ISSUE 44, WINTER 2002 XCELL JOURNAL XILINX, INC.

Issue 44
Winter 2002

Xcell journal

THE AUTHORITATIVE JOURNAL FOR PROGRAMMABLE LOGIC USERS

The New ISE 5.1i Software Is Here!

PERSPECTIVE

Lowest Cost FPGAs

ISE SOFTWARE

Get Faster Timing Closure

Control Hierarchy Retention

New Architecture Wizards

ChipScope Pro Integrated Bus Analyzer

Architectural Synthesis

NEWS

The First True Digital Guitar

NEW PRODUCTS

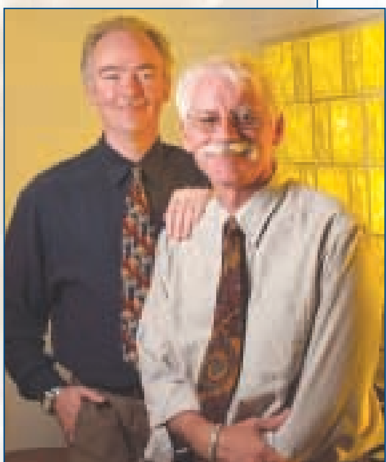
MicroBlaze Development Kits

COVER STORY

Searching for the Elusive Higgs Boson at 1.5 Terabytes per Second

XILINX®

What's Important Now?



Based on customer feedback, there are two primary challenges that dominate our industry in this time of economic uncertainty: lowering costs, and increasing productivity. In this issue of Xcell we explore both subjects – we show you how Xilinx programmable logic technology is truly the lowest cost and the most productive solution for most new designs in today's highly competitive marketplace.

For example, our devices span a wide range of needs, including the lowest cost and the highest performance devices you can get. And, our new ISE software is, by far, the most productive set of tools in the industry – tools that not only help you complete your designs faster but also produce faster designs. This combination of devices and software, along with our education and consulting services, gives you the lowest overall cost, with the fastest time to market. For most products, there is simply no better design method.

Our devices are now being used in a wide array of low-cost consumer applications, from cell phones and automobiles, to set top boxes and even guitars. The inherent flexibility of programmable logic makes it the ideal choice for a changing marketplace where standards quickly evolve, and new products must be introduced quickly.

Our devices are also being used in ultra-high performance applications at the very frontier of pure science. The Fermi National Accelerator Laboratory is using an array of 582 Xilinx FPGAs in the search for the last subatomic particle – the Higgs boson.

See for yourself. The advantages are clear. Xilinx technology is already your best design solution, and it just keeps getting better.

I hope you enjoy this edition of Xcell. Please write and tell me what you think – your comments and suggestions are always welcome.

Carlis Collins

Carlis Collins
Editor-in-Chief

EDITOR IN CHIEF	Carlis Collins editor@xilinx.com 408-879-4519
MANAGING EDITOR	Tom Durkin tom.durkin@xilinx.com 530-271-0899
XCELL ONLINE EDITOR	Tom Pyles
CONTRIBUTING EDITORS	Jane Middleton Charmaine Hussain Peter Bergsman
DESIGN & ILLUSTRATION	Scott Blair Dan Teie

Xcell journal

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124-3400
Phone: 408-559-7778
FAX: 408-879-4780

Correction

In the Fall/Winter 2001 edition of *Xcell Journal* (Issue 41), there was an error in Table 1 (page 72) of the article "Two Virtex-II FPGAs Deliver Fastest, Cheapest, Best High-Performance Image Processing System." The cost of the "Off-the-Shelf ASIC-Based Solution" from Catalina Research Inc. should have read $48,000, not $480,000. *Xcell Journal* makes every effort to ensure the accuracy of the articles we publish, and we regret this error.

View from the Top Page 4

Using Technology to Manage Costs in an Uncertain Economy

Xilinx uses advanced technology to help reduce your cost and manage resources.

New Technology Page 10

The New ISE 5.1i Software Soars Higher and Faster

Efficient design methodologies help lower your project costs and finish your designs faster.

New Products Page 24

ChipScope Pro Integrated Bus Analyzer

Powerful Debugging Tools for Virtex-II Pro FPGAs

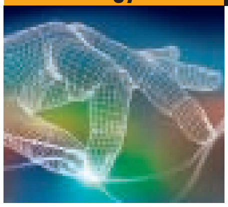
Debugging bus transactions is now faster and easier than ever before.

New Technology Page 30

Architectural Synthesis: Unleashing the Power of FPGA System-Level Design

Architectural synthesis shifts complex system-level design to a higher level.

Technology Focus Page 58



How to Manage Power Consumption in Advanced FPGAs

...various sources of power consumption and techniques for power management

Cover Story Page 68

500+ Xilinx FPGAs Search for Elusive Higgs Boson at 1.5 Terabytes per Second

Scientists at the Fermi National Accelerator Laboratory hope to track down the last subatomic particle - the Higgs boson.

Contents Winter 2002

Xcell journal

Subscribe to the Xilinx Xcell Journal at: ***www.xilinx.com/forms/literature.htm.***

Visit Xcell Online for the latest news and information, as soon as it arrives at: ***www.xilinx.com/publications/xcellonline/.***

Using Technology to Manage Cost in an Uncertain Economy

Xilinx uses advanced technology to help you reduce cost and manage resources.

by Wim Roelandts
CEO, Xilinx Inc.

In today's uncertain economy, device cost is the most important issue in the minds of many people; time-to-market issues appear to be less important now, and programmable logic appears to be expensive in the long run. These misguided ideas do not take into account that companies will have to react quickly when the economy improves again. They also fail to realize that the same technology advances we've made in density and performance have also allowed Xilinx to drive down system costs in many areas. Programmable logic is not only the fastest way to develop new products, it's also the lowest cost alternative in most applications.

Your Time to Market Still Matters

When business is slow, many people think they have the extra time they need to create products using ASIC technology, trying to get the lowest possible production cost. Their rationale is that the long design cycle of an ASIC won't hurt them because the slow economic conditions allow more time before new products need to be introduced. In fact, these very same slow economic conditions may put pressure on companies to very quickly come up with innovative products when their end markets recover. When business starts to improve, it is the companies that get their new products to market first that will reap the rewards of the upturn.

If you've been designing with ASICs, you won't be able to quickly modify your product to meet new market needs. Only programmable logic gives you the flexibility to quickly develop products that will allow you to realize the maximum profit from an improving economy.

Technology Advances Drive Down Cost

Using advanced technology to develop new products has allowed Xilinx to make tremendous advances in programmable logic density and performance. Just four years ago, our largest device contained one million system gates – today our largest device contains more than eight million system gates. Advances in device technology and software tools have increased system performance of our FPGAs by 40% in just the last year alone.

Our technology has allowed Xilinx to create the most advanced, feature-rich FPGAs in the world – devices that allow dramatic reductions in system cost through massive integration. Embedded PowerPC™ processors and RocketIO™ multi-gigabit serial transceivers are now included as standard features in our Virtex-II Pro™ FPGAs, with no cost increase compared to the previous generations of Virtex devices. In fact, Virtex-II Pro devices have a smaller die size than any competitive FPGA of similar logic density, even though they include all these advanced features.

Moving to advanced technology has also allowed us to dramatically reduce cost and bring programmable logic within the reach of many more cost-conscious customers than before. For example, a 300,000 system gate device cost more than $200 in 1998. Today, it's under $20. This better than ten-fold cost reduction in just four years is the result of advances in device architecture as well as an aggressive move to 300 mm wafer technology.

Within the last four years, our sales into consumer and automotive applications have gone from almost nothing, to as much as 15% of Xilinx revenue. Our Spartan™ FPGAs are rapidly becoming the solution of choice for leading-edge consumer products such as home networks, set-top-boxes, DVD recorders, and plasma TV displays. In addition, our CoolRunner™-II RealDigital CPLDs (which eliminated the use of power-hungry sense amps that require special processes) now use leading edge CMOS technology to deliver the best CPLD performance and the lowest power at the lowest cost. By leveraging the same technology advances as our FPGAs, we can improve the costs of our CPLDs more rapidly than other suppliers.

Our Virtex-II EasyPath™ solutions save cost not by using different silicon, but by using special testing methods that verify the silicon for a single design image, giving dramatic cost savings with no engineering risk. They offer the advantages of FPGAs in development and initial production with dramatic cost savings in high volume production, but none of the risk and cost of an ASIC conversion effort.

ASICs Are Not Always Cheap

An ASIC will always have a lower unit cost than PLDs for very high volumes over a very long time frame. If you are building a million identical units a year for five years, an ASIC would be the lowest cost device, overall. The problem is, most systems don't stay the same for that long and most don't have high enough volumes to recoup the up-front engineering investment. Programmable logic devices are easier to get, easier to use, and they are far easier to inventory because you can use one device in many different applications.

Life cycle issues can severely affect your profitability as you phase out one product to introduce a new one. Having low unit cost doesn't save you money if you have excess inventory left over at the end of a product's life. Many ASIC users are faced with obsolete inventory issues, while FPGA users can use the product inventory on a new product and avoid inventory write-down costs.

You wouldn't build a computer with just one set of programs in it and no ability to load new ones, so why build a system with no provision to change the software and hardware as market needs change? With in-system configurable programmable logic from Xilinx, you can update your system's hardware as easily as you would a software driver. When using our Virtex-II Pro FPGAs with the embedded PowerPC processor, this field upgradability extends into the embedded software domain as well. In fact, you could change the partitioning of hardware and software functions in your system without ever replacing your hardware at all – and you can do it all remotely, over the Internet. This can save you a lot of money and give your products a critical advantage.

Software Is a Key Factor in Cost Savings

The shorter design cycles and time-to-market advantages of FPGAs also mean that you need less engineering resources. This allows you to make the best use of your staff when poor economic conditions restrict your ability to hire more engineers. Our fast, efficient, and highly productive ISE software tools help you get the job done in less time, and they make each engineer more productive. Our ISE software will also produce designs that run faster than ever before, so you can often save money by using slower speed grade devices. Plus, we partner with the leading EDA software suppliers, development board manufacturers, and intellectual property producers to bring you the best solutions from the best minds in our industry.

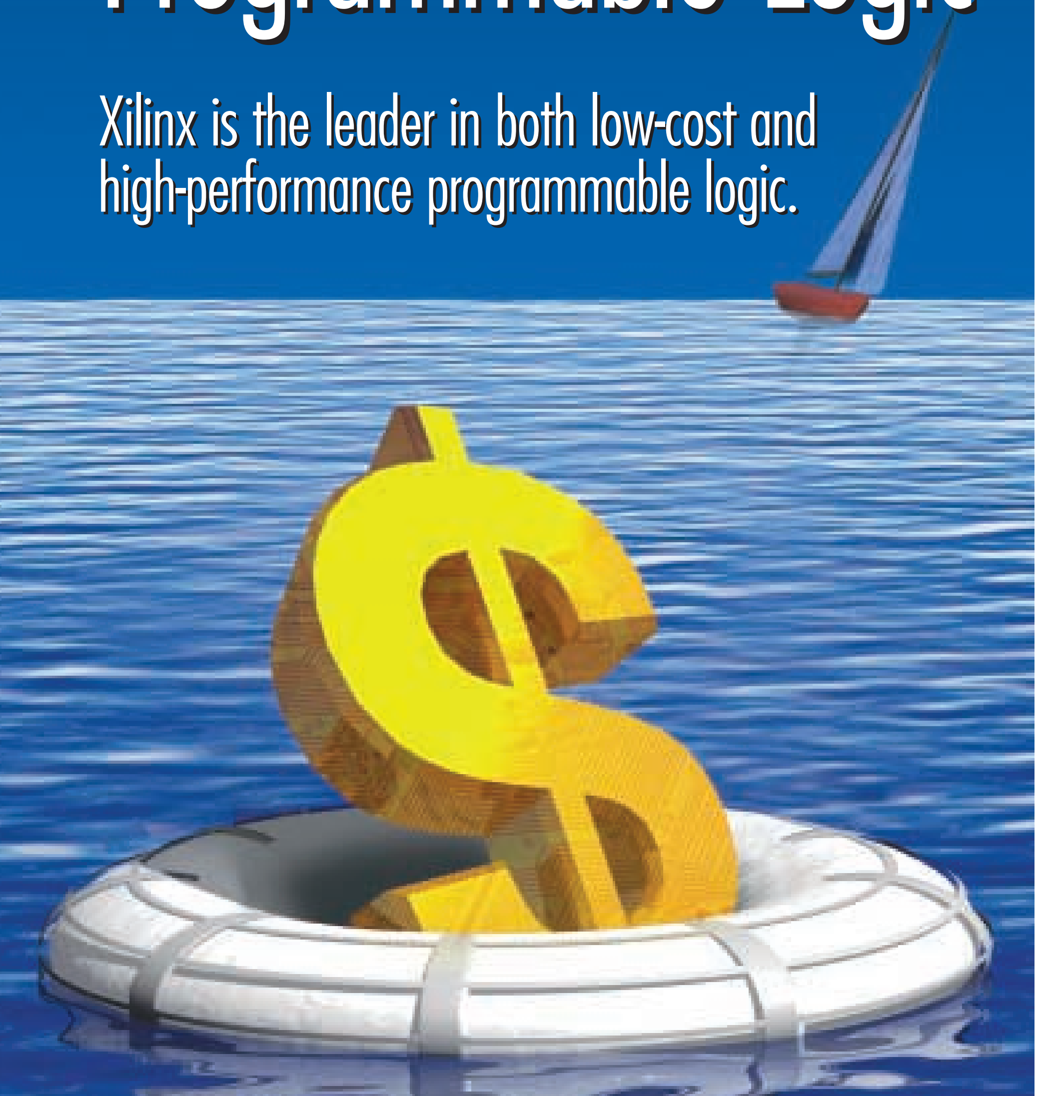
Debugging your design is far easier and less expensive than ever before as well. The Xilinx design methodology integrates devices and software with our ChipScope Pro logic and bus analyzer to provide a debugging environment that offers unparalleled, real-time access to your system; it reduces debugging times by as much as 50%.

Conclusion

If you think Xilinx technology is expensive, think again. Our technology uses the most advanced processes along with optimal architectures and unique testing programs to offer you not only the highest performance and the lowest power devices, but also the lowest cost devices. Combined with our industry-leading development software, Xilinx gives you an overall solution that not only saves you money up front in design, and in later in production, but it also helps to make your products last longer and make more profit. There is no better or less expensive way to develop your next product.

Xilinx Delivers Lower Cost While Continuing to Redefine Programmable Logic

Xilinx is the leader in both low-cost and high-performance programmable logic.



by Steve Sharp
Senior Manager, Programmable Logic Solutions
Worldwide Marketing
Xilinx, Inc.
steve.sharp@xilinx.com

Since we invented the FPGA in 1988, we have aggressively advanced our technology in many ways. You are probably familiar with the extreme performance and the advanced features of our Virtex™ family. However, you may not be aware that our technology advances have also allowed us to develop very low cost devices, processes, and features that save you a significant amount of money.

Our commitment to lowering your costs has opened many new, high-volume applications for our products. Our FPGAs and CPLDs are now used in a wide range of low-cost, high-volume applications, from cell phones and digital cameras, to automobiles and DVD players. So, as you can see, while we have redefined the standards for programmable logic with major advances in performance, features, speed, density, power, flexibility, tools, and cores, we have also set new standards for cost effectiveness. As the industry's technology leader, our customers have come to expect these kinds of advances from Xilinx.

