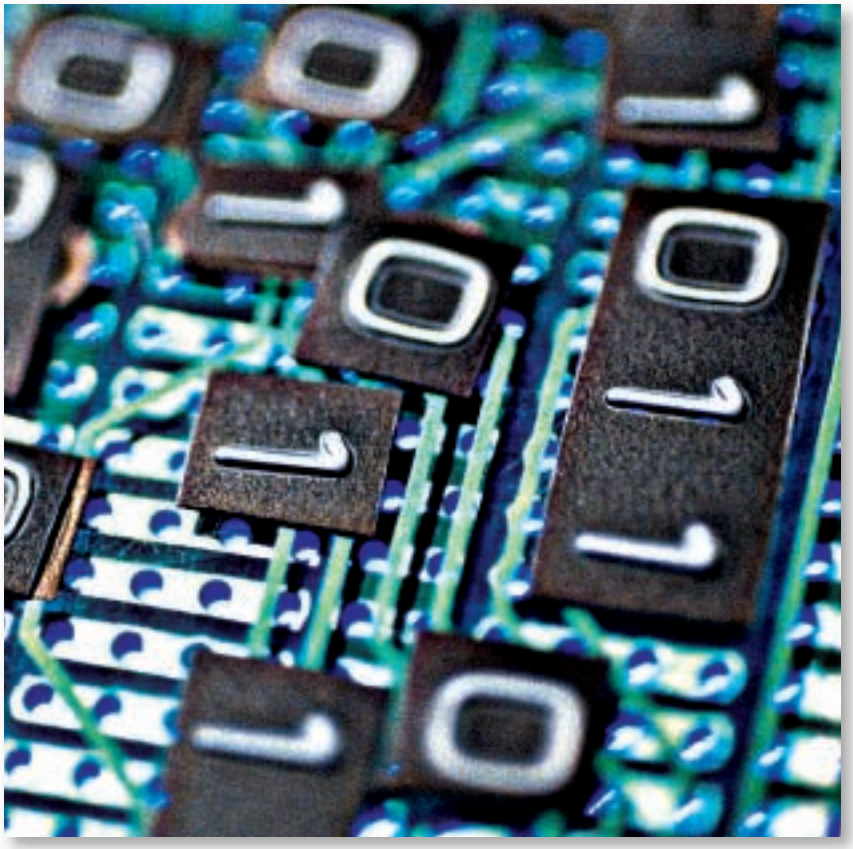
The Internet's explosive popularity, and the resulting surge in demand for bandwidth, has created immense competitive pressure for communications equipment makers. To complicate matters, the communications market is a minefield of specifications evolving unpredictably toward standardization. Schemes for carrying voice and video over the Internet, or the Internet over cable, or any number of variations on the Internet theme bring with them a multitude of implementation details that often must either be ironed out by committees or decided by the market.

Such last minute unpredictability makes bringing complex communications equipment to market quickly even more difficult. Equipment vendors who hope to exploit the high-density and high-volume economics of ASIC technology risk everything should a last minute protocol change force a silicon re-spin. Realizing this, increasing numbers of equipment manufacturers are instead opting for the fast turnaround and flexibility offered by FPGAs.

The quick re-configurability of FPGAs carries other benefits in addition to accommodating shifting specifications. By exploiting the option for in-system programmability, companies can easily and affordably make field upgrades, correct bugs and add features over the Internet, to name just a few possibilities.

### The Impact of FPGA Technology Advances

FPGA technology used to be one or two generations behind ASICs. However, the increase in FPGA density and performance has steadily outpaced that of ASICs for some time. Because of the increasing NRE costs of leading-edge ASICs, the typical ASIC design is falling farther behind the leading edge of process technology, while FPGAs are becoming process drivers.



However, taking full advantage of state-of-the-art FPGA technology today presents new and difficult challenges. As FPGA process technology progresses well into the deep submicron realm, interconnect-dominant delay now poses the same difficulties for FPGA designers that previously plagued their ASIC counterparts. Traditional schematic or logic synthesis-based programmable design solutions, similar to the ASIC methodologies of three to four years ago, lack the ability to adequately account for interconnect effects early in the design cycle. With today's highly complex circuits and relentless market pressure, it is more important than ever that accurate interconnect-related performance information be integral to early design processes.

### Evolution of Software

Design automation is key to enabling this new era of FPGA design. Unfortunately, simply extending interconnect-aware ASIC design technology to the FPGA domain won't work. The interconnect configurations and options unique to FPGA architectures cannot be comprehended utilizing ASIC physical modeling techniques. Instead, new FPGA-targeted design automation technology is needed. Such technology must address the difficult task of bringing accurate information about the physical interconnection of a programmable circuit into the design process without extending design cycles.

Fortunately, FPGA design technology is evolving to accommodate the needs of designers in the deep submicron era. Of particular significance is FPGA-based physical synthesis technology. Physical synthesis factors a design's physical characteristics into the synthesis process. During synthesis, a design is optimized and implemented based upon not only traditional timing constraints, but also physical constraints. The nature of FPGA architectures makes it possible to perform physical optimization techniques during synthesis, for example, moving registers across regional boundaries to increase performance.

Physical synthesis offers significant productivity as well as performance advantages to FPGA designers. First, the use of physical constraints during synthesis results in more accurate timing estimation, eliminating time-consuming and tedious design iterations common with traditional approaches. Likewise, physical optimization during synthesis makes it possible to physically optimize a circuit for the best possible performance. Combined physical synthesis and optimization techniques can have significant cost benefits, enabling designers in many cases to implement a device in a lower-cost speed grade.

FPGA physical synthesis forms the critical link between state-of-the-art programmable technology and designers seeking to leverage its unique advantages. Such technology opens the door of opportunity for those faced with today's tough market realities.

### Conclusion

Market forces have created a risky environment where the winners and the losers are often separated by only a small gap of innovation. Therefore, it is increasingly important to produce next-generation designs on time and within budget, using limited engineering resources. Today's programmable logic technology and the development tools that support them are your keys to success in this dynamic marketplace.

# Maximizing FPGA Design Performance Using Amplify from Synplicity

In very large designs, interconnect delays are becoming more and more predominant. Amplify™ is the first physical synthesis tool for FPGAs that helps you optimize signal routing delays and achieve timing closure.

**by Philippe Garrault**
Technical Marketing Engineer, Xilinx, Inc.
philippe.garrault@xilinx.com

Amplify is a new product from Synplicity® that allows you to add physical constraints to your design, using the Synplify® PRO synthesis engine. It links your RTL code to a floorplan of the target device. You can assign your RTL code to physical regions by graphically dropping the logic into these regions. During the synthesis process, Amplify will use a set of rules to optimize the RTL code based on estimated placement and interconnect delays (into and between regions).

Amplify generates an optimized netlist (.edf file) and a Xilinx constraint file (.ncf file) based on the specified physical constraints. These files are then used by the Xilinx Alliance Series 3.1i software tools to implement the design.

### Improving Maximum Frequency

With Amplify you can iterate the implementation twice. In the first pass, you determine the critical paths after place and route. In the second pass, you assign physical constraints on these critical paths to optimize the netlist. By iterating the second pass, the physical constraints can be refined according to the place and route post-layout timing report, until your timing requirements are met.

Another approach is to set several regions prior to implementation. These regions could be a representation of the RTL hierarchy of the design and each could contain one block (or module) of the design.

Finally, you could also mix these techniques to get a floorplan that would not only be a hierarchy representation but you could also add physical constraints on the critical paths regardless of any hierarchy consideration. This is what we did in the following example.

### Design Example

The following example shows how to interface Amplify with the Xilinx place and route tools. The design presented is a network application, which is divided into a top module driving nine similar sub-modules. This code is implementing FIFOs and large busses. We targeted a Virtex-E, XCVE1000-7 device.

On the first pass, the project is synthesized then implemented with global timing constraints only (without physical constraints). Figure 1 shows a floorplanned view of the design after the first pass. Note that the logic is spread over the chip because, without placement constraints, the place and route algorithm has no information about what logic is crucial for grouping into a region (or there are too many possibilities). The timing constraints were not met, and the best frequency obtained was 104.6MHz.

After this first implementation, we analyzed the post-layout timing report to gather information on the critical path, such as:

- The number of critical paths.
- The start and end points (single or multiple).
- The type of critical path (link with I/O or purely internal, wire or bus).
- The number of logic levels.
- The device resources (flip-flop, combinatorial, block RAM, and so on).

On the second pass, the different critical paths were assigned in separate regions through the Amplify user interface. Figure 2 shows the physical constraints entered in the synthesis environment for our example.

The regions are floorplanned according to the required design resources, such as block RAM and high fanout nets. . By re-synthesizing the design with these physical constraints, a new netlist file along with a constraint file were created. The place and route software uses these optimized files to constrain the logic on these particular areas by placing the critical logic together, shortening net delays to meet the timing requirements.

If the constraints are not met, the Xilinx floorplanner can give useful information about the final layout and determine the precise logic utilization within a region, or view the exact placement of logic on the critical path. This can help to resize regions, remove constraints on non-critical paths, or re-place regions closer together to drive or share common logic or busses.

Figure 3 shows the floorplanned view of the constrained design in our example. Note that the logic is gathered according to the constraint file. The best frequency obtained was 120.1MHz, which is a 12.9% improvement. Using the post-layout timing report and the floorplanner helped us focus on the critical parts of the design, thus saving time by applying more accurate constraints, and reducing the number of iteration needed to meet the timing constraints.

### Interfacing Amplify with Xilinx Tools

Here are some guidelines to help you get the best performance from these tools:

- Put different critical paths into different regions. This usually gives better results.
- If the critical path contains lots of logic, two regions can be overlapped: one small one containing the most critical logic that needs to be placed very close together and a bigger one containing the rest of the critical path.
- Amplify does not currently write constraints for block RAM or black boxes, thus if the critical path includes these objects, use the Constraints Editor and constrain black-boxes and block RAM in the UCF file within or close to the region constraining the rest of the critical path. Use the following command:

```
INST p1.qram1 LOC = RAMB4_R0C1:
RAMB4_R7C1, RAMB4_R*C2;
```

- Applying physical constraints to critical paths is more likely to improve results if the ratio between routing and logic is in favor of routing (this ratio is given by the post-layout timing report).
- Slightly moving a region can affect the maximum frequency of your design.

### Conclusion

Synplicity's Amplify Physical Optimizer in conjunction with the Xilinx Alliance Series 3.1i software can significantly improve your design performance. As a result of the new features and capabilities, you have a more efficient way to visualize and constrain critical paths of your designs, saving time in multiple iterations while getting better speed performance.

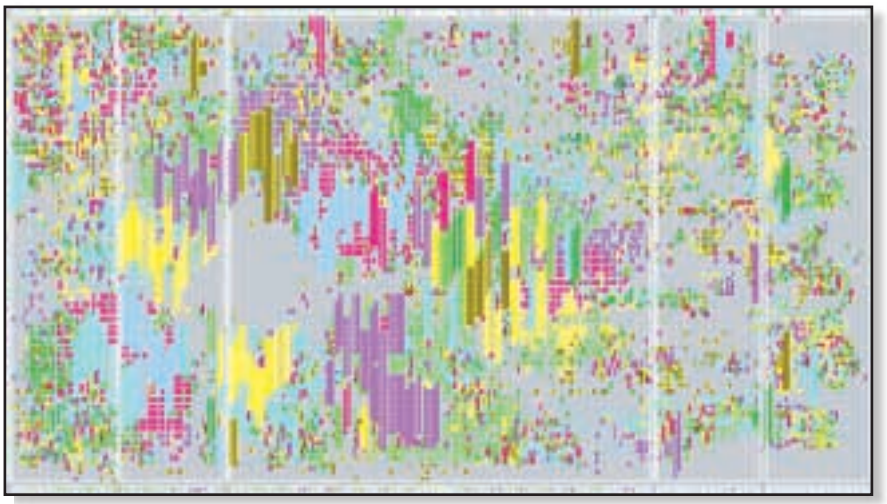


*Figure 1 - floorplanned view of the design on the first pass.*

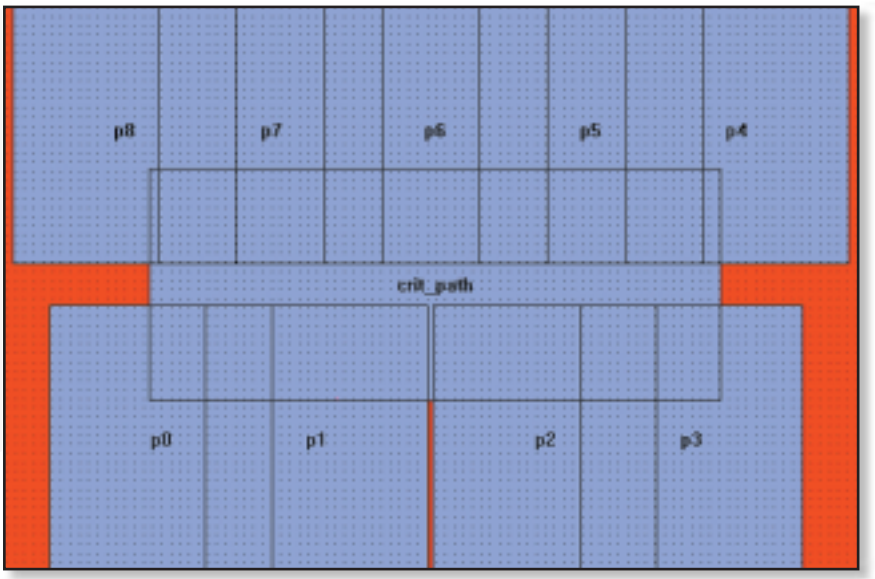



*Figure 2 - Physical constraints entered in the synthesis environment.*

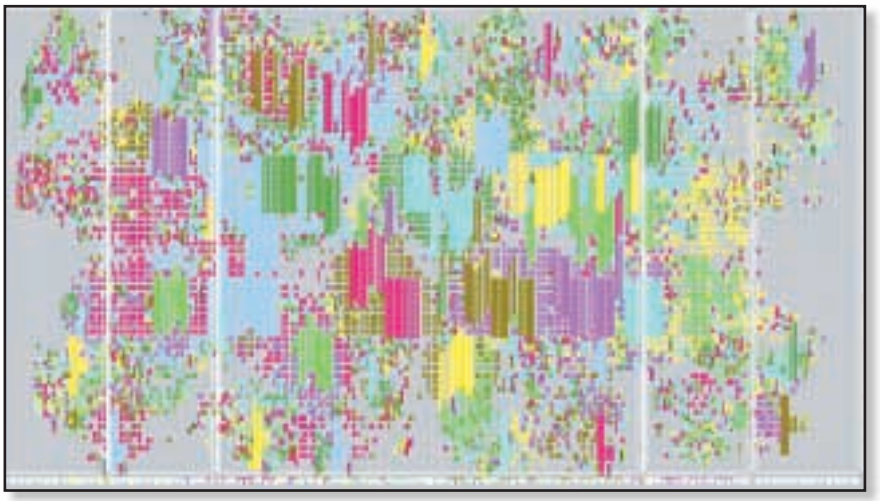


*Figure 3 - Floorplanned view of the constrained design.*

# PowerPC Processors in Virtex-II FPGAs

## IBM and Xilinx combine cutting edge technologies to create a revolutionary new product.

by Ann Duft
Manager of North American PR, Xilinx
annd@xilinx.com

IBM and Xilinx recently announced a partnership to create a new generation of devices for use in communications, storage, and consumer applications. Under the agreement, we are working together to embed IBM PowerPC processor cores in Xilinx VirtexTM-II FPGAs. The pairing of a low-cost, high performance PowerPC processor core with customizable FPGA circuitry allows you to create custom chips for your particular application at reduced cost and with faster time-to-market. Availability dates for the new chips will be announced later this year by Xilinx.


The new devices will initially be fabricated by IBM for Xilinx using advanced IBM chip manufacturing technologies, including copper interconnects which adds to their performance. This will enable Xilinx to broaden its manufacturing volume and geographic diversity, and leverage common foundry manufacturing processes.

"This joint effort will bring about a new design era combining programmable logic time-to-market advantages with the cost benefits of standard cell technology," said Andrew Allison, semiconductor industry analyst. "This is a potent combination."

"IBM and Xilinx are committed to meeting each customer's unique blend of requirements for cost, design time, and individualized function," said John Kelly, general manager, IBM Microelectronics Division. "This requires a variety of chip design options, from standard, off-the-shelf parts to FPGAs to ASICs. This agreement creates both a new approach in chip design, as well as a unique collaboration between the world's leading ASIC and programmable logic providers."

The complementary marketing and technology agreement will enable customers who choose FPGA solutions from Xilinx to more easily migrate to IBM ASIC and standard product solutions. Customers will factor performance, cost, time-to-market, and volume requirements in making a determination as to the best option for a given application, while using the same industry standard PowerPC and IBM's system-on-a-chip CoreConnect bus technology across all solutions.

"The combination of technologies will lead to a new level of performance and flexibility in the semiconductor market," said Wim Roelandts, president and CEO of Xilinx. "IBM's process technology is the most advanced in the industry and the PowerPC architecture has become the standard in communications, enabling us to deliver the highest performance and highest density products into the market at the leading edge of technology."

Under the multi-year agreement, Xilinx will license IBM's high-performance PowerPC processor cores and CoreConnect bus for integration into Xilinx FPGAs. IBM and Xilinx will map the resulting designs to IBM's advanced chip manufacturing processes, keeping Xilinx FPGAs on the leading edge of technology. IBM will license IP from Xilinx to quickly move leadership process technology to the marketplace. We also plan to explore other areas of cooperation that could benefit customers of both companies.

### About IBM

IBM Microelectronics is the world's leading ASIC supplier and a key contributor to IBM's status as a premier information technology provider. IBM Microelectronics develops, manufactures, and markets state-of-the-art semiconductor and interconnect technologies, products, and services. Its superior integrated solutions can be found in many of the world's best-known electronic brands. More information about IBM Microelectronics can be found at www.chips.ibm.com.

# The MathWorks and Xilinx take FPGAs into Mainstream DSP

## Now you can develop high-performance programmable DSP systems with Xilinx FPGAs using system design and verification tools from The MathWorks, Inc.

by Per Holmberg
Sr. Product Marketing Manager, Xilinx
per.holmberg@xilinx.com

Anne Mascarin
DSP Market Segment Manager, The MathWorks
amascarin@mathworks.com

The MathWorks and Xilinx have entered a strategic exclusive alliance and joint development agreement for the system-level creation of FPGA-based DSP designs. The first product will be the Xilinx System Generator for Simulink(tm) software. Used with The MathWorks' popular Simulink system-level design tools, and the Xilinx CORE Generator and LogiCORE DSP algorithms, this software is the first to bridge the gap between system-level DSP design and FPGA implementation, allowing you to easily design high-performance DSP applications in Xilinx FPGAs. The overall system flow is illustrated in Figure 1.

### System Overview

The Xilinx System Generator automatically generates hardware description language (HDL) code from a system representation model in Simulink. The HDL design can then be synthesized for implementation in Xilinx FPGAs using standard hardware synthesis software. To maximize predictability, density, and performance, the System Generator automatically maps blocks in the system design into optimized LogiCORE algorithms (cores). With only one representation of the design and no manual intervention when translating the system-level design to HDL, a common source of errors is removed.

You can significantly reduce development time by quickly iterating between the system-level model in Simulink and the hardware implementation. This is especially important for DSP applications, because many system-level design trade-offs are based upon the results of the hardware implementation. Consequently, your development time is significantly reduced, and moreover, even if you are new to FPGAs, you can use the familiar tools from The MathWorks, along with the Xilinx FPGA development tools, to create FPGA-based DSP applications. This combination not only gives you the ease-of-use and time-to-market advantages of FPGAs but also the highest possible performance.

### The System Generator

The System Generator consists of two components to help you proceed from a system model to actual hardware. The Xilinx Block Set (XBS) allows you to embed bit-true and cycle-true models of FPGA-specific circuitry into a Simulink design, while the System Generator translation software converts the Simulink model into synthesizable VHDL, with Xilinx FPGA hardware as the target.

The XBS provides these elements:

- Parametric blocks for DSP, arithmetic, and logic functions.
- Gateway blocks to communicate with the Simulink environment, where you have access to the extensive set of Simulink DSP libraries.*
- Special tokens to support user-defined black boxes and to invoke the System Generator interface to the Xilinx FPGA software.

The XBS provides the functionality of:

- Simulink S-functions - the Simulink representation of the XBS.
- Synthesizable VHDL - the hardware representation of the XBS, including the use of Xilinx cores where appropriate.

The System Generator software token activates the translation software that converts a Simulink model built from XBS elements into synthesizable VHDL. The VHDL generated may include cores for appropriate functions, as well as corresponding simulation models.

The System Generator graphical user interface (GUI) allows you to customize the Simulink simulation. For example, it can hierarchically override fixed-point values with doubles. This is particularly useful during design and debugging.

### Simulink

You can model a VHDL design using any combination of Simulink blocks. By using the XBS black box token, you can then instantiate the design into a generated VHDL model. The black box customization GUI encapsulates the design information necessary for the compiler to create the correct instantiation interfaces. This black box support allows the abstraction of commonly used control signals and ports, and then infers them within the generated VHDL.

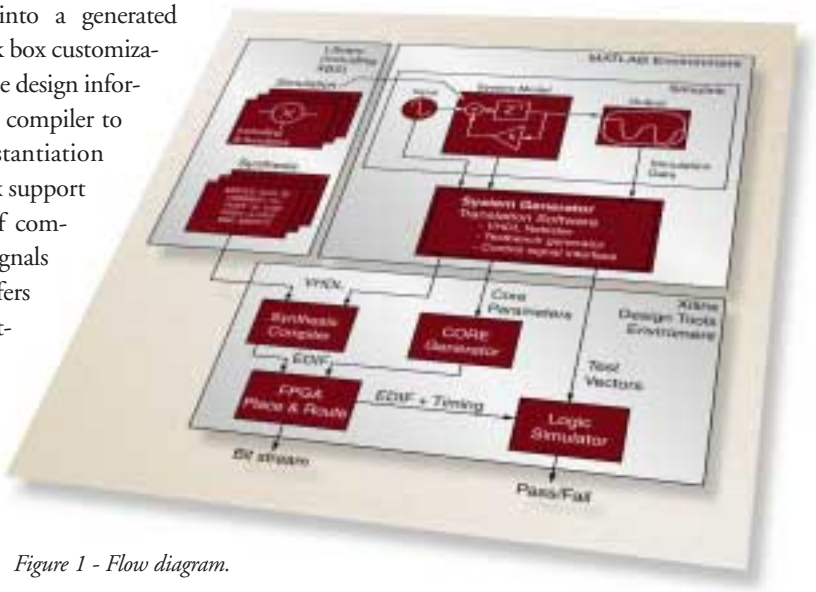



*Figure 1 - Flow diagram.*

Working in Simulink, you create a model of the hardware system as well as one or more test environments in which to simulate the model. When the model is first entered, simulation is typically performed using floating-point data to verify its theoretical performance. You then add the appropriate XBS elements, and then convert data types to the bit-true representations used in the hardware implementation. Then the model is re-simulated to verify its performance with quantized coefficient values and limited data bit widths, which can lead to overflow, saturation, and scaling problems. User-defined black boxes can also be incorporated in the modeling and elaboration process. When the model has been converted to a form realizable in the FPGA and its performance meets specification, you can invoke the netlist and the test bench generator.

### The Netlister and Mapper

The netlister extracts a hierarchical representation of the model structure annotated with all the element parameters and signal data types. A mapper then analyzes the elements in the hierarchy and creates a VHDL description of the design. Where possible, the mapper uses the Xilinx CORE Generator to generate optimized LogiCORE implementations for specific design elements.

When an element or its parameter values imply functionality unavailable in the CORE Generator, the mapper instantiates a reference to a parameterized, synthesizable entity in a synthesis library or your own supplied model.The actual hardware entities used have additional inputs and outputs for control signals that are not evident at the level of abstraction used in Simulink. The mapper inserts the necessary control ports and connects them to control logic blocks.



Multi-rate clocking is supported through time step information provided during simulation. Each control logic block is given a default synthesizable behavior which may require alteration to achieve an efficient implementation.

### The Testbench Generator

The testbench generator is an interactive tool that runs in the MATLAB environment, in which you capture the input stimuli and system outputs of selected simulation runs for conversion to test vectors. The generator converts the captured simulation data into VHDL code that will exercise the implemented model and test its outputs against the expected results.

### Logic Synthesis

The Xilinx Foundation Series, or any synthesis compiler supported through the Xilinx Alliance Series software tools, can be used to synthesize the control logic and elements for which no hardware macros exist, and combine all the pieces into a fully realized netlist. The outputs of this back-end process are a bit stream and an EDIF structural netlist of the hardware, annotated with timing information. This netlist can be simulated with the test vectors produced previously from system simulations to verify the performance of the completed FPGA hardware realization.

### Conclusion

With the introduction of the Xilinx System Generator for Simulink, you now have a tool that makes the job of incorporating FPGAs into your DSP designs easier than ever before. Today, Xilinx FPGAs provide you with the world's highest-performance programmable DSP solution, supporting applications equivalent to 160 billion MACs. As FPGA technology continues to advance, it is expected that by the year 2002, you will have access to a ten million gate FPGA offering 0.6 Tera MACS per second.

***See www.xilinx.com for more information.***


# The World's Highest-performance Programmable DSP Solution. A New Paradigm in High Performance Digital Signal Processing...

by Rufino Olay
Sr. DSP Product Marketing Engineer, Xilinx
rufino.olay@xilinx.com

Xilinx DSP consists of the Virtex and Spartan series FPGAs, a wide range of DSP algorithms, and a comprehensive set of software tools and prototyping boards. This is a complete DSP solution giving you the high performance and system integration of ASICs and ASSPs plus the reconfigurability and quick turnaround of standard processors. This ultimate combination provides a comprehensive and robust platform to help you create the highest performing reprogrammable signal processing applications imaginable.

We've created a complete design flow that guides you through the conceptual architectural design, verification, and implementation. Design techniques such as parallel processing and distributed arithmetic, coupled with industry-leading hardware platforms, increases sampling rates by an order of magnitude over that of traditional approaches; there is no faster DSP solution, anywhere. Table 1 shows a sample listing of algorithm benchmarks.

| Algorithm | Benchmark | Unit of Measure |
|---|---|---|
| 16-bit | 160,000,000,000 | MAC/s |
| 256 Tap FIR | 160 | MSPS |
| 1024 pt FFT | 41 | usec |
| Reed Solomon Decoder | 87 | MHz |
| JPEG Codec | 21 | MHz |
| ADPCM | 16 | MHz |

*Table 1 - Performance Matrix*

### Fast, Flexible, and Easy

With Xilinx DSP you can easily create customized architectures that give you the best speed and area utilization for your particular needs.

### Parallel Operations

Xilinx DSP helps you create the most robust DSP applications by exploiting the parallelism that is inherent in DSP mathematical models. Using the vast logic resources that are present in the Virtex FPGAs, you can create fully parallel structures that give you the utmost in computational power.

### Customizable Data Structures

Unlike fixed-width processors or ASICs, Xilinx FPGAs give you the freedom to create custom word lengths for your particular situations, including different parameters within the same design. For example some channels in your system might require more bits of precision than others; you can easily change the algorithm and the Xilinx DSP software will easily accommodate the new data configuration.

### Multiple Data Paths and Channels

Individual arithmetic logic can be linked to separate data paths or mutually coupled to run parallel operations during individual cycles. This approach is ideal for applications consisting of multiple subtasks that need little or no interdependency.

### Logical Operations

You can easily implement a variety of bit- and byte-wide operations such as barrel shifting, comparison, rotation, and accumulation by instantiating any of our numerous cores or by writing your own HDL code to customize a particular process.