Programmable Logic Costs Less

Xilinx programmable logic solutions offer many unique cost advantages over competing technologies. When you look at the total cost of creating and manufacturing your products, you'll see that device cost alone is not the only factor. Some of the cost advantages of programmable logic include:

- **Ease of use.** There simply is no easier way to develop digital products. Our software tools are fully optimized for our devices, which reduces your risks and helps you create better designs that run faster, with fewer engineers – saving you a significant amount of money.
- **Faster development time.** Time to market is a critical factor in the profitability of most products. The sooner you get your product manufactured, the more money you make. There is no faster way to get from idea to finished product – significantly increasing your profitability.

- **Field reprogrammability**. Time in market is another key factor in profitability; the longer your product stays viable, the more money you make. Our programmable logic devices can easily be reprogrammed, in the field, over the Internet. You can fix bugs, add new features, or adapt to changing market trends with ease. This will make your customers happy, save you a lot of engineering time and expense, and give you a superior product – a unique and significant cost advantage.
- **Comprehensive support services.** The more you know, the more productive you can be. You significantly reduce your risks and your development problems when you fully understand the devices and tools you use; plus you can create more, faster. Our education and support services are the best in the industry, helping you do more – with far less cost.

As you can see, programmable logic technology – in general – is cost effective. However, we also strive to lower your specific device costs in every way we can, making our devices attractive in many new, low-cost applications.

Spartan-IIE FPGAs – Your Best Value for Today's Digital Consumer Applications

When we introduced the Spartan™-IIE family of cost-optimzed FPGAs in November 2001, we delivered the optimum combination of performance, flexibility, and value. Designed for today's cost-sensitive digital consumer applications, the Spartan-IIE family includes advanced features such as low voltage differential signaling (LVDS), high-speed dual-port block RAM, and digital delay-locked loops (DLLs), with up to 300,000 system gates of programmable logic. Supporting such popular IP cores as PCI interfaces and our MicroBlaze™ soft processor, these solutions are the ideal alternative to gate arrays in applications such as broadband access, set-top boxes, and plasma displays.

Spartan-IIE devices lower your costs even further because your can quickly develop your designs using our industry-leading ISE 5.1i software. Your time to market is significantly reduced because our comprehensive tools are fast, efficient, and thorough, making your job far easier than ever before. Plus we offer a wide selection of time saving cores, optimized for the Spartan architecture. Our cores and our new ISE 5.1i software help you complete your designs faster than ever before.

In high volume production applications, Spartan-IIE devices cost less than any competitive solution. You get an outstanding value because we integrate many of the expensive system functions normally found in standalone ASSP devices, plus we use advanced 300 mm wafer fabrication technology that reduces our manufacturing costs to the minimum. Spartan series solutions are the lowest cost FPGAs in the industry today.

When you add it all up, there is no faster, easier, or lower cost method for creating high volume designs that give you all the benefits of programmable logic.

CoolRunner-II RealDigital CPLDs – Redefine Low-Power Technology and Value

In January, 2002, we introduced the CoolRunner™-II family of RealDigital CPLDs. This family defined a new class of CPLDs, combining high performance, ultra-low power, and advanced system features with the most competitive prices in the industry. What makes these CPLDs unique is that we removed the traditional power-hungry analog sense amplifiers, replacing them with low-power digital CMOS circuitry. Now we can offer the best performance in the industry with standby power that is 100 times lower than any competing device. We also added many powerful system features normally associated with FPGAs, such as clock management and multiple-voltage I/O capability. CoolRunner-II CPLDs are available in tiny, low-cost packages as well, which makes them ideal for any portable, battery-powered, high-volume application.

The all-digital technology used in CoolRunner-II devices also allows us to use the same process technology that we pioneered for our FPGAs, gaining economy of scale and leveraging the cost benefits of using the latest manufacturing technology. Thus, we can offer you the most competitive prices in the industry. Now you don't have to choose multiple CPLD solutions to get the best performance, power, features, or price – you get it all in our CoolRunner-II CPLDs.

Virtex-II Pro FPGAs – What Was Once Optional Is Now Standard

When we introduced the Virtex-II Pro FPGA family in March 2002, we delivered the industry's first platform for programmable systems. Because we embedded IBM PowerPC™ processor cores and 3.125 gigabit per second RocketIO™ serial transceivers into the industry leading Virtex-II programmable logic fabric, it is now possible for you to design a true programmable system on a single programmable device.

As with our other FPGAs and CPLDs, your Virtex designs are completed quickly in a comprehensive development environment that combines the highest performance silicon and software tools, the widest range of IP cores, and the most flexible system debugging environment in the industry.

Our overall mission is to deliver the most advanced technology in each new generation of devices, while driving down prices. We continue this strategy with the Virtex-II Pro family, which is not only opening the door to programmable system design in the future, it is also delivering more capability at lower cost for any user of programmable logic today.

Virtex-II EasyPath Solutions Reduce Cost While Minimizing Risk

To further reduce the costs of using our Virtex-II devices we developed a special testing program that can reduce device costs by as much as 80% in large volume applications.

Our new Virtex-II EasyPath™ devices use the same silicon as Virtex-II FPGA devices, but they are tested to your specific design image only, resulting in higher yields and significantly lower costs. This cost reduction approach is completely risk free. Your production devices will work exactly like your prototypes, because these devices are exactly the same as their general purpose cousins – the only difference is the testing.

Virtex-II EasyPath solutions give you a volume conversion strategy with no risk, no investment of your engineering resources, and the fastest conversion time of any competing high-volume strategy for high-density FPGA designs. This software-based approach to cost reduction has met with universal acclaim from our customers for its simplicity and effectiveness.

ISE 5.1i Software – Reducing Your Development and Production Costs

Our new ISE 5.1i software can save you money in a number of ways:

- **It's fast**. You can complete designs much faster than with any previous solution, and that's like putting money in the bank.
- **It's comprehensive**. Everything you need is provided, and it all works together in a seamless environment that makes your life easier.
- **It's easy to use**. The tools are well designed, thorough, and seamless. Plus we offer comprehensive training at your site, online, or at one of our training facilities.
- **It produces faster designs**. Because the software is optimized for the device architecture, your Virtex-II designs can run up to 15% faster than before. This means that you can often use a slower speed grade device, for a significant cost savings.
- **It's backed by XPERTS**. Xilinx XPERTS are people who are certified by Xilinx to have a deep knowledge of our software. If you need an extra hand, or if you need to quickly solve design problems, Xilinx XPERTS will save you time and money.

ISE 5.1i

Tying all our new solutions together and offering cost savings of its own, our new ISE 5.1i software sets new standards for speed, productivity, and capability. The ISE tools now encompass logic design, embedded software design, and system design. Building on its position as the most widely used logic design system in the industry, the addition of new embedded design tools from Xilinx and key partners, such as Wind River Systems, make the ISE 5.1i tool set more powerful than ever, delivering cost savings as well.

For logic designers, being more productive and getting designs completed and debugged faster translates to lower development cost. ISE 5.1i delivers this through improved incremental design capabilities, a powerful macro builder for design reuse, a graphical pinout and area constraints editor (PACE), and ChipScope™ Pro analyzers, the industry's most flexible and powerful system debugging solution.

ISE 5.1i also delivers production cost savings as well. Virtex-II designs will achieve system speeds an average of 15% higher than with our previous software. This translates into a lower speed grade requirement for production, and considerable cost savings over the life of a program. ISE 5.1i also includes architecture wizards that simplify integration of complex functions into the powerful digital clock managers (DCMs) and RocketIO serial transceivers in Virtex-II and Virtex-II Pro FPGAs. By integrating more functions into the FPGA, you can directly reduce the bill of materials cost in you systems.

If you are using the embedded PowerPC processors in our Virtex-II Pro FPGAs, you'll also need to develop embedded software. The ISE 5.1i embedded design tools, incorporating the industry-leading Wind River Systems tools, make it easy for you to take full advantage of the powerful Virtex-II Pro platform for programmable systems.

Conclusion

Xilinx technology is not always expensive; in fact it will save you money in many ways. We offer the highest performance devices you can get, and that performance does come at a high price in the most advanced applications. However, we also offer a full range of high-performance devices that are well suited to low-cost applications as well. There are many reasons to choose programmable logic, and lower overall cost is one of best.

CoolRunner-II There's a new class of CPLD, and it's the breakthrough in high performance and low power you've been looking for. Introducing the RealDigital CPLD from Xilinx. The CoolRunner™-II RealDigital CPLD family offers a 100% digital core, eliminating power-hungry analog sense-amp technology.

The RealDigital CPLD makes everything else obsolete

Competitive CPLDs are old news. The new CoolRunner-II RealDigital CPLDs get rid of analog sense amps, which means you'll never again have to sacrifice low power for speed. Featuring our unique Fast Zero Power™ technology, the new 1.8V CoolRunner-II series achieves clock speeds over 400 MHz with standby power of <100µA (1/20th of the nearest competitive device).

1.8V CORE VOLTAGE CPLD COMPETITIVE CHART

Manufacturer	Xilinx	Lattice	Altera
Device Family	CoolRunner-II	ispMACH4000C	None
Standby Current	<100 µA	2 mA	N/A
Clock Divider	YES	NO	N/A
Clock Doubler	YES	NO	N/A
I/O Standards Support	LVTTL, LVCMOS, HSTL, SSTL	LVTTL, LVCMOS	N/A
I/O Banks (max)	4	2	N/A

FREE downloadable software

The new CoolRunner-II RealDigital CPLDs are fully supported by the easy-to-use ISE WebPACK™, downloadable FREE via the Internet. Or choose our lightning-fast ISE 4.2i software — the same development system that supports all Xilinx devices, including the Virtex products you use today.

Find out more about the RealDigital CPLD, plus get your free CoolRunner-II resource CD by visiting **www.xilinx.com/digitalcpld** today.

XILINX®

The Programmable Logic Company℠

www.xilinx.com/digitalcpld

FORTUNE® 2002
100 BEST COMPANIES TO WORK FOR

The New ISE 5.1i Software

Efficient design methodologies help lower your project costs.

by Lee Hansen
Product Marketing Manager
Xilinx, Inc.
lee.hansen@xilinx.com

As the economic downturn drags on, the pressures are now greater than ever before to lower your project costs. ISE 5.1i, the latest release of our design software, offers a number of productivity technologies that shorten your logic design flow, freeze design results, shorten implementation and verification cycles, and provide interactive design assistance – while at the same time enabling you to realize even faster design performance. The end result to you is cost savings across your entire project.

ISE 5.1i, released this past August, delivers all the potential of the leading-edge Xilinx programmable device families by incorporating methodologies that reduce logic design bottlenecks. Built on the ProActive Timing Closure technology introduced in ISE 4.1i last year, ISE 5.1i gives you:

- Designs with 15% higher-performance (2x faster than ISE 4.1i) – allowing the use of slower devices to achieve a cost savings of at least one device speed grade
- PACE (Pinout and Area Constraints Editor) – a graphical pin assignment and area editing tool
- Incremental Design to lower design recompile times
- Macro Builder to freeze performance for design reuse
- Architecture Wizards for fast and easy programming of advanced device features
- ChipScope Pro™ 5.1i for on-chip, real-time debugging.

Xilinx continues to deliver the fastest design performance available anywhere in the PLD industry. Internal benchmark data shows that ISE, coupled with Virtex-II Pro™ FPGAs, is 30 % faster than any competing PLD design solution. Higher performance from ISE means you hit your design target faster and earlier than with any other PLD software – and spend less time reaching design closure.

PACE – Pinout and Area Constraints Editor

ISE 5.1i includes the new Pinout and Area Constraints Editor tool (PACE), delivering new functionality to pin and area management in an easy-to-use, graphical environment. PACE helps speed you through the design flow by streamlining a difficult and time-consuming process.

Pin Management Made Easy

PACE includes graphical pin management that is both powerful and easy to use. You can drag-and-drop pin assignments onto a graphical map of the device, group pins logically and by color-coding for easy recognition, specify I/O standards and banks, prohibit I/O locations, and verify legal pin assignments using the built-in design rules-checking. You can assign and place differential I/Os, and much more. Even as devices grow larger, PACE brings a new level of ease to the difficult task of assigning and verifying your design pins, and quickly moves you forward in the design flow.

Area Definition Moves Forward in the Design Flow

Because PACE uses the Native Generic Database (NGD) file, it can be used in the design flow from the very beginning of the design process. PACE allows you to edit both location and area constraints, define logic areas graphically, and display I/Os on the periphery for connectivity checking. PACE also allows area mapping by examining the defined HDL hierarchy and checks logic areas against expected gate size, making area definitions quick, accurate, and easy.

Incremental Design – Minimizing the Impact of Design Changes

ISE 5.1i includes Incremental Design, a next-generation technology that shortens design recompile times and helps to minimize the time and cost impact of late-arriving design changes. With Incremental Design, you begin by using either PACE or the ISE Floorplanner to plan your design along hierarchical boundaries. The design process then proceeds as usual. Should a change then be required, Incremental Design ensures that only the area in question need be reimplemented. The rest of the design stays locked and intact. Incremental Design lets you use your project time where it's most needed: concentrating on verification, thoroughly testing a critical area of your project, or simply using your time savings to get to market faster.

Macro Builder

Built into the ISE 5.1i Floorplanner is a new feature that allows you to build and save "macros," or blocks of logic, to be reused in a later design. Once a design has been floorplanned and placed, you can execute the "write RPM to NCF" command within the ISE Floorplanner. This saves both the design EDIF (Electronic Design Interchange Format) file and placement information. The new macro, including relative placement information, can then be reused in a future design. Macro Builder lets you leverage your existing investment in HDL development and delivers excellent performance every time; and during project downtime, your engineers can be developing HDL code for later use.

Architecture Wizards

Each new hardware capability released in a Xilinx device family results in an associated learning curve for you. As a designer, you must learn all the programming attributes necessary to make the best use of those new features.

ISE 5.1i includes new Architecture Wizards that help you quickly and easily program both the Digital Clock Manager (DCM) in Virtex™-II and Virtex-II Pro FPGAs, and the 3.125 Gigabit Multi-Gigabit Transceiver (MGT) RocketIO™ pins in Virtex-II Pro devices.

The Architecture Wizards provide a simple, graphics-based way to specify the device feature. By setting dialog box switches appropriately according to the way the device is to be used, HDL code is output in either VHDL or Verilog format for use in the design source files. The Architecture Wizard is great for the first-time designer, for designers new to Virtex-II and Virtex-II Pro devices, or to speed you through device setup, enabling everyone to make the best use of feature-rich device capabilities.

Unsurpassed On-Chip Real-Time Debugging

Released in October, the new ChipScope Pro 5.1i debugging and verification software takes on-chip verification to new levels. ChipScope Pro includes all the functionality of the ChipScope ILA release, plus new enhancements that support even greater debugging potential. These include a new IBA (Integrated Bus Analysis) core that supports debugging of the IBM CoreConnect™ bus (for the Virtex-II Pro IBM PowerPC™ 405 processors); enhancements for logic analysis; and CORE Generator™ and Core Inserter tools for placing the necessary cores into either the HDL source or directly into the design netlist.

The new Agilent Trace Core (ATC), also included in the ChipScope Pro software, is a result of the pioneering relationship between Xilinx and Agilent, the leader in test and measurement equipment. The ATC core links FPGA debugging to the Agilent FPGA Trace Port Analyzer (available separately from Agilent). This test equipment/FPGA combination yields deeper trace debugging with ample sampling memory, more complex triggering options, and support for remote debugging over the Internet.