### The MathWorks Simulink Integration

Xilinx and The MathWorks (the leader in DSP algorithm tools) have created a strategic alliance that allows you to build high performance DSP systems in Xilinx FPGAs using the system design and verification tools with which you are already familiar. The result is the Xilinx System Generator™ tool-set which bridges the gap between your conceptual architectural design and the actual translation and implementation of your FPGA-based DSP system.

With the Xilinx System Generator you can easily experiment with various DSP functions and quickly see the algorithmic tradeoffs between performance and silicon area.

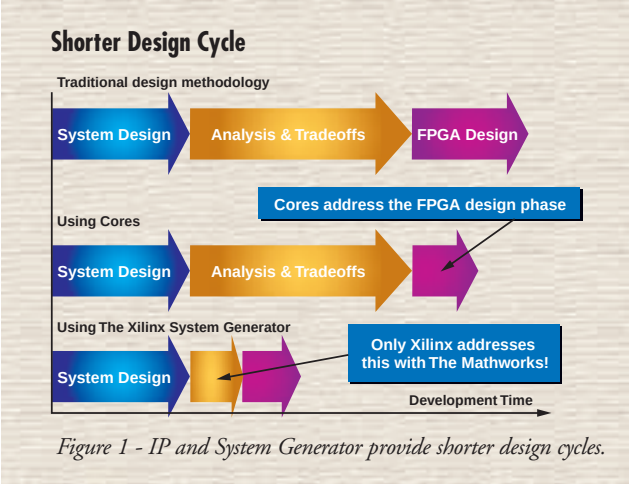



*Figure 1 - IP and System Generator provide shorter design cycles.*

Then you can easily compare the cost and speed with off-the-shelf DSP devices.

Key Features include:

- Seamless integration; no manual redesign is required.
- No risk of error introduction.
- Only one source code to maintain.
- Floating-point and fixed-point system simulation.
- Automatically generates HDL description for Xilinx FPGAs.

### Comprehensive IP Offering

Our extensive selection of IP, including filters, correlators, transforms, FFTs, FECs, integrators, DDS, and math functions, gives you the power to build large, complex designs quickly and effortlessly. Plus, our IP is optimized and parameterized for implementation in our Virtex and Spartan FPGAs so you get the most efficient and fastest implementation.

The intuitive GUI in our CORE Generator guides you through the various options to help you customize the IP for your specific design requirements. The features include:

- Scalable IP to fit your particular application.
- Millions of possible permutations.
- Minimal learning curve.

### New FIR Filter Generator

Xilinx recently announced the FIR Filter Generator, a new tool for creating fully optimized and parameterized algorithms for FPGA-based complex single-rate, half-band, Hilbert transform, and interpolated filter designs. Design techniques such as distributed arithmetic are employed to optimize filter structures for high-end DSP applications such as wireless and xDSL modems, medical imaging, and radar signal processing. The operational performance of the FIR filters, in the Virtex family, exceeds 160 billion multiply accumulates (MACs) per second.

The FIR Filter Generator allows you to choose from millions of parameter combinations to match your unique DSP design requirements, from fast, fully parallel systems to cost-effective designs optimized for lower sampling rates, as shown in Figure 2. The available parameters include:

- From 2 to 1024 taps.
- From 1 to 32 bit input data and coefficient precision.
- Signed or unsigned input data.
- Fully serial, parallel, or a combination of serial/parallel filter implementations with the ability for multi-clocking of output data.
- Time multiplexing of data for multiple channel structures.

### Smart-IP Technology

By employing the Xilinx Smart-IP Technology, the CORE Generator maintains constant performance over the entire range of FPGA densities; Smart IP gives you predictable timing and optimal an implementation for area and speed. This predictability, unique to Xilinx, is essential for incorporating entire systems on an FPGA.

### In-House Expertise

We've gathered together a highly knowledgeable team of DSP experts to create the best possible DSP tools and to provide you the best possible design expertise. In addition, Xilinx DSP FAEs have extensive design experience which gives them particular insight and knowledge to the challenges and intricacies of high performance DSP applications.

### Third-Party Programs

To help you further reduce your development time, Xilinx has partnered with various DSP development companies to provide a wide range of services, from algorithm development to full turnkey operation, giving you:

- A large and growing selection of system IP from our AllianceCORE partners.
- Access to the leading DSP experts in various application areas.
- Design services and expertise from the Xilinx XPERTS partners.
- A wide range of in-house and third-party design expertise and application experience.

In addition, numerous third-party vendors such as GV & Associates, Nallatech, and Lyr Signal Processing also offer prototyping hardware.

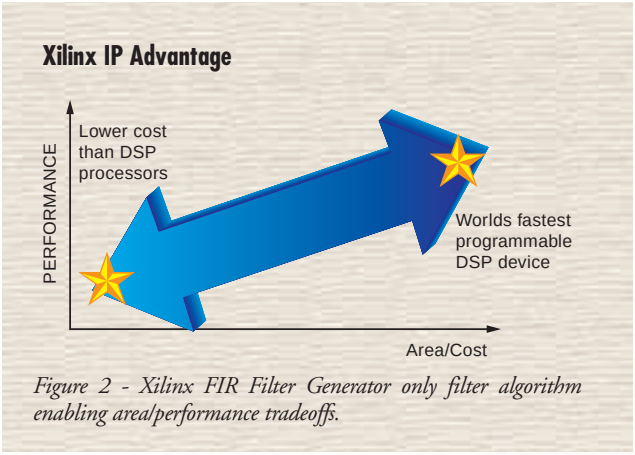



*Figure 2 - Xilinx FIR Filter Generator only filter algorithm enabling area/performance tradeoffs.*

### Conclusion

With the Xilinx DSP solution, you get faster DSP designs that are customized for your exact needs. Plus, there are no NRE charges or limitations to your creativity. There simply is no better way to create DSP designs.

*For more information see www.xilinx.com/dsp*


# IRL Makes Console Switching Faster and More Reliable.

by Robert Rowe
Product Manager, Apex
robert.rowe@apex.com

One of the most pressing concerns facing Information Systems Managers today is managing the volume and diversity of servers under their authority. As corporations increase the number of servers they deploy, the infrastructure needed to support this growth requires more efficient management of physical space, server information, and staff time.

Administrators have recognized the importance of reducing the amount of space occupied by computing and peripheral devices, and as a result, are implementing console switches to manage and operate multiple servers from a single console. This provides more space for servers, routers, hubs, and printers by reducing the quantity of keyboards, monitors, and mice.

Apex, the market leader in console switches, chose Xilinx to provide the FPGAs to create their next generation switch. By using Xilinx FPGAs and the Xilinx Internet Reconfigurable Logic (IRL) technologies, Apex could develop a console switch that can be upgraded with new services or functionality, in the field, without having to supply customers with new boards or EPROM chips. Not only does this provide convenience for their installed base of customers, it also provides flexibility for developing and rapidly deploying new services and for remotely reconfiguring data in the console switch.

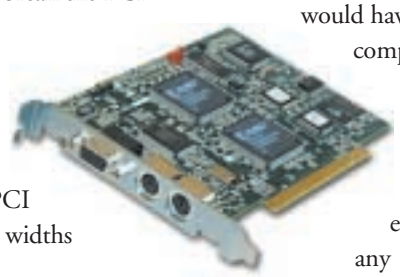
### Faster Time-to-Market

By populating the console switch with Xilinx FPGAs, Apex engineers developed 80% of the base solution to get the system up and running. Then, by the time the switches were deployed to customers, the remaining 20% of the code was completed and transmitted to the systems in the field. This gave Apex a lead in deploying the product to the field before the competition-a real time-to-market advantage and a key benefit for their customers.

### Flexibility and High Throughput

In most customer environments, the console switch is located in the same facility as the servers, but with the flexibility offered by Xilinx FPGAs, the console can be located anywhere in the world. With the Apex Emerge2000, for remote access server control, you can manage the graphical consoles of your server population from any location.

The basic architecture is based on the PCI bus (Figure 1). By using two Xilinx FPGAs in each Emerge2000, one for PCI communications that is loaded at power-up and remains loaded, and the other FPGA loaded dynamically at run time, Apex console switches never have to break the PCI communication. The Xilinx PCI core provides up to 33 MHz operation, is easy to use, and is easy to migrate to other PCI cores with higher bus widths if the need arises.



Xilinx also provides an easy migration path from low to high density devices allowing Apex to include more robust features in their console switch products as their requirements evolve.

### Rapid Prototype Development

By employing Xilinx FPGA technology, the Apex engineers could continue developing the design right up to the time of board delivery. Once the boards became available, the engineers could load the FPGA configuration at the work bench and verify the design. If they had used custom ASICs, it would have required much more time to complete the design. In addition, due to the ability to remotely upgrade the design over the Internet, the engineers were able to ship the product early, with the assurance that any new features could easily be added in the field.

**...DUE TO THE ABILITY TO REMOTELY UPGRADE THE DESIGN OVER THE INTERNET, THE ENGINEERS WERE ABLE TO SHIP THE PRODUCT EARLY, WITH THE ASSURANCE THAT ANY NEW FEATURES COULD EASILY BE ADDED IN THE FIELD.**

### Technical Support from Xilinx

Support from a supplier is a critical path item, often overlooked when designers view supplier technology. Obtaining the most up-to-date information, in a precise format, can become a critical factor when moving at the rapid pace our market demands. The Xilinx support organization is fast and responsive, and delivered the lat-

r-

)-

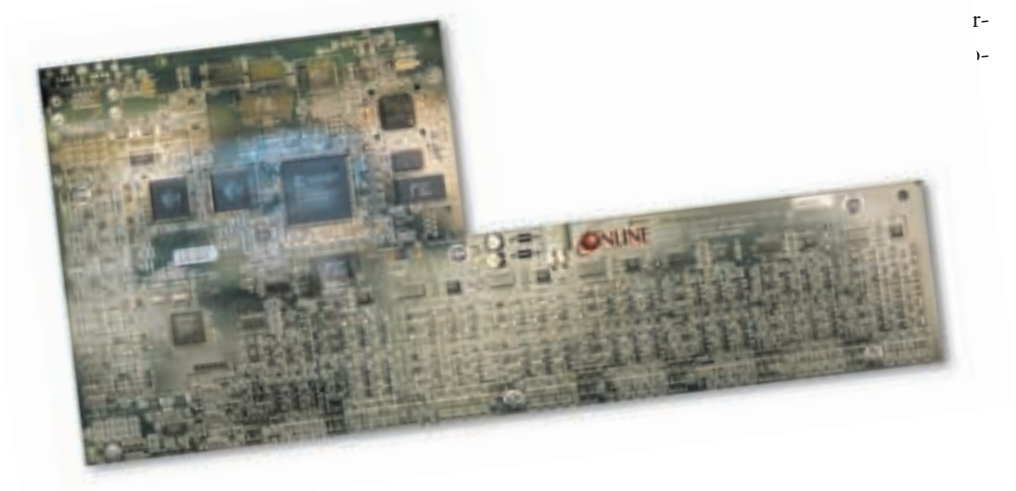


*Figure 1 - The Emerge2000 board*

### Conclusion

Xilinx FPGAs, along with the Xilinx IRL technologies, allows companies like Apex to build flexible, next-generation equipment that can be upgraded over the Internet. The time to market advantages are obvious and the unique design possibilities are limited only by your imagination.

## About Apex

**Apex designs, manufactures and markets stand-alone electronic switching systems and integrated server cabinet solutions for the client/server computing environment. Apex supplies stand-alone switching systems to some of the largest PC manufacturers in the industry for integration into their product offerings, providing switching systems to OEMs (Compaq, Dell, IBM, and Hewlett-Packard) representing 43 percent of all PC servers and 66 percent of all "super" servers shipped worldwide.Go to http://www.apex.com for more information.**

# LavaCORE - A Configurable Java Processor

Create Java-enabled appliances, mobile devices, secure Internet devices, and embedded network computers that can be dynamically configured for application-specific computing.

by Bhaskar Bose, Ph.D. and M. Esen Tuna
President, Vice President - R&D, Derivation Systems, Inc.
bose@derivation.com, mtuna@derivation.com

LavaCORE™ is a 32-bit configurable Java™ processor core targeted to the Xilinx Virtex FPGA architecture. The processor executes Java bytecode directly in hardware eliminating the need for software-based interpreters or code translators. LavaCORE is a clean-room implementation of the Java Virtual Machine and is provided as a synthesizable "soft-core" with a suite of tools for parameterized core generation, hardware/software co-design, co-verification, and custom Java application development.

LavaCORE is a revolutionary product that provides unparalleled flexibility for a wide range of prototyping and embedded applications. Direct Java bytecode execution keeps the runtime to a minimum making the processor ideal for embedded applications that require small memory foot-prints and flexible reconfigurable architectures. Both the application and the operating system code can be developed in Java and compiled to native code by standard Java compilers.

LavaCORE is designed to bring embedded Java technology to the reconfigurable marketplace. The customizable core and dynamic reconfiguration take full advantage of the features of programmable hardware.

## LavaCORE Features

The processor core features a 32-bit architecture with 32-bit address and data paths. Our reference design is implemented in a Virtex XCV300 BG352-4 FPGA. Figure 1 shows a block diagram of the architecture.

Some of the features are highlighted below:

- 32-bit direct execution Java processor.
- Executes Java Virtual Machine bytecode in hardware.
- Stand-alone (system on a chip) or core configuration.

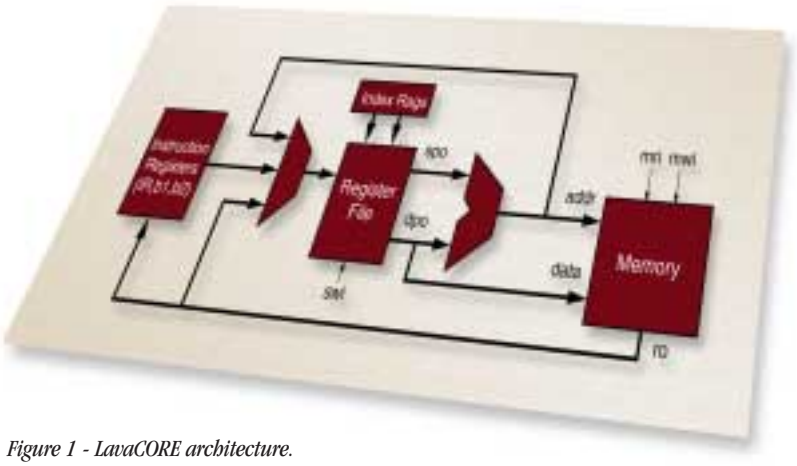



*Figure 1 - LavaCORE architecture.*

A 16x32-bit dual-ported RAM implements the register-file. The instruction register is composed of three 8-bit registers denoting the instruction, byte one, and byte two of the instruction stream. A 32-bit ALU computes arithmetic and logical operations. Additional signals include clock, reset, signal interrupt, memory interface, and a set of observability pins for the state and flags.

- Built-in hardware encryption module.
- Fully synthesizable FPGA core.
- Parameterized Core Generator automatically synthesizes VHDL, Verilog, or EDIF gate-level netlists.
- Software support includes Parameterized Core Generator, LavaOS Runtime Environment, Co-Simulation/Verification Tools, and Hardware Debugger.

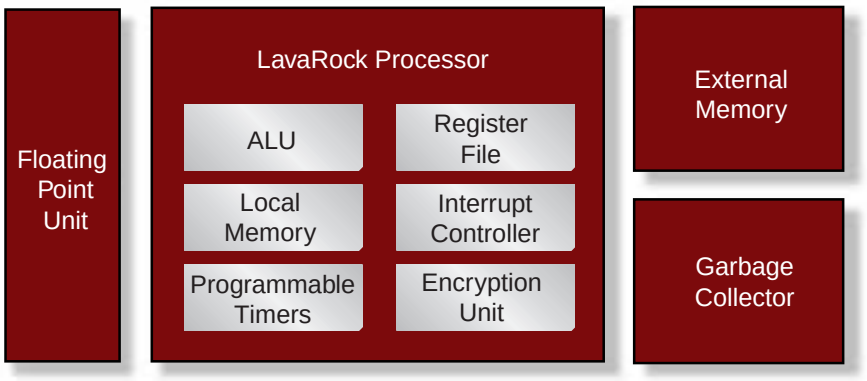



*Figure 2 - LavaCORE system architecture.*

### Parameterized Core Generator

The LavaCORE Parameterized Core Generator operates via an intuitive graphical interface and includes user selectable parameters for a wide range of configurable options. Every implementation of LavaCORE can be application specific, leading to optimal solutions in terms of power, speed, and area.

Typically, specialized embedded applications use only a subset of the full JVM instruction set. By analyzing the application, you can determine which Java bytecode instructions can be omitted or moved from hardware to software, improving various cost criteria.

Additional options allow you to include functional components of the synthesized architecture such as the built-in encryption engine, programmable timers, and interrupt controllers. For larger systems, floating point and external memory interfaces are defined (Figure 2).

Once parameterized options are selected, the LavaCore Generator automatically synthesizes a gate-level implementation in either VHDL, Verilog, or EDIF netlist formats. Along with this softcore, an HDL testbench and a customized runtime called LavaOS™ are generated.

The synthesized core can then be directly input to Xilinx Foundation Series or Alliance Series software for place and route. The HDL testbench is used both to test the hardcore and softcore. Figure 1 depicts the dataflow architecture of a LavaCore instance.

### Linker/Application Builder

Standard Java-class files, generated from third party commercial Java development environments, are statically resolved to build executables. In addition, the linker builds the boot tables, class initialization codes, and assigns the interrupt and trap handlers.

The executable image incorporates the runtime environment. It is designed for embedded applications and is small enough to be implemented in the internal block RAM. In a standalone configuration, the LavaCORE system architecture can incorporate the entire runtime environment and a Java application within the Virtex block RAM.

### Application Debugging Tools

The LavaCore application debugging tool set consists of a core simulator and a hardware interface (Figure 3). The simulator has a built-in debugger that features single stepping, memory and register file monitors, and conditional break points. The simulator and the debugger share a graphical user interface to display and debug the LavaCore execution both at instruction and micro steps.

The LavaCore simulation environment consists of three separate components that validate all three aspects of the LavaCore:

- The instruction tester.
- The application simulator.
- The hardware testbench for the synthesized core.

The hardware interface provides an implementation debugging bridge for the synthesized hardware. The same debugging environment and user interface are used for the testing of the target hardware.

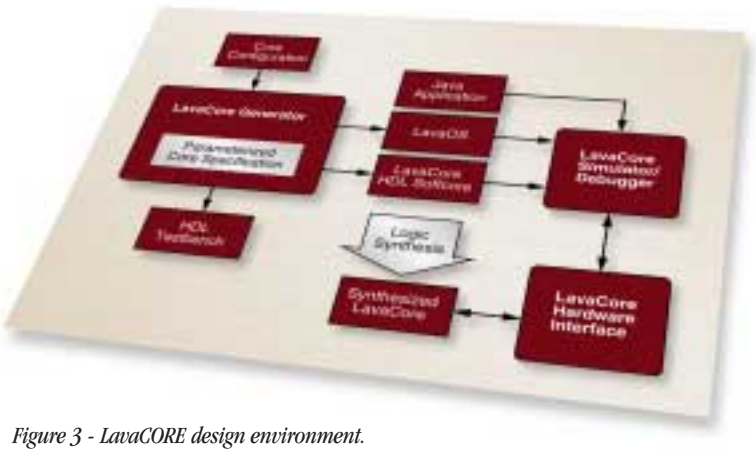



*Figure 3 - LavaCORE design environment.*

### Conclusion

The LavaCORE Configurable Java Processor core enables the deployment of Java technology for a new generation of embedded reconfigurable systems. Native execution of Java bytecode provides reliable hardware execution, denser code, and minimum runtime environment. The LavaCORE parameterized core generator allows you to configure the capabilities of the processor core and synthesize an optimal LavaCORE for your application.

LavaCORE provides a fast design solution for embedded Java processors targeted to programmable hardware. With our configurable Java processor core and co-design/verification environment this unique IP product will reduce design time leading to faster time to market.

For more information about Derivation Systems, or the LavaCORE products, see www.derivation.com

# Inferring Read Only Memory in FPGA Compiler II and FPGA Express

by Alan Ma
Senior Corporate Applications Engineer, Synopsys, Inc.
alanma@synopsys.com

Prior to the recent advancements in FPGA technology, you had to rely on external RAM or ROM. Now, with the introduction of million-gate FPGAs such as the Xilinx Virtex devices, you have access to abundant on-chip memory resources. FPGA Compiler II/FPGA Express (FCII/FE) takes advantage of Virtex resources such as Look-Up Tables (LUTs) MUXF5s, MUXF6s, and on-chip block SelectRAM to provide the highest quality of results for ROM functions.

### Coding Styles

Version 3.4 of FCII/FE recognizes a ROM description using CASE statements in both Verilog and VHDL, and by using constant arrays in VHDL.

### Using CASE Statements

FCII/FE infers ROM when the inputs to the CASE statement are constant and all the states are specified. Figure 1 shows an example of an 8x4 ROM in Verilog. Figure 2 illustrates its VHDL equivalent.

### Using Constant Arrays

You also have the option of using constant arrays in VHDL. Figure 3 describes the coding style for the same 8x4 ROM where CONV_INTEGER is a built-in function that converts std_logic_vector to integer.

### General Implementations

FCII/FE generally implements ROM using LUTs, MUXF5s, and MUXF6s when targeting Xilinx Virtex devices. The actual resources used are closely related to the width of the address port. If the address has less than or equal to four bits, then the ROM will be implemented using LUTs. If the address has five bits, then MUXF5s, (which provide multiplexer functions in one half of a Virtex Configurable Logic Block), will be used in addition to the LUTs. If the address has between six to nine bits, the ROM will be implemented using LUTs, MUXF5s, and MUXF6s which provide multiplexer functions in a

```
module rom_8x4          (
                        address,
                        rom_out
                        );

input   [2:0]           address;
output  [3:0]           rom_out;

reg     [3:0]           rom_out;

always @(address)
        case (address)
                3'b000  : rom_out <= 4'b0001;
                3'b001  : rom_out <= 4'b0010;
                3'b010  : rom_out <= 4'b0100;
                3'b011  : rom_out <= 4'b1000;
                3'b100  : rom_out <= 4'b1000;
                3'b101  : rom_out <= 4'b0100;
                3'b110  : rom_out <= 4'b0010;
                3'b111  : rom_out <= 4'b0001;
        endcase
endmodule
```

*Figure 1 - Using CASE statements for ROM in Verilog.*

```
library ieee;
use ieee.std_logic_1164.all;

entity rom_8x4 is port  (
                        address: in std_logic_vector(2 downto 0);
                        output: out std_logic_vector(3 downto 0)
                        );
end rom_8x4;

architecture rtl of rom_8x4 is
begin
        process (address)
        begin
                case address is
                        when "000"      => output <= "0001";
                        when "001"      => output <= "0010";
                        when "010"      => output <= "0100";
                        when "011"      => output <= "1000";
                        when "100"      => output <= "1000";
                        when "101"      => output <= "0100";
                        when "110"      => output <= "0010";
                        when "111"      => output <= "0001";
                        when others     => output <= "0000";
                end case;
        end process;
end rtl;
```

*Figure 2 - Using CASE statements for ROM in VHDL.*

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity rom_8x4 is port  (
                        address: in std_logic_vector(2 downto 0);
                        output: out std_logic_vector(3 downto 0)
                        );
end rom_8x4;

architecture rtl of rom_8x4 is

constant num_word       : integer := 8;
constant num_bit        : integer := 4;
signal int_add          : integer range 0 to 7;

type ROMARRAY is array (0 to (num_word-1)) of
                std_logic_vector((num_bit-1) downto 0);

constant ROM: ROMARRAY := (
        "0001",
        "0010",
        "01                             00",
                                "1000",
                                "1000",
                                "0100",
                                "0010",
                                "0001"
                                );
begin
        int_add <= CONV_INTEGER(address);
        output <= ROM (int_add);

end rtl;
```

*Figure 3 - Using constant arrays for ROM in VHDL.*

full Virtex CLB. When the address has ten or more bits, FCII/FE implements ROM using on-chip block SelectRAM resources if certain conditions are met.

### Using Block SelectRAM

The Virtex series provides dedicated blocks of on-chip, dual port synchronous RAM, with 4096 memory cells (bits). These resources can also be used for ROM if certain conditions are met. Our research indicates that when the address of the ROM has ten bits and the data has more than or equal to three bits, quality of results can be improved by mapping the ROM to block SelectRAM if the output of the ROM is registered. However, if the data has less than or equal to two bits (when the address has ten bits), using LUTs, MUXF5s, and MUXF6s, as described in the previous section, yields better results. For ROM whose address has more than or equal to eleven bits, block SelectRAM will always be used if the output is registered. Table 1 summarizes the conditions for ROM inference.

To reserve block SelectRAM for user-defined functions, you can use the scripting command "set_chip_instantiated_blockram". For example, the following reserves 4096 bits of block SelectRAM for inferred ROM:

```
set_chip_instantiated_blockram 4096
```

While set_chip_instantiated_blockram applies to the current project, you can use the variable proj_user_instantiated_blockram to reserve block SelectRAM globally for all the subsequent projects. For example, the following reserves 4096 bits of block SelectRAM for all future projects:

```
proj_user_instantiated_blockram 4096
```

### Conclusion

Inferring ROM is easy with FPGA Compiler II and FPGA Express, which take full advantage of the abundant on-chip memory resources of the Xilinx Virtex devices.

Visit the Synopsys FPGA website at www.synopsys.com/fpga for other information on the latest FPGA synthesis technologies.

| | | LUT | MUXF5 | MUXF6 | SelectRAM |
|---|---|---|---|---|---|
| ADD <= 4 Bits | Any Data Width | ◆ | | | |
| ADD = 5 Bits | Any Data Width | ◆ | ◆ | | |
| 6 Bits <= ADD <= 9 Bits | Any Data Width | ◆ | ◆ | ◆ | |
| ADD = 10 Bits | 1 Bit <= DATA <= 2 Bits | ◆ | ◆ | ◆ | |
| ADD = 10 Bits* | DATA >= 3 Bits* | | | | ◆ |
| ADD >= 11 Bits* | Any Data Width* | | | | ◆ |

*Table 1 - LUTs, MUXF5s, MUXF6s, and block SelectRAM utilization.*

* Note that LUTs, MUXF5s, and MUXF6s will be used if the output of the ROM is not registered.

# Using Block-Level Incremental Synthesis in FPGA Compiler II and FPGA Express

## Block-Level Incremental Synthesis allows you to modify a subset of a design and then re-synthesize just the modified subset.

by Alan Ma
Senior Corporate Applications Engineer, Synopsys, Inc.
alanma@synopsys.com

Advancements in the capacity and speed of FPGAs have made this technology increasingly attractive for million-gate designs. As the complexity of the designs grows so does the need for advanced synthesis and place-and-route tools. One of the highly sought features is the ability to recompile only the modified portion of a design after modifications have been made. Such a feature is needed not only to reduce compilation time, but also to preserve the timing behavior of certain sections of a design when other sections of the design are changed.

To address these requirements, FPGA Compiler II and FPGA Express (FCII/FE) version 3.4 introduce Block-Level Incremental Synthesis (BLIS) to allow you to modify a subset of your design and then re-synthesize just the modified subset. A design can be divided into "blocks," the smallest subset to which BLIS can be applied. FCII/FE generates an optimized netlist for each block, which does not change unless the design associated with that block has been modified. The netlist of each block is then presented individually to the place-and-route tool which is capable of recompiling only the modified netlists. Not only does this increase the likelihood of preserving post place-and-route timing behavior for the unmodified blocks, overall compilation time for both synthesis and place-and-route is significantly reduced.

### Identifying Blocks

A block is the smallest subset to which BLIS can be applied. It can be a Verilog module, a VHDL entity, an EDIF netlist, or a combination of these, as long as they form a tree within the design hierarchy. The top-level module/entity/netlist of this tree is, by definition, the block root. The components of a block include the block root and all modules/entities/netlists in its tree of the design hierarchy that do not include another block root.

For example, the top-level design "TOP" in Figure 1 has two subdesigns A and B. A instantiates C and D. B instantiates E and F. By definition the top-level design TOP is a block root. If you also decide to designate A and E as block roots, then the entire design will have three blocks:

- Block 1: TOP, B, F
- Block 2: A, C, D
- Block 3: E

Any modifications to any module/entity/netlist contained in a block cause the entire block to be re-synthesized. For example, if F is modified then every member of Block 1 (TOP, B, F) will be re-synthesized even though TOP and B have not been changed.

Blocks can be identified in both the Graphical User Interface (GUI) and the shell of FCII/FE. In the GUI, right-click an elaborated implementation in the Chips window and select Edit Constraints. Select the Modules

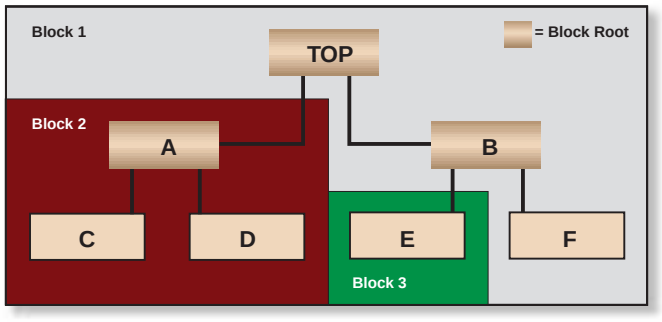



*Figure 1 - Example of blocks and block roots*

tab to display the Modules Constraint Table. You can then specify any subdesigns as block roots in the Block Partition column. To remove a block root designation, click the particular cell and select Remove. Note that the top-level design is always identified as the

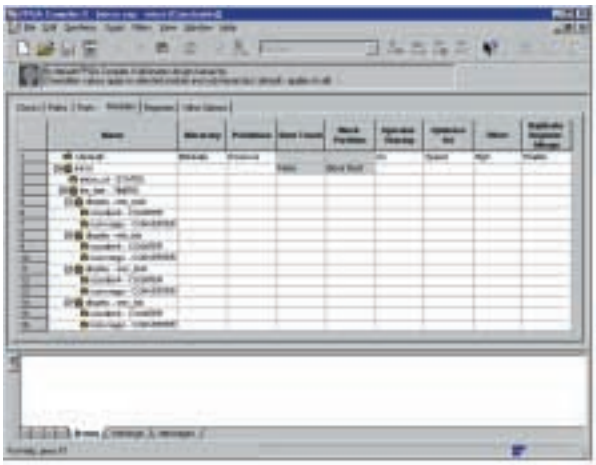


*Figure 2 - Modules Constraint Table*

block root by definition and it cannot be removed. Figure 2 shows the Modules Constraint Table.

In the shell, use the command set_module_block followed by the option "true" and the path to the module/entity/netlist to be specified as the block root. For example:

```
fc2_shell/fe_shell> set_module_block true /TOP/A
```

In the interactive mode, you can remove any block root designations by using the "false" option. For example:

```
fc2_shell/fe_shell> set_module_block false /TOP/A
```

In the batch mode, the same can be achieved simply by removing the set_module_block commands associated with the block roots to be removed in the fc2_shell or fe_shell scripts.

Please refer to the man page of set_module_block for additional usage and syntax information. To access the man page:

```
fc2_shell/fe_shell> man set_module_block
```

Block roots can only be designated on user-created modules/entities/netlists. For example, attempting to modify the setting on a primitive or the top-level design will result in the following error message:

```
Error: Cannot set block option on module 'AND'
```

Furthermore, the concept of block and block root only applies when the target architecture supports BLIS. Attempting to apply this feature on an architecture which is not supported by BLIS will result in the following error message:

```
Error: block assignments are not supported for the target technology of this chip
```

BLIS is currently available for the Xilinx Virtex architecture.

### Design Planning

The advancement in synthesis and place-and-route technologies certainly plays an important role in Quality of Results (QoR) but thorough design planning can never be replaced. In order to take full advantage of BLIS it is important to understand that:

- FCII/FE uses a time stamp to determine if an implementation is out-of-date. If the time stamp of any of the analyzed design source files is newer than the elaborated implementation then the elaborated implementation needs to be updated.
- A block is the smallest subset to which BLIS can be applied and any changes in any member of a block cause the entire block to be re-synthesized.
- A block represents "hard" boundaries which FCII/FE does not optimize across.

Therefore it is recommended that each module/entity/netlist in a design be described in its own design source file. Doing so ensures that modifying a module/entity/netlist will not affect the time stamp of other modules/entities/netlists which could potentially be members of other blocks.

In Figure 3, A and B are described in the same design source file. Modifying A not only causes the entire Block 2 to be re-synthesized, it also affects the time stamp of B for being in the same design source file as A. The newer time stamp of B causes the entire Block 1 to be re-synthesized. Needless to say, this is not a desired behavior.

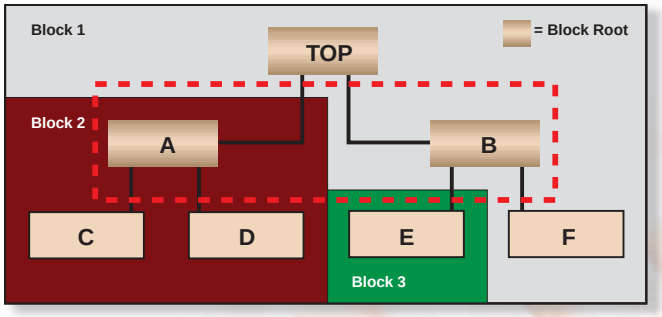



*Figure 3 - Example of design planning*

It is obvious that BLIS is not useful in the extreme case where all modules/entities/netlists are described in a single design source file.

Since FCII/FE does not incrementally optimize across block boundaries it is important not to break combinatorial logic into different blocks. For best QoR, it is recommended to observe conventional Register Transfer Level (RTL) coding style when partitioning designs. This involves grouping related combinatorial logic within a module/entity and registering all outputs of such a module/entity.

Note that BLIS is most effective when modifications are done within a module/entity. Changing the partition or the pins of the modules/entities of the design causes the entire design to be re-synthesized. This is why thorough planning in the early stage of the design process is vital to success.

### Implementation Update

This section describes what needs to be done in the GUI and the shell to update an implementation after a design has been modified. It is assumed that blocks have been defined as described previously, an optimized implementation has been created, and place-and-route netlists have also been generated. For those who are not familiar with the synthesis process of FCII/FE, please refer to the *FPGA Compiler II/FPGA Express Getting Started* manual for step-by-step instructions.

In the GUI, a red question mark next to a design source file in the Design Sources

window indicates that the file has been modified. Right-clicking the modified file and selecting Update File reanalyzes the file for syntax errors.

Similarly, a red question mark next to an elaborated implementation in the Chips window indicates that the implementation is out-of-date with respect to its design source files. Right-clicking the elaborated implementation and selecting Update Chip re-elaborates the implementation.

Once the design has been re-elaborated, right-clicking the optimized implementation and selecting Update Chip re-optimizes the design. If blocks are properly defined as described previously, BLIS will be automatically enabled. In this situation only the blocks that are associated with modified files are re-optimized. Note that there has to be at least two blocks defined to enable BLIS.

Instead of individually updating the design source files, the elaborated implementations, and the optimized implementations, you can update the entire project in one step by right-clicking the project icon in the Design Sources window and selecting Update Project. It is important to understand that FCII/FE generates one netlist in EDIF format for each block defined. The names of the netlists are the same as the module/entity/netlist name of the respective block roots. Performing an Update Project ensures that only the netlists associated with modified blocks are regenerated. Note that explicitly choosing Export Netlist forces all netlists to be regenerated. This step is therefore only recommended when the design is synthesized for the first time.

If Export Timing Specifications is selected in the Export Netlist dialog box, FCII/FE generates a Synopsys Constraint File (.scf) to pass timing constraints entered in FCII/FE to the place-and-route tools. Since the Xilinx place-and-route tool does not read .scf directly, the information in this file must be transferred to a User Constraint File (.ucf) for timing constraints to be considered during place-and-route.

Figure 4 shows a sequence of equivalent fc2_shell/fe_shell commands to be used in the interactive mode. Please refer to the man pages for complete usage and syntax information.

### Limitations

As mentioned previously, time-stamping is used to determine if an implementation is out-of-date. FCII/FE processes time stamps in whole seconds. If both analysis and elaboration finish within one second, then the analyzed design source files and the elaborated implementation will have the same time stamp. When the time stamp of the elaborated implementation is older than or equal to the analyzed design source files, the existing elaborated implementation is discarded and re-elaborated upon Update Chip or Update Project. The new elaborated implementation will then have a newer time stamp than the existing optimized implementation. This causes the existing optimized implementation to be discarded and re-optimized when actually none of the design source files have been modified. Because designs in general take longer than one second to be analyzed and elaborated, this limitation should not present any problems.

Note that BLIS is driven only by the difference in time stamps, the existence of at least two defined blocks, and any change of block roots. Other operations, such as modifying timing constraints, do not cause block-level incremental re-synthesis of the implementation.

### Conclusion

FPGA Compiler II and FPGA Express version 3.4 introduce Block-Level Incremental Synthesis allowing you to modify a subset of a design and then re-synthesize just the modified subset. They generate an optimized netlist for each block and the netlist of a block does not change unless the design associated with that block has been modified.

The place-and-route tools that support block-level incremental place-and-route are able to recognize and recompile only the netlists that have been changed. Not only does this increase the likelihood of preserving post place-and-route timing behavior for the unmodified blocks, overall compilation time for both synthesis and place-and-route can be significantly reduced.

*Please visit http://www.synopsys.com/products/fpga/ for the latest in FPGA synthesis technology.*

```
create_project TOP

add_file -library WORK -format VHDL c:/designs/top/TOP.vhd
add_file -library WORK -format VHDL c:/designs/top/A.vhd
add_file -library WORK -format VHDL c:/designs/top/B.vhd
add_file -library WORK -format VHDL c:/designs/top/C.vhd
add_file -library WORK -format VHDL c:/designs/top/D.vhd
add_file -library WORK -format VHDL c:/designs/top/E.vhd
add_file -library WORK -format VHDL c:/designs/top/F.vhd

analyze_file

create_chip -name TOP -target VIRTEX -device V800FG680 -speed -6 -frequency 50 TOP

current_chip TOP

set_module_block true /TOP/A
set_module_block true /TOP/B/E

optimize_chip -name TOP-Optimized

export_chip -dir .

# design has been modified

update_project

list_message
```

*Figure 4 - fc2_shell/fe_shell commands*

# Lucent Technologies Gets a Time-to-market Advantage

When Lucent Technologies set out to design the Cajun P880 Routing Switch, designers quickly determined that a traditional ASIC design would require too much time.

by Tamara Snowden
Corporate PR Manager, Xilinx
tamaras@xilinx.com

The Cajun P880 Routing Switch is the newest member of Lucent's next-generation, enterprise data networking solutions that provide network managers with an easy and flexible way to optimize their network designs; it had an extremely aggressive development schedule.

"We originally designed the P880 with an ASIC at the heart of the switching fabric," said Brian Ramelson, design manager at Lucent. "We soon realized that if we stuck with an ASIC design, the product would be late to market." Ramelson had recently attended a Xilinx Virtex series presentation and decided to contact Xilinx personnel to find out more.

As a result, a close working relationship developed between Ramelson and John DePapp, Xilinx field applications engineer, over the next several weeks. The two spent over 100 hours working together on the project. "The design process went very smoothly," said Ramelson "the support provided by DePapp was second-to-none."

### Virtex FPGAs - a Test Case

Ramelson's team selected the Virtex XCV150 and XCV800 FPGAs to supply the functions they needed. "At the time we were skeptical that any FPGA could implement these functions," said Ramelson. "We decided to try Xilinx FPGAs for this project as a test case for future designs." The Virtex series devices range from 50,000 to 1,000,000 system gates at clock speeds up to 200 Mhz and include many new features that address system-level design challenges. Fully supported by the Alliance Series software, the Virtex family offered a complete solution for Lucent, ready to meet the design challenges for their groundbreaking product.

> "By going with Xilinx we were able to ship the P880 months earlier than we would have with a traditional ASIC design"

### The Software Solution

Lucent's design team chose Synopsys FPGA Express for synthesis, VCS for a Verilog simulator, and Xilinx Alliance Series tools for place and route. Ramelson especially appreciated the ability of the Alliance Series software to perform multi-pass timing-driven place-and-route. "If a layout initially failed to meet timing requirements, we let the system work on the problem," said Ramelson. "In one case we let the software perform about twenty successive iterations over a weekend resulting in two or three workable layouts."

The Alliance Series provides the flexibility to select the best EDA design environment for a specific application. Combining the advanced implementation technology of Xilinx with the strengths of its partners provides a powerful overall design solution, the highest clock performance and the highest densities in the industry.

### The Cajun P880 Switch

The 17-slot Cajun P880 is a major step up from Lucent's P550 7-slot switch, although the two share a common architecture. The P880 can use any of the company's existing 50-series boards for Ethernet, Fast Ethernet, or Gigabit Ethernet. This provides net managers an obvious migration path. By mid-2000 even more capable 80-series boards will become available and can be deployed beside the 50-series boards in the same P880 chassis.

The P880's backplane scales upward from 56 Gigabits per second to 139 Gigabits per second providing the P880 with the capability of switching or routing from 41 to 106 million packets per send. Designed with no single point of failure, the highly reliable Cajun P880 supports up to 768 10/100 megabits per second Ethernet ports, up to 384 fiber Fast Ethernet ports, and up to 128 Gigabit Ethernet ports in a fault-tolerant, modular chassis.

The P880's switching fabric consists of six switch elements, each using two XCV150 devices for a total of 12 per chassis. The controller used a single XCV800 device. The design allows for high redundancy to maximize reliability. A second backup controller is optional with the P880, while a seventh switch element can be added for redundancy. Ramelson estimates a typical place and route for the controller took just about two hours.

### Conclusion

Ramelson notes this was a far different experience from the last time he worked with an FPGA. "The element that best represents the success of this program was how fast we were able to complete the design. End to end the design process was completed within six weeks—we had a working model in about a month."

Because of the breadth of its capabilities, the P880 spans both the "internetwork" and "wiring-closet" market sectors. This makes the Lucent switch a potent competitor. But specs mean little unless you can deliver product on time and that was where Xilinx FPGAs entered the picture. "By going with Xilinx we were able to ship the P880 months earlier than we would have with a traditional ASIC design," concluded Ramelson.

*More information on the Cajun P880 can be found at http://www.lucent.com/ins/products/p880/*

# New HDLC and ADPCM Cores

## A strategic partnership with ISS produces advanced communications products.

by Katie DaCosta
IP Product Marketing, Xilinx
katie.dacosta@xilinx.com

To help our customers reduce design cycles and get products to market quickly, Xilinx and ISS (Integrated Silicon Systems, Ltd.) have entered into a strategic development and OEM agreement to provide versions of ISS multimedia and communications cores optimized for Xilinx FPGAs. The IP cores will be available for purchase from Xilinx as LogiCORE products, downloadable from the Xilinx IP Center, and optimized for Xilinx architectures and design tools. The cores in this program include Single-channel HDLC, 32-channel HDLC, and 32-channel ADPCM, and are available now.

### A Strategic Alliance

The deliverables of this agreement are optimized and supported cores from ISS's extensive IP portfolio for the latest Xilinx devices including Virtex, Virtex-E, and Spartan-II FPGAs, as well as future FPGA families. Xilinx will deliver these LogiCORE products using the Xilinx CORE Generator tool to smoothly integrate the cores into the Xilinx design flow.

"This agreement broadens our long-term and successful relationship with Xilinx," said James G. Doherty, CEO of ISS. "The strength of the Xilinx Virtex architecture coupled with their commitment to partnerships provides a perfect environment to expand the usage of ISS cores in wireless and wired communications, and digital video and imaging applications. With this agreement, Xilinx can now fully utilize our expertise in multimedia communications to provide the highest-performance, off-the-shelf design solutions available in the market."

"ISS has achieved industry-wide recognition for solving the needs of the communications and multimedia user community," said Mark Aaldering, senior director for the IP Solutions Division at Xilinx. "By partnering with ISS, Xilinx is able to address the demand for a one-stop source of high-quality, cost-effective, optimized FPGA cores. This agreement expands the successful long-term relationship between ISS and Xilinx, which started with membership in the Xilinx

AllianceCORE third party IP development program and the co-development partnership for Reed-Solomon Forward Error Correction cores. The agreement reinforces the Xilinx commitment to providing leading-edge IP for multi-million gate FPGA designs."

### The New Cores

The first three cores to result from the Xilinx-ISS agreement are focused on accelerating the design cycle of high-density FPGAs in communications applications.

"The strength of the Xilinx Virtex architecture, coupled with their commitment to partnerships, provides a perfect environment to expand the usage of ISS cores in wireless and wired communications, and digital video and imaging applications."

The HDLC (High-level Data Link Control) protocol controller cores and the ADPCM (Adaptive Differential Pulse Code Modulation) codec core are used in a number of telecom applications ranging from internet routers and switches to VoIP (Voice over Internet Protocol) gateways.

"The Xilinx HDLC and ADPCM LogiCORE products address the exploding demand from telecom and network developers for complex Virtex-based IP for data and voice processing," said Babak Hedayati, Director of Marketing and Business Development at Xilinx IP Solutions Division. "The HDLC protocol is proven as one of the most popular methods of transmitting data over WAN systems. The availability of these cores on the Xilinx IP Center provides a one stop source to integrate telecom IP solutions into Xilinx FPGAs."

### HDLC Cores

The HDLC cores conform to the ITU Q.921 and X.25 recommendations for full duplex, point-to-point and multi-point operation. The cores function at data rates over 40 Mbps and include a direct connection to pulse code modulation (PCM) networks. The HDLC cores are ideal for public and private packet switched data networks such as Frame Relay switches, cable modems, Broadband ISDN, T1/E1, T3/E3, Packet-based DSL Access Multiplexers (DSLAMs), Remote/multi-service access concentrators, and Sonet networks.

### ADPCM Cores

The 32 Channel ADPCM speech codec performs the ITU G.726 conversion of 64 kbit/s A-law or μ-law PCM channels to and from 40-, 32-, 24- or 16-Kbit channels using the Adaptive Differential Pulse Code Modulation transcoding technique. The core supports up to 32 duplex encoding/decoding channels or up to 64 encoding and/or decoding channels, and can operate in burst or continuous modes. It is on-line configurable for A-law or μ-law PCM encoding as well as various ADPCM compression rates including G.721 or G.723 mode operation. The ADPCM codec will be used in applications such as CO DSLAMs, satellite communications, digital audio storage, H.323 VoIP gateways, access servers, and computer telephony and cellular networks where high quality voice compression is important.

### Availability

The HDLC and ADPCM cores are available now. Suggested resale pricing is $7,200 for the 32 Channel HDLC controller core, $3,900 for the Single Channel HDLC controller core, and $14,400 for the 32 Channel ADPCM codec core. Licensing information and instructions for downloading the cores, and information on all Xilinx LogiCORE products can be found at the Xilinx IP Center at: www.xilinx.com/ipcenter. Information on other Xilinx LogiCORE and third-party AllianceCORE products is also available from the IP Center.

### Conclusion

The strategic partnership between Xilinx and ISS is already producing optimized cores for the communications market, and many more are on the way. Xilinx continues to provide the most advanced FPGA and IP technologies to help you create the next generation of high performance equipment and get your products to market as soon as possible.

## About ISS

**Integrated Silicon Systems Ltd. (ISS) is the leading supplier of application-specific virtual components (ASVCs) for multimedia and communications FPGA, ASIC, and System-on-a-Chip (SoC) integrated circuits. ISS delivers video/image compression, audio compression, and channel coding solutions for consumer and communications applications. Using proprietary techniques for direct-mapped implementations of digital signal processing functions and algorithms in hardware, ISS delivers solutions that realize 10X to 1000X improvements in performance compared to conventional implementations using software-programmable DSP microprocessors. ISS is a privately held company with European headquarters in Belfast, Northern Ireland, UK and worldwide sales and marketing operations in San Jose, California. More information about ISS and its products may be obtained at http://www.iss-dsp.com**

# Moving the M8051Ewarp ASIC Core to a Virtex FPGA

Mentor Graphics implements their M8051Ewarp™ core in a Virtex FPGA.

by Kevin Rowley
Design Engineer, Mentor Graphics - IP division
kevin_rowley@mentorg.com

There is an ever increasing demand for digital IP cores that have been proven in FPGAs as well as ASIC test silicon. This is due to the obvious prototyping advantages and increasing market share of FPGAs. In this article I explain the strategy we used and the results we obtained when implementing the Mentor Graphics M8051Ewarp™ core, which was originally created for ASIC development, in Xilinx Virtex™ technology.

### Test Strategy

The usual method we use to prove the functionality of the M8051Ewarp core in ASIC gates is to envelope the synthesized core in a wrapper which has instantiated all the necessary components for simulation of the core. These components typically are program memory, data memory, and extra peripherals. However, after FPGA place and route the core will have I/O pads on its interface.

The delays inherent in these pads preclude the usual ASIC test method and a new synthesizable wrapper is required which, after place and route, will have the M8051Ewarp core, memories, and required peripherals in a single netlist which can then be simulated. The synthesizable wrapper we used is illustrated in Figure 1.

The most important point about Figure 1 is that all memories are on-chip, which means there are no I/O pads between the M8051Ewarp core and its memory modules, and thus delays on inputs and outputs to and from memory are avoided. The test code for the core is contained in the Program ROM. Also shown are the data memory RAM block and the internal data memory (IRAM). The peripherals are the wait-state generator necessary for the test suite and the External Special Function Registers (ESFRS) also needed by the test suite.

Data captured from the wrapper is written to a simulation-listing file for comparison

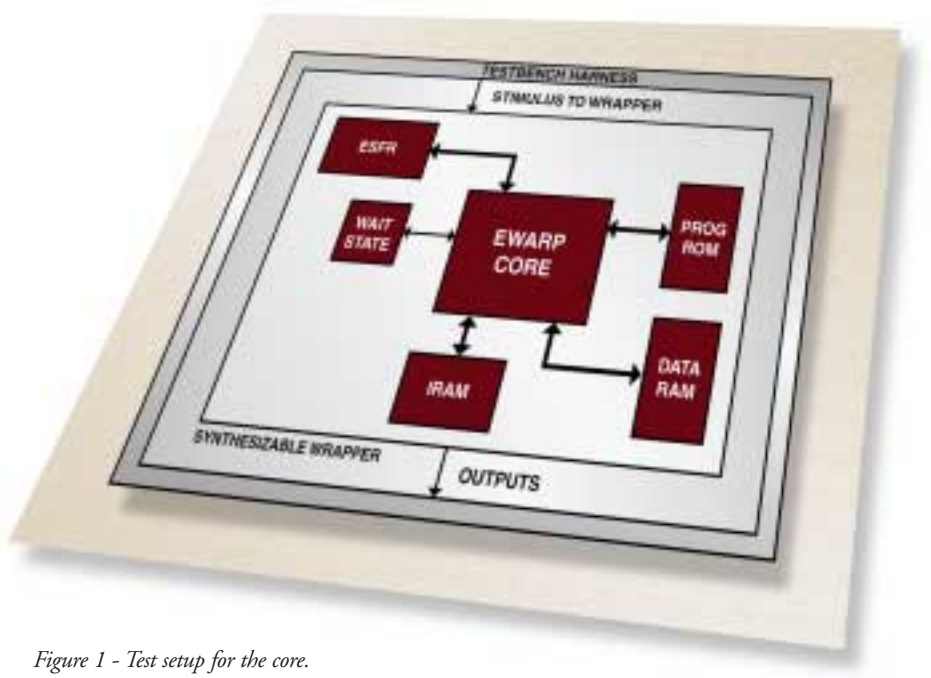



*Figure 1 - Test setup for the core.*

to reference listings. The test bench is a self-checking type which will setup and initialize the M8051Ewarp core and then check the program execution (using the test program stored in ROM).

### Memory Requirements

Successful simulation of the system test suite required the following memory :

- 4k bytes of synchronous ROM.
- 1k bytes of synchronous single-port RAM.
- 256 bytes of dual-port synchronous RAM.

The CORE Generator tool from the Xilinx 2.1i Alliance Series software suite was used to design and generate the memory modules.

### Specifying Memory Contents

The RAM cores had all their contents initialized to zero by the CORE Generator, however the ROM module had to have test opcodes stored in it for simuation. There are two ways of specifying ROM contents with the CORE Generator:

- .MIF file - Requires a line for each ROM location but also each byte has to be in binary format.
- .COE file - Allows the ROM locations to be specified in hex format.

Because the .COE format is closer to Intel hex format we decided to use a .COE file to specify the ROM. First however it was necessary to write a special program which would take the Intel hex format file for the test opcodes and convert it into a .COE file for the CORE Generator.

The final COE file looked like the following (abridged) :

```
Component_Name=crom;
Data_Width=8;
Depth=4096;
Radix=16;
Default_Data=0;
Memory_Initialization_Vector =
01,
80,
00,
c2,
a8,
...
```

The CORE Generator then generated an EDIF netlist with this .COE file for the ROM; it also generated EDIF netlists for the RAM modules. The EDIF netlists would be dragged-in later at the place and route stage for the wrapper.

### Synthesis Strategy

We used Mentor Graphics Leonardo™ for synthesis. The target operating speed of the M8051Ewarp core in the Virtex FPGA was 30 Mhz. Therefore, the synthesis constraints were setup accordingly. The part we targeted was the XCV200BG352. To synthesize the wrapper we first synthesized the M8051Ewarp core separately and then read it in, during the synthesis of the wrapper, as a separate file.

### Wrapper Synthesis

For synthesis, all the memory modules are treated as "black boxes." In addition to producing an EDIF file of the synthesized wrapper for place and route, synthesis also produced an .NCF file which Xilinx place and route uses to determine the timing constraints of the circuit. The timing analysis by Leonardo produced the following results:

```
Clock Frequency Report

Clock        : Frequency
-----------------------------------------------------------------

SCLK         : 33.8 MHz
CCLK         : 33.4 MHz
PCLK         : 33.4 MHz


Critical Path Report

There are no paths that violate user specified
options or constraints
And the expected area utilization report:

****************************************
Device Utilization for v200bg352
****************************************
Resource        Used  Avail Utilization
-----------------------------------------------------------------
IOs          97   260   37.31%
Function Generators  3653  4704  77.66%
CLB Slices       1827  2352  77.68%
Dffs or Latches    653  4704  13.88%
-----------------------------------------------------------------
```

This shows that the wrapper fitted into the XCV200 device and, according to Leonardo, was expected to run correctly at 30 Mhz.

### Place and Route of the Wrapper

Because several runs were required to pass static timing during synthesis and place and route, it was necessary to automate the place and route stage using the following script :

```
ngdbuild -p v200bg352-6 ewarp_f.edf ewarp_f.ngd

map ewarp_f

par -d 1 -ol 5 -pl 5 -rl 5 -w ewarp_f ewarp_f_out.ncd ewarp_f.pcf

trce ewarp_f_out ewarp_f.pcf -v 3 -o ewarp_f_out

ngdanno ewarp_f_out ewarp_f.ngm

ngd2ver ewarp_f_out -w
```

The file ewarp_f.edf is the EDIF file for the wrapper after synthesis. Note "ngdanno" produces the SDF file which must be back-annotated with the FPGA netlist for gate-level simulation. The "ngd2ver" program produces a verilog netlist of the SimPrims primitives for verilog gate-level simulation. The FPGA device utilization figures we achieved are detailed in the output file from the "par" program:

```
Device utilization summary:

   Number of External GCLKIOBs   3 out of 4   75%
   Number of External IOBs      94 out of 260  36%

   Number of BLOCKRAMs          11 out of 14  78%
   Number of SLICEs           2073 out of 2352 88%

   Number of GCLKs               3 out of 4   75%
   Number of TBUFs              16 out of 2464  1%
```

The "trce" program listed static timing information and constraints applied for place and route. This produced a lot of violations on paths which upon closer inspection turned out to be multi-cycle path exceptions. This was because multi-cycle path exceptions setup for the M8051Ewarp synthesis were not included in the .NCF file generated from wrapper synthesis. Some effort was needed to examine all violating paths from "trce" and make sure they were path exceptions.

We found that the maximum speed with the netlist was 31.25 Mhz. Above this speed the setup time required for PROGA feeding into the ROM was being violated;

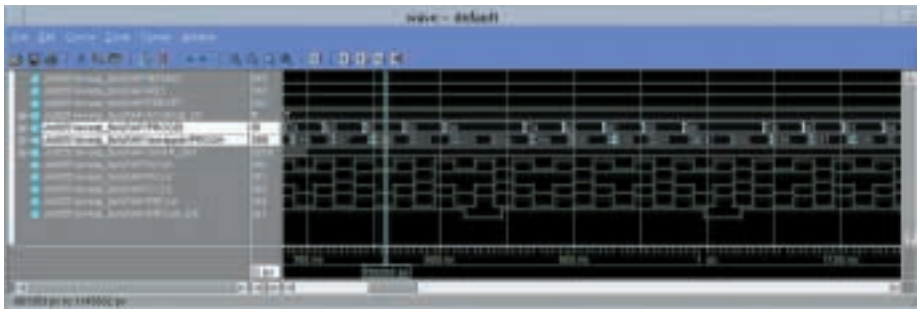


*Figure 2 - Waveform view.*

### Verification

Illustrated in Figure 2 is a waveform view of the M8051Ewarp core cycling through the test code stored on ROM at 30 Mhz. Program opcode can be seen coming into M8051Ewarp core on the PROGDI input. The program address is shown in signal PROGA. The test code was setup by the COE file at ROM generation.

Running the complete simulation through 4k bytes of test code results in 12100 vectors. If the core has simulated correctly, it will finish at PROGA equal to FFF hex. If there has been a problem then the simulation will not reach program address FFFh and it will be stuck in a loop at an earlier program address and will terminate after a certain time-out period.

The output delays on some of the M8051Ewarp core output ports exceeded the strobe period at 30 Mhz. Usually core outputs are strobed out to a listing file twice per clock cycle, the strobe period being just less that half a clock cycle. Therefore, since some output delays exceeded the strobe period it was not possible to do a straight unix "diff" between the Virtex netlist simulation listing and the reference listing for this test. However inspection of the listing file from simulation and comparison with the assember code listing for the test showed the circuit to be functioning correctly.

PROGA was becoming valid too late before the next clock cycle.

### Conclusion

We successfully placed and routed a complete M8051Ewarp core plus memory and peripherals on a Xilinx Virtex device and got it to work at speeds up to 31.25 Mhz. The key to our success was using the on-chip memory of the Xilinx part, and using the Xilinx CORE Generator software to design the memories. Synthesis was performed by Leonardo and the post-synthesis EDIF netlist of the M8051Ewarp wrapper was placed and routed by Xilinx 2.1i Alliance Series software. The resulting verilog netlist and SDF file, after place and route, ran through the test suite successfully at the required speed.

**For more information on the M8051Ewarp core see the Mentor Graphics website at: www.mentor.com/inventra/8051e_warp.html**

*References*

*[1] ASIC/FPGA market share , www.xilinx.com, June 2000*

# Xilinx WebPACK Software Now Includes ModelSim

## A complete design environment, including simulation, for Xilinx CoolRunner and XC9500 CPLDs...

by Anita Schreiber
Staff Applications Engineer, Xilinx
anita.schreiber@xilinx.com

Simulation is one of the keys to creating fast compact designs, with the least time and effort. Now you can simulate your functional and post-route VHDL and Verilog CPLD designs using the Xilinx WebPACK software, which includes the ModelSim software from Model Technology, Inc. (MTI).

Using the MTI ModelSim simulator gives you the ability to use HDL testbenches which allow you to behaviorally describe the stimulus for your design. Describing stimulus behaviorally allows the simulation to more accurately represent the system conditions and to vary based on the response received by the device. HDL testbenches also enable concurrent stimulation of different functions within the design and can be defined to compare the results from the device with expected results, eliminating the need to inspect waveforms to insure that the device is functioning properly.

### MTI ModelSim Xilinx Edition (MXE)

The inclusion of ModelSim in WebPACK is through an exclusive OEM arrangement between MTI and Xilinx. MTI licenses to Xilinx a special edition of ModelSim called the ModelSim Xilinx Edition (MXE). MXE Starter is a free trial version available to Xilinx WebPACK customers that allows you to run up to 500 debuggable lines of code before performance reductions occur. Upgrades to the ModelSim Personal Edition (PE) and the Elite Edition (EE) are available from MTI and both upgrades are compatible with WebPACK.

### Downloading MXE

MXE is included in the latest backPACK module of WebPACK as shown in Figure 1. Downloading the required design entry and CPLD fitter modules of WebPACK and the MXE backPACK module results in a tight integration between WebPACK's Project Navigator design environment and MTI's simulation tools. WebPACK is available at: www.xilinx.com/products/software/webpowered.htm.

### MXE Licensing

Licenses are required for MXE and can be obtained at the end of the installation process or after the product has been installed. With either method, your Web browser will be directed to an online license request form. Upon completion of this form, the MXE license and instructions for installing the license will be quickly e-mailed to you.

### Using MXE with the WebPACK Project Navigator

Once you have installed MXE, you can import HDL testbenches into design projects in the same manner as other source files. When an HDL testbench is selected in Project Navigator, the ModelSim Functional and Post-route simulation processes are available as shown in Figure 2. You can create simulation command files (ModelSim ".do" files) as Project Navigator invokes the MXE simulator, or you can specify existing .do files.

Execution of either of the simulator processes results in ModelSim simulating the design and displaying the specified windows as shown in Figure 3.

### Conclusion

With ModelSim and WebPACK you can quickly and easily create CPLD designs that work perfectly, the first time.

A detailed application note (XAPP338) instructing you on the operation of MXE within WebPACK for both CoolRunner and XC9500 CPLDs, leads you through a simple and complex design example using MXE for both functional and post-route simulations. XAPP338 is available on the Xilinx website at:

**www.xilinx.com/xapp/xapp338.pdf.**

*Figure 1 - MXE backPACK Module.*

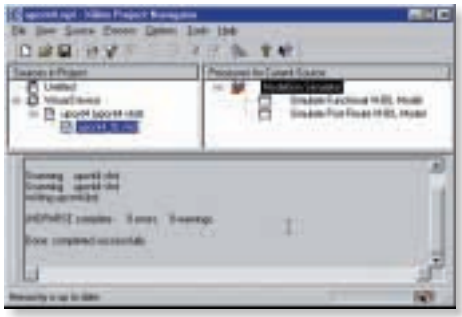


*Figure 2 - ModelSim Simulator processes.*

*Figure 3 - Resulting ModelSim windows.*

# New WebPACK Integrated Synthesis Environment (ISE)

## Free, comprehensive design environment for CPLDs available over the Web.

by Larry McKeogh
CPLD Software Sr. Technical Marketing Engineer, Xilinx
larry.mckeogh@xilinx.com

Xilinx WebPACK™, the popular Internet-enabled Project Navigator-based CPLD design environment just got better. The new WebPACK ISE release extends your productivity and performance capabilities even further with an easier to use design interface and greater design control.

The free WebPACK tool modules include:

- Design Entry Module - Provides VHDL, Verilog, and ABEL HDL support as well as schematic capture design entry which is new to WebPACK with ISE.
- Fitter - Fitters for either the XC9500 or CoolRunner families.
- Programming - For device programming control.

In addition, the recently introduced BackPACK modules have been augmented for the WebPACK ISE release.

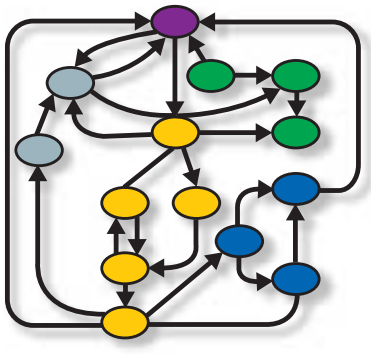
### New BackPACK Capabilities

Xilinx introduced the concept of a WebPACK "BackPACK," or productivity enhancement module in April 2000. BackPACK modules are not required for CPLD design completion, but are useful in extending design flow functionality. The first BackPACK modules introduced were ChipViewer and Model Technology's ModelSim XE Starter Edition. WebPACK ISE includes updates to these two tools as well as offering schematic design libraries for use with schematic capture.

WebPACK ISE also introduces two new tools developed by Visual Software Solutions (VSS) for inclusion in Xilinx software: StateCad (www.statecad.com) and HDL Bencher (www.testbench.com).

### StateCad

StateCad automates the state machine design process. It automatically looks for common design problems such as stuck-at-state, conflicting state assignment, and inde-



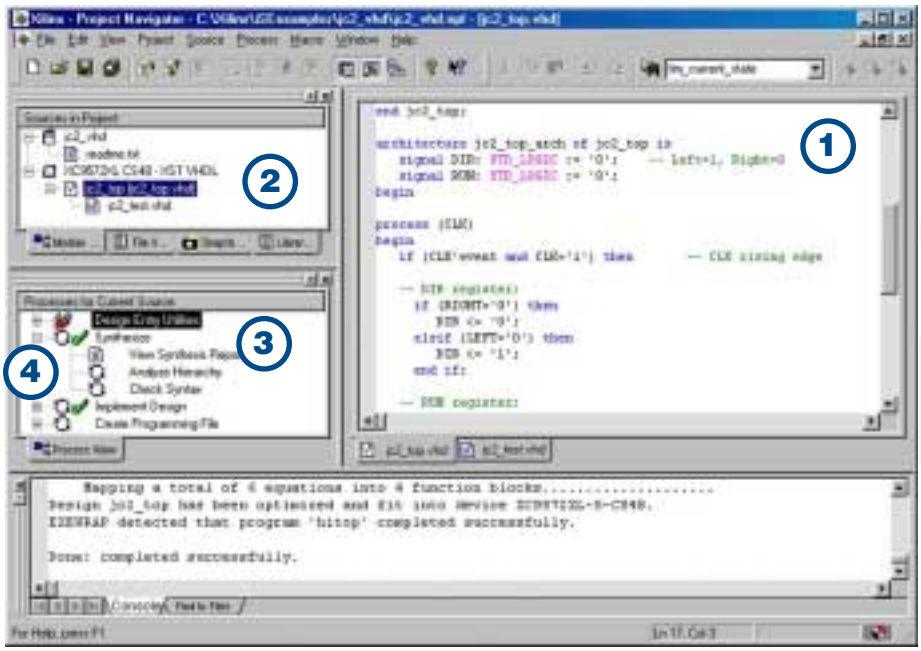
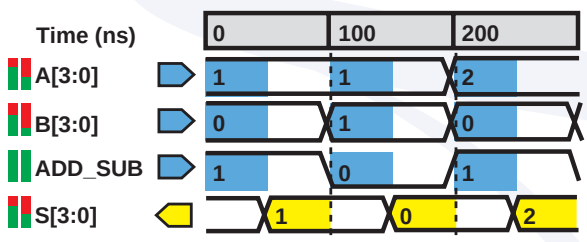



*Figure 1 - Project Navigator design environment.*

terminate conditions. Its automated error analysis insures that designs are logically consistent which reduces the simulation requirements and improves product reliability. StateCAD automatically produces HDL code for synthesis and simulation that eliminates manual translation efforts and coding errors. It simplifies complex state machine design allowing you to achieve peak hardware performance with less effort than before.

### HDL Bencher

HDL Bencher is an automatic test bench generator allowing test benches to be created in minutes. A VHDL or Verilog design




file can be imported to HDL Bencher either manually or through the Project Navigator design environment. HDL Bencher analyzes the design I/O and creates a default stimulus waveform for each. The waveform may be easily modified to simulate the expected design performance. Final waveforms may be exported as a VHDL or Verilog file for use in many popular EDA simulators. The automatic nature of HDL Bencher allows test benches to be automatically updated as your design changes, thus eliminating stale test cases. Use of HDL Bencher ensures that your complex hardware designs are more robust and up-to-date than if coded manually all in a fraction of the time.

### Easier to Use Design Interface

The upgraded Project Navigator design environment, shown in Figure 1, has a new look and feel. The basic Project Navigator structure is maintained and includes significant new flexibility enhancements, allowing you to fully customize the design environment.

Other ease of use improvements (as shown in Figure 1) include:

1 - HDL editor integration.

2 - Synthesis flow switching.

3 - Double-click design process property and options access.

4 - Augmented design processes grouped into four basic areas allowing as much or as little interaction with the design flow as you want.

### Greater Control

In addition to the improved design environment and new tool integration the CPLD tools are augmented with the following new features and improvements:

- XC9500XV output banking support automatically assigns device outputs based on the desired output voltage standard.
- Improved design analysis included in the fitter and timing reports.
- Operating code cleanup and efficiency improvements.
- The ABEL language was upgraded to version 7.3 with the following changes:
  - ABEL-XST synthesis. ABEL designers can now choose ABEL-XST synthesis for similar synthesis results with improved software stability
  - WYSIWYG Support. New option for ABEL designers who want to assign the exact design implementation and not have the synthesis or fitting tools re-optimize the design
  - True bus notation for ABEL designs
- Improved design analysis included in fitter and timing reports.
- Operating code cleanup and efficiency improvements.

### Conclusion

WebPACK ISE builds on the success of the original WebPACK with improvements to the Project Navigator design environment for improved ease of use. The Xilinx CPLD design implementation tool update included in WebPACK ISE provides design control, while the tight integration of new tools such as ECS, HDL Bencher, and StateCad extend the WebPACK functionality making WebPACK ISE the industry leader in design environments available over the Internet.

*WebPACK ISE is a modular design tool available for free from the Internet at http://www.xilinx.com/products/software/webpowered.htm.*

# Low Power CoolRunner CPLDs

## An Ideal Fit for Smart Card Readers


Coolrunner CPLDs can be used to implement various functions within Smart Card Readers, and are especially beneficial in handheld, lightweight, battery-powered applications.

by Karen Parnell
European Marketing Manager,
High Volume Products, Xilinx
karen.parnell@xilinx.com

Smart Cards are becoming a common part of our every day lives, and new ways to use this technology are being developed at a rapid pace. Because they hold more than 100 times the information contained on a standard magnetic strip card, Smart Cards can store small denominations of cash, accumulate medical records, be used as security cards, and can greatly ease on-line buying. Their applications and acceptance are increasing daily; the American Express® Blue Card now comes with a free Smart Card reader that connects to your PC so you can interrogate your account securely on-line, and you don't have to enter your account details every time you purchase from a website.

### The Smart Card Reader

One of the fastest growing uses for Smart Cards is to replace the standard credit card; this has lead to a dramatic increase in portable battery powered Smart Card readers. These readers can be used in restaurants for payment at your table, in taxis and buses for payment on the move, and in on-line Web-based or main street stores.

Figure 1 shows the basic components of a Smart Card reader. These functions may also be integrated into cash registers, vending machines, public pay phones, set-top boxes, mobile phones, utility meters, and many other devices that require an authentication or secure payment system.

The functional blocks that make up the system are:

- Main data processing - typically a 16- or 32-bit microprocessor (MCU) for computational functions.
- Memory - to store data (operating system, variables, data storage) and microprocessor boot code.
- Security logic - to aid data encryption.
- Card reader interface - for both the Smart Card reader (contact and contactless) and the magnetic card reader.
- Keypad and keypad decoder - for entering Personal Identification Numbers (PINs) and other data, and the associated logic to decode the input.
- LCD Display driver - for user feedback.
- Modem and modem interface - for interfacing to wireless, cellular, and radio modems (usually PCMCIA type).

### How it Works

In a typical consumer transaction:

- The merchant inserts the Smart Card into the card reader and power is applied to the card.
- The reader communicates with the Smart Card MCU to perform the card authentication cycle.

- During the initial read function the Smart Card interface logic passes the data to the card reader microprocessor via the security logic. (The CoolRunner device is a non-volatile EEPROM CPLD so it is secure and reliable and thus ideal for implementing security logic and MCU decoding.)
- The card reader instructs the user to enter a PIN via a message on the LCD. The user enters their PIN via the keypad; this is authenticated by the reader MCU. The PIN is verified by the MCU in the card which compares the PIN stored in it's RAM with the one presented. If the comparison is negative the CPU will refuse to work. The Smart Card keeps track of how many wrong PINs are entered and if it is over a predetermined number, typically three attempts, the card blocks itself against any future use.

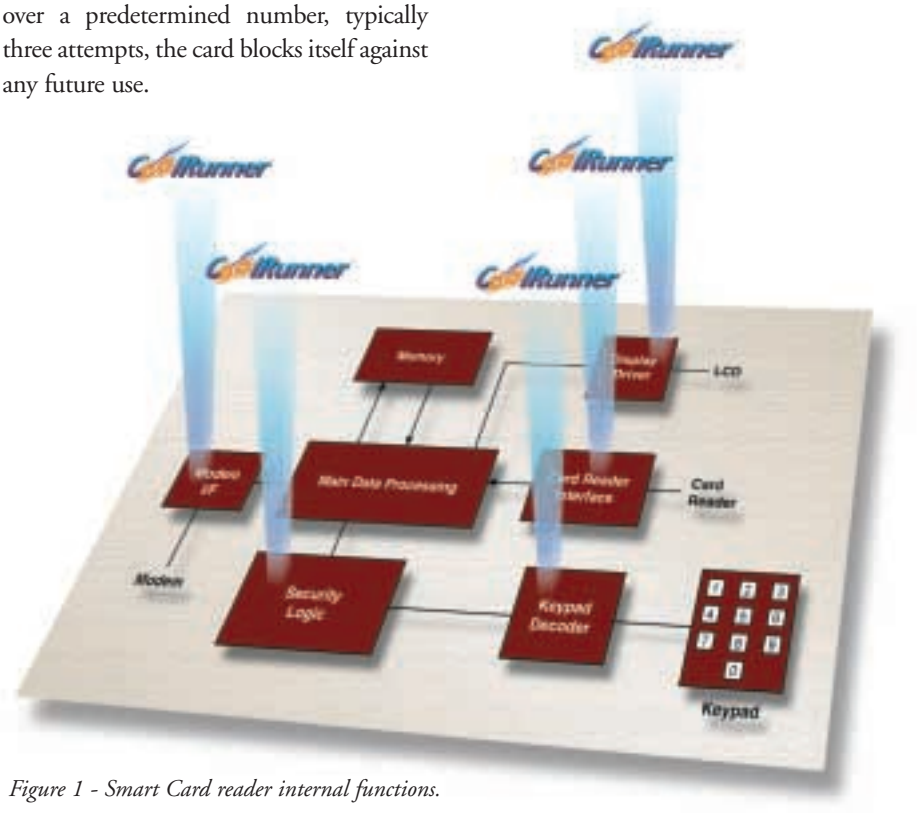



*Figure 1 - Smart Card reader internal functions.*

- When the PIN is verified, the purchase amount is entered by the merchant and the Smart Card is interrogated to see if it has enough stored value. If so, the amount entered is deducted from the stored value on the card.
- If it is not an SVC (Stored Value Card) transaction then the amount to be debited from the bank account will be verified using the modem (wireless, cellular, or radio).
- When the transaction is complete the card is ejected and removed. The Smart Card reader is then ready for the next transaction.

### CoolRunner Technology is Key

The Coolrunner family of ultra low power, low cost CPLDs is ideal for this application. Not only do the CoolRunner devices consume less power than any other CPLD but they come in very small form factor packages, such as the 56 pin, 0.5mm pitch Chip Scale Package device shown in Figure 2. CoolRunner devices are available in both 3.3V and 5V versions, enabling Smart Card readers to easily accommodate both the new 3.3V Smart Cards and the older 5V types.

CoolRunner CPLDs are perfect for performing the interfacing and decoding functions in the Smart Card reader. The main CoolRunner tasks are the memory interfacing, input/output expansion, keypad decoder logic, LCD interfacing, modem interfacing, and interfacing to the physical card reader itself. Because they are reprogrammable, CoolRunner devices allow the Smart Card reader to be updated in the field, thus increasing the effective system life. By integrating all of the logic into a small form factor, ultra-low-power CoolRunner CPLD, you can dramatically reduce the total PCB area and number of layers required.

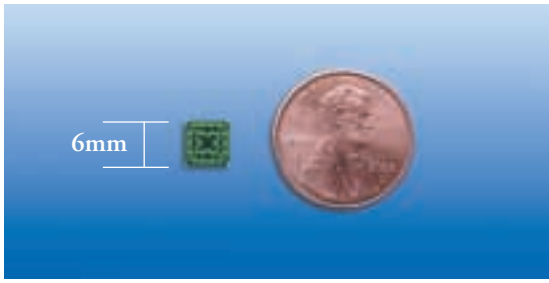



*Figure 2 - CoolRunner XPLA3 64-macrocell 56-pin, 0.5mm pin pitch Chip Scale package.*

### Conclusion

The Smart Card market is on the brink of realising its full world-wide potential for cashless transactions, store loyalty schemes, access control systems, medical record cards, identity cards, drivers licenses, and many other applications. This year we will see personal computers shipped with Smart Card readers as standard equipment which will unlock widespread world-wide acceptance of multi-application Smart Cards, and hand-held battery powered Smart Card readers in taxis and buses will become common place.

Xilinx high-volume CoolRunner CPLD devices provide you with cost effective solutions that retain the traditional PLD time to market advantage but with the added benefit of ultra low power operation, very small form factor packages, and a secure, reliable, non-volatile process technology.

| | XCR3032XL | XCR3064XL | XCR3128XL | XCR3256XL | XCR3384XL |
|---|---|---|---|---|---|
| Macrocells | 32 | 64 | 128 | 256 | 384 |
| Usable Gates | 1000 | 2000 | 4000 | 8000 | 12000 |
| $t_{PD}$ (ns) | 5 | 6 | 6 | 7.5 | 7.5 |
| $f_{SYS}$ (MHz) | 200 | 166 | 166 | 133 | 133 |
| Packages (Max. User I/Os) | 44VQ (32)<br>48CS (32) | 44VQ (32)<br>*48CS (32)<br>56CP (44)<br>100VQ (64) | 100VQ (80)<br>144TQ (104) | 144TQ (104)<br>208PQ (160)<br>280CS (160) | 280CS (216) |

*Table 1 - The CoolRunner family of CPLDs.*

# PLDs-Your Competitive Edge for High Volume Production

## With a compound annual growth rate of over 29% from 1990 to 2000 the PLD industry is one of the fastest growing segments of the IC industry.

by Rebecca Burr
Director, Market Analysis, Xilinx, Inc.
rebecca.burr@xilinx.com

According to the Semiconductor Industry Association (SIA), the programmable logic market is forecasted to grow by 42.9% in 2000 to $4.1 billion. To place this in perspective, in 1990, PLD shipments of $416 million represented roughly 5% of all logic devices sold. In the year 2000, PLDs will account for more than 14% of all logic devices sold. This tremendous market expansion only further reinforces the growing acceptance and viability of PLDs as a system design standard. Fueling this explosive growth are several industries whose success was in part enabled by the PLD industry.

### Time-to-Market Advantages

Winning companies have succeeded by getting their products to market before their competition. They meet their customers' needs quickly and therefore establish a market position that is very difficult to challenge; this position also gives them a market advantage for subsequent generations of products. This so-called "time-to-market" paradigm is the watchword for many entrepreneurial companies.

Reducing their time to market remains a key focus for competitive companies. This focus is growing in complexity because of virtual manufacturing and because system designers are increasing their collaboration with suppliers and customers–the steps required for prototyping, design change, quality testing, and change execution are being further compressed. Programmable logic has proven itself a very effective solution for dealing with these challenges by helping companies deliver products to market as fast as possible.

### Time-to-Volume Advantages

Winning companies must not only develop new products quickly (time to market), they must also be able to manufacture the product, and quickly ramp production to meet customers' demands. This is called "time-to-volume." Component supply through electronics distributors, and flexible contract manufacturing, are factors that have enabled companies to respond to soaring customer demand. Furthermore, PLDs have been a major factor in providing an off-the-shelf platform for not only prototyping and early production, but also manufacturing through the entire product life cycle.

As standard, off-the-shelf products, PLDs can be produced inexpensively, in high volumes; there is no delay for production ramp up as is often the case with ASICs which often require a long lead time and added risk. We maintain that time to volume is a more critical concern than time to market for electronic equipment designers and manufacturers today.

Designing with standard, off-the-shelf programmable logic devices gives you the key advantage of flexibility, allowing you to immediately address market demands for high volume production; programmable logic has been proven effective for all applications, not just low volume systems and prototyping.