The combination of Virtex-II Pro devices, ChipScope Pro software, and ISE delivers the most powerful design and real-time debugging solution available, shortens verification cycles, and lowers associated project costs.

Conclusion

ISE continues to define the standard of logic design, concentrating on performance and productivity. ISE delivers the time efficiency demanded by today's high-pressure design environments, and helps you get the highest performance from your logic devices. Go to *www.xilinx.com/xcell_ise* to find out more about ISE 5.1i. To get your copy of ISE 5.1i today, contact your local sales office.

Now Get Faster Timing Closure with ISE 5.1i

ISE Macro Builder, the new timing closure capability available in the latest Xilinx ISE 5.1i release, allows you to reuse design blocks and IP cores easily with assured timing performance.

by Davin Lim
Technical Marketing Manager
Xilinx, Inc.
davin.lim@xilinx.com

Justine Chen
Product Marketing Manager
Xilinx, Inc.
justine.chen@xilinx.com

As design complexity increases, achieving timing closure becomes more challenging than ever. In an effort to reduce time to market, designers typically reuse as many existing design blocks and IP cores as possible. Of course, you can never be sure that the reused design block will perform at the same speed in the new context, right?

Wrong. Xilinx, the leading provider of timing closure and design reuse solutions since the company's founding, has made another important breakthrough: Now you can easily lock in the design performance of a reused design block. With the new ISE (Integrated Software Environment) Macro Builder, you can easily capture both your HDL design code and the placement information of a "known-good" design block – and maintain the performance of the captured design block for reuse in future designs.

Macro Builder Technology

Macro Builder technology is based on relationally placed macros (RPMs) that enable you to control the placement of components of your design relative to each other. By using RPMs, you not only obtain a high degree of control over the final design performance, but also significantly reduce place-and-route runtimes for functions defined as RPMs.

One of the most important uses of RPMs is the creation of user-defined blocks and IP cores for design reuse. In addition, RPMs also make it possible to instantiate design blocks and IP cores multiple times in a top-level design. The performance of each RPM instance is highly predictable and repeatable. Moreover, place-and-route runtimes for those portions of the design are typically very fast.

Before the 5.1i release, creating an RPM involved manually entering the relative location constraints (RLOCs) of each component in the RPM. For small RPMs with few components, this does not constitute a significant problem. However, for large RPMs containing hundreds or more individual elements, entering RLOCs can be a time-consuming and error-prone task.

ProActive Timing Closure

The Macro Builder works with the ISE 5.1i Floorplanner to further facilitate the creation of RLOCs. The ISE 5.1i Floorplanner can be used to locate the logic of your user-defined IP core and then save the placement information in the form of RPM RLOCs in a netlist constraint file (NCF). The NCF, along with the original EDIF netlist, gives you a complete description of your "known-good" design block.

Simple Steps to Create a Reusable Design Block

The following shows the steps to create a reusable design block with ISE 5.1i:

1. Write HDL description (e.g., core.v or core.vhd) for the design block function.
2. Synthesize the core HDL – without I/O insertion – to get the netlist (e.g., core.edf).
3. Use the ISE Constraint Editor to create and apply timing constraints to the design block via UCF (e.g., core.ucf).
4. ISE Translate and NGDBuild will take EDIF and UCF to generate NGD (e.g., core.ngd).
5. (Optional) Use ISE Floorplanner to define area group to constrain the core in a fixed "shape."
6. ISE Implement via MAP/PAR generates NCD (e.g., core.ncd).
7. Make necessary iterations to meet the timing goals.
8. Open the core design in ISE Floorplanner.
9. Read in "placed" NCD and the physical constraints from the placement to make the floorplan match the post-PAR placement.
10. Save via "Write RPM to NCF ..." command on File menu of ISE 5.1i Floorplanner (e.g., core.ncf).

Figure 1 shows the flow chart of the reusable design block creation process.

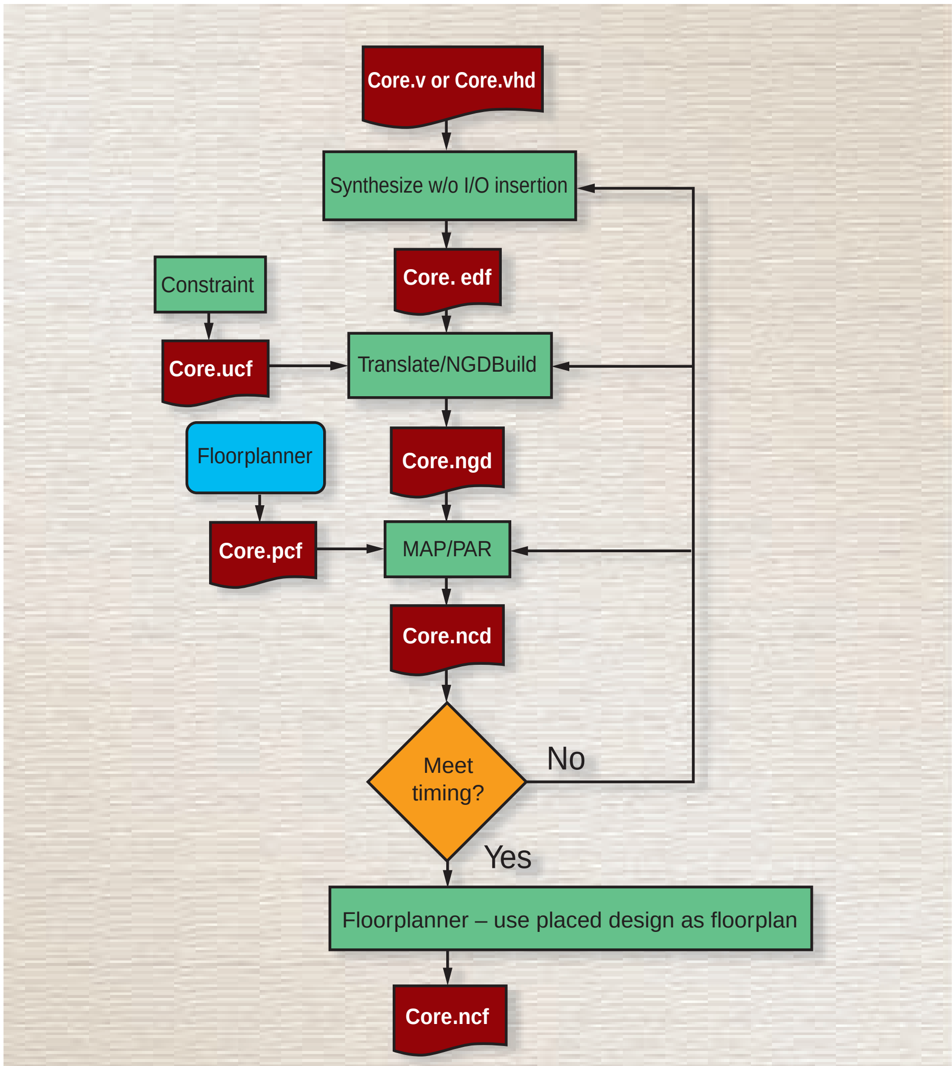


Figure 1 - Macro Builder flow chart

Reuse Predefined Design Blocks

Instantiating predefined design blocks (typically, Netlist (core.edf) and NCF (core.ncf) in any of your projects is now very simple. All you have to do is write the top-level HDL description containing one or more instances of the design block, then synthesize and implement your design as usual. ISE NGDBuild automatically searches for the NCF (e.g., core.ncf) when it processes the netlist (e.g., core.edf), and annotates each element of each instance in the top-level design with the RLOCs from the NCF.

Figure 2 shows the flow chart of the predefined design block reuse process. ISE 5.1i PAR uses the RLOCs during placement, thereby preserving the performance of the original core implementation. Placement runtime of the reused design logic is typically very fast.

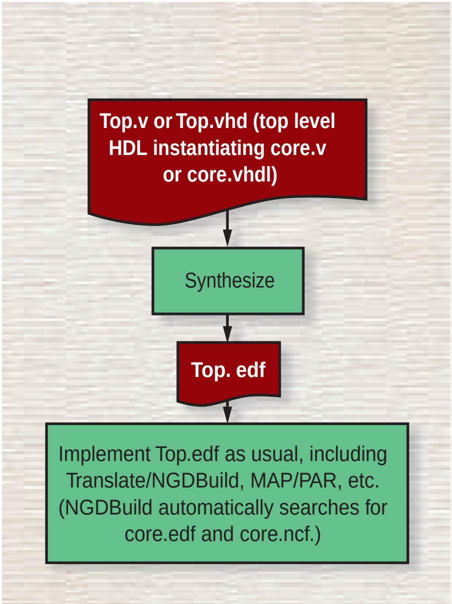
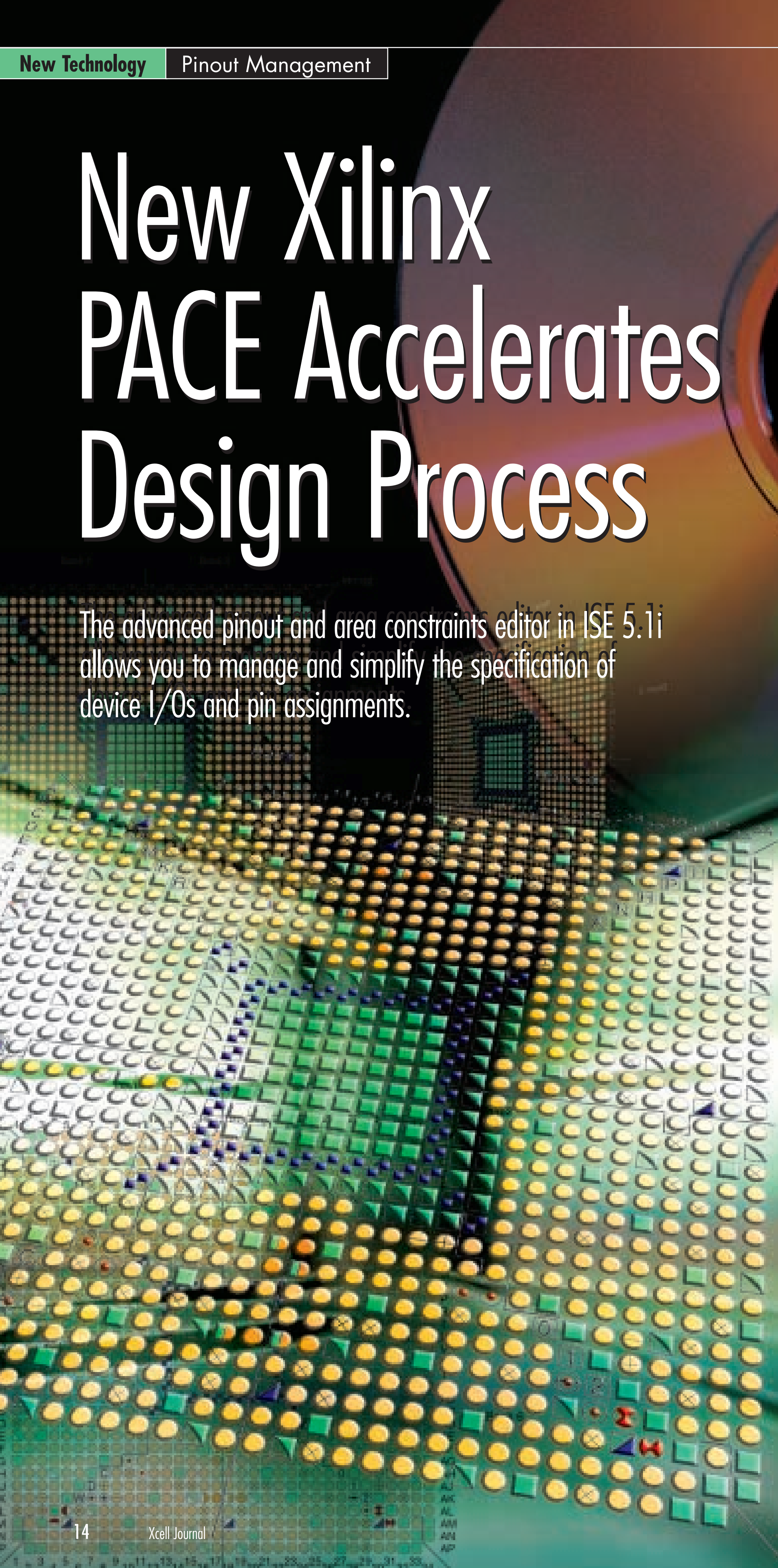


Figure 2 - Design block reuse flow chart

Conclusion

We believe that Macro Builder, the new ProActive Timing Closure capability available in ISE 5.1i, will enable you to easily create any large design block or IP core design for reuse with highly repeatable performance. The Macro Builder can help you achieve your timing requirements quickly, significantly shorten the design process, speed up time to market, and reduce development cost. To find out more about ISE 5.1i or to obtain an evaluation copy of ISE 5.1i, visit *www.xilinx.com/ise.*



New Xilinx PACE Accelerates Design Process

The advanced pinout and area constraints editor in ISE 5.1i allows you to manage and simplify the specification of device I/Os and pin assignments.

by Mark Goosman
Software Product Marketing Manager
Xilinx, Inc.
mark.goosman@xilinx.com

Today, team-based design is essential to achieve faster time to market. The inability to describe device I/O and area constraints until late in the design process has become a major problem for FPGA designers dealing with larger board designs. The new, more complex boards and devices with larger pin packages offer far greater functionality than was previously available – but they also make it more difficult to define I/O logic, specify pin assignments, and understand area constraints. Even minor changes in the middle of the design process can alter pinouts and resource requirements.

The Xilinx PACE (Pinout and Area Constraints Editor – Figure 1) is an interactive graphical application included in the new ISE 5.1i software. PACE allows you to define I/O logic and devices, make pin assignments, and create area constraints at the very beginning of your design process – enabling you to finish faster.

PACE Gets Your Design Off to a Good Start

PACE supports I/O layout via an NGD file, so you can use it at the design entry stage. The NGD file is a native generic database file that describes your logical design after it has been reduced to its Xilinx primitives. PACE reads the NGD file and writes a user constraints file (UCF).

With PACE's advanced features, you can easily:

- View and edit location constraints for I/O and global logic
- Create area constraints for hierarchical symbols in your design
- Evaluate the connectivity and resource requirements of your design
- Explore the resource layout of your target FPGA
- Determine how your design maps onto the FPGA via location and area constraints.

Xilinx PACE supports Virtex™, Virtex-E, Virtex-II, Virtex-II Pro™, Spartan™-II, and Spartan-IIE FPGAs.

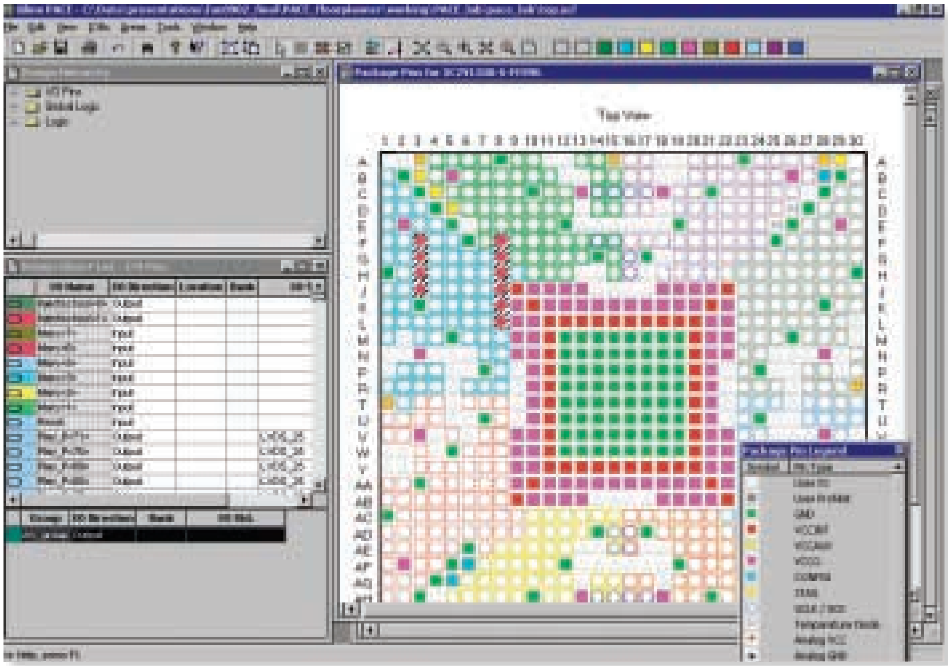


Figure 1 - Xilinx pinout and area constraints editor (PACE)

Pin Assignment Made Simple

The sophisticated PACE pin assignment function assigns I/O locations, specifies I/O banks and I/O standards, prohibits certain I/O locations, and creates legal pin assignments.

The PACE Pin Package window displays the appropriate pin layout for your specified FPGA package (BG, PG, FG, PQ, or CS) – making it easy to define and explore device I/O assignments. You simply drag and drop I/Os into the Pin Package window. Pins already floorplanned are displayed in the same color in which they appear in the Design Hierarchy window.

The Pin Package window also allows you to prohibit assignments to designated pins, including automatically prohibiting assignment of user I/Os to "special" pins, such as those used for configuration (for example, SelectMAP or JTAG) or as voltage reference (VREF) pins for certain SelectI/O™ modes.

A Bird's-Eye View of Available Resources

The Device Architecture window gives you an abstract view of the resources available for your target device. Internal logic resources are represented as a grid of "tiles," with each tile representing a CLB (configurable logic block), or device "slice." The logic elements in these tiles reflect the architecture of the device. For example, Virtex devices are represented with CLB tiles, each composed of four LUTs, four registers, four carry multiplexers, two BUFTs, and miscellaneous logic gates. This window also displays abstract representations of I/O logic, as well as specific global logic, such as clock buffers, block RAM, and so forth.

You can designate resources or CLB/slice tiles in the Device Architecture window as being either "prohibited" or "allowed." Prohibited resources are grayed out (Figure 2).

Area Constraints

The Area Constraints function is useful for defining and displaying a high-level abstraction (Device Architecture View) of the elements within your design. You can also create area constraints for logic in your

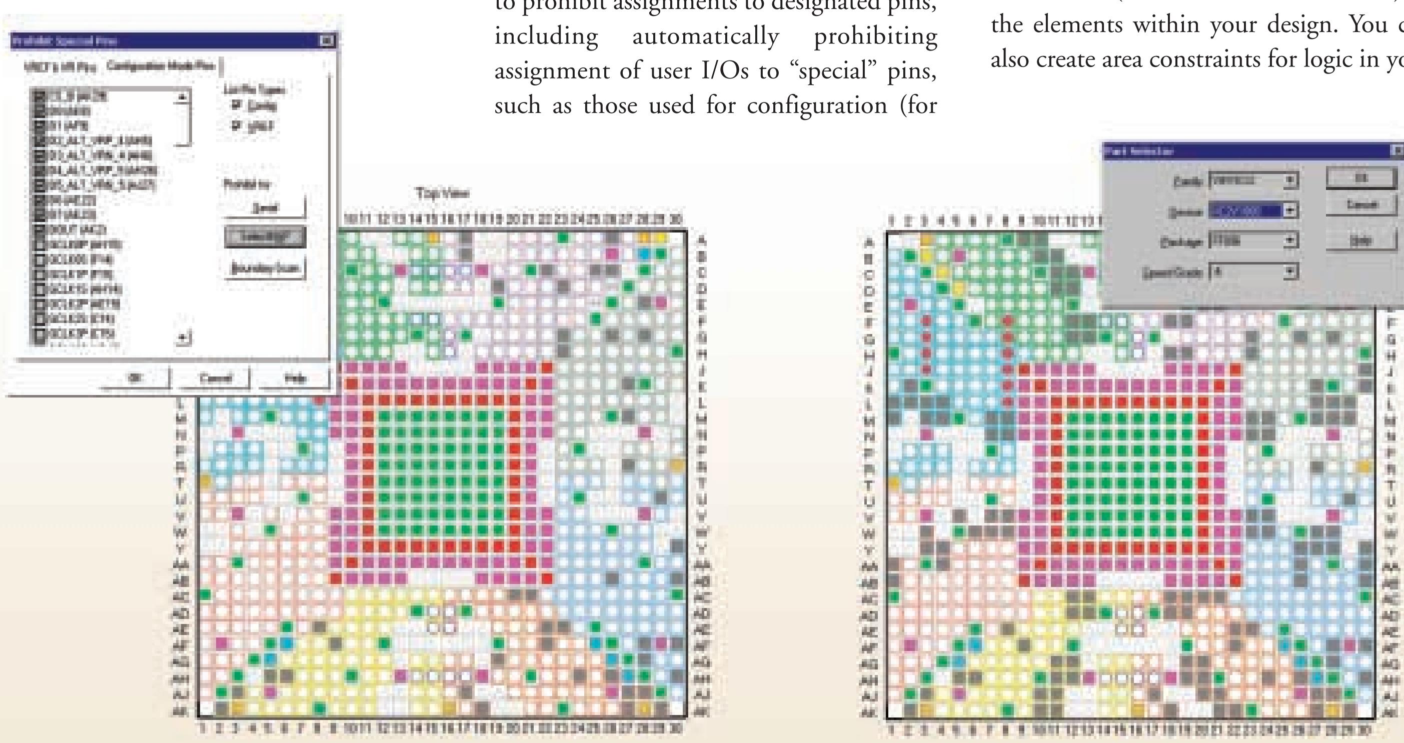


Figure 2 - Prohibit selected pins (left) – Prohibit specific pins for smaller devices (right)

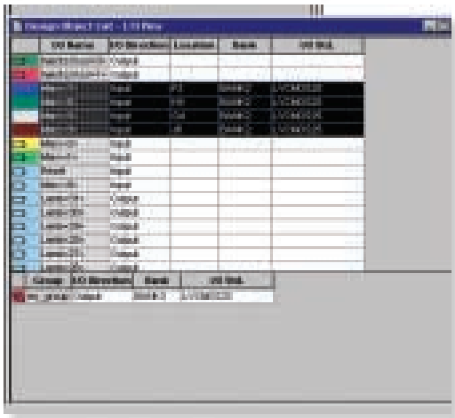


Figure 3 - Area Constraints function of the Device Architecture View window

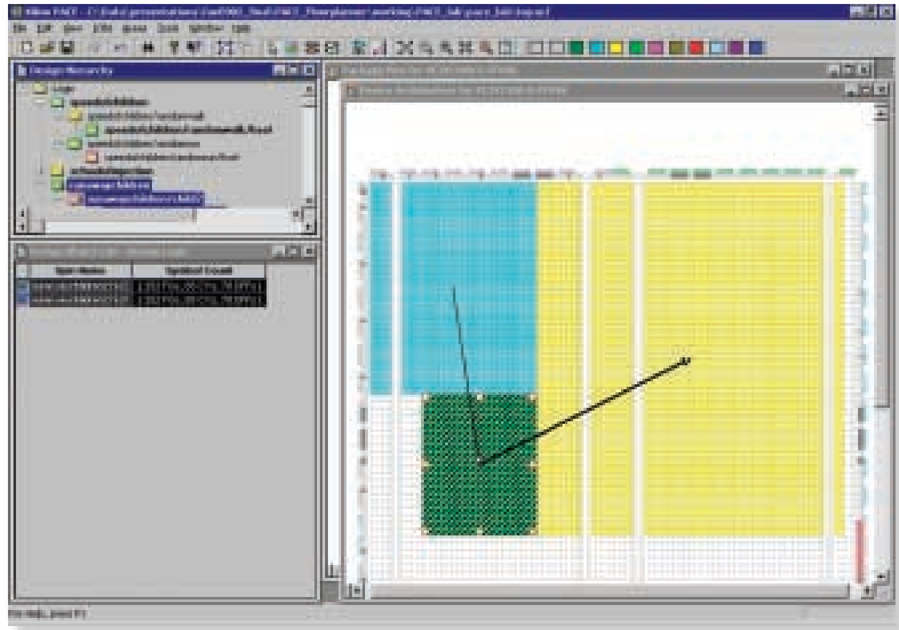


Figure 4 - Definition of complex area constraints

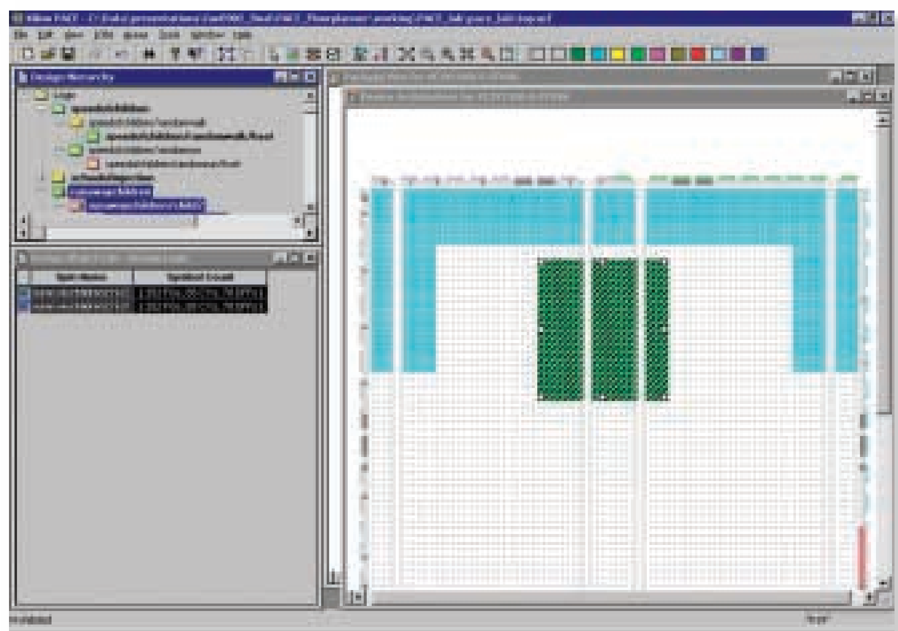


Figure 5 - Design Hierarchy window

design, and display I/Os on the periphery to show connectivity (Figure 3).

Complex area constraints are depicted as a set of rectangles, any one of which can be easily created, moved, resized, or deleted. It's also easy to add a new rectangle to a selected area constraint, append an area group to another rectangle to make the shape more complex, and create non-rectangular area groups, such as T-shapes, L-shapes, and other arbitrary shapes (Figure 4). You can even define area groups by hierarchical boundaries – which is especially useful in an incremental design methodology.

Area constraints associated with nodes of your design hierarchy are also displayed as rectangles. The rectangles can overlap, providing the overlap does not leave a deficit of resources for the constrained logic. Handles make it easy to move and resize areas.

Whenever you resize a rectangle, PACE automatically estimates the requirements of the new area, based on your designated padding value and the size of existing rectangles. The size of the new rectangle will meet this minimum size requirement. If the minimum requirement has already been met, the new area size is one tile.

Design Hierarchy Browser

The Design Hierarchy window (Figure 5) is extremely useful for managing your design's overall hierarchy and grouping structures. You can use the Design Hierarchy window to navigate your design hierarchy, create custom I/O groups, and quickly navigate between various sections of your design.

The Design Hierarchy window contains hierarchy elements for I/O pins and logic, including global logic. To manipulate the hierarchy, simply drag and drop I/O symbols from the Design Hierarchy window into the Package Pin or Device Architecture window.

Conclusion

The addition of PACE to ISE 5.1i makes it easy to create constraints early in the design process by defining FPGA pin assignments and area group boundaries. These capabilities, along with the PACE point-and-click interface, will help you finish designs much faster and easier. In addition, PACE's built-in capabilities ensure that pinouts and area groups are correct, significantly reducing the risks associated with making changes in your design – even late in the design cycle.

How to Control Hierarchy Retention in Xilinx ISE 5.1i

With ISE 5.1i, you can break down your design into a hierarchical structure of modules, reducing the complexity of construction, analysis, and verification.

by Brian Philofsky
Software Technical Marketing Engineer
Xilinx, Inc.
brian.philofsky@xilinx.com

The hierarchical modularity typical of modern PLD designs contributes substantially to the efficiency and reliability of front-end RTL design verification. However, for earlier versions of the Xilinx ISE, preserving this hierarchy information through the design flow was problematic. Much of this hierarchy information was often lost by the time the design flow reached the timing analysis and back-end verification. Without this information, these verification steps had to rely on a monolithic rather than a modular examination of the structural design. This verification method was inefficient, more error-prone, and time consuming.

Xilinx ISE Release 5.1i implements several improvements throughout the design flow to permit you to select between exact retention of design hierarchy for those areas of the design where ease of analysis is paramount. ISE 5.1i also allows hierarchy flattening when the design demands optimization across hierarchy boundaries. For each area of the design, you can employ the optimum tradeoff between design visibility for back-end verification, and the best possible performance and area optimization.

5.1i Design Flow Enhancements

Modern PLD designs are typically so complex that it is unrealistic to design and verify them as monolithic objects. Breaking the design down into a hierarchical structure of modules reduces the complexity of construction, analysis, and verification to manageable levels. Even less complex designs routinely benefit from hierarchical structure with improved understanding, documentation, and code reusability.

In previous versions of the Xilinx ISE software, hierarchy retention was an optional switch for simulation netlist creation. However, the actual hierarchy created did not always correlate well to the input design hierarchy because of synthesis and place-and-route (PAR) tool optimizations. Although some boundary optimizations can be controlled from the synthesis tool with the use of synthesis directives or global optimizations, this information was never communicated to PAR, which, as a result, was unable to determine which modules were intended and preserved by synthesis and thus should be preserved and recreated for post-PAR simulation.

Also, a synthesized design often contained both user-created and unintended hierarchies. Unintended hierarchies were created by mechanisms such as generate statements, primitive instantiations, and third-party IP hierarchy. Xilinx ISE 5.1i expands the use of an existing user attribute, KEEP_HIERARCHY, to communicate to the back-end PAR tools which hierarchies were preserved in synthesis and are intended to be preserved throughout the tool flow. This ensures the hierarchy that improves design verification will be preserved – and the hierarchy that does not improve design verification – will be flattened for improved path optimization. Figure 1 illustrates how the design flow has changed in the 5.1i release.