### Affordability Meets Desirability

Many factors affect whether a product is viewed by the marketplace as being both desirable and affordable. Figure 1 illustrates market potential, which is the junction representing those consumers who desire and can afford a given product.

Consumers must balance the purchasing power of their income with the cost of a given product. Purchasing power is affected by a broad spectrum of factors. For example, consumers' income levels, other expenses, and inflation play a large part in determining purchasing power. At the same time, they must rationalize the cost of ownership of a product (maintenance, and so on) and any ongoing usage costs. Therefore, a product must offer a high level of utility through features and performance. The combination of product appeal and practicality equates to market size.

### A Tale of Two Products

Let's illustrate how the manufacturing model, and consequently the drive for time to volume, has changed over the years by contrasting TVs with DVDs (digital video disk players). Television entered the US market in 1936 with very little market impact, because there was little content and also little disposable income with which to buy the TV sets. By 1945 there were probably less than ten thousand TV sets in use, yet that number was destined to grow enormously as

post-war incomes rose and broadcast networks were able to supply the content that drove demand. By 1950, the number of TV sets in consumer's hands had grown significantly to six million sets, and then to sixty million by 1960. In 1998, according to the Consumer Electronics Manufacturing Association (CEMA), penetration of color televisions reached 98% of U.S. households. Figure 2 presents television penetration over time.

Contrasting the acceptance and penetration of television to that of DVD player market illustrates why time to volume has become a fundamental market force for designers and system manufacturers. The DVD market was in an ideal market position at its inception. The content (movies) was already available and the market requirement for high-quality movies to be played on big-screen TVs and home theater systems had already developed. Moreover, the cost to purchase a DVD player and the DVD disks was low compared to when TVs were initially introduced to the market. The result was explosive growth in product demand. Industry prognosticators forecast that the US consumers will purchase ten million DVD players in 2000. This will represent an increase in market penetration from 4% to 12% in twelve months.

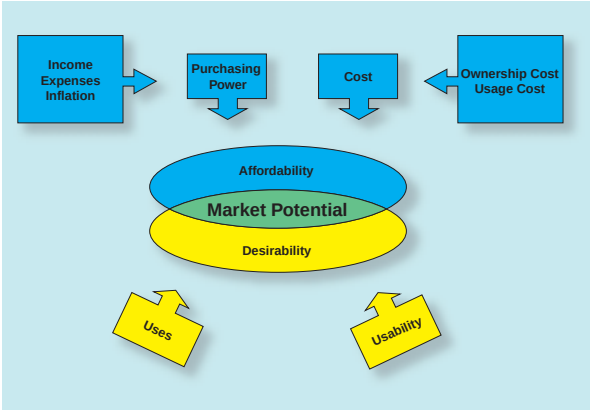



*Figure 1 - Market potential is the overlap of affordability and desirability.*

Customer demand has driven this growth of advanced products. TVs were interesting and unique, yet with the lack of disposable income there was not a significant demand for televisions until the 1950s. For consumer products, the affordability threshold is attained when the end equipment price falls between 1.1 and 1.8 weeks of household income. Table 1 presents US median household income and corresponding price points for consumer goods acceptance.

## Industrial Markets

The concepts of market potential apply to non-consumer, industrial equipment as well. For example, Internet-based corporations of all sizes must respond to their customers' needs for more bandwidth. Both internal customers for Intranet support, and external customers demanding faster, more seamless Internet interfaces, are demanding greater performance. The result is that companies are demanding ever-increasing performance and more features from their network system suppliers. Those network suppliers, in turn, have to respond with faster, better, cheaper products in much shorter time than ever before. The stakes are colossal, because if a network company cannot service its customers, then some other supplier will step in and take its place as the market leader.

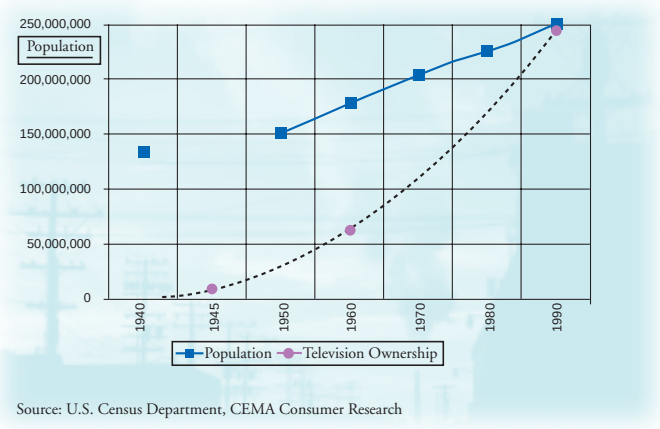



*Figure 2 - Television penetration time to volume*

## Conclusion

Programmable logic will continue to outpace other electronic component market segments because there is no faster or better way to develop and manufacture your products; this applies to cellular phones, base stations, electronic test equipment, medical devices, telephone switching systems, and a vast array of consumer and industrial products. In the year 2000, PLDs will account for more than 14% of all logic devices sold.

Programmable logic has proven itself a very effective solution for dealing with the challenges of delivering product to market on a timely basis.

| Year | Total Income | Product Affordability Range | | |
|---|---|---|---|---|
| 1969 | $33,072 | $728 | to | $1,191 |
| 1979 | $34,666 | $763 | to | $1,248 |
| 1983 | $32,941 | $725 | to | $1,186 |
| 1989 | $36,598 | $805 | to | $1,318 |
| 1993 | $33,660 | $741 | to | $1,212 |
| 1996 | $35,172 | $774 | to | $1,266 |

*Table 1 - Product affordability based on consumer income (in 1996 dollars).*

# Design Reuse Strategy for FPGAs

## The design productivity gap creates an opportunity for competitive advantage.

by Carol Fields
Methodology Manager, Design Solutions Group, Xilinx
carol@xilinx.com

With today's technology, you are faced with more usable gates then ever before; that's both a blessing and a curse. Moore's law, stating that "the number of transistors per square inch on an integrated circuit doubles every two years", has been holding true since 1965, allowing a higher level of integration to occur on one piece of silicon. However, you can only take advantage of this exponential growth in densities if it is matched by a similar growth in design productivity.

Why worry about this gap in productivity? History has shown that on average, for a given product market, the first entrant takes 70% of all sales over the lifetime of the product, the second takes 15%, and the later ones share the remaining 15%, as illustrated in Figure 1. It pays to be first.

### Design Reuse Fills the Productivity Gap

The most promising way to fill this productivity gap, and be the first to the market with your new product, is to reuse existing designs or Virtual Components (VCs). Design reuse is not a new idea; designers have always reused code, scripts, testbenches, and so on. What is new (from two years ago) is the growth in intellectual property (IP) infrastructure and the formalization of Design Reuse methodologies.

Design Reuse exists in several levels from "ad hoc" reuse of a previous project to purchasing a design from a third-party vendor. If an in-house VC has the potential of being reused more than twice, it is wise to put some effort into developing a Design Reuse methodology. This methodology addresses all aspects of the process including legal and business practices, VC specification, designing and coding, testing strategies, design storage and retrieval, and design metrics.

The amount of effort or lack of effort has a direct effect on the VC's reusability, especially if the person reusing the module is not the original designer. If a third-party VC is to be purchased, it is important to understand the total cost and actual usability of the VC.

The key to Design Reuse is trust. For an internally generated VC you need to build-in trust, and when purchasing a third party VC you need to define what makes the VC trustworthy.

### Design Reuse Strategy Today

The most popular design reuse methods are for ASIC designs. This makes sense because the abundance of gates provided by Moore's Law is allowing such a high level of integration as to place an entire System-on-a-Chip (SoC). In 1995 the Dataquest definition of SoC included a compute engine (microprocessor, microcontroller, or digital signal processor), at least 100K of user gates, and significant on-chip memory. Five years later we still do not find an abundance of systems on an ASIC chip. However, what we are seeing is an increased use of System Level Integration (SLI). What we are also finding is more SLI occurring on FPGA devices with many of the same design issues as SoC.

The predictions are that by 2003 the bulk of the industry revenue growth will be because of SLI. The existing Design Reuse methodologies today are closely linked to SoC. In reality it makes more sense to tie Design Reuse methodology to both SoC and SLI since a formalized Design Reuse methodology would benefit both SoC and SLI designers.

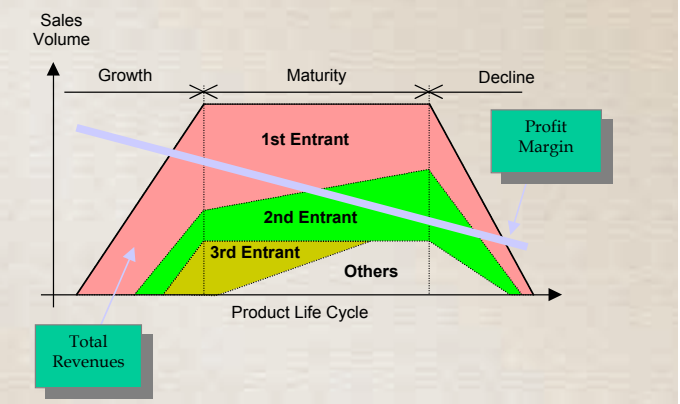



*Figure 1 - Time-to-market product life cycle.*

### Systems-On-a-Reprogrammable-Chip

FPGAs have changed dramatically since they were first introduced and used as glue logic just 15 years ago. In the late 1980 and early 1990s, FPGAs were primarily used for prototyping and lower volume applications while custom ASICs were used for high volume, cost sensitive designs. FPGAs had been too expensive and too slow for many applications, let alone for Systems-on-a-Chip. Plus, the development tools were often difficult to learn and lagged the features found in ASIC development systems.

Silicon technology now allows us to build FPGAs consisting of tens of millions of transistors allowing for more features and capabilities in programmable technology. With today's deep sub-micron technology, it is possible to deliver over three million usable system gates in a FPGA. Today's average ASIC design operating at 30 to 50MHz can be implemented in an FPGA using the same RTL synthesis design methodology. By the year 2003, a state-of-the-art FPGA will exceed 10 million system gates, and will operate at internal speeds far surpassing 200 MHz. Many designs that once could only be implemented in an ASIC due to speed, density, or pricing are converting to a much more flexible and productive FPGA solution.

It is a safe bet that more systems will be implemented in FPGAs in the future, especially given Moore's law and the ingenuity of FPGA R&D engineers. The industry analysts predict that in 2003 FPGAs will begin to replace standard cell ASICs in all but very high volume applications. However, there are many cases where the volume is not high enough for Standard Cell technology, such as the adoption of gigabit Ethernet, and areas where there is a demand for higher performance products to meet increasing traffic flow from the Internet.

Because Design Reuse is about planning for the future, the term Systems-on-a-Reprogrammable-Chip (SoRC) is used for SoC implemented in an FPGA. This term is used to define both full and partial system-level designs since the challenges facing these FPGA designers are almost identical. Although it is rare to find entire systems on an FPGA today, there is an increasing amount of SLI designs occurring due to the newer FPGA architectures (such as the Virtex family containing system-level features).

### Planning for the Future

Today most major digital design companies are in the process of defining (or redefining) their Design Reuse strategy. This generally includes the creation of an internal Design Reuse Department, similar in structure to the existing EDA Department used to manage CAD tools. In the late 1980's many major companies had already formed their EDA Department to support the multitude of ASIC EDA tools. At this point FPGA technology was just emerging with designers using either a proprietary EDA tool or a simple schematic capture tool; these tools hardly required a department to manage them. By the mid-1990s FPGAs were being widely used by these same companies for higher density glue logic and SLI, and FPGA support was retrofitted into the EDA Departments.

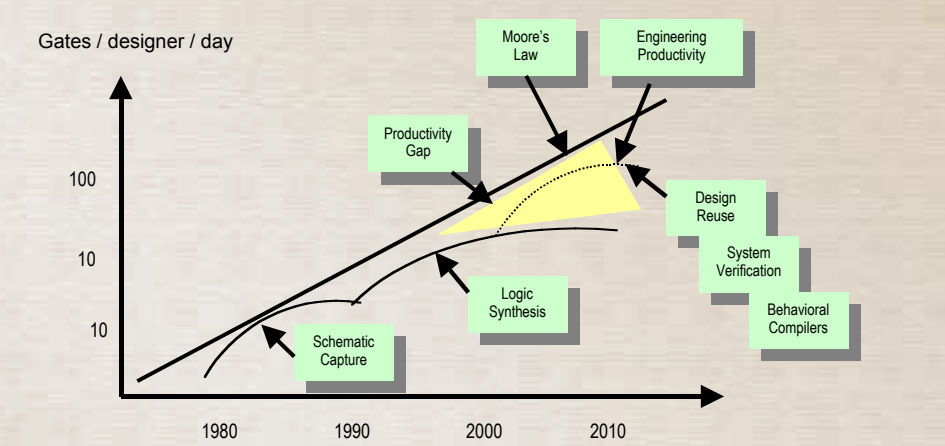



*Figure 2 - The productivity gap*

Today, we have the opportunity to define a reuse strategy that can not only co-exist for FPGAs and ASICs but can also work seamlessly between the two technologies. The decision to include FPGAs in a Design Reuse strategy must be made up-front because it affects almost all phases of the Design Reuse process, from design specification to verification planning.

### Sharing RTL Design Methods

One of the most exciting outcomes of the dramatic improvements in FPGA architectures, pricing, and design tools is that this technology advancement has made it possible for ASIC and FPGA designers to share a common RTL design methodology. A common RTL design methodology is the basis for a common design reuse methodology.

Though ASICs will continue to provide higher levels of design integration, higher speeds, and new EDA environments, FPGAs are never far behind. The major FPGA and EDA companies have made a conscious decision to keep their design environments the same, from the end users point of view, to make it easy for users to move from one technology to the another. This was illustrated by the wide adoption of RTL synthesis tools and verification tools in the mid-1990s. In the case of RTL synthesis, existing ASIC methodology was kept the same and the synthesis algorithms where changed to target specific FPGA devices. Today we are seeing higher-level EDA tools such as Floorplanners and team-based design tools using the Internet.

### Conclusion

In 1999, the number of ASIC design starts peaked at only 1000 designs, and despite all the publicity over the multimillion gates designs, most of these design starts were under 200K transistors. The average FPGA design start in 1999 was between 10K and 50K gates, with the fastest growing size range between 50K and 100K gates. Considering that FPGAs are more widely used than ASICs in digital designs today, it makes sense to include FPGAs in a design reuse strategy.

There are many benefits of sharing a common design reuse strategy; one of the most compelling is the flexibility it gives the designer to choose the IC technology late in the design cycle. It provides the flexibility to choose the best method to implement an SLI design without the overhead of retraining the design teams. In this fast pace market it is difficult to predict what features your product will need and what technology you should use.

A Design Reuse strategy is more than RTL code and synthesis. Many companies reusing designs have found more value in the design and test specifications than the actual RTL design. If you are currently an ASIC user, the good news is that many of the elements of a good design reuse methodology can easily incorporate FPGAs with minimal modifications.

*Xilinx has joined efforts with Qualis Design Corporation to create the first Reuse Design Guide for FPGA users. This new FPGA Reuse Field Guide will walk you through the elements of building a design reuse strategy and is available, free of charge, from the Xilinx website at: www.xiliinx.com/ipcenter.*

# Year 2000 Worldwide Xilinx Event Schedules

### Year 2000 North American Event Schedule

| | |
|---|---|
| **Sept 6-7** | Embedded Internet Conference 2000<br>*San Jose, CA* |
| **Sept 20** | SNUG 2000 Boston<br>*Boston, MA* |
| **Sept 24-28** | Embedded Systems Conference 2000<br>*San Jose, CA* |
| **Sept 26-28** | MAPLD 2000<br>*Laurel, MD* |
| **Oct 16-19** | ICSPAT 2000<br>*Dallas, TX* |
| **Oct 16-18** | NCF / InfoVision 2000<br>*Chicago, IL* |
| **Oct 18-21** | Frontiers in Education 2000<br>*Kansas City, MI* |

### Year 2000 European Event Schedule

| | |
|---|---|
| **Sept 4-15** | Xilinx Roadshow<br>*Throughout Germany* |
| **Sept 5-8** | EUSIPCO 2000<br>*Tampere, Finland* |
| **Oct 12-13** | Nokia Expo<br>*Espoo, Finland* |
| **Oct 23-24** | IP2000 Europe<br>*Edinburgh, Scotland* |
| **Nov 21-24** | Electronica 2000<br>*Munich, Germany* |

### Year 2000 South East Asian Event Schedule

| | |
|---|---|
| **Sept 27-28** | IIC 2000<br>*Seoul, Korea* |
| **Oct 3-4** | EDA&T<br>*Hsinchu,Taiwan* |
| **Nov 2000** | Xilinx Technical Seminar<br>*Sydney, Singapore, Seoul, and Shanghai* |

### Year 2000 Japanese Event Schedule

| | |
|---|---|
| **Nov 2000** | Micon System Tool Fair 2000<br>*Tokyo, Japan* |
| **Nov 2000** | Xilinx Technical Seminar<br>*Tokyo, Osaka, Yokohama, and Tachikawa, Japan* |

For more information about Xilinx Worldwide Events, please contact one of the following Xilinx team members or see our website at: http://www.xilinx.com/company/tradeshows.htm

- US Shows: Darby Mason-Merchant at: darby@xilinx.com or Jennifer Makin at: jenn@xilinx.com
- European Shows: Andrea Fionda at: andrea.fionda@xilinx.com.
- Japanese Shows: Renji Mikami at: renji.mikami@xilinx.com
- SouthEast Asian Shows: Mary Leung at: mary.leung@xilinx.com

# CoolRunner CPLDs - Your Best Choice for Battery Operation

Recent studies prove that the Xilinx CoolRunner™ family gives you far longer battery life than any other CPLD on the market.

by John Hubbard
CPLD Applications Engineer, Xilinx
john.hubbard@xilinx.com

The hand-held consumer electronics market is currently experiencing unprecedented growth, and it has become imperative for manufacturers to maximize battery longevity to enhance their competitive edge. With the CoolRunner CPLD family you get high performance, extremely low power, and field upgradeability, in a very small package—clearly the best solution for the next generation of low cost portable equipment.

### CPLD Comparison

Your battery-operated design will run much longer with CoolRunner CPLDs. To illustrate the dramatic difference, Figure 1 compares the impact on battery life using CPLDs from Altera®, Cypress®, Lattice®, Vantis® and Xilinx. This comparison was implemented with the following conditions:

- Two Energizer® No. E91 AA 1.5V alkaline batteries were used as the power source.
- The CPLD was the only device loading the batteries.
- Each CPLD was fully populated with 16 bit binary counters.
- All counters were clocked at 20 MHz.
- All outputs were unloaded.

### Dynamic Power Consumption

As you can see from Figure 1, CoolRunner CPLDs extend battery life far beyond the competition. In fact, the competition was given a power advantage by measuring their CPLDs in low power mode. However, the competition's CPLDs run much slower in low power mode. CoolRunner CPLDs do not need a low power mode and they always operate at full speed.

### Static Power Consumption

Since the CoolRunner CPLD draws 1/1000th the power that the competition requires at standby, your power management requirements will be dramatically reduced or eliminated. As shown in Figure 2, in standby mode, the CoolRunner CPLD can extend battery life up to 390 times that of the competition.

### Total Power Consumption

By modulating the dynamic operation (full power) with static operation (standby mode), you will considerably extend the battery life of your design using CoolRunner CPLDs. Figure 3 illustrates battery longevity in the form of duty cycle where, for example, a 75% duty cycle represents 75% dynamic and 25% static operation.

### Conclusion

CoolRunner CPLDs are clearly the low power leader—for your company to remain competitive in the rapidly growing portable market, your best choice is Xilinx.

See the articles on page ___ and ___ for more information. You can find the full story on CoolRunner CPLDs at: http://support.xilinx.com/products/xpla3.htm and http://support.xilinx.com/products/coolpld.htm

**You can extend your battery life up to 200 times just by using CoolRunner CPLDs.**

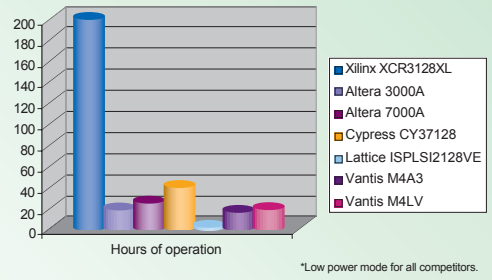



*Figure 1 - Dynamic CPLD battery life comparison.*

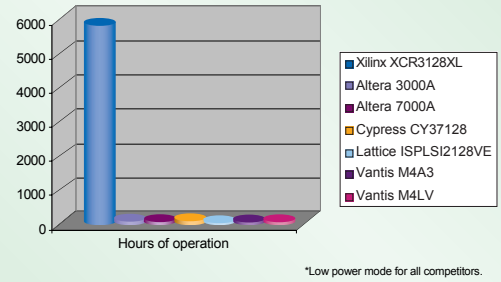



*Figure 2 - Static CPLD battery life comparison.*

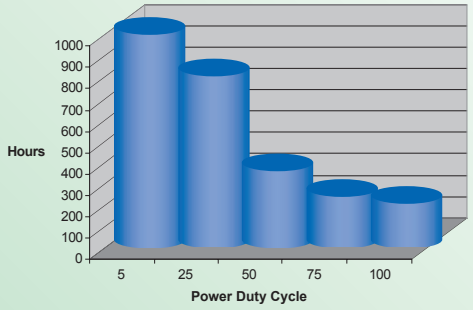



*Figure 3 - CoolRunner duty cycle effect on battery life.*

# Choices, Choices, and Opinions

## How to choose the best Xilinx programmable logic technology for your application.

by Peter Alfke
Applications Engineering, Xilinx
peter@xilinx.com

Xilinx offers a wide variety of programmable logic devices, using different architectures and technologies. Here is a high-level overview of the different families and some suggestions for the prospective user.

### 5V or 3.3V / 2.5V?

The standard logic supply for 30 years has been 5 volts, but modern processes with smaller geometries demand lower voltages, such as 3.3V, 2.5V, 1.8V, and even lower in the future. While Xilinx strives to make inputs tolerant to voltages higher than Vcc, we discourage you from starting any new project with 5V devices.

Xilinx will continue offering 5V devices for years to come, but these devices will not benefit from the traditional performance enhancing and cost-reducing redesigns, and process enhancements. To benefit from the latest technology, do not use 5V devices for new designs unless there is a special reason.

### CPLDs or FPGAs?

CPLDs with their PAL-derived, easy-to-understand, AND-OR structure offer a single-chip solution with fast pin-to-pin delays, even for wide input functions. And, once programmed, the design can be locked and thus made secure. Most CPLD architectures are very similar, so it is important to evaluate the subtle nuances.

In-system-programmability (ISP) is a must for today's designs, and the ability to maintain pin-outs during design modifications ("pin-locking") is crucial. The limited complexity (<300 flip-flops) means that most CPLDs are used for "glue logic" functions. For most CPLDs, the relatively high static (idle) power consumption prohibits their use in battery-operated equipment. Xilinx CoolRunner devices are the notable exception, offering the lowest static power consumption (<50 microamps) of any programmable device.

FPGAs offer much higher complexity, up to 70,000 flip-flops, and their idle power consumption is reasonably low. Because the configuration bitstream must be reloaded every time power is re-applied, design security is an issue, but the advantages and opportunities of dynamic reconfiguration, even in the end-user system, are an important advantage. FPGAs offer more logic flexibility than CPLDs, and more sophisticated system-level features (clock management, on-chip RAM, programmable I/O levels).

### Recommendations

Use CPLDs, such as the XC9500XL family, for small designs where "instant-on", fast and wide decoding, in-system programmability, and design security are important. Use CoolRunner CPLDs when idle-power consumption is important, as in battery-operated equipment.

Use FPGAs for larger and more complex designs.

### FPGA Families

Xilinx has a wide range of FPGAs to choose from. Here's an overview of our complete product line, from the beginning.

#### XC2000 and XC6200

The Xilinx XC2000 family was introduced in 1985, and has outlived its useful life. The XC6200 family embodied an innovative architecture that was popular in academic research, but found no commercial use. These families are no longer available.

#### XC3000, XC3100, and XC5200

These families are not recommended for new designs, because several newer families offer better functionality and performance at a lower price. The XC3000L is still the FPGA family with the lowest static (idle) power consumption of <100 microamps, and it offers an on-chip crystal-oscillator driver, not available in any other FPGA family. These families stay in production, but are not recommended for new designs.

#### XC4000 - E, EX, XL, XLA, and EV

Today, this is the industry's most popular series of FPGA families. The XC4000E is a superset of the XC4000 family, with higher speed, more routing, and edge-triggered synchronous write into the LUT-based RAM. The XC4000EX extends the XC4000E family to 3000 flip-flops, and adds a generous amount of routing resources. The XC4000,

XC4000E, and XC4000EX are 5V families, and as such are not generally recommended for new designs, because newer families offer better performance and lower cost.

The 3.3V XC4000XLA is an upgrade of the very popular 3.3V XC4000XL family, and the 2.5V XC4000XV extends the family to 18,000 flip-flop capacity. The XC4000XLA devices should be used where the more advanced features of the Virtex series (BlockRAM, clock management, and versatile I/O) are not needed. Use Virtex-E instead of XC4000XV for new designs.

**Spartan**

Spartan devices are functionally a subset of the XC4000E family, offering up to 2000 flip-flops at a significantly lower price. They are mainly used in cost-sensitive, high-volume (consumer) applications. Spartan FPGAs achieve lower manufacturing cost in several ways:

- The die are smaller.
- The manufacturing flow is streamlined.
- Speed, temperature, and package options are more limited.
- Configuration modes are bit-serial only.
- The pricing structure favors high-volume sales.

Spartan is a 5V family, and as such is not generally recommended for new designs.

**Spartan-XL and XC4000XLA**

The Spartan-XL and XC4000XLA families offer similar features and performance, where Spartan-XL covers the range of 360 to 2000 flip-flops, while XC4000XLA offers 1,500 to 7,000 flip-flop capacity. These 3.3V families should be used where the more advanced features of the Virtex series and Spartan-II are not needed (such as BlockRAM, clock management, and versatile I/O).

**Virtex, Virtex-E, and Virtex-EM**

The Virtex family is the biggest design project in Xilinx history and, judging by the number of early design-wins, is also the most successful. The Virtex architecture is rooted in XC4000 concepts (4-input look-up tables, usable as synchronous RAM), but the design started with a clean slate:

- The interconnect structure is generous, and is optimized for short and predictable delays.
- DLL-based fully digital clock-management eliminates on-chip and on-board clock delays.
- The device pins are compatible with many board-level I/O standards.
- Up to several hundred dual-ported BlockRAMs of 4Kb each.

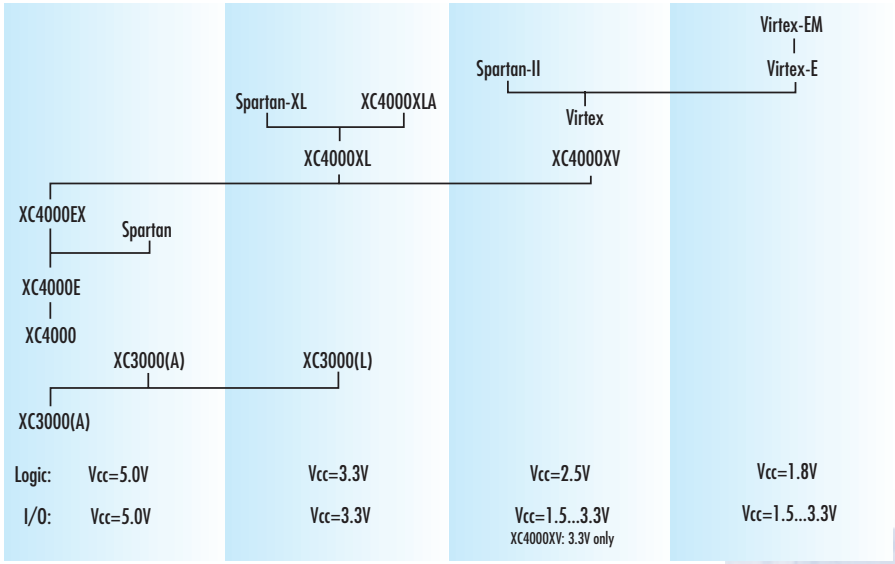



*Figure 1 - Xilinx FPGA Genealogy*

**The 2.5V Virtex family** covers the range from 1,800 to more than 27,000 flip-flops. The 2.5V Virtex pins are 5V tolerant, and the devices can implement 5V PCI.

**The 1.8V Virtex-E family** is an enhanced superset of the original Virtex family with two or three times the amount of BlockRAM, as well as support for differential I/O standards such as LVDS, BusLVDS, and LVPECL. At the high end, Virtex-E offers 73,000 flip-flops (>3 million system gates). The enhanced 0.18 micron process provides higher performance, but requires 1.8V for the core. The I/O uses up to 3.3 V, and is not 5V tolerant.

**The 1.8V Virtex-EM family** includes two devices that are electrically and architecturally identical with Virtex-E, but have significantly more BlockRAM (over 1 million bits in the XCV812E). Virtex-EM is also the first FPGA family using copper interconnect technology for lower interconnect resistance, higher speed, and better resistance to metal-migration problems.

The Virtex-E and -EM families are highly recommended for new designs, where they offer not only high speed and high capacity, but also valuable system-level features such as clock management, a versatile I/O structure, and substantial amounts of dual-ported BlockRAM. Virtex-EM is ideal for memory-intensive applications.

**Spartan-II**

Spartan-II extends the advanced features of the Virtex family, with 400 to 4,700 flip-flops. Spartan-II uses streamlined manufacturing methods and limited speed, temperature, and package options to address the cost-sensitive high-volume (consumer) market.

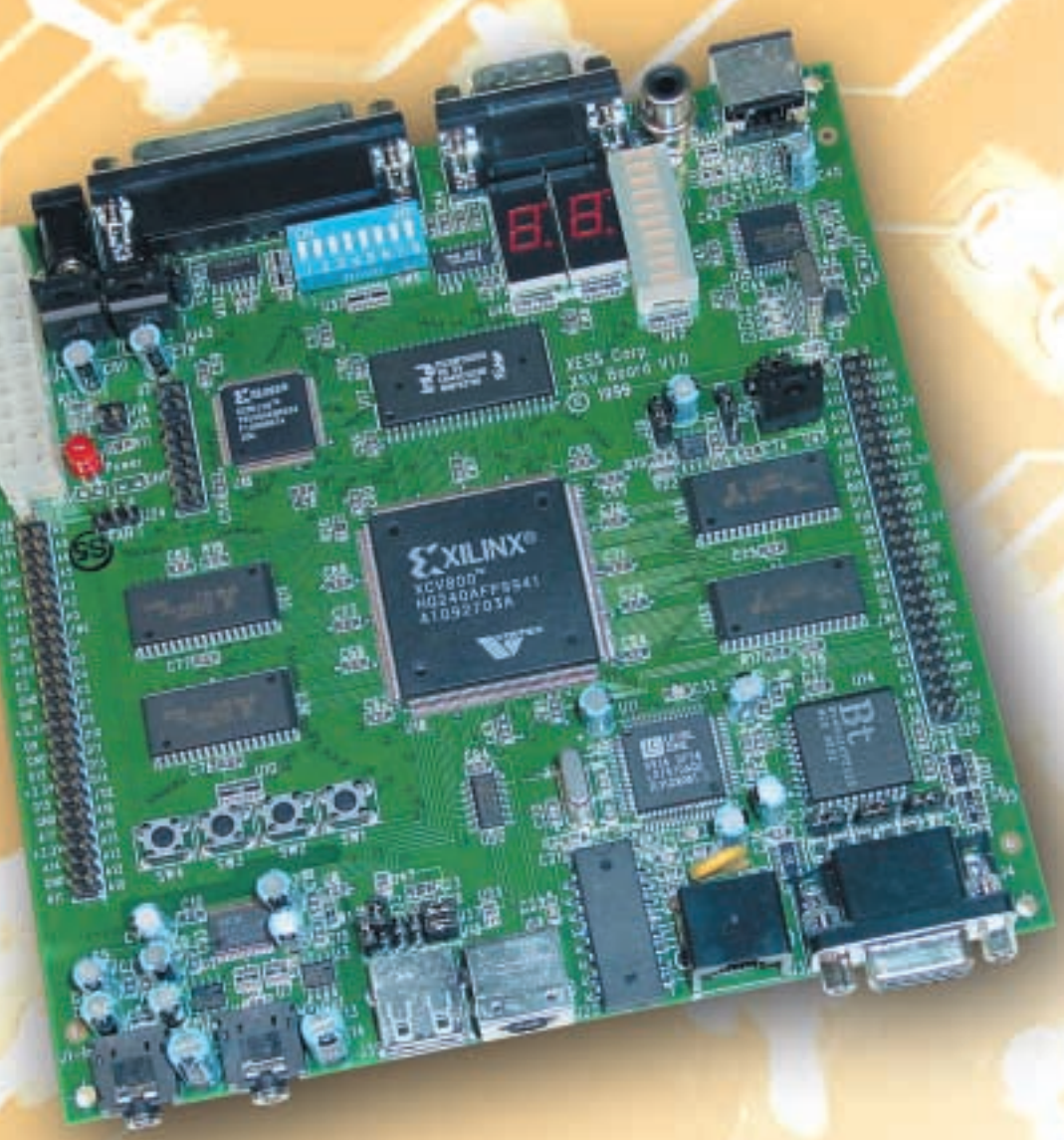
**Conclusion**

Xilinx offers many different programmable logic families to meet the needs of a wide range of applications. Make sure you are using the right technology today, so your designs will continue to be competitive tomorrow.

# New Virtex XSV Development Board

## for Creating Intellectual Property

The XSV Board, designed for the SIP developer community, provides a flexible, low-cost platform that supports a large set of interesting applications.




by Dave Vanden Bout
Product Manager, XESS Corp.
devb@xess.com

With the advent of large FPGAs such as the Xilinx Virtex series, the semiconductor intellectual property (SIP) market is poised for rapid expansion. Developers of open SIP serve a large segment of this market. With open SIP, the HDL source code and schematics for a functional core are made freely available and the developers generate revenue through service and support. But open SIP developers need an affordable, widely available platform for their cores to run on, just as the PC provided the platform for the growth of the open Linux OS. That's why we created the XSV Development Board

### XSV Features

The XSV Development Board from XESS Corp., shown in Figure 1, is a general-purpose Virtex FPGA development platform with extensions for multimedia I/O and network connectivity. The XSV houses a single Virtex chip in a 240-pin PQFP package, ranging from the XCV50 (50 K gates) up to the XCV800 (800 K gates). The wide range of device sizes in the same low-cost package lets XESS tailor the price of the board to meet the price requirements of

open SIP developers and users. Therefore you can move from smaller to larger FPGAs on the XSV Board without the need to rearrange your pin assignments.

### Video and Audio Capabilities

The XSV Board surrounds the Virtex chip with circuitry that supports many of today's important applications. A video decoder chip digitizes NTSC, PAL, and SECAM formats with up to nine bits of resolution. The FPGA can process the digitized video and store it in one of two independent banks of 512K x 16 SRAM, or it can send the video to a 24-bit RAMDAC for output on a VGA monitor. Video needs sound to go with it, so a 20-bit stereo codec lets the FPGA input and output two channels of audio.

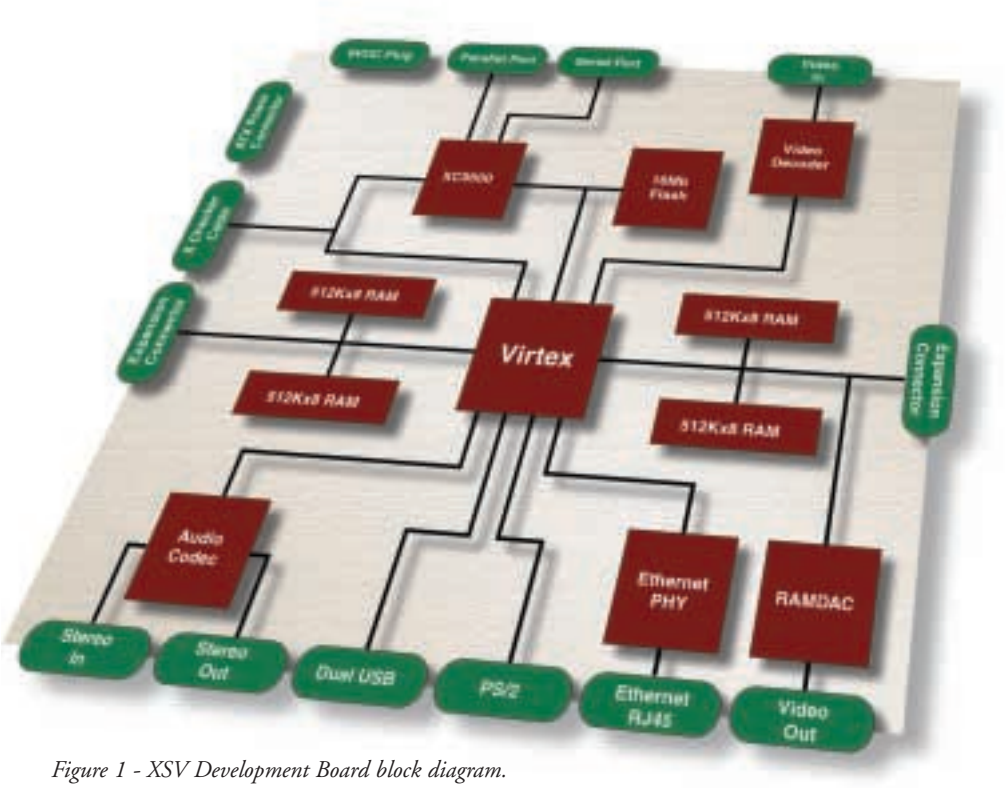



*Figure 1 - XSV Development Board block diagram.*

### Connectivity

To provide connectivity, the XSV Board has interfaces for the parallel, serial, and PS/2 keyboard/mouse ports common on PCs. There is also a USB port that connects a wide variety of devices to the XSV Board at up to 12 Mbps through a USB core in the FPGA. For even higher speeds, an Ethernet physical-level interface links the FPGA to a network at up to 100 Mbps. The FPGA can also communicate with external systems through two 50-pin I/O headers (38 general-purpose I/O, 12 power/ground).

### Configuration

A Xilinx XC95108 CPLD manages the configuration of the XSV Board. The CPLD accepts bitstreams via the parallel port and loads them into the FPGA. Or it can program the bitstream information into a 16 Mb Flash RAM on the XSV Board. The Flash RAM stores multiple bitstreams that are selectable with a DIP switch. The CPLD configures the FPGA with the selected bitstream upon power-up.

### Using XSV

The XSV Board is attached to a PC with a commonly available 25-wire parallel port cable, and is powered with a standard PC-ATX power supply. The Xilinx Foundation Series or Alliance Series development software is used to generate Virtex bitstreams which are loaded into the XSV Board with the GXSLOAD utility from XESS. XESS also provides utilities to run self-test diagnostics and to set the frequency of the programmable oscillator on the XSV Board.

XSV Board users can get example designs and application notes at www.xess.com and can subscribe to an email list of users who provide assistance to one another.

### Conclusion

Using the XSV Board, you can easily develop Intellectual Property for Xilinx FPGAs. This flexible, low-cost platform supports a variety of applications and helps you to quickly realize your designs. For developing IP, there is no faster or easier method.

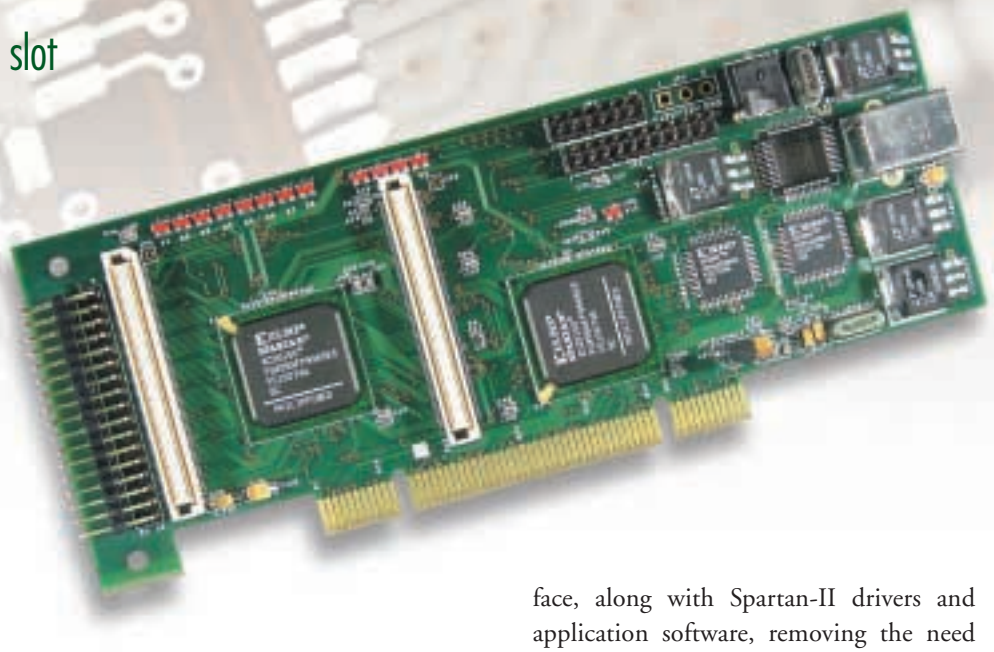
For more information contact XESS online at: www.xess.com, or send e-mail to: fpga-info@xess.com

## About XESS

**XESS Corporation is the leading provider of hybrid programmable logic/microcontroller development boards. Targeted at the educational market, XESS boards allow universities and students to build systems which combine elements of hardware and software and then quickly explore tradeoffs between the two. XESS sells its products directly to customers worldwide. Founded in 1990, XESS is privately held and is headquartered in Apex, NC.**

# Strathnuey - The Xilinx Technologies Integrator

## An entry level PCI card with DIME slot capability using Spartan-II FPGAs.



by Derek McAulay
Design Engineer, Nallatech Ltd
d.mcaulay@nallatech.com

The low profile PCI/USB card, Strathnuey, is the latest carrier card to be added to Nallatech's award winning DIME module product line. The Strathnuey pulls together, in a compact form factor, many of the exciting new technologies being produced by Xilinx for FPGAs. Support is included for the high speed SelectMAP interface (which forms the basis for partial reconfiguration), Chipscope ILA (giving built-in logic analyzer capability), Xilinx PCI LogiCORE development (for custom application development), and DSP IP core verification.

Using two of the latest Spartan-II devices the Strathnuey provides a versatile platform for application development. The Spartan-II family is used in diverse applications such as wireless digital communication systems, digital TV, high-speed DSL, cable modems, and medical imaging systems.

### DIME - DSP and Image processing Modules for Enhanced FPGAs

The DIME module slot, illustrated in Figure 1, gives access to the wide range of DIME modules, which you can use to customize the card to your selected application.

### DIME Select

To complement the Strathnuey, a new range of low profile modules, called DIME Select, are now available from Nallatech. These new modules enable you to customize the Strathnuey for your application while maintaining the low cost and features of the DIME standard. DIME Select modules will include AD/DA conversion, Bluetooth, SDRAM, QDR ZBT, and Ethernet.

### DIME Professional

If you require a higher level of performance and more processing power, the Virtex-based DIME Professional modules can also be mounted onto the Strathnuey. These DIME modules are all populated with the Virtex family and include a wide range of functions, such as video capture and display, high speed communications, data capture/generation, and complex DSP algorithms.

Whether you are using DIME Professional or DIME Select modules, the Strathnuey enables you to construct full custom FPGA-based systems using standard off the shelf products, guaranteeing risk reduction and a significant reduction in development costs and time to market.

### Configuration Software

The Strathnuey kit comes complete with compiled designs for the PCI/USB interface, along with Spartan-II drivers and application software, removing the need for required expertise in PCI or USB interfacing. The advanced system-level tools supplied are those from the established DIME range, which include plug and play capability and a standard interface which provides transparent migration to the more powerful and scalable DIME platforms such as the Ballynuey.

A key component of the tools is the integrated high speed configuration mechanism which allows the on board Spartan-II, or other attached FPGAs, to be configured directly over the PCI or USB buses, thus eliminating the need for dedicated download cabling and PROM programming. The configuration of the Spartan-II or Virtex-E FPGA (using the DIME slot) can be performed via the Xilinx SelectMap protocol, or alternatively via the JTAG chain, which allows access to the data and control registers, enabling partial reconfiguration. This feature naturally provides the mechanism to develop dynamically reconfigurable systems giving this card the capability of having multiple personalities.

The software tools run on all Windows platforms (95/98/NT/2000) and is one of the first FPGA-based system platforms to be supported under Linux.

### Communication Interface

Communication and control of the Strathnuey is provided by two separate mechanisms. Using the universal PCI interface enables the card to be used inside standard PC systems for embedded applications. Alternatively, for standalone or remote applications, the card can use the USB interface allowing it to be connected directly to a laptop, for example.

### IP Core Demonstration Platform

The flexibility and scalability of the Strathnuey make its range of applications wide and varied. In particular, the Strathnuey is an ideal demonstration environment for IP core suppliers to show their cores' capabilities working "live" with real hardware and data.

As an example, IP developers can easily send data to and from their cores via the straightforward infrastructure interfaces such that demonstration GUIs could be constructed to elegantly demonstrate any special features of their particular cores. Additionally, if required, a DIME module can provide the appropriate physical front end or back end interface to hook up the core with other systems for a complete demonstration system.

Integrated Silicon Systems (ISS), the leading multimedia IP core provider, selected DIME to demonstrate the capabilities of its applications-specific virtual components (ASVCs). "Performance and flexibility were key selection criteria when ISS went looking for a suitable hardware platform to demonstrate its multimedia and communications IP cores working in Xilinx Virtex technology." commented Stephen Farson, Engineering Manager, ISS. "Nallatech's DIME solution excelled in these areas."

### Educational Lab Experiments

The Strathnuey provides an excellent educational platform for Electronic Engineering courses, with the ability to go far beyond education in FPGA technology to complete system design. The simplicity of the software interface to the uncommitted secondary Spartan-II FPGA allows very elaborate Computer Based Training (CBT) courses to be based around the Strathnuey. These can be developed to educate the student in many areas and the integration of technologies such as the Xilinx Chipscope ILA allows the student to have a full debug workbench on his PC without the expense of additional test equipment.

The addition of DIME Select Modules will ensure that lecturers can educate their students in many areas including:

- FPGA technology.
- DSP theory.
- Communications theory.
- ADC/DAC principles including aliasing.
- Control theory.
- Hardware and software partitioning.

The new development system is the result of a close collaboration between Nallatech and the reconfigurable logic group led by Patrick Lysaght at the University of Strathclyde. The group has a 70-seat laboratory for teaching undergraduates digital design and has been using Xilinx FPGAs since 1989.

### PCI Development Platform

The Strathnuey can also be used to prove your own PCI interface designs when you are using the Xilinx 32-bit PCI LogiCORE or an in house PCI core. Support for 3.3V and 5V PCI is incorporated with auto detection circuitry to select the appropriate bit streams from the flash-based XCV1800 PROMs. These PROMs can easily be reconfigured with the user-specific bit stream.

### Conclusion

The Strathnuey gives you a complete platform that links the exciting new technologies from Xilinx. In addition, the Strathnuey provides an ideal entry-level DIME-based system development platform, which can also be used as a stepping stone to the high performance Ballynuey DIME Professional products.

*For more information on the Strathnuey or other DIME products, contact Nallatech at www.nallatech.com.*

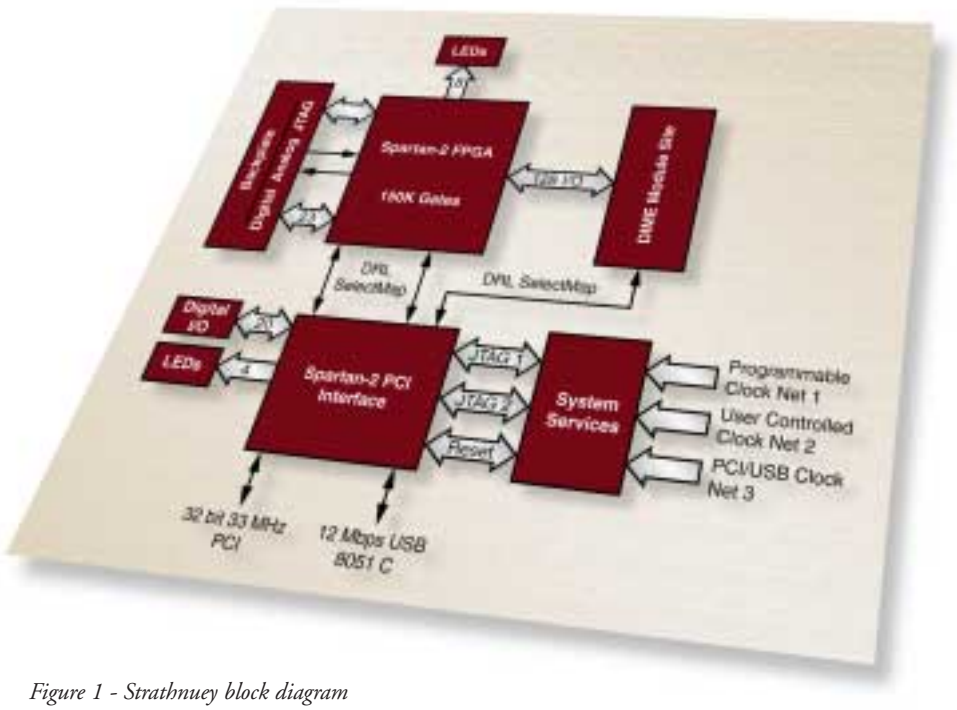
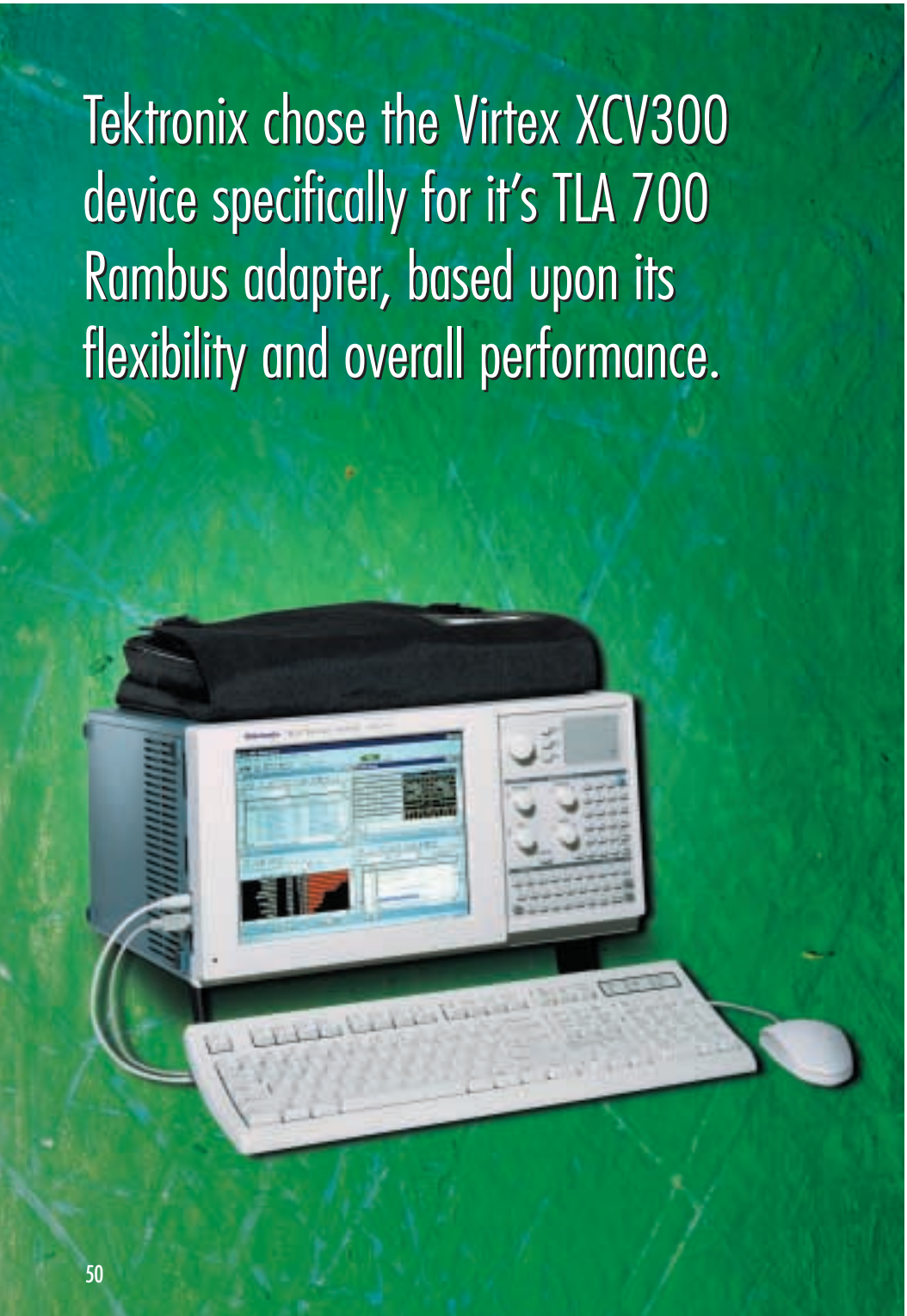



*Figure 1 - Strathnuey block diagram*

# Tektronix Logic Analyzers

## 800 Mbit/sec Using Virtex FPGAs

Tektronix chose the Virtex XCV300 device specifically for it's TLA 700 Rambus adapter, based upon its flexibility and overall performance.



by Tamara Snowden
Corporate PR Manager, Xilinx
tamaras@xilinx.com

Technology leader Tektronix, headquartered in Wilsonville, Oregon, recently announced its new suite of instruments designed for the most challenging leading-edge digital design applications. The integrated tool set is composed of new performance-leading instruments: the TLA 714/720 portable and benchtop logic analyzers, the TDS694C digital storage oscilloscope (DSO), and complementary connection devices. The instruments were designed to work together to provide specialized features and optimized performance.

### An Integrated Solution

"A digital design engineer operates in an environment of accelerating technological change under extreme time-to-market conditions," said Steve Jennings, director of marketing of the Tektronix' Measurement Business Division. "This new integrated solution provides superior measurement and analysis capabilities for even the most challenging design areas like Rambus memory systems and next-generation microprocessors."

### Choosing an Upgradable FPGA Solution

"The Virtex FPGA provides the flexibility necessary to implement protocol detection and reorganize the data for a delightful presentation to the Logic Analyzer and in so doing, delighting the customer," said Brainard Brauer, a Tektronix design engineer. "An ASIC was not an option in this design. The Virtex FPGA provides field upgradeabli-

ty of the code, which is critical in this fast moving, cutting-edge application."

"Virtex FPGAs have allowed Tektronix to de-serialize the Rambus data bus," said Brauer. Virtex devices accept Rambus data after presampling logic has converted the serial channel from 26 bits on both edges at 800Mbit/sec to 56 single edge signals to 400Mbits/sec. "Tektronix has been successful at using the Virtex FPGA to accept this 400MBits/sec data directly into the Virtex device." This was achieved by providing multi-phase clocks to the device (four phases). Each clock was connected to a different global clock input, routed over its own internal global route with separate DLL's (Delay Locked Loops).

"The device achieves a setup/hold specification better than 1.2ns worst case across temperature and at any pin of the device," said Brauer. "It is critical to verify that the tools have routed each data input to an IOB (Input/Output Block). In the Tektronix implementation, four input pins were consumed for each data line. As a result, four IOBs were required (one for each clock phase). The package loading must be carefully understood and compensated for to avoid signal slew rate issues.

### Choosing a Software Solution

Brauer used the Xilinx Alliance Series software to design the TMS810 Rambus interface adapter, which provides the crucial front end interface between Rambus and the TLA700 logic analyzer. The Alliance Series is the industry's leading open systems software that provides the flexibility to select the best EDA design environment for a specific application. Combining the advanced implementation technology of Xilinx with the strengths of its partners provides a powerful overall design solution, the highest clock performance and the highest densities in the industry.

### A New Family of Logic Analyzers

The new family of logic analyzers, the TLA 714 and the TLA 720, are replacing the original TLA 704/711. They offer an industry-leading combination of acquisition speed, channel width and memory depth, all of which are essential for supporting next-generation microprocessor designs. Now offering up to 16M, the TLA 714/720 have the deepest memory configuration in the industry plus an innovative hardware-assisted display system to simplify the management of such a large memory.

"High register count is a key to our bus decoding implementation, once deserialized into a wide parallel pipe line architecture. The Virtex device provided the lower power consumption we needed with much higher useable register densities than previous devices," said Brauer.

> "AN ASIC WAS NOT AN OPTION IN THIS DESIGN. THE VIRTEX FPGA PROVIDES FIELD UPGRADEABLITY OF THE CODE, WHICH IS CRITICAL IN THIS FAST MOVING, CUTTING-EDGE APPLICATION."