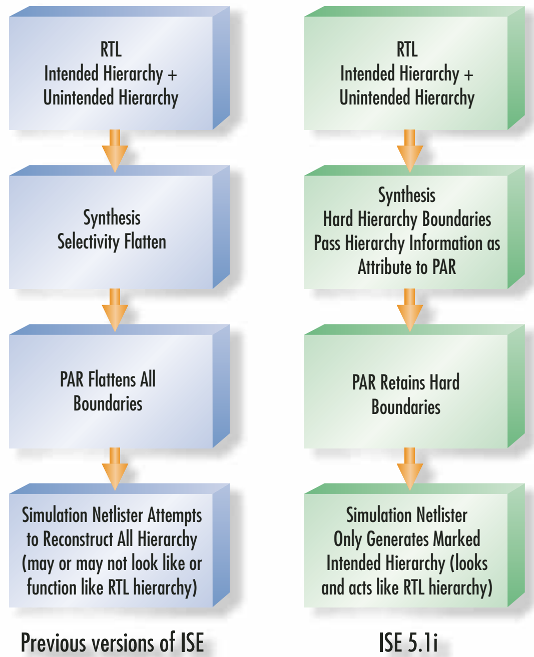


Figure 1 - Flow changes in ISE 5.1 for hierarchy retention compared to previous versions of the software

How to Use KEEP_HIERARCHY Attribute

The current version of the Xilinx Synthesis Tool (XST), as well as future releases of Synplicity and other synthesis tools, will pass the KEEP_HIERARCHY attribute automatically when hierarchy is preserved during synthesis.

For versions that do not currently pass the attribute automatically, the KEEP_HIERARCHY attribute may be passed manually, within the RTL code, within a synthesis constraint file, or within a user constraints file (UCF). Table 1 shows example syntax for passing the attribute.

For alternative flows, such as incremental design, modular design, or any bottom-up synthesis flow that generates separate EDIF files for each level of hierarchy, use the -insert_keep_hierarchy switch in Ngdbuild to automatically place the KEEP_HIERARCHY attribute on each input EDIF file.

For the ISE Project Navigator interface, you can specify the KEEP_HIERARCHY attribute by selecting the "Preserve Hierarchy on Sub Module" option in the "Advanced Process" menu. Figure 2 shows the location of this switch in the ISE Project Navigator tools.

The KEEP_HIERARCHY methodology works exceptionally well with these design flows by supporting modular design techniques that

Example UCF syntax (assuming hierarchy was preserved during synthesis):

```
INST hierarchy_name KEEP_HIERARCHY=TRUE;
```

Example of Synplicity SDC file syntax:

```
define_attribute {v:module_name} syn_hier {hard}
define_attribute { v:module_name} xc_props {KEEP_HIERARCHY =TRUE}
```

Example of Synplicity Verilog code syntax (module instantiation):

```
module_name instance_name(port_mapping) /* synthesis syn_hier="hard"
xc_props="KEEP_HIERARCHY=TRUE" */;
```

Example Synplicity VHDL code syntax (placed in architecture of preserved hierarchy):

```
attribute syn_hier : string;
attribute syn_hier of architecture_name: architecture is "hard";
attribute xc_props : string;
attribute xc_props  of architecture_name: architecture is "KEEP_HIERARCHY=TRUE";
```

Table 1 - Example syntax of how to pass KEEP_HIERARCHY for Synplicity

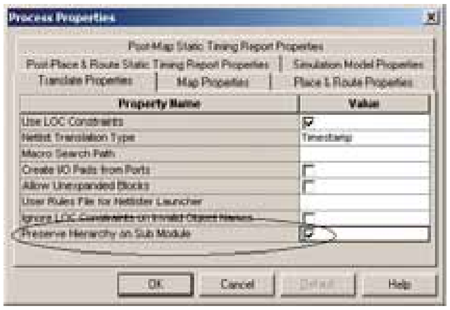


Figure 2 - Location of "Preserve Hierarchy on Submodule" option in ISE 5.1i

lend themselves to incremental design creation, implementation, and verification.

Once the attribute is placed on a module, the back-end PAR tools will preserve the instance names, port declarations, and logic functions within the hierarchy boundary. The resulting simulation netlist produced by the VHDL or Verilog™ netlister will contain the desired hierarchy structure for back-end timing and functional verification.

Convergence of Timing and RTL Simulation Methodologies

An important benefit of the new KEEP_HIERARCHY feature is the similarity that it establishes between the front-end and back-end verification methodologies. The structural simulation timing netlist may be loaded into any HDL simulator and used similarly to the RTL netlist. Hierarchical forces, breakpoints, probes, and watch points created during RTL simulation should exist at all hierarchical ports.

Simulator scripts, the waveform viewer, or the testbench itself may reference hierarchical signals just as in the RTL simulation. Testbenches and RTL simulation scripts should be generally reusable. Identifying signal locations in the hierarchy browser of the simulator should be easier and schematic views of the structural design should look similar to those in the original design source code.

This new methodology should also improve the productivity of structural timing simulations. Figures 3 and 4 show how the hierarchical netlists can be used in popular simulators to improve back-end verification.

Conclusion

As PLD design size and complexity increase, design verification is becoming a significant bottleneck within the design process. Xilinx ISE 5.1i will dramatically improve your control over hierarchy retention, which will help alleviate this bottleneck by increasing the efficiency of verification and shortening the verification cycle for PLD designs.

The new ISE 5.1i software products are now available for purchase at the ISE website at *www.xilinx.com/ise/*.

Further improvements to this feature are planned for future releases of Xilinx ISE software, including the ability to write out separate structural netlists and SDF files for each level of preserved hierarchy, allowing for true modular verification of the design.

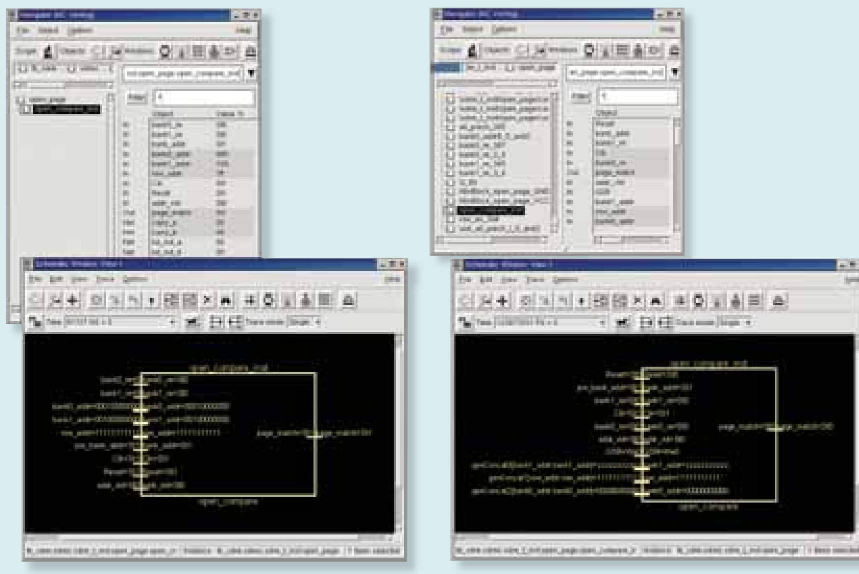


Figure 3 – Views of a sub-level module in Cadence NC-Sim Navigator window and Schematic window; left side is RTL design and right side depicts structural view of same design.

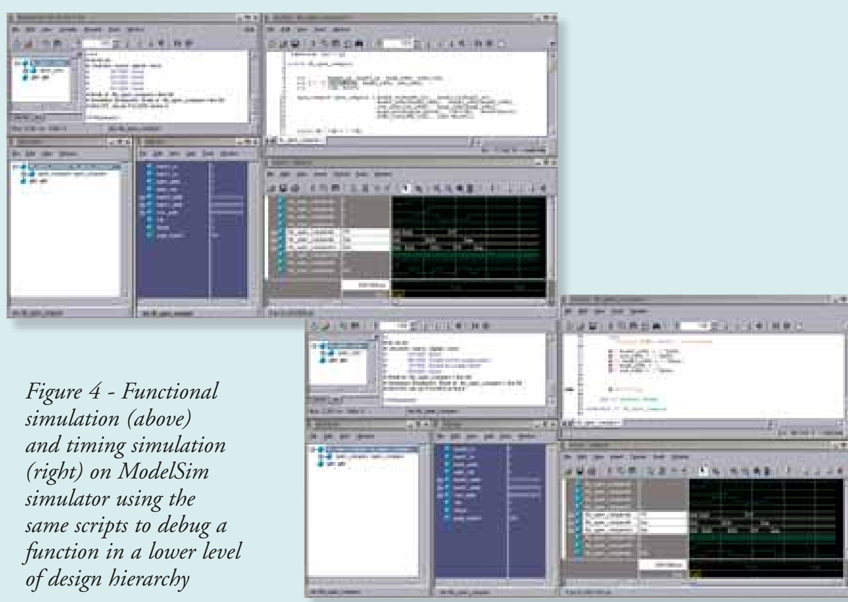
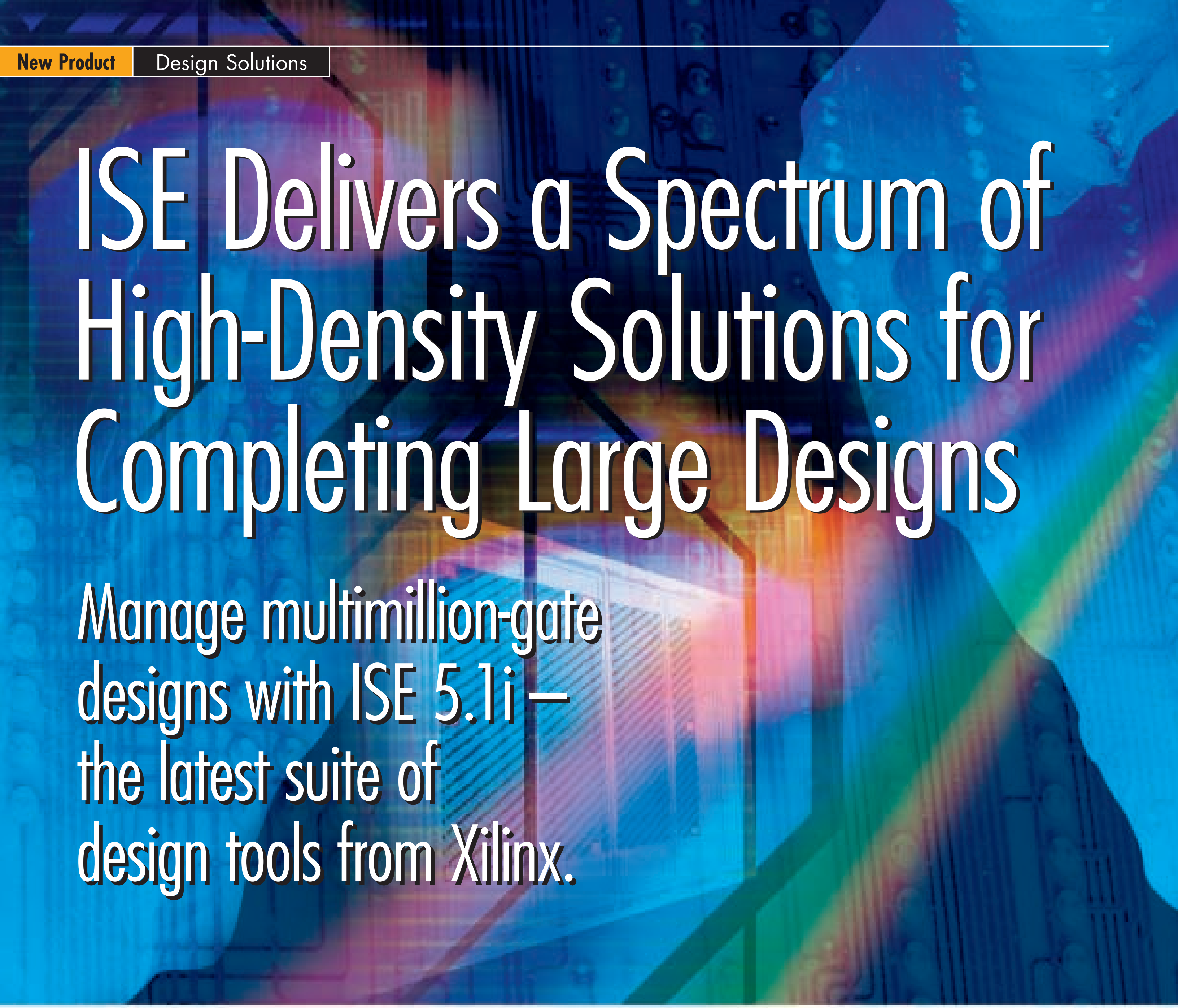


Figure 4 - Functional simulation (above) and timing simulation (right) on ModelSim simulator using the same scripts to debug a function in a lower level of design hierarchy



ISE Delivers a Spectrum of High-Density Solutions for Completing Large Designs

Manage multimillion-gate designs with ISE 5.1i – the latest suite of design tools from Xilinx.

by Lee Hansen
Product Marketing Manager
Xilinx, Inc.
lee.hansen@xilinx.com

As logic design sizes now routinely exceed 1 million gates, new pressures unique to high-density designs are being felt by logic engineers worldwide. ISE (Integrated Software Environment) from Xilinx is responding with a spectrum of technology for larger designs, including Incremental Design, available in ISE 5.1i. These technologies augment the larger design flow with "divide and conquer," and performance-locking strategies that help bring large, daunting projects under control.

Area Groups – Quick-and-Easy

Area groups is a quick-and-easy way to bring a measure of control to your project. Engineers can map areas of logic for the target FPGA using either the new PACE tool (Pinout and Area Constraints Editor) in ISE 5.1i, or the ISE Floorplanner. PACE lets you create area maps around hierarchical HDL boundaries automatically, or let PACE give area estimates for target logic that you can either use or modify and draw by hand. Figure 1 demonstrates PACE being used to define a logic area.

Defining area groups delivers many design advantages. First, and most simply put, being able to "see" the different areas of logic can help delineate regions where different design entry methods are being used, partition out areas for design reuse or IP placement, or point out where the "known problem" areas of the design will occur. But defining area groups also offers technical advantages as well. Most important, area planning the design correctly can accelerate timing closure by keeping critical logic cells and paths together, and by minimizing the number of interface ports between modules.

Using area groups is a good and fast methodology to help gain some advantage over a large design, but area groups do not provide control over design changes.

Incremental Design – Change Without Risk

In the middle of the high-density technology spectrum is a new capability called Incremental Design, available at no cost in ISE 5.1i. Incremental Design combines the quick-and-easy aspects of area groups with performance-locking, to offer a measure of immunity to late-cycle design changes.

With Incremental Design, engineers can use PACE to assign area groups along hierarchical HDL boundaries, as previously discussed. The overall design is then completed as usual. Should a design change occur after or close to completion – Incremental Design guarantees that only the area that needs to change has to be re-implemented. The remainder of the design stays locked and intact.

Incremental Design reduces the overall design re-compilation time by focusing the implementation cycle on the module that needs to change. During debug and verification this speedup offers a number of advantages; including more debug iterations possible, faster overall verification cycles, and allowing engineers to focus on the real design problems rather than recompiling the entire design over and over.

Incremental Design also delivers faster design completion when late design changes must occur. A recent informal survey of Xilinx customers indicated that every one of their logic design projects underway in 2001 had at least one late-cycle design change that occurred after design freeze, negatively affecting the overall completion date. Incremental Design delivers an overall completion advantage for this common problem, and can help with large and midrange design sizes as well.

Modular Design – Divide and Conquer Management

At the high end of the high-density spectrum is Modular Design, an optional team design technology that can be purchased and then added to your ISE software environment. Modular Design implements a "divide and conquer" approach for corporate environments that deploy teams of engineers on large designs.

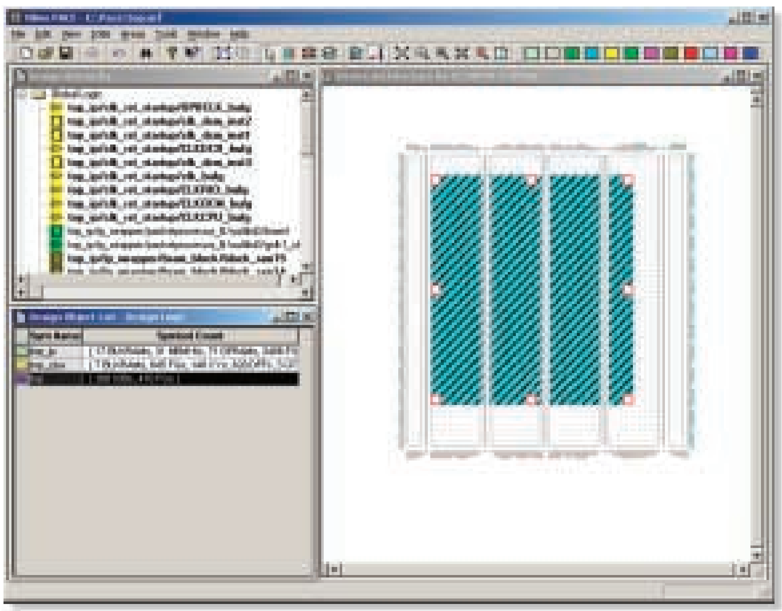


Figure 1 - PACE Area Management

Modular Design requires the design manager to plan the larger design ahead of time, based on knowledge of which engineers will be assigned to each portion of the design. The design manager can then use the ISE Floorplanner to partition the overall larger design into smaller "modules," which are then implemented independently. All of the ISE design tools and flows can be brought to bear on the smaller modules individually and in parallel. Engineers are focused solely on the smaller and more direct task of completing just their respective modules. Once a module is finished, its place and route results are locked while the manager waits for all modules to be completed.

Modular Design delivers full planning control and faster project completion over the larger design, implementing a true bottoms-up design approach that completes the larger design via smaller modules implemented in parallel.

Macro Builder – Locked Performance

Also included in ISE 5.1i, the new Macro Builder function lets you generate design macros for saving your design away. Using ISE Floorplanner on a design that has been placed, the "write RPM to NCF" command saves away the placed floorplan along with the design file. This new macro, including relative placement information, can now be registered with your IP cataloging tool and then reused in later designs. Macro Builder helps corporate environments leverage their HDL development, reduces overall costs for future designs, and lets you save away "known-good" designs.

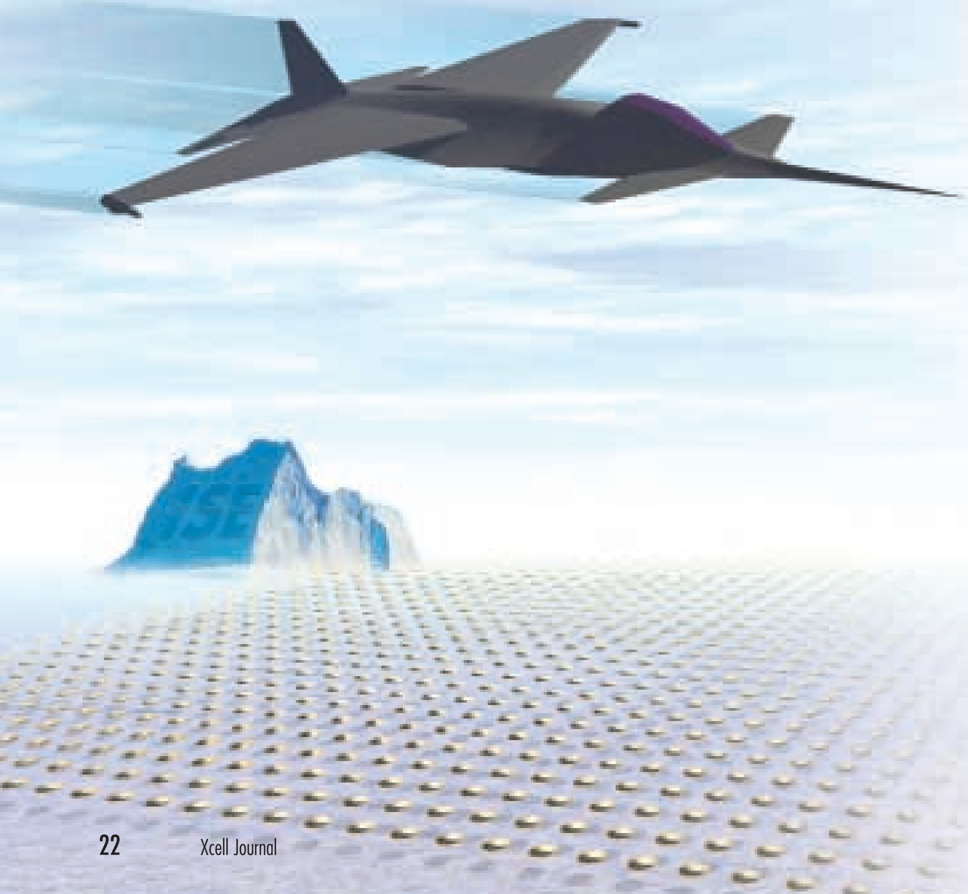
In this tight economy, Macro Builder helps managers make more efficient use of their HDL investments. When a new project is started, design time is saved by reusing proven functions, and not having to re-create design sections that previously worked before. And engineering resources can be utilized during project downtimes, to create modules for future use, starting the next project with an even greater completion time advantage.

High-density Design Made Easier

ISE 5.1i includes a spectrum of strategies that bring larger design sizes under control. From quick-and-easy area management to team-based "divide-and-conquer" methodologies, ISE offers technology that streamlines your high-density design process and works the way you expect it to. For more information on ISE 5.1i visit *www.xilinx.com/xcell_ise* and contact your local sales representative to order ISE 5.1i

Finish Faster with ISE 5.1i Architecture Wizards

The latest release of ISE 5.1i allows you to complete Virtex-II Pro designs faster with Architecture Wizards for multigigabit I/O configurations and complex clocking schemes.



by Craig Willert
Product Line Marketing Manager
Xilinx, Inc.
craig.willert@xilinx.com

Flexibility and performance are just two of the many reasons why engineers today are choosing Xilinx Platform FPGAs more than any other logic devices. Now, Xilinx ISE (Integrated Software Environment) 5.1i logic design tools, and new Architecture Wizards, make it easy to take advantage of such advanced functionality as programmable RocketIO™ interfaces and advanced digital clock management. These features enable high clock speeds within the chip, and high bandwidth when communicating between chips and across high-speed backplanes.

Flexibility by Design

The Xilinx Virtex-II Pro™ platform for programmable systems can be used to enhance your design's performance, both internal and external to the chip. Today's multiclock systems with internal clock rates exceeding 300 MHz will benefit from the added functionality of the Xilinx Digital Clock Manager (DCM). With DCM, you can synthesize a number of low-skew clock signals and customize almost every aspect of the clock's behavior with such features as:

- **Frequency synthesis** – supports user-programmable clock multiplication and division, with several options
- **Phase shifting** – configurable for both coarse and fine-grained shifting with dynamic phase shift control
- **Clock de-skew** – both on-chip and off-chip with user-designated clock references.

Based on your application, you might use multiple configurations – and even combinations of DCMs – in your design. With these complex clock management capabilities, you can create higher performance designs than ever before.

Xilinx Virtex-II Pro FPGAs provide as many as 24 RocketIO transceivers to support several emerging serial connectivity standards, including PCI Express, serial RapidIO™, InfiniBand™, Fibre Channel, and 10 Gigabit Ethernet XAUI. Each RocketIO transceiver delivers 622 Mbps to 3.125 Gbps baud rate – as well as channel bonding to aggregate multiple channels – thus supporting a wide range of baud rates within these standards.

In addition, the transceivers have several configurable features, such as bypassable 8B/10B encoder/decoder, scalable FPGA data path interface, programmable output voltage swing, and so on. Choose from among the 100 predefined configurations to manage the myriad I/O standards and proprietary interfaces prevalent today. In fact, by leveraging the flexibility offered by RocketIO multi-gigabit transceivers in Virtex-II Pro FPGAs, Xilinx customers have already delivered working designs incorporating new serial standards demonstrated at Programmable World 2002 and SuperComm 2002.

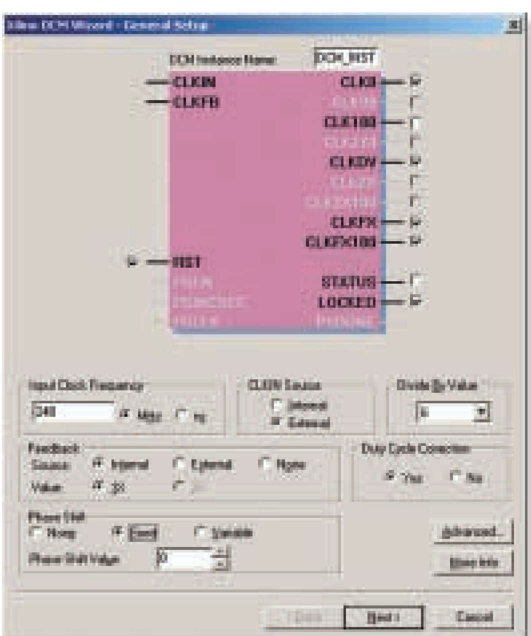


Figure 1 - Digital Clock Manager Wizard

What Design Gap?

One of the challenges plaguing design engineers over the years has been the "design gap" – that gap between the capabilities available in leading-edge ICs and a design team's ability to take advantage of them. For example, even though Xilinx customers don't have to deal with the complexity of physical design in ASICs, the immense flexibility of such advanced features as DCM and RocketIO blocks can make it difficult to use their capabilities fully – introducing a programmable logic design gap.

Xilinx Platform FPGAs have always put you on a design platform closer to your goals. We keep the design gap to a minimum – and give you a jump-start on developing leading-edge systems – by making it easy to use device features. Continuing in this tradition, ISE 5.1i offers new Architecture Wizards that enable you to configure and harness the full capabilities of the advanced features in Virtex-II Pro FPGAs.

The Architecture Wizards

You don't have to be a rocket scientist to design with RocketIO and DCM. Driven by an intuitive graphical user interface (GUI), one Architecture Wizard walks you through the process of customizing the RocketIO or DCM capabilities and generating HDL. The Architecture Wizard ensures that it's done right the first time, and that your design will interface with all leading HDL synthesis and simulation tools from Xilinx and our partners.

Figure 1 illustrates some of the clock management capabilities provided by the DCM Wizard, and the way you can use it to define DCM inputs and outputs. A second DCM Architecture Wizard provides control over the definition of the generated clock.

Figure 2 depicts the general setup dialog box of the RocketIO Wizard. You can see that the GUI makes it easy to define the transceiver's configuration. Additional dialog boxes are used to define advanced configuration features, such as channel bonding.

Figure 2 - RocketIO Wizard

The Architecture Wizards are tightly integrated with ISE's Project Navigator, and they are also available for standalone operations. From within the Project Navigator, Architecture Wizards are initiated by adding new source code to your ISE project. To operate it as a standalone, simply type the command line arwz, and the intuitive GUI makes it easy to define the appropriate parameters for your application.

Conclusion

ISE's 5.1i Architecture Wizards enable you to take advantage of the programmable logic industry's most advanced features quickly and easily. The ISE Architecture Wizards help you finish faster by automating the creation of synthesizable HDL – helping you to build world-class designs that include multi-gigabit transceivers and advanced clocking schemes. Go to *www.xilinx.com/xcell_ise/* to find out more about the latest and greatest features of ISE 5.1i.

New ChipScope Pro Integrated Bus Analyzer

Powerful Debugging Tools for Virtex-II Pro FPGAs

Debugging bus transactions is now faster and easier than ever before.

by Brent Przybus
Product Marketing Manager
Xilinx, Inc.
brent.przybus@xilinx.com

Debugging complex system-level Virtex-II Pro™ designs can be quite a challenge. These FPGAs contain many advanced features, including PowerPC™ processors and RocketIO™ multi-gigabit serial transceivers – all the signals you need to verify are inside the device and inaccessible to the usual logic analyzers. However, now there is an elegant way to trace any internal Virtex-II Pro signal; it's called the ChipScope™ Pro Analyzer.

Two years after defining on-chip debugging, the ChipScope engineers have created the latest solution called IBA (Integrated Bus Analyzer). Using this new core, you now have point access to the bus transactions that occur in the IBM CoreConnect™ structure – this is the critical interface between processor peripheral logic cores and the processor itself. Because the CoreConnect bus is implemented in programmable logic, ChipScope Pro cores have access to every available signal. Add to this complete knowledge of the IBM CoreConnect standard definition, and you have a powerful ally in your quest to debug and verify your next FPGA design.

Point Access to System Busses

ChipScope Pro cores support true system-level debugging that includes bus-level monitoring and debug capabilities. The 5.1i release of ChipScope Pro tool includes the first of several new IBA cores designed specifically for the IBM CoreConnect Bus Architecture. The first core available has been predefined and built specifically for the On-chip Peripheral Bus (OPB) and is based on the IBM CoreConnect standards specification. The OPB bus is designed to alleviate congestion and system performance bottlenecks on the Processor Local Bus. Many common peripherals such as UARTS, GPIO, system timers, and other devices will use the OPB to interface to PowerPC or MicroBlaze™ processors.

The ChipScope Pro IBA core provides point access to each of the 32-bit address and data buses as well as control signals associated with the OPB, allowing you to view individual transactions. In addition, ChipScope Pro tools provide OPB protocol error violation detection, capable of detecting and reporting any of the 79 different OPB protocol violations that can occur.

CoreConnect IBA Core Features

ChipScope CoreConnect IBA cores are optimized for the Virtex-II series fabric and include the following features and capabilities:

- **Fast** – CoreConnect IBA cores are designed to operate at the CoreConnect OPB frequency
- **Small** – IBA cores use at little at 3% to 4% of available logic and memory resources in Virtex-II devices.
- **Flexible** – You can use multiple cores in a single design, and place multiple IBA cores to access processor OPB busses associated with PowerPC and or MicroBlaze processors.

ChipScope Pro Analyzer

ChipScope Pro 5.1i features a completely redesigned project-centric user interface that not only supports the new cores available in 5.1i but also provides a convenient interface for system level debugging. The new ChipScope Pro Analyzer provides the following features and benefits:

- Project-centric interface allows you to set up and view data from multiple cores in different windows within the ChipScope Pro Analyzer.
- Advanced trigger setup dialogs support bus analysis as well as logic analysis. You can use the advanced trigger setup capabilities to define complex bus and logic trigger statements; this is ideal for debugging system busses with multiple control signals.
- A powerful listing viewer is now an alternative to the traditional waveform display, and provides the opportunity to view descriptive bus transactions in order of execution.
- Optional time-stamps in the cores allow you to display signal activity and bus transactions referenced to absolute time or by transaction number.

Figure 1 - ChipScope Pro System

- Advanced data display options in the new Analyzer will allow you to plot data-versus-time and data-versus-data; this is a valuable tool for debugging DSP applications.

Easily Add ChipScope Pro Cores to New and Existing Designs

Whether you are working with an existing design or are specifying your next project, ChipScope Pro Analyzer provides quick, easy-to-use tools that allow you define and generate the debugging cores you need.

You can generate cores using the standalone ChipScope Pro Core Generator™ tool or specify ChipScope Pro cores from within the Core Generator tools, part of the Xilinx ISE design environment. These tools will create an HDL file that you can add to the project design HDL for synthesis and implementation.