The new TLA 700 features an easy-to-use Windows 98 user interface and a PC platform with expanded openness in response to customers' strong acceptance of the original TLA 700's open platform. Tektronix built in an industry-standard computer and operating system to provide the user with a familiar interface. Because it works just like any other PC-based software, the user can focus on the problem, rather than the tool. Tektronix has also created the Embedded Systems Tools Partners Program to deliver development and debug solutions for the TLA 700 Series. The solutions range from providing software, analysis tools and physical processor connections, to disassembly software that runs on the logic analyzer.

## About Tektronix

**Tektronix is a portfolio of measurement, color printing, and video and networking businesses dedicated to applying technology excellence to customer challenges. Headquartered in Wilsonville, Oregon, Tektronix has operations in 26 countries outside the United States. Founded in 1946, the company had revenues of $2.1 billion in fiscal 1998. For more information, visit the website at www.tek.com.**

At the heart of all of the TLA 700 Series logic analyzer modules is a breakthrough called MagniVu™ acquisition technology, a super-high-speed sampling architecture that dramatically changes the way logic analyzers work and what functionality they offer. All incoming data is oversampled at a 2GHz rate, regardless of how the logic analyzer is being used. The oversampled data is then processed in real time to perform timing acquisition, state acquisition, and triggering without missing the slightest piece of crucial timing information on any channel.

# McData Uses Xilinx FPGAs for Fibre-channel Switch

When IBM needs state of the art Fibre-Channel hardware, it thinks McDATA. When McDATA needs FPGAs, it thinks Xilinx.

by Tamara Snowden
Corporate PR Manager, Xilinx
tamaras@xilinx.com

McDATA Corporation, headquartered in Broomfield, CO, is the architect of the first enterprise-wide SAN (Storage Area Network) solution. McDATA specializes in highly available, scalable and centrally managed enterprise SANs, providing customers with hardware and software that allow dynamic connection between the data center and edge servers in large enterprise data centers. McDATA also offers design, implementation planning, integration testing, and training services for companies building enterprise SANs.

### Designing a Bridge Card

In the late 1990s McDATA designed a 128-port 9032 Model 5 Director for IBM- a super switch for interconnecting mainframe computers and high speed peripherals like disk and tape arrays. IBM came to McDATA again when the company needed an internal bridge card to manage the optoelectronic, framing, and format conversion from newer, IEEE One Gigabit Fibre-Channel connections (or FICON) for the 9032 (Shown in Figure 1). Originally designed with the 200-Megabit ESCON protocol in mind, to the Model 5 would be the first of IBM's switches to support FICON.

Each new bridge card would perform the work of up to eight ESCON channels. Up to 16 FICON Bridge cards (equivalent to 128 ESCON ports or one half of the Model 5's capacity) would be supported in a single 9032 Model 5 Director.

Retrofitting the FICON bridge card would allow data centers to use the much higher speed of FICON without replacing the 9032 Directors. It would also give IBM's customers the ability to optimize for either raw speed or through multiplexing-larger arrays of storage devices. Except for the costs associated of the bridge card itself, upgrading the Director would be nearly painless, and conserve the customer's investment in a costly piece of high-performance hardware-the 9032 Model 5 has a backplane with full-duplex, 1-Gigabit capacity.

> "It was McDATA's positive experience that made Xilinx the choice of the engineering team. They also knew that they didn't need to push Xilinx technology; there was capability in reserve."

### Choosing FPGAs Over ASICs

Roughing out the block diagram for the new FICON card was straightforward. The bridge card would handle multiplexing and format conversion; IBM would perform protocol conversion in the mainframe. The 1-Gigabit optical input is steered to an optoelectronic converter, and from there to a 32-bit, 26-Mhz parallel bus. Next is a framer, then a Fibre-Channel high-level protocol converter, followed by eight IBM-supplied ESCON engines the output of which connect to the 9032 Director's backplane.

Xilinx FPGAs form the heart of the bridge card's framer and protocol conversion sections. Originally, an ASIC had been targeted for these roles. But ASIC development is risky and time consuming and project engineer Paul Hwang needed to minimize risk and get the product to market as quickly as possible. Hwang decided to use Xilinx FPGAs and design tools rather than develop an ASIC. The team would take advantage of FPGA reprogrammability to design, develop, and debug as they went along.

Using multiple FPGAs sped up the effort further. McDATA found that by partitioning the design into modules, each with its own FPGA, they could assign each part to a different designer or team for easy to manage and faster parallel development.

Xilinx FPGAs were also valuable because there were so many unknowns in the project. In fact, once the decision had been made to use FPGAs, McDATA decided to optimize the bridge card architecture around them, to get the most out of the programmability. This helped minimize the impact of specification changes, bugs, and implementation changes. Use of the FPGAs drastically cut development risk, and development moved along far faster than the ASIC approach.

"We knew Xilinx' product philosophy, having been a Xilinx customer for five years, and were very comfortable with it and with the company's product," says Hwang . "It was McDATA's positive experience that made Xilinx the choice of the engineering team. They also knew that they didn't need to push Xilinx technology; there was capability in reserve."

### The Design Process

The design process went smoothly using standard Xilinx Alliance Series( software tools. "We used the scripting capabilities to the fullest so that we could automate as much of the process as possible," says Hwang. In addition, McDATA used the Xilinx EPIC FPGA editor to help debug the design by bringing out internal signals to unused I/O pins, something that would not have been possible with an ASIC—another time saver and another advantage of the FPGA approach.

McDATA used several different Xilinx products to implement the bridge card, among them the XC4013XL, XC4028XL, and XC4044XL FPGAs, and an XC9500 CPLD. The CPLD was used to interface to an Intel i960 processor which downloaded the bitmaps into the FPGAs. As the design evolved, McDATA started with relatively large FPGAs, but as they altered and optimized the design, they moved to smaller die sizes especially with respect to the XC4013s. McDATA is also moving to Xilinx XLA technology FPGAs to "greatly help with cost reduction, while continuing to minimize program risks," according to Hwang.

The process went so well that McDATA is sticking with its original FPGA implementation, especially since this allows for the design to be further refined. As is usually the case, McDATA originally anticipated going to an ASIC to minimize costs. But now, production of the FICON Bridge Card is well underway using Xilinx FPGAs. The cost reductions made using Xilinx XLA-class FPGAs have been significant, and consequently, there are no immediate plans to switch to a custom chip.

> "The risk was too high with the ASIC approach. In hindsight, we made exactly the right decision to go with Xilinx FPGAs."

### Conclusion

Looking back on the development Hwang says, "The risk was too high with the ASIC approach. In hindsight, we made exactly the right decision to go with Xilinx FPGAs."

# The "Flancter"

## How to set a status flag in one clock domain, clear it in another, and never have to use an asynchronous clear for anything but reset.

by Rob Weinstein
Senior Member of Technical Staff,
Memec Design Services

There are times when it's important to generate a status flag that is set by an event in one clock domain and reset by an event in a different clock domain. Using a D-type flip-flop, where a "1" is clocked in from one clock domain and its asynchronous reset is pulsed by logic in the second clock domain, is the time-honored method to achieve this function. While there is nothing logically wrong with this, it introduces other problems such as combinational logic driving an asynchronous reset pin, uncertainty in timing constraint boundaries, and muddying the global reset function.

Here, I present an alternative method for generating a multiple clock domain flag register that mitigates these problems. It's called the "Flancter" (named by my colleague, Mark Long), and is shown in Figure 1.

As you can see, it's made up of two D-type flip-flops, an inverter, and an exclusive OR (XOR) gate. Notice that the asynchronous reset inputs to the flip-flops are shown unconnected for clarity only. Normally, these would be tied to the global set/reset net in the system.

Operation of the Flancter is simple; when FF1 is clocked (rising edge of SET_CLK while SET_CE is asserted), OUT goes high. When FF2 is clocked (rising edge of RESET_CLK while RESET_CE is asserted), OUT goes low. Note that this circuit must be used in an interlocked system where the flip-flops won't be continuously clocked by the two clock domains. Also, the output must be synchronized with additional flip-flops to mitigate metastability when crossing clock domains (more about this later).

### How It Works

To explain its operation, I like to rearrange the circuit as shown in Figure 2.

This is the same circuit as shown in Figure 1, but untwisted so that the two inputs to the XOR gate are clearly visible. You can see that the XOR gate's upper input is labeled Q1, while its lower input is labeled Q2. Also, Q1 and Q2 are the Q outputs of FF1 and FF2, respectively. Now for the trick part of this magic trick: the D input to FF1 comes from an inverter, so whenever FF1 is clocked, Q1 assumes the opposite state of Q2 and the output of the XOR

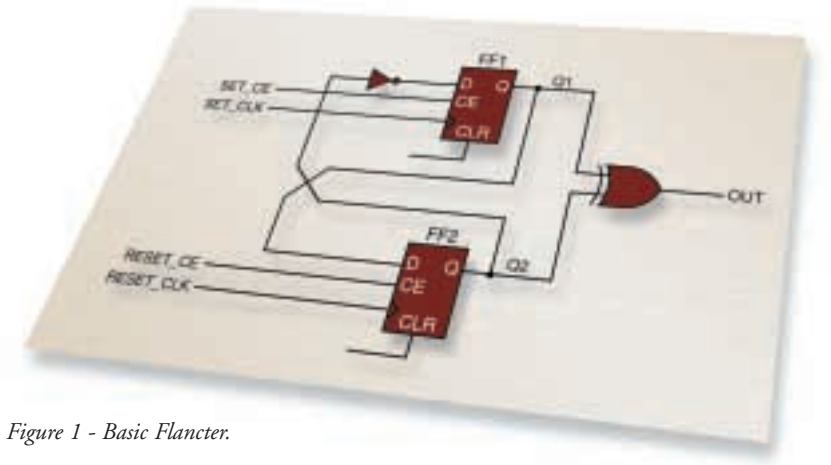



*Figure 1 - Basic Flancter.*

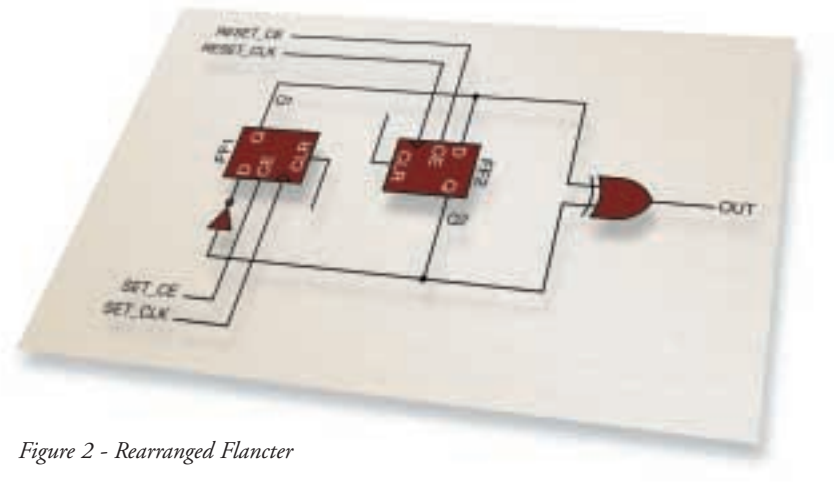



*Figure 2 - Rearranged Flancter*

gate will go high. When FF2 is clocked, Q2 becomes the same as Q1, and the output will go low. In summary, clocking FF1 causes OUT to go high and clocking FF2 causes OUT to go low.

A timing diagram will help describe the Flancter's operation. Figure 3 shows the basic timing diagram.

The basic points of interest in the timing diagram are:

- SET_CLK and RESET_CLK are asynchronous to each other.
- At point A, the rising edge of SET_CLK while SET_CE is high causes Q1 to go high because it gets the inverted value of Q2. Also, OUT goes high because it is the XOR of Q1 and Q2.
- At point B, the rising edge of RESET_CLK while RESET_CE is high causes Q2 to go high because it gets the value of Q1. Also, OUT goes low because it is the XOR of Q1 and Q2.
- At point C, Q1 again gets the inverted value of Q2, causing OUT to go high.
- At point D, Q2 goes low because it gets the value of Q1, causing OUT to go low.

### So What's Wrong With It?

There are a few things wrong with the Flancter from the start. One problem is that it uses two flip-flops to create a single flag bit. This is a minor fault when you consider that most FPGAs have an abundant supply of flip-flops. A more serious issue is how to use the output. Remember that the output can change synchronously to either clock domain. You need to resynchronize the output to whichever clock domain needs to see it; often both clock domains. It is common to use two flip-flops in series as a metastability-resistant synchronizer.

However, the most serious drawback to the Flancter is that operating the set and reset flip-flops must be mutually exclusive in time. This means that when logic in clock domain 1 sets the Flancter, it doesn't attempt to set the Flancter again until it sees that it has been reset. Likewise, the logic in clock domain 2 never attempts to reset the Flancter unless it sees that it has been set. Establishing this kind of interlocked protocol guarantees that both of the Flancter's flip-flops won't be clocked simultaneously (or within each other's setup and hold time windows).

### Applications of the Flancter

There are many applications of the Flancter, but a very common application is interfacing a microprocessor to an FPGA. Typically, the microprocessor and FPGA logic run on separate clocks. When the microprocessor writes a control register within the FPGA, the Flancter can be used as a status flag to tell an internal state machine that new data is available.

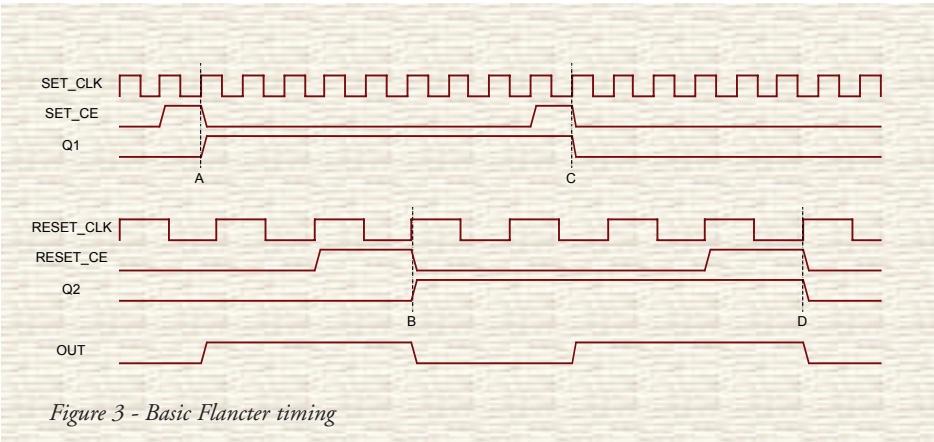



*Figure 3 - Basic Flancter timing*

Likewise, a state machine within the FPGA can use the Flancter to generate an interrupt to the microprocessor that is subsequently cleared by a read or interrupt-acknowledge cycle from the microprocessor, as shown in Figure 4.

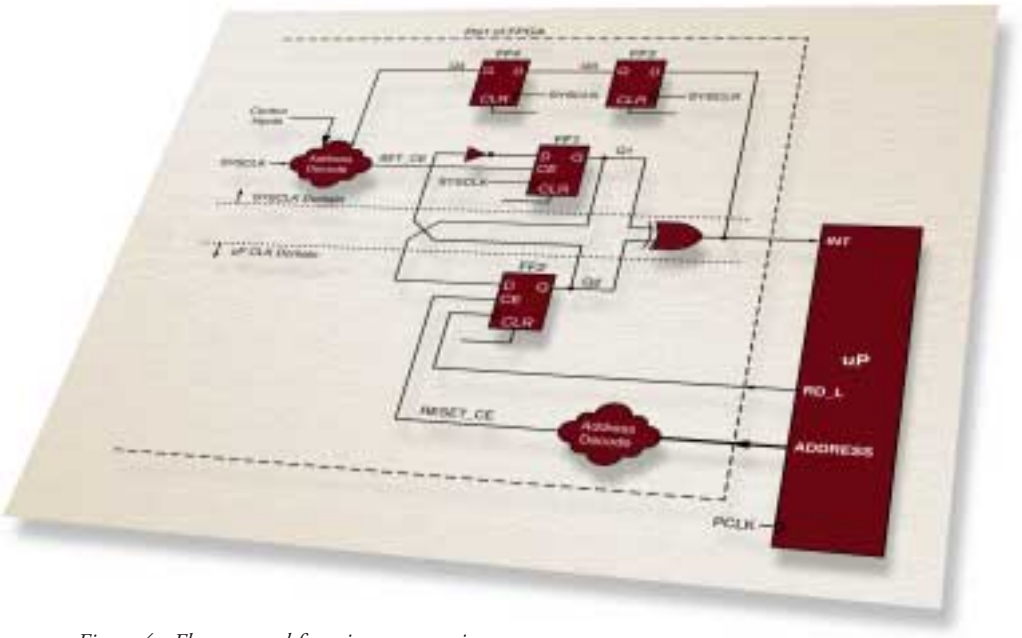



*Figure 4 - Flancter used for microprocessor interrupt*

Things to notice in Figure 4:

- The Flancter is made up of FF1, FF2, the inverter, and the XOR gate.
- The state machine, FF1, FF3, and FF4 are all synchronous to SYSCLK.
- The microprocessor (µP) runs off its own clock, PCLK.
- The state machine pulses SET_CE for one SYSCLK cycle when it needs to request an interrupt.
- The microprocessor performs a read cycle from a predefined address to reset the interrupt. Although not shown, reading from this address may also cause a status register to be driven onto the microprocessor's data bus allowing simultaneous reading of status and resetting of interrupt.
- FF3 and FF4 are resynchronizing flip-flops used to filter any metastable logic conditions from propagating into the state machine.
- The particular microprocessor used in this example employs metastable resistant techniques on its INT input.
- The interrupt sequence is defined such that setting and resetting the interrupt flag cannot occur simultaneously.

The following timing diagram helps illustrate the operation:

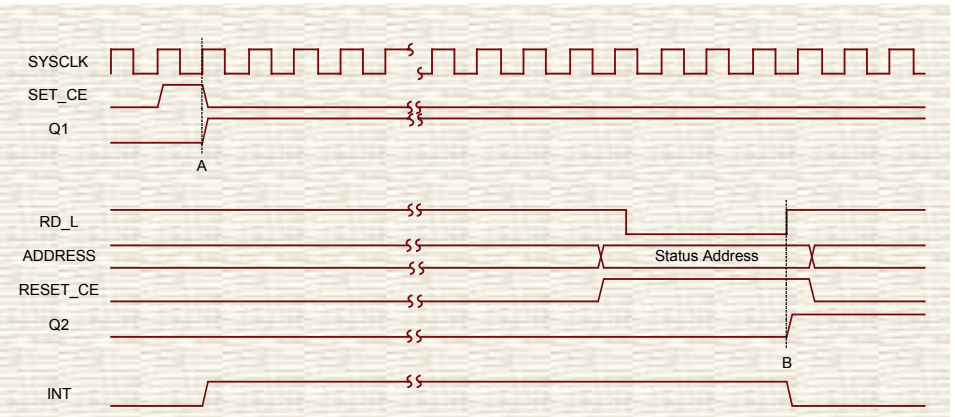



*Figure 5 - Flancter interrupt timing diagram*

Things to notice in Figure 5:

- The FPGA's state machine sets the interrupt request (INT) at point A.
- Sometime later, the microprocessor responds to the interrupt by reading a status register, thus resetting the interrupt request at point B.

### Variations on a Flancter

While the basic Flancter is very simple, many useful variations are possible. I describe some of the more interesting variations in the Flancter app-note available on our website, http://www.memecdesign.com/resources/guides/. Some of the variations you'll find there include the 3-way Flancter, the n-way Flancter, the async-reset emulating Flancter, and the default-to-active-high Flancter. The app-note also has VHDL and Verilog listings for the basic Flancter.

### Conclusion

I turned 37 this year and I got to thinking that all the "greats" did their best work long before they reached my age. As I look back on my design career, I realize that I haven't designed any circuits or developed any theorems that will bear my name like the Pierce Oscillator or Shannon's Sampling Theorem. The best I can do is to offer the Flancter as my legacy. Perhaps someday, the Weinstein-Flancter will be found in the indexes of engineering tomes right between Watt-Hour and Wien-Bridge.

# Distribution Adds Value:

## Intel StrongARM Supported with Xilinx Spartan-II FPGAs

Avnet is a key member of the Xilinx sales force, helping our customers create solid designs using innovative development tools

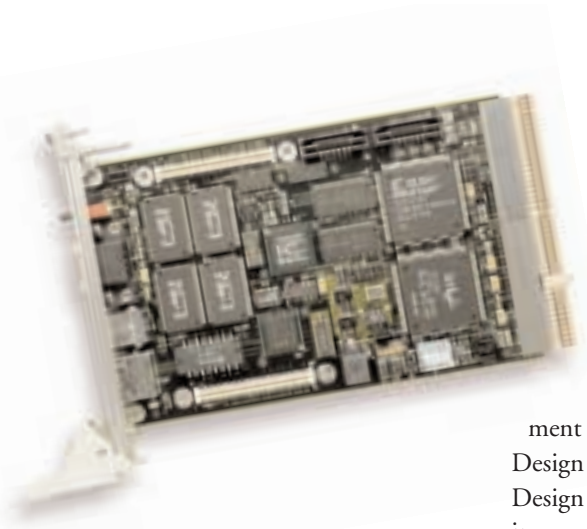
by Russ Sinagra
Marketing Manager GPD, Xilinx
russell.sinagra@xilinx.com

A number of Avnet customers, in various locations across North America, were looking for a simple and flexible way to develop a PCI interface to an Intel StrongARM processor, and several Avnet FAEs (Field Applications Engineers) across the country helped their customers develop unique solutions for their individual designs. Then, during an Avnet FAE design review where the PCI requirements of different customers were discussed, it became apparent that there were a number of common requirements for a PCI interface—and they realized there was a common solution to many different problems.

By using Spartan-II FPGAs from Xilinx, the Avnet FAEs knew they had a great opportunity to provide an aggressively priced yet flexible PCI connection that would meet the needs of many customers. The FAEs knew that this would be a great way to highlight a number of Avnet Design Services (ADS) strengths as well.

### The Catapult™ StrongARM PCI Development Kit

Avnet combined their ADS board-level solution services, their FPGA services, and their ability to create Intellectual Property, to create the Catapult StrongArm PCI Development Kit, a solution that covers many needs (Fig.1). This kit gives you a time-to-market boost by integrating all the key product technologies into a single development environment.



A StrongArm SA1110 Processor provides the processing power and a Xilinx Spartan-II XC2S100 implements a glueless interface between the SA1110 processor bus and the industry standard PCI bus. Along with these hardware resources, Avnet Design Services provides IP cores for PCI, SDRAM control, and the StrongArm interface as a starting point for your specific application. In addition to the IP core, software is provided to help develop your embedded application. Applications development software is also available from Avnet Design Services or third parties. Avnet Design Services can also provide engineering consulting services, helping you to quickly customize your design to further speed your time to market.

The kit sells for $8,995.00 and is available now from your local Avnet distributor. For more information, visit the Avnet site at: http://www.ads.avnet.com

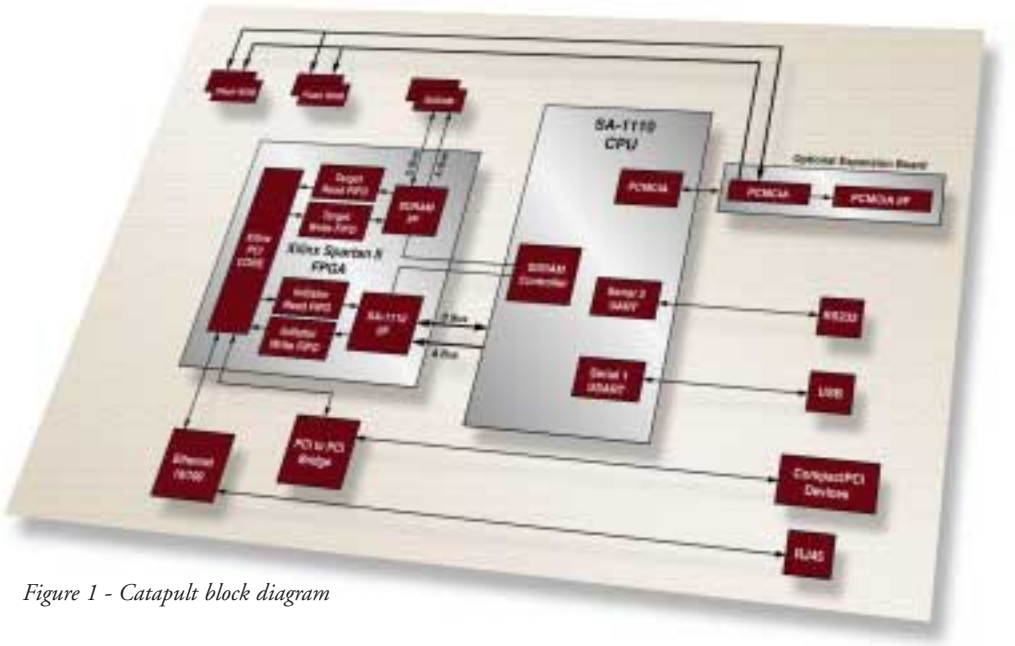



*Figure 1 - Catapult block diagram*

# Virtex and XC4000X Series FPGAs

Each Virtex family has its own unique features to meet different application requirements. All devices have both distributed RAM and block RAM, and between four and eight DLLs for efficient clock management.

- The Virtex family, consisting of devices that range from 50K up to 1 million logic gates, supports 17 I/O standards, and offers 5V PCI compliance.
- The Virtex-E family offers the highest logic gate count available for any FPGA, ranging from 50K up to 3.2 million system gates, and supports 20 I/O standards including LVPECL, LVDS, and Bus LVDS differential signaling.
- The Virtex-EM Extended Memory family consists of two devices that have a high RAM-to-logic gate ratio that is targeted for specific applications such as gigabit per second network switches and high definition graphics.

The XC4000X Series is part of the broad spectrum of Xilinx "XL" products unveiled September, 1998. As a result, Xilinx offers the broadest choice of 3.3 volt and 2.5 volt devices available from a single supplier, with densities ranging from 800 to 500,000 system gates. With 12 family members ranging from 30,000 to 500,000 system gates, the devices feature patented SelectRAM memory, with a highly flexible arrangement of logic, single-port, or dual-port memory.

Designed in an advanced 0.25 micron process, the XC4000X series delivers industry-leading performance while significantly reducing power consumption.

***See www.xilinx.com for more information.***

**FPGA Product Selection Matrix**

| | | DENSITY | | | | | | | FEATURES | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DEVICES | KEY FEATURES | Logic Cells | Maximum Logic Gates | Typical System Gate Range | Max. RAM Bits | CLB Matrix | CLBs | Flip-Flops | Max. I/O | Output Drive (mA) | PCI Compliant | 1.8 Volt | 2.5 Volt | 3 Volt | 5 Volt |
| XC4013XLA | XC4000 Series: Density Leadership/ High Performance/ SelectRAM Memory | 1368 | 13K | 10K-30K | 18K | 24x24 | 576 | 1536 | 192 | 12/24 | Y | – | – | – | X |
| XC4020XLA | | 1862 | 20K | 13K-40K | 25K | 28x28 | 784 | 2016 | 205 | 12/24 | Y | – | – | – | X |
| XC4028XLA | | 2432 | 28K | 18K-50K | 33K | 32x32 | 1024 | 2560 | 256 | 12/24 | Y | – | – | – | X |
| XC4036XLA | | 3078 | 36K | 22K-65K | 42K | 36x36 | 1296 | 3168 | 288 | 12/24 | Y | – | – | – | X |
| XC4044XLA | | 3800 | 44K | 27K-80K | 51K | 40x40 | 1600 | 3840 | 320 | 12/24 | Y | – | – | – | X |
| XC4052XLA | | 4598 | 52K | 33K-100K | 62K | 44x44 | 1936 | 4576 | 352 | 12/24 | Y | – | – | X | * |
| XC4062XLA | | 5472 | 62K | 40K-130K | 74K | 48x48 | 2304 | 5376 | 384 | 12/24 | Y | – | – | X | * |
| XC4085XLA | | 7448 | 85K | 55K-180K | 100K | 56x56 | 3136 | 7168 | 448 | 12/24 | Y | – | – | X | * |
| XCV50 | Virtex Family: Density/ Performance Leadership BlockRAM Distributed RAM SelectI/O 4 DLLs | 1728 | 21K | 34K-58K | 56K | 16x24 | 384 | 1536 | 180 | 2/24 | Y | – | – | X | * |
| XCV100 | | 2700 | 32K | 72K-109K | 78K | 20x30 | 600 | 2400 | 180 | 2/24 | Y | – | – | X | * |
| XCV150 | | 3888 | 47K | 93K-165K | 102K | 24x36 | 864 | 3456 | 260 | 2/24 | Y | – | X | I/O | * |
| XCV200 | | 5292 | 64K | 146K-237K | 130K | 28x42 | 1176 | 4704 | 284 | 2/24 | Y | – | X | I/O | * |
| XCV300 | | 6912 | 83K | 176K-323K | 160K | 32x48 | 1536 | 6144 | 316 | 2/24 | Y | – | X | I/O | * |
| XCV400 | | 10800 | 130K | 282K-468K | 230K | 40x60 | 2400 | 9600 | 404 | 2/24 | Y | – | X | I/O | * |
| XCV600 | | 15552 | 187K | 365K-661K | 312K | 48x72 | 3456 | 13824 | 512 | 2/24 | Y | – | X | I/O | * |
| XCV800 | | 21168 | 254K | 511K-888K | 406K | 56x84 | 4704 | 18816 | 512 | 2/24 | Y | – | – | X | * |
| XCV1000 | | 27648 | 332K | 622K-1,124K | 512K | 64x96 | 6144 | 24576 | 512 | 2/24 | Y | – | – | X | * |
| XCV50E | Virtex-E Family: Density/ Performance Leadership BlockRAM Distributed RAM SelectI/O+ 8 DLLs LVDS, BLVDS, LVPECL | 1728 | 21K | 47K-72K | 88K | 16x24 | 384 | 1536 | 176 | 2/24 | Y | X | I/O | I/O | ** |