Alternatively, you can add cores to an existing design using the ChipScope Pro Core Inserter tool. This tool allows you to identify existing signals and nodes within a design and generate ChipScope Pro cores. The ChipScope Pro Core Inserter tool generates the cores needed and creates a design netlist that is merged with your design netlist.

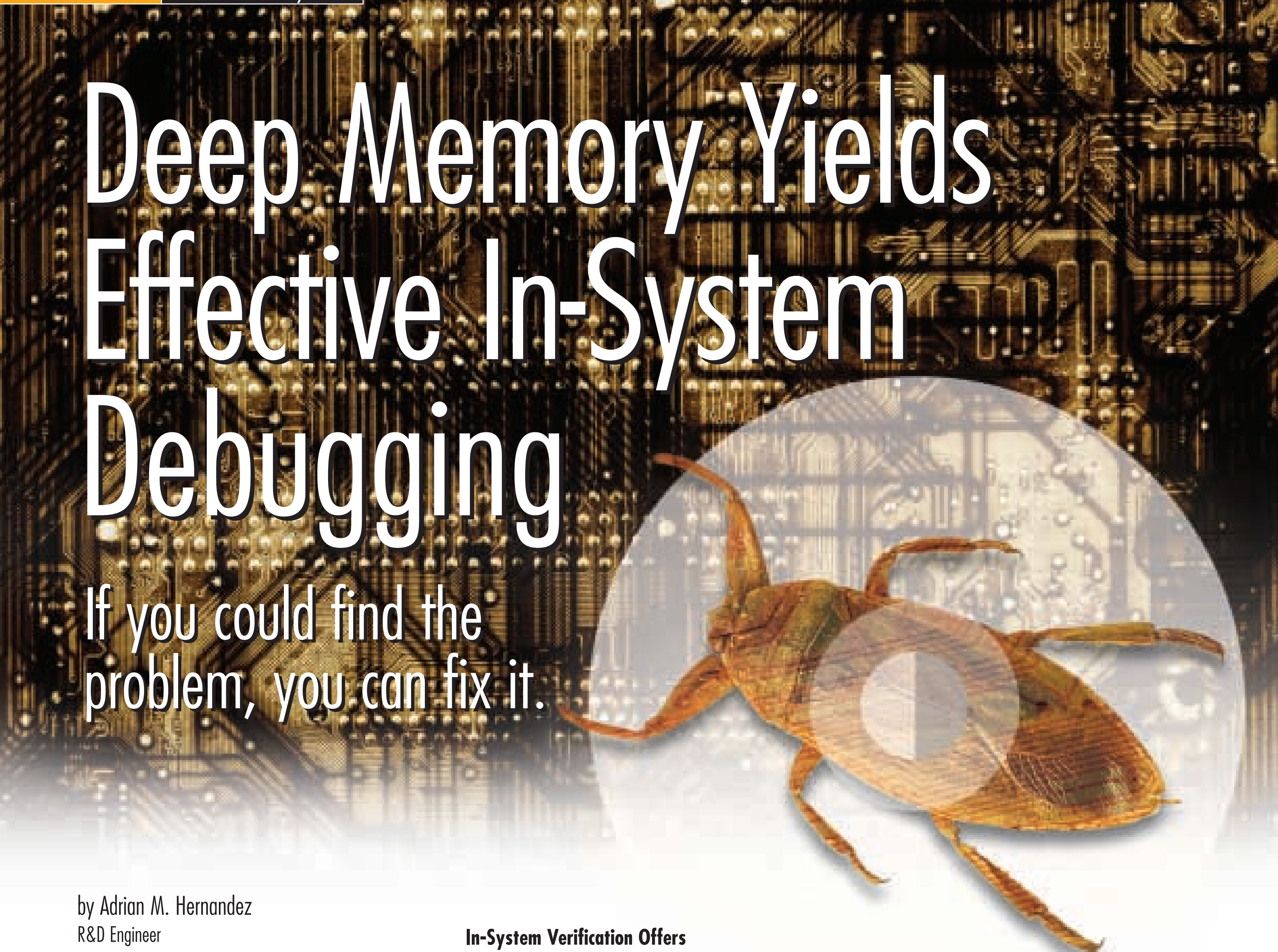
Additional functionality is provided via the FPGA editor tools available in the ISE design environment. Using the FPGA editor, you can reassign signals to existing ChipScope Pro cores without having to create new cores and with minimal impact to your design.

Complete On-Chip Debugging and Verification System

In addition to providing the latest generation CoreConnect IBA core, ChipScope Pro features a new, enhanced ILA (Integrated Logic Analysis) Pro core and the new ATC (Agilent Trace Core) developed by Agilent Technologies. Together these tools make up a complete on-chip system (see Figure 1) that supports core generation, insertion, device configuration, debugging, and verification.

Conclusion

Don't let debugging and verification get the best of your time in your next design. The new ChipScope Pro 5.1i tools are available today and can help you greatly reduce overall development time by shortening the critical debugging and verification phase of a design. Visit *www.xilinx.com/chipscopepro/* today to download a free 30-day full-featured evaluation copy of the tools.



Deep Memory Yields Effective In-System Debugging

If you could find the problem, you can fix it.

by Adrian M. Hernandez
R&D Engineer
Agilent Technologies
adrian_hernandez@agilent.com

FPGAs that can incorporate whole systems have definitely made in-system debugging more challenging. On-chip debugging methodologies aren't always adequate to provide trace memory that is deep enough to capture a sufficient event history. Plus, critical internal nodes may not be readily accessible to external logic analyzers. A new solution comprising Xilinx ChipScope™ Pro tools, the Agilent FPGA Trace Port Analyzer, and the Agilent Trace Core combines the key advantages of internal and external logic analysis. Virtex™-II and Virtex-II Pro™ users now have a solution that combines the best of on-chip debugging with high-speed, deep, external trace using a limited number of pins.

In-System Verification Offers Real-Time FPGA Debugging

Although simulation continues to play an important role in verification of complex FPGAs, in-system verification provides strong complementary value. The primary advantages of in-system verification are that it runs at real-world speeds, enjoys the benefits of real-world stimulus, and has real-world modeling accuracy.

In-System Today

Logic analyzers remain the dominant tool for in-system debugging, with internal nodes routed to the pins. This manual process consumes significant routing resources and – most important – precious pins, but it does offer powerful triggering capabilities, deep memory, and time-correlation. Plus, the logic analyzer can be used for other tasks.

Some designers prefer on-chip debugging that uses an internal logic analyzer, such as ILA (Integrated Logic Analyzer), to develop an internal trace. Here, a logic analysis core is inserted into the FPGA design, and block RAM is used to store resulting traces. JTAG (Joint Test Action Group) is used to set up the logic analyzer and to move the trace buffer from the FPGA to a PC for analysis. The popularity of this emerging method derives from two factors:

- It requires no additional pins outside of JTAG.
- The tools are inexpensive.

Shallower trace depths, triggering that is more limited than with external analyzers, and the lack of time correlation are the primary tradeoffs of this methodology.

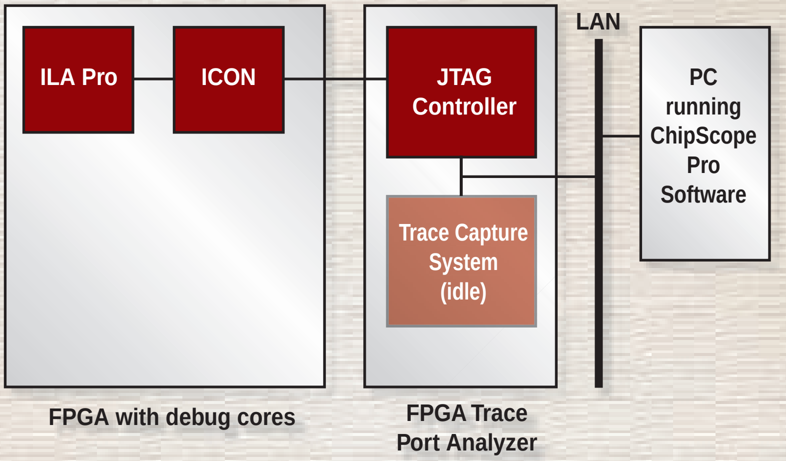


Figure 1 - FPGA Trace Port Analyzer connection to ILA Pro

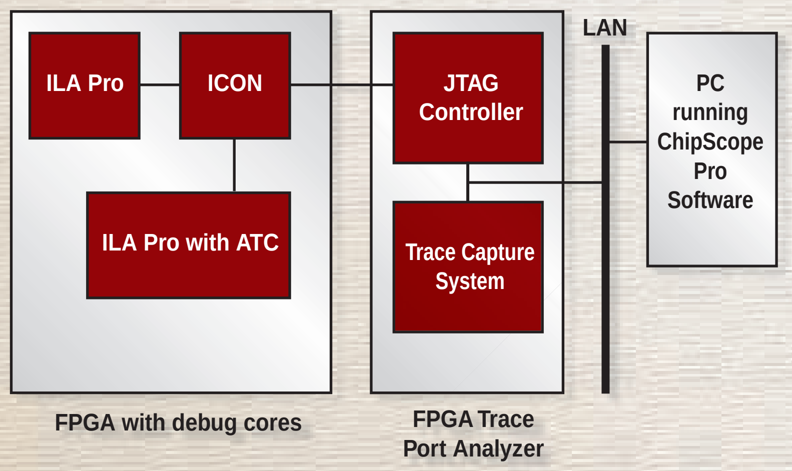


Figure 2 - FPGA Trace Port Analyzer connection using deep external trace memory

The In-System Solution with Deep Memory

The Xilinx/Agilent collaboration has produced a solution that offers the real-time/real-world benefits of in-system debugging with deep trace depth and enhanced triggering. The following describes the different components of this solution.

The Agilent E5904B Option 500 FPGA Trace Port Analyzer provides up to two million states of trace depth for each signal probed, at acquisition speeds up to 200 MHz. This is roughly 60 times deeper than the maximum trace depth offered by ILA Pro (32K) using block RAM. The additional trace depth is especially beneficial in capturing elusive events where symptom and cause may be separated by a long period of time. Another benefit of external trace storage is that it allows you to retain internal FPGA memory for the design instead of dedicating this valuable resource to debugging.

Reduces Dedicated Debugging Pins

The ChipScope Pro 5.1i software ships with a version of ILA Pro connected to an Agilent Trace Core (ATC). The ATC uses time division multiplexing to reduce the number of pins required to pass trace information to the FPGA Trace Port Analyzer for storage. With time division multiplexing, the internal data is accelerated, so a wide bus can be sent out on a few pins. The choices for acceleration, 1x, 2x, and 4x, represent the number of internal nodes sent through a single pin. While single-ended signals can be driven at 200 MHz on the pins, an internal circuit may run at 50 MHz. In this case ATC would produce a 4:1 pin compression ratio. Up to 75 signals can be probed using just 20 pins to pass the trace to the FPGA Trace Port Analyzer.

High-Speed LAN-Based Cable Capabilities

The FPGA Trace Port Analyzer consists of two blocks: a trace acquisition sub-system, and a JTAG control sub-system. The JTAG controller provides a high-speed LAN cable interface between ChipScope Pro and ILA blocks (Figure 1). The controller, which can run up to 30 MHz, is used to configure FPGAs, set ILA triggers, and read back the stored trace data from the ILA block RAM control sub-system. This enables the FPGA Trace Port Analyzer to work with stand-alone ILA Pros, or a combination of ILA Pro with ATC ILA Pro core and one or more ILA Pros (Figure 2).

The Agilent Trace Port Analyzer lets you debug FPGAs remotely via a LAN, which means you can drive ChipScope Pro from your desk and control a design board located in a remote lab. This feature can be quite powerful, especially in conditions where many designers share a single prototype.

Connecting the FPGA Trace Port Analyzer to the Target System

The AMP MICTOR (Matched Impedance Connector) is designed into your target system via a connection with an Agilent Trace Port Analyzer. The MICTOR is a high-speed board connector capable of operating at clock rates above 200 MHz; it has a predefined pinout for the Agilent Trace Port Analyzer (Figure 3). This particular connector and pinout is compatible with both the IBM PowerPC™ 405 CPU trace connector and the Agilent logic analyzer connector. Thus, three instruments can use the same connector for debugging, one instrument at a time – a powerful advantage, especially when the FPGA under design is a Virtex II Pro device.

Debugging an FPGA in the Context of a Larger System

When debugging the FPGA in-system, it is often necessary to time-correlate FPGA events to other system events. The Agilent Trace Port Analyzer enables you to determine quickly whether your FPGA is operational. Using a design as simple as a counter connected to an ILA Pro with Agilent Trace Core, you can validate the FPGA programming and I/O interface in one step. Issues such as JTAG chain connections, FPGA pin configurations, non-functioning system clocks, and stuck traces, can be identified very simply by using the trace output to monitor the activity of the FPGA inputs.

The most complex FPGA malfunction occurs at the PCB (printed circuit board) level. The malfunction can come from a variety of devices or conditions external to the FPGA. For example, when an external processor is used, the FPGA must be able to work appropriately with the processor bus and handle all the additional devices on the bus. But because there can be many real-time situations that occur on a processor bus – interrupts, long burst cycles, and so on – it is difficult to simulate such a situation in software. In fact, sometimes the errors on buses or system boards are not logical, but physical. For this reason signal integrity issues, such as cross-talk, are usually difficult to simulate in software but very apparent in hardware.

Check for Signal Integrity

To facilitate PCB and system measurements, the Agilent FPGA Trace Port Analyzer has two ports, “Trig Out” and “Break In.” The Trig Out port is an output port that signals other instruments, such as Agilent oscilloscopes and logic analyzers (performance has been validated with Agilent instruments only), to complete their measurement. The Break In port is an input port other instruments

Target Header Pin-Out for the MICTOR Connector

No Connect*	1	2	No Connect*
No Connect	3	4	No Connect*
No Connect*	5	6	ATCLK
No Connect*	7	8	No Connect*
No Connect*	9	10	No Connect*
TDO	11	12	Vref
No Connect*	13	14	No Connect
TCK	15	16	ATD19
TMS	17	18	ATD18
TDI	19	20	ATD17
No Connect*	21	22	ATD16
ATD15	23	24	ATD7
ATD14	25	26	ATD6
ATD13	27	28	ATD5
ATD12	29	30	ATD4
ATD11	31	32	ATD3
ATD10	33	34	ATD2
ATD9	35	36	ATD1
ATD8	37	38	ATD0

*Pins 1, 2, 3, 4, 7-10, 13, and 21 *must* be true no-connects. Pins 1-4 are driven when a logic analyzer is connected to the target system through the header connector. Pins 7-10, 13, and 21 are driven by the Trace Port Analyzer.

For designs with less than 20 trace data pins, any unused ATD pins *must* be connected to ground.

Figure 3 - FPGA debug connector

use to signal the FPGA Trace Port Analyzer to complete its measurement. The combination of the Trig Out and Break In ports enables you to make various complex measurements, such as checking for signal integrity issues on data lines.