| XCV100E | | 2700 | 32K | 105K-128K | 118K | 20x30 | 600 | 2400 | 176 | 2/24 | Y | X | I/O | I/O | ** |
| XCV200E | | 5292 | 64K | 215K-306K | 186K | 28x42 | 1176 | 4704 | 284 | 2/24 | Y | X | I/O | I/O | ** |
| XCV300E | | 6912 | 83K | 254K-412K | 224K | 32x48 | 1536 | 6144 | 316 | 2/24 | Y | X | I/O | I/O | ** |
| XCV400E | | 10800 | 130K | 413K-570K | 310K | 40x60 | 2400 | 9600 | 404 | 2/24 | Y | X | I/O | I/O | ** |
| XCV600E | | 15552 | 187K | 679K-986K | 504K | 48x72 | 3456 | 13824 | 512 | 2/24 | Y | X | I/O | I/O | ** |
| XCV1000E | | 27648 | 332K | 1,146K-1,569K | 768K | 64x96 | 6144 | 24576 | 660 | 2/24 | Y | X | I/O | I/O | ** |
| XCV1600E | | 34992 | 420K | 1,628K-2,189K | 1062K | 72x108 | 7776 | 31104 | 724 | 2/24 | Y | X | I/O | I/O | ** |
| XCV2000E | | 43200 | 518K | 1,857K-2,542K | 1240K | 80x120 | 9600 | 38400 | 804 | 2/24 | Y | X | I/O | I/O | ** |
| XCV2600E | | 57132 | 686K | 2,221K-3,264K | 1530K | 92x138 | 12696 | 50784 | 804 | 2/24 | Y | X | I/O | I/O | ** |
| XCV3200E | | 73008 | 876K | 2,608K-4,074K | 1846K | 104x156 | 16224 | 64896 | 804 | 2/24 | Y | X | I/O | I/O | ** |
| XCV405E | Virtex Extended Memory Capabilities | 10800 | 130K | 1,068K-1,307K | 710K | 40x60 | 2400 | 9600 | 404 | 2/24 | Y | X | I/O | I/O | ** |
| XCV812E | | 21168 | 254K | 2,569K-3,062K | 1414K | 56x84 | 4704 | 18816 | 556 | 2/24 | Y | X | I/O | I/O | ** |

* I/Os are 5V tolerant
** 5 Volt tolerant I/Os with external resistor
X = Core and I/O voltage
I/Os = I/O voltage supported


Say hello to a new level of performance; the Spartan-II family now includes devices with over 200,000 system gates, and you get 100,000 system gates for under $10, at speeds of 200Mhz and beyond, giving you design flexibility that's hard to beat. These low-powered, 2.5-V devices feature I/Os that operate at up to 3.3V with full 5-V tolerance. Spartan-II devices also feature multiple Delay Locked Loops, on-chip RAM (block and distributed), and versatile I/O technology that supports over 16 high-performance interface standards. You get all this in an FPGA that offers unlimited programmability, and can even be upgraded in the field, remotely, over any network.

### Robust Feature Set

- Flexible on-chip distributed and block memory.
- Four digital Delay Locked Loops for efficient chip-level/board-level clock management.
- Select I/O Technology for interfacing with all major bus standards such as HSTL, GTL, SSTL, and so on.
- Full PCI compliance.
- System speeds over 200 MHz.
- Power management.

### Extensive Design Support

- Complete suite of design tools.
- Extensive core support.
- Compile designs in minutes.

### Advantages Over ASICs

- No costly NRE charges.
- No time consuming vector generation needed.
- All devices are 100% tested by Xilinx.
- Field upgradeable (remotely upgradeable, using Xilinx Online technology).
- No lengthy prototype or production lead times.
- Priced aggressively against comparable ASICs.

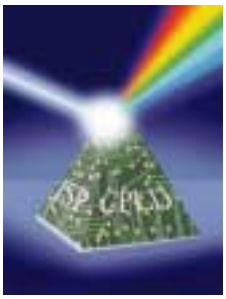
***For more information see www.xilinx.com/products/spartan2***

**FPGA Product Selection Matrix**

| | | DENSITY | | | | | | | FEATURES | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DEVICES | KEY FEATURES | Logic Cells | Maximum Logic Gates | Typical System Gate Range | Max. RAM Bits | CLB Matrix | CLBs | Flip-Flops | Max. I/O | Output Drive (mA) | PCI Compliant | 1.8 Volt | 2.5 Volt | 3.3 Volt | 5.0 Volt |
| XCS05 | Spartan Family: High Volume ASIC Replacement/ High Performance/ SelectRAM Memory | 238 | 3K | 2K-5K | 3K | 10x10 | 100 | 360 | 77 | 12 | Y | – | – | – | X |
| XCS10 | | 466 | 5K | 3K-10K | 6K | 14x14 | 196 | 616 | 112 | 12 | Y | – | – | – | X |
| XCS20 | | 950 | 10K | 7K-20K | 13K | 20x20 | 400 | 1120 | 160 | 12 | Y | – | – | – | X |
| XCS30 | | 1368 | 13K | 10K-30K | 18K | 24x24 | 576 | 1536 | 192 | 12 | Y | – | – | – | X |
| XCS40 | | 1862 | 20K | 13K-40K | 25K | 28x28 | 784 | 2016 | 205 | 12 | Y | – | – | – | X |
| XCS05XL | Spartan-XL Family: High Volume ASIC Replacement/ High Performance/ SelectRAM Memory | 238 | 3K | 2K-5K | 3K | 10x10 | 100 | 360 | 77 | 12/24 | Y | – | – | X | * |
| XCS10XL | | 466 | 5K | 3K-10K | 6K | 14x14 | 196 | 616 | 112 | 12/24 | Y | – | – | X | * |
| XCS20XL | | 950 | 10K | 7K-20K | 13K | 20x20 | 400 | 1120 | 160 | 12/24 | Y | – | – | X | * |
| XCS30XL | | 1368 | 13K | 10K-30K | 18K | 24x24 | 576 | 1536 | 192 | 12/24 | Y | – | – | X | * |
| XCS40XL | | 1862 | 20K | 13K-40K | 25K | 28x28 | 784 | 2016 | 224 | 12/24 | Y | – | – | X | * |
| XC2S15 | Spartan-II Family: High Volume BlockRAM Distributed RAM SelectI/O 4 DLLs | 432 | 8K | 6K-15K | 22K | 8x12 | 96 | 384 | 86 | 2/24 | Y | – | X | I/O | * |
| XC2S30 | | 972 | 17K | 13K-30K | 36K | 12x18 | 216 | 864 | 132 | 2/24 | Y | – | X | I/O | * |
| XC2S50 | | 1728 | 30K | 23K-50K | 56K | 16x24 | 384 | 1536 | 176 | 2/24 | Y | – | X | I/O | * |
| XC2S100 | | 2700 | 53K | 37K-100K | 78K | 20x30 | 600 | 2400 | 196 | 2/24 | Y | – | X | I/O | * |
| XC2S150 | | 3888 | 77K | 52K-150K | 102K | 24x36 | 864 | 3456 | 260 | 2/24 | Y | – | X | I/O | * |
| XC2S200 | | 5292 | 103K | 71K-200K | 130K | 28x42 | 1,176 | 4704 | 284 | 2/24 | Y | – | X | I/O | * |

* I/Os are tolerant
X = Core and I/O voltage
I/Os = I/O voltage supported

# XC9500 and CoolRunner CPLDs



Whether performing high speed networking-based or power-conscious portable designs, Xilinx CPLDs provide you with a complete range of value oriented products.

**XC9500 -** Offers industry-leading speeds, while giving you the flexibility of an enhanced customer-proven pin-locking architecture along with extensive IEEE Std. 1149.1 JTAG Boundary-Scan support.

**CoolRunner -** Offers the patented Fast Zero Power (FZP() design technology, combining low power and high speed. These devices offer standby currents of less than 100 microamps, operating currents 50-67% lower than traditional CPLDs, and pin-to-pin speeds of 5.0ns.

**WebPOWERED Software Solutions -** Offer you the flexibility to target the XC9500 and CoolRunner Series CPLDs on-line or on the desktop, including:

- **WebFITTER -** an on-line device fitting and evaluation tool that accepts HDL, ABEL, or netlist files and provides all reports, simulation models, and programming files, along with price quotes.
- **WebPACK -** downloadable desktop solutions that offer free CPLD software modules for ABEL/HDL synthesis and simulation, device fitting, and JTAG programming.

Through leading performance, free internet-based WebPOWERED software and the industry's lowest power consumption, Xilinx has the right CPLD for every designer's need.

See www.xilinx.com/products/cpldsolutions/index.htm for more information

| CPLD Product Selection Matrix | | | | Density | | Features | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Core Voltage | CPLD Family | Devices | Key Features | Macrocells | Max. I/O | Pin-to-Pin Delay (ns) | System Frequency | Individual OE Ctrl | JTAG | Ultra Low-Power | Philips Part Number |
| 3.3 Volt ISP | XC9500XL | XC9536XL | Best Pin-Locking JTAG w/Clamp High Performance High Endurance | 36 | 36 | 4 | 200 | √ | √ | | |
| | | XC9572XL | | 72 | 72 | 5 | 178.6 | √ | √ | | |
| | | XC95144XL | | 144 | 117 | 5 | 178.6 | √ | √ | | |
| | | XC95288XL | | 288 | 192 | 6 | 151 | √ | √ | | |
| | XPLA3 | XCR3032XL | Ultra Low Power JTAG Increased Logic Flexibility | 32 | 32 | 5 | 200 | | √ | √ | |
| | | XCR3064XL | | 64 | 64 | 6 | 166 | | √ | √ | |
| | | XCR3128XL | | 128 | 104 | 6 | 166 | | √ | √ | |
| | | XCR3256XL | | 256 | 160 | 7.5 | 133 | | √ | √ | |
| | | XCR3384XL | | 384 | 216 | 7.5 | 133 | | √ | √ | |
| | XPLA-Enhanced | XCR3032A | Ultra Low Power JTAG | 32 | 32 | 6 | 111 | | √ | √ | PZ3032A |
| | | XCR3064A | | 64 | 64 | 7.5 | 95 | | √ | √ | PZ3064A |
| | | XCR3128A | | 128 | 96 | 7.5 | 95 | | √ | √ | PZ3128A |
| | XPLA2 | XCR3320 | Ultra Low Power High Density | 320 | 192 | 7.5 | 100 | | √ | √ | PZ3320C |
| | | XCR3960 | | 960 | 384 | 7.5 | 100 | | √ | √ | PZ3960C |
| 5 Volt ISP | XC9500 | XC9536 | Best Pin-Locking JTAG High Endurance | 36 | 34 | 5 | 100 | √ | √ | | |
| | | XC9572 | | 72 | 72 | 7.5 | 83.3 | √ | √ | | |
| | | XC95108 | | 108 | 108 | 7.5 | 83.3 | √ | √ | | |
| | | XC95144 | | 144 | 133 | 7.5 | 83.3 | √ | √ | | |
| | | XC95216 | | 216 | 166 | 10 | 66.7 | √ | √ | | |
| | | XC95288 | | 288 | 192 | 10 | 66.7 | √ | √ | | |
| | XPLA-Enhanced | XCR5032C | Ultra Low Power JTAG | 32 | 32 | 6 | 111 | | √ | √ | PZ5032C |
| | | XCR5064C | | 64 | 64 | 7.5 | 105 | | √ | √ | PZ5064C |
| | | XCR5128C | | 128 | 96 | 7.5 | 100 | | √ | √ | PZ5128C |

XC18V
XC17V
XC17S

FPGA Configurations


Xilinx offers a full range of configuration memories optimized for use with Xilinx FPGAs. Our PROM product lines are designed to meet the same stringent demands as our high-performance FPGAs and CPLDs, taking full advantage of the same advanced processing technologies. In addition, they were developed in close cooperation with Xilinx FPGA designers for optimal performance and reliability.

**XC18V:** Our in-system re-programmable family provides a feature-rich, fast configuration solution unmatched by any other configuration PROM available today, and provides a cost-effective method for re-programming and storing large Xilinx FPGA bitstreams. This family is JTAG ready and Boundary-Scan enabled for exceptional ease-of-use, system integration, and flexibility.

**XC17V/XC17S:** Our low-cost XC17V and XC17S families are an ideal configuration solution for cost-sensitive applications. XC17V00 PROMs are pin-compatible with our 18V family to allow for a cost-reduction migration path as your production volumes increase. The XC17S family is specially designed to provide a low-cost, integrated solution for our Spartan families of FPGAs.

**3.3V Configuration PROMs**

| PROM | Density | PD8 | SO20 | PC20 | PC44 | VQ44 | JTAG ISP |
|---|---|---|---|---|---|---|---|
| XC1765EL(X) | 64Kb | X | X | X | X | | |
| XC17128EL(X) | 128Kb | X | | X | X | | |
| XC17256EL(X) | 256Kb | X | | X | X | | |
| XC17512L | 512Kb | X | X | X | | | |
| XC1701L | 1Mb | X | X | X | | | X |
| XC1702L | 2Mb | | | | X | X | |
| XC1704L | 4Mb | | | | X | X | |
| XC18V128 | 128Kb | | | | X | X | X |
| XC18V512 | 512Kb | | X | X | | X | X |
| XC18V01 | 1Mb | | X | X | | X | X |
| XC18V02 | 2Mb | | | | X | X | X |
| XC18V04 | 4Mb | | | | X | X | X |

Note: XC1700EL parts are marked with an "X" instead of "EL"

**3.3V Configuration PROMs for Spartan-XL/Spartan-II**

| FPGA | Configuration Bits | Family | PROM Solution | PD8 | VO8 | SO20 |
|---|---|---|---|---|---|---|
| XCS05XL | 54,544 | Spartan XL | XC17S05XL | X | X | |
| XCS10XL | 95,752 | Spartan XL | XC17S10XL | X | X | |
| XC2S15 | 197,696 | Spartan II | XC17S15XL | X | X | |
| XCS20XL | 179,160 | Spartan XL | XC17S20XL | X | X | |
| XCS30XL | 249,168 | Spartan XL | XC17S30XL | X | X | |
| XC2S30 | 336,768 | Spartan II | XC17S30XL | X | X | |
| XCS40XL | 330,696 | Spartan XL | XC17S40XL | X | X* | X |
| XC2S50 | 559,232 | Spartan II | XC17S50XL | X | | X |
| XC2S100 | 781,248 | Spartan II | XC17S100XL | X | | X |
| XC2S150 | 1,041,128 | Spartan II | XC17S150XL | X | | X |
| XC2S200 | 1,335,872 | Spartan II | XC17S200XL* | X | | X |

* In development

**Configuration PROMs for Virtex-E**

| FPGA | Configuration Bits | XC17xx/XC18Vxx Solution | PD8 | PC20 | SO20 | PC44 | VQ44 |
|---|---|---|---|---|---|---|---|
| XCV50E | 630,048 | 01 | X** | X | X | | X*** |
| XCV100E | 863,840 | 01 | X** | X | X | | X*** |
| XCV200E | 1,442,106 | 02 | | | | X | X |
| XCV300E | 1,875,648 | 02 | | | | X | X |
| XCV400E | 2,693,440 | 04 | | | | X | X |
| XCV405E | 3,430,400 | 04 | | | | X | X |
| XCV600E | 3,961,632 | 04 | | | | X | X |
| XCV812E | 6,519,648 | 04+04 or 08* | | | | X | X |
| XCV1000E | 6,587,520 | 04+04 or 08* | | | | X | X |
| XCV1600E | 8,308,992 | 04+04 or 08* | | | | X | X |
| XCV2000E | 10,159,648 | 04+08* or 16* | | | | X | X |
| XCV2600E | 12,922,336 | 16* | | | | X | X |
| XCV3200E | 16,283,712 | 16* | | | | X | X |

* In development
** Available on XC17xx only
*** Available in XC18Vxx only

**Configuration PROMs for Virtex**

| FPGA | Configuration Bits | XC17xx/XC18Vxx Solution | PD8 | PC20 | SO20 | PC44 | VQ44 |
|---|---|---|---|---|---|---|---|
| XCV50 | 558,048 | 01 | X* | X | X | | X** |
| XCV100 | 780,064 | 01 | X* | X | X | | X** |
| XCV150 | 1,038,944 | 01 | X* | X | X | | X** |
| XCV200 | 1,334,688 | 02 | | | | X | X |
| XCV300 | 1,750,656 | 02 | | | | X | X |
| XCV400 | 2,544,896 | 04 | | | | X | X |
| XCV600 | 3,606,816 | 04 | | | | X | X |
| XCV800 | 4,714,400 | 04 + 512 | | | | X | X |
| XCV1000 | 6,126,528 | 04 + 02 | | | | X | X |

* Available on XC17xx only
** Available on XC18Vxx only

# QPRO QML-Certified FPGAs and PROMs

The Xilinx QPRO family of Radiation Hardened FPGAs and PROMs are finding homes in many new satellite and space applications. Both the XQR4000XL and XQVR Virtex products are being designed into space systems that will utilize reconfigurable technology. Numerous communications and GPS satellites, space probe, and shuttle missions are included on the growing list of programs that will be flying these devices.

The Virtex QPRO family of High Reliability products is experiencing a high degree of success in the defense market. As designers find it more and more difficult to find components suitable for the harsh environments seen by defense systems, they are discovering that they can incorporate the functions of obsolete parts into Virtex QPRO products. This has the added long term advantage of significantly reducing the costs of future re-qualifications, because their systems can retain consistent form, fit, and function through the use of Virtex QPRO FPGAs. This cannot be achieved with costly and inflexible ASICs or custom logic.

Please visit http://www.xilinx.com/products/hirel_qml.htm for all the latest information about these products, including some new applications notes.

## FPGA Product Selection Matrix

| | | DENSITY | | | | | | | FEATURES | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DEVICES | KEY FEATURES | Logic Cells | Maximum Logic Gates | Typical System Gate Range | Max. RAM Bits | CLB Matrix | CLBs | Flip-Flops | Max. I/O | Output Drive (mA) | PCI Compliant | 1.8 Volt | 2.5 Volt | 3 Volt | 5 Volt |
| XQR/XQ4013XL | XC4000 Series: Density Leadership/ High Performance/ SelectRAM Memory | 1368 | 13K | 10K-30K | 18K | 24x24 | 576 | 1536 | 192 | 12/24 | Y | – | – | X | * |
| XQR/XQ4036XL | | 3078 | 36K | 22K-65K | 42K | 36x36 | 1296 | 3168 | 288 | 12/24 | Y | – | – | X | * |
| XQR/XQ4062XL | | 5472 | 62K | 40K-130K | 74K | 48x48 | 2304 | 5376 | 384 | 12/24 | Y | – | – | X | * |
| XQ4085XL | | 7448 | 85K | 55K-180K | 100K | 56x56 | 3136 | 7168 | 448 | 12/24 | Y | – | – | X | * |
| XQV100 | Virtex Family: Density/ Performance Leadership BlockRAM Distributed RAM SelectI/O 4 DLLs | 2700 | 32K | 72K-109K | 78K | 20x30 | 600 | 2400 | 180 | 2/24 | Y | – | X | I/O | * |
| **XQVR/XQV300 | | 6912 | 83K | 176K-323K | 160K | 32x48 | 1536 | 6144 | 316 | 2/24 | Y | – | X | I/O | * |
| **XQVR/XQV600 | | 15552 | 187K | 365K-661K | 312K | 48x72 | 3456 | 13824 | 512 | 2/24 | Y | – | X | I/O | * |
| **XQVR/XQV1000 | | 27648 | 332K | 622K-1,124K | 512K | 64x96 | 6144 | 24576 | 512 | 2/24 | Y | – | X | I/O | * |

* I/Os are tolerant
** XQR and XQVR devices are Radiation Hardened
X = Core and I/O voltage
I/Os = I/O voltage supported

(1) All devices specified from -55°C to +125°C
(2) Selected XQ4000E/EX devices also available

## QPRO QML-certified PROMs

| | | Package | | | |
|---|---|---|---|---|---|
| Device | Density | DD8 | SO20 | CC44 | PC44 |
| XC1736D | 36Kb | X | | | |
| XC1765D | 64Kb | X | | | |
| XC17128D | 128Kb | X | | | |
| XC17256D | 256Kb | X | | | |
| XQR/XQ 1701L* | 1Mb | | X | X | |
| XQR/XQ 1704L* | 4Mb | | | X | X** |

* XQR devices are Radiation Hardened.
** XQ devices only.

Xilinx development systems give you the speed you need. With version 3.1i solutions, Xilinx place and route times are as fast as 2 minutes for our 200,000 gate, XC2S200 Spartan(r)-II device, and 30 minutes for our 1 million gate, system-level XCV1000E Virtex(tm)-E device. That makes Xilinx development systems the fastest in the industry.

And with the push of a button, our timing-driven tools are creating designs that support I/O speeds in excess of 800 Mbps, and internal clock frequencies in excess of 300 MHz.

These development solutions combine powerful technology with a flexible, easy to use graphical interface to help you achieve the best possible designs within your project schedule, regardless of your experience level.

**Lead Product Feature Comparison**

| | Alliance Series | | Foundation Series | | | | ISE Series | | |
|---|---|---|---|---|---|---|---|---|---|
| | ALI-STD | ALI-ELI | FND-EXP | FND-ELI | FND-BAS | FND-BSX | ISE-ELI | ISE-EXP | ISE-BSX |
| **Design Entry** | | | | | | | | | |
| Schematic Entry | | | √ | √ | √ | √ | √ | √ | √ |
| HDL Entry | | | √ | √ | √ | √ | √ | √ | √ |
| ABEL Entry | | | √ | √ | √ | √ | √ | √ | √ |
| HDL Editor | | | √ | √ | √ | √ | √ | √ | √ |
| State Diagram Editor | | | √ | √ | √ | √ | VSS* | VSS* | VSS* |
| **Environment** | | | | | | | | | |
| Operating System | PC/UNIX | PC/UNIX | PC | PC | PC | PC | PC | PC | PC |
| **Simulation** | | | | | | | | | |
| Gate Level Timing Simulation | | | √ | √ | √ | √ | | | |
| HDL Simulation | MTI** | MTI** | MTI** | MTI** | MTI** | MTI** | MTI** | MTI** | MTI** |
| HDL Test Bench Generator | | | | | | | VSS* | VSS* | VSS* |
| **Synthesis** | | | | | | | | | |
| Xilinx Synthesis Technology (XST) | | | | | | | √ | √ | √ |
| FPGA Express | | | √ | √ | √ | √ | √ | √ | √ |
| Incremental Synthesis | | | √ | √ | √ | √ | √ | √ | √ |
| **System Features** | **ALI-STD** | **ALI-ELI** | **FND-EXP** | **FND-ELI** | **FND-BAS** | **FND-BSX** | **ISE-ELI** | **ISE-EXP** | **ISE-BSX** |
| Constraints editor | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| Floorplanner | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| CPLD ChipViewer | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| Pin Editor | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| FPGA Editor | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| Core Generator included | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| Configuration by cable | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| Error navigation to Xilinx Web | | | | | | | √ | √ | √ |
| Command line operation | | | | | | | √ | √ | √ |
| HTML timing reports | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| Data Book I/O timing | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| Project archiving | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| **System Interfaces** | | | | | | | | | |
| EDIF in | √ | √ | √ | √ | √ | √ | | | |
| PROM file generation | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| JTAG download software | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| IBIS | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| STAMP | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| VHDL, Verilog | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| HDL Simulation libraries | √ | √ | √ | √ | √ | √ | √ | √ | √ |

*VSS - Delivered as part of the ALLSTAR program or as a backPACK module in WebPACK

**Lead Product Device Support Comparison**

| | | Alliance Series | | Foundation Series | | | | ISE Series | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Family | Part Number | ALI-STD | ALI-ELI | FND-EXP | FND-ELI | FND-BAS | FND-BSX | ISE-ELI | ISE-EXP | ISE-BSX |
| **Virtex** | XCV50 only | | | | | √ | √ | | | √ |
| | All devices up to XCV1000 | √ | √ | √ | √ | | | √ | √ | |
| **Virtex-E** | XCV50E only | | | | | √ | √ | | | √ |
| | All devices up to XCV1000E | √ | √ | √ | √ | | | √ | √ | |
| | All devices up to XCV3200E | | √ | | √ | | | √ | | |
| **Virtex-EM** | XCV405EM | √ | √ | √ | √ | | | √ | √ | |
| | XCV812EM | √ | √ | √ | √ | | | √ | √ | |
| **Spartan** | XCSxx (All devices) | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| **Spartan XL** | XCSxxXL (All devices) | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| **Spartan-II** | XC2SxxXL (Includes XC2S200) | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| **XC9500 Series** | XC9500 XV/XL (All devices) | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| **XC400 Series** | XC4000E/L/EX (All devices) | √ | √ | √ | √ | √ | √ | √ | √ | √ |
| | XC4000XL/XLA (All devices up to XC4020) | | | | | √ | √ | | | √ |
| | XC4000XL/XLA (All devices) | √ | √ | √ | √ | | | √ | √ | |
| | XC4000XV (All devices) | √ | √ | √ | √ | | | √ | √ | |
| **XC3000 Series** | XC3x00A/L (All devices) | √ | √ | √ | √ | √ | √ | | | |
| **XC5200 Series** | XC5200 (All devices) | √ | √ | √ | √ | √ | √ | | | |

***Note**: CoolRunner Series is only available in WebFITTER and WebPACK at this time.

# You can bet your ASIC on it.


The Spartan®-II family offers a winning combination of performance and price. In fact with over 200,000 system gates, clock speeds beyond 200MHz, and lightning fast compile times, Spartan-II FPGAs give you the best dollar value in the industry.

### Get rid of the risk

Now you've got the design flexibility you've been waiting for, without any annoying NRE costs, strung-out production times, or the design risks associated with a typical ASIC. To save system costs, there's also multiple Delay Locked Loops, on-chip RAM (block and distributed), and our unique SelectI/O™ interface technology.

### Xilinx IP Center completes the total solution

The Xilinx IP Center brings you all the FPGA cores, reference designs, design services and third party support you could wish for, all easily available via the web. Find out more today by visiting us at ***www.xilinx.com/sp1.htm.*** You'll soon see why you can bet your ASIC on us.

XILINX®

The Programmable Logic Company℠

**www.xilinx.com**