To check signal integrity, set up an oscilloscope on the suspect data lines. Next, add an ILA with ATC into your design on the data lines being probed by the oscilloscope. The FPGA Trace Port Analyzer's external port, Trig Out, is connected to the oscilloscope via a cable. The oscilloscope is then configured to stop its measurement when the port out signal from the FPGA is asserted. Now that you have completed this setup, you make your measurement. To do this, first start the oscilloscope. With the oscilloscope running, and the trigger on the ILA with ATC set to the bad data, you then begin the measurement on the FPGA Trace Port Analyzer. When the FPGA Trace Port Analyzer triggers, it will assert the Trig Out signal, which, in turn, signals the

oscilloscope to stop its measurement. Once stopped, you can inspect the oscilloscope waveform, going back in time to where the suspect data can be found. This measurement enables you to determine the root of a signal integrity issue.

Conclusion

The Agilent FPGA Trace Port Analyzer, combined with ChipScope Pro tools, is an affordable solution that enables effective in-system debugging. The combination of these two powerful tools gives FPGA designers internal node visibility during in-system debugging. It gives FPGA designers flexibility to take wide, shallow 32K state deep traces or narrow 2M state deep traces. It also provides remote debugging through the network capabilities of the Agilent FPGA Trace Port Analyzer. These features, along with the FPGA Trace Port Analyzer's ability to work with other Agilent instruments, make a powerful solution for debugging Xilinx FPGAs in-system.

Xilinx Redefines the FPGA... Once Again.

- *Highest density: 125,000 logic cells*
- *Highest performance: 400+ MHz clock rates*
- *Over 10 Mb on-chip RAM*
- *Over 200 IP cores*
- *Fastest and most productive ISE software tools*
- *TeraMAC DSP performance*
- *24 x 3.125 Gbps Rocket I/O transceivers*
- *2000+D-MIPS IBM PowerPC™ processing power*
- *Lowest system cost*

The Virtex-II Pro™ FPGAs provide the highest logic performance, density, and memory capacity in the industry. Plus there are up to four IBM PowerPC™ processors and up to 24 Rocket I/O™ transceivers included at no additional charge. Supported by the industry-leading ISE software and over 200 IP cores, Xilinx delivers more value than ever.

Unbeatable Leadership in Logic and Memory

Virtex-II Pro logic designers can take advantage of superior logic performance (400+ MHz clock rates). And with 125,000 logic cells, 10Mb embedded RAM, and 1.7 Mb distributed RAM, Virtex-II Pro devices provide the highest density and performance in the industry, period.

Highest System Performance

Virtex-II Pro FPGAs extend performance and integration into the system realm with TeraMAC DSP performance, over 2000 D-MIPS of PowerPC processing power, and up to 24 3.125 Gbps Rocket I/O serial transceivers. Our SelectI/O™ Ultra delivers 840 Mbps LVDS performance, all with the world's leading FPGA logic fabric.

Lowest System Cost

The Virtex-II Pro solution delivers the industry's lowest system cost by reducing your development and production costs. Our ISE tools speed you through design and debug, extracting the maximum performance and density out of the Virtex-II Pro architecture. Our system integration capabilities reduce your overall bill of materials and provide the lowest production cost. And with 300mm wafer technology and Virtex-II Pro EasyPath solutions for cost reduction, we ensure you'll always have a system cost advantage.

Industry-Leading Software Tools and IP Cores

Driving the Virtex-II Pro FPGA is Xilinx's ISE 5.1i software and over 200 IP cores. ISE 5.1i includes incremental design, a macro builder, our intuitive Architecture Wizard, and compile times up to 6x faster than our nearest competitor, making it the industry's fastest and most productive tool set.

Visit *www.xilinx.com/virtex2pro* today and get all the value of working with a leader.

XILINX®

The Programmable Logic Company℠

www.xilinx.com/virtex2pro

FORTUNE® 2002
100 BEST COMPANIES TO WORK FOR

Architectural Synthesis

Unleashing the Power of FPGA System-Level Design

Architectural synthesis shifts complex system design to a higher level.

by Don Davis
Manager, High Level Language Tools
Xilinx, Inc
don.davis@xilinx.com

When you combine the capabilities of powerful Virtex-II™ Pro FPGAs with the wide range of hardware cores now available (from soft processors such as MicroBlaze™ to bus interfaces such as PCI), you've got all you need to develop complete systems on a single device. However, with all of this capability comes added design complexity. How do you take advantage of these vast resources and deal effectively with the added system complexity?

FPGAs have evolved beyond glue logic into fundamental system elements. To remain relevant, development methodologies must respond to this changing role by providing the appropriate abstraction levels and tools needed to manage this complexity. You need:

- High-level languages to support the capture of complex design functionality in an abstract manner
- Profiling and characterization to explore solution space tradeoffs
- Debugging and verification tools to ensure design integrity
- Compilers and optimizers to produce high quality implementations.

A New Approach

A compelling design methodology based on Architectural Synthesis (AS) offers a comprehensive strategy for managing all of these issues. AS streamlines design, verification, and implementation of complex systems by leveraging powerful development tools and advanced FPGA devices. AS enables you to define system functionality at a high level of abstraction (using a software-based design entry point) and then debug, synthesize, and verify a range of architecture implementations that meet the system specification. What makes AS so exciting is that it is based on a single system specification, so you can easily explore a variety of hardware implementations to achieve the optimal cost/performance point for your application – and even change the hardware/software partitioning – without having to modify the source specification.

The System Design Challenge

Effective system design in the era of platform FPGAs requires a holistic approach. No longer is FPGA design simply about mapping your algorithm to LUTs. In the first place, today's design complexity has grown to such an extent that you need higher-level methods of algorithm specification and design capture. What's more, with embedded processors of both the hard and soft varieties, your implementation options are vastly expanded: Should I implement this piece of functionality in the FPGA fabric or on the embedded processor? What is the impact on the system with respect to performance? How should the various processing elements communicate?

Complicating matters still further, the answer to any one of these questions can affect the answer to the others. A local optimization, for example, may not be a system optimization. You must be able to explore these interactions quickly and cover the entire solution space with minimal effort if you're to have any hope of achieving an optimal solution.

By contrast, the current process is a lot like throwing darts blindfolded. Your designer intuition will usually get you facing the dartboard, but hitting the bulls-eye is mostly a matter of luck. You pick a hardware/software partition based on your experience and some limited modeling or profiling, and hand it off to the rest of the design team. Barring catastrophic circumstances, the partition is fixed at that point; it is simply too difficult to go back and rework all the hardware implementations and interfaces because they're all products of manual translations of the specification.

Synopsys Provides the Multiple Levels of Abstraction Necessary for AS

"Higher levels of abstraction are crucial for exploring system specifications, for reaching hardware/software architecture closure early in the design cycle, and for decreasing implementation cycles," reports Joachim Kunkel, vice president of IP and systems marketing, Synopsys.

Synopsys' CoCentric System Studio application, for example, gives you the multiple levels of abstraction needed to accomplish a range of tasks: Untimed Functional for data exchange; Timed Functional for computational and communicational delay; and Transaction Level, the natural meeting point for hardware and software designers to achieve cycle-true platform performance analysis.

SYNOPSYS®

These abstraction levels offer verification speeds that are higher than those offered by RTL, by orders of magnitude – and yet they give you sufficient detail to do platform analysis and come to closure on the hardware/software architecture early. (Although high-level abstraction offers an effective way to deal with today's increasingly complex designs, and definitely offers design-cycle time benefits, it does not replace the pin-accurate models necessary for an automated path to hardware implementation.)

Because this methodology consists of design, debug, verification, and implementation in hardware and software, the company calls it "SystemC Design and Verification." But the CoCentric solution also gives you the option to use RTL, giving you complete design control, especially in cases where the required hardware architecture is very well understood.

Using the appropriate level of abstraction and automation for your analysis and implementation – and combining that with a unified hardware/software methodology and the unique dual programmability of Platform FPGAs – enables you to create differentiated products cost-effectively.

Your team may never know if your choice was a good one, or whether this was an optimal partition. You just have to make it work, so you spend lots and lots of time optimizing and tweaking code (hardware or software) in a struggle to make sure your design meets the system specification.

This is a major weakness in the flow. On the software side, the translation from the software specification (usually written in C) into the C implementation for the embedded processor is straightforward. But translating the software specification into hardware implementations (usually Verilog, VHDL, or RTL hardware description languages) is another matter entirely.

Typically, you or your team interpret the specification and tediously convert it into a hardware implementation that (you hope) will meet the system specifications. Here again, this approach gives you another chance at "blindfolded darts." The level of resource sharing, number of pipeline stages, and amount of loop unrolling are just a few examples of the many decisions that are difficult to change at the RTL – and which you have to make up front.

All of these decisions affect the performance and area of the final implementation, and all of these issues offer a range of options for a given algorithm. Your ability to explore the solution space is severely limited if you have to re-code the HDL by hand because you know that each re-code takes time, both in terms of the design itself and the subsequent verification of the new implementation.

Architectural Synthesis to the Rescue

AS comprises a suite of technologies designed to meet the challenges associated with system-level design – and help you realize its benefits. (See the sidebar stories for an overview of the different ways our partners are incorporating AS into their design flows.) AS offers an

Using Forte's Cynthesizer and AS to Improve Process and Outcome

Architecture modeling and synthesis allows groups to produce better designs faster. Combining the power of AS with C++ design, verification, and software development marks a significant step forward in the design process.

With AS methodology and appropriate constraints and directives, you can create multiple RTL implementations from one C++ model in minutes, each implementation representing a unique tradeoff between performance, area, and power. Forte's Cynthesizer customers have found that designs created and verified in C++ typically yield a 20x to 30x reduction in lines of code, simulate faster by orders of magnitude, and reduce the design schedule by 50% or more.

Imagine your group is creating a cellular phone chipset. Among the design elements you'll want to consider are the tradeoffs between performance and die size, and between hardware and software implementations of a JPEG algorithm. To get an accurate hardware estimate, you'll first need to produce RTL code, and then apply RTL estimation tools or logic synthesis. Using traditional methods, that process would take your team several calendar months – 50 to 100 engineering months – to create one RTL implementation.

With AS, on the other hand, you can automatically create a range of RTL implementations from high-level C++. Armed with this data, you can trade off hardware and software implementations with confidence.

The exploration capability of AS, coupled with the sheer productivity gain in moving to behavioral C++, makes the AS/C++ combination the designer's power tool for the complex systems of tomorrow.

Savvy Design Teams Are Re-Evaluating Their Design Practices

Using programmable SOCs (system-on-chips) in combination with higher-level building blocks, design teams can now optimize an entire system's performance throughout the development process – eliminating the performance issues that cause costly delays. Mentor Graphics and Xilinx have teamed up to provide an advanced EDA (electronic design automation) and silicon solution, setting the stage for true platform-based design.

Multimillion-gate FPGAs with embedded processors and high-speed interfaces require architectural solutions tailored to specific design needs. Issues such as hardware/software partitioning and validation, board interconnect, and system-level verification can all lengthen your time to market. The key to efficient and effective design is to employ an integrated flow that brings together hardware, software, and board and layout engineers early in the design process.

Mentor's comprehensive, system-level FPGA design solution, including design creation and management, hardware/software co-verification, simulation, synthesis, and PCB (printed circuit board) analysis and layout, empowers the complete design team. All team members can take advantage of the advanced building blocks found in today's FPGAs and avoid costly delays. At Mentor Graphics we are committed to delivering complete and integrated solutions that support AS. Our goal is to help you eliminate performance issues and shorten your time to market.

impressive array of tools that address design, partitioning and optimization, debugging and verification, and reuse. Let's look at each one in turn.

Design

AS improves up-front design decisions because it's based on a high-level language specification. It's a lot easier to manage your design functionality when you don't have to worry about register and interface timing. AS works to capture the functionality and get it verified quickly, and then automatically compiles to an implementation that meets your system specifications. AS also makes it easier to trade off design constraints against performance goals. For example, if you run a compilation and the resulting implementation doesn't meet your performance criteria, you simply rerun the compiler and ask it to improve the performance by using more hardware resources – it's simple and painless.

Partitioning and Optimization

Improved partitioning and optimization are core benefits of an AS flow. AS enables you to define hardware/software partitions easily, push a button, and have the tools automatically generate the software-executable and hardware bitstream, as well as all of the routing required to enable the hardware and software components to communicate effectively. This includes the synthesis of buses and bus interfaces, as appropriate, as well as the software drivers necessary to support them.

The ability of AS to accomplish this automatically, with minimal user interaction, is key. The more you have to do by hand, the less you're going to iterate to find the optimal solution. The power of AS is that these automatic tools make it easy to explore the entire solution space quickly, enabling you to find the best solution for your application. To evaluate potential candidates in the solution space effectively, the tool must be able to profile the candidates with respect to such design considerations as throughput, memory usage, and FPGA area. You'll also appreciate being able to evaluate design bottlenecks. You can easily answer questions such as: "Is the system constrained because there is too much traffic over the system bus, or because the memory accesses are taking too long?"

On the hardware side of the partition, selection is even more complex. In other words, a typical software implementation most often involves a single performance point. If you want better performance, you need a faster processor. Code-tuning can provide improvements at the cost of memory, but overall the range of implementations is limited. An FPGA implementation, on the other hand, can involve a wide range of performance points based on varying the hardware architecture. More gates typically (but not always!) equals better performance. An example of a classic hardware tradeoff is adding pipeline stages to dramatically improve throughput at the expense of latency and area.

Here again, your ability to explore potential hardware architectures thoroughly is directly related to making the optimal design choices. AS enables you to do this automatically, simply by rerunning the compiler with different preferences. You don't have to rewrite specification code, which saves both design and verification time. Perhaps even more important, AS prevents you from introducing errors into this verified design.

Pipelining

It is in this area that high-level language-based methodologies truly shine. For example, changing the level of pipelining later in the traditional design flow is such a huge undertaking it's usually not even considered. If a design doesn't meet specifications, the entire design team typically spends significant time and energy trying to tweak the design to achieve the specification performance.

In an AS flow, you don't need to decide in the beginning about what level of pipelining is appropriate – you can make your decision at compile time. The resulting implementation is far more likely to converge on the optimal architecture for your system specifications.

Debug and Verification

The AS approach likewise expands your options for system debugging and verification. Traditional approaches to design verification rely exclusively on RTL simulation, which, while accurate, is unacceptably slow for large system designs. In addition, the interfaces to software code and ISS (Instruction Set Simulators) are clumsy and inelegant.

With AS, the system specification is automatically synthesized to implementations, so functional verification of the system specification is equivalent to functional verification of the implementation. By contrast, traditional methods of implementation involve a manual translation step, so verification of the system specification tells you little about the functional correctness of the ultimate implementation.

In addition, an AS debugging and verification toolset provides multiple layers of fidelity. The first level is the software paradigm. Here you can use traditional software debugging techniques, setting breakpoints and stepping to code, to chase down bugs. The next level creates a co-simulation environment, integrating simulation tools and ISS to delve into the details of how the hardware interacts with the software. Finally, you can even implement your design on a target FPGA and use tools, such as ChipScope™, to explore the details of the actual implementation running on real hardware.

Another key benefit of the AS approach is that it enables you to work at a level of abstraction appropriate for the level of design you're working on. For example, working at the clock cycle level, when your goal is simply to verify that the algorithm functions correctly, provides too much detail and will actually get in the way of understanding the algorithm's performance. Moreover, working at higher levels of abstraction is typically many times faster than working at the lower levels, enabling faster iteration time in the code-compile-debug cycle. Of course, for those times when you need to figure out how to remove "just one more" clock cycle to meet your throughput specification, working at the cycle-accurate simulation level, even though it is slower, will give you the detail you need.

Reuse

AS also facilitates efficient design reuse. A powerful tool in any designer's toolbox is the large set of IP you can use for standard system functions. Xilinx and its partners provide a wide range of IP products. These products plug directly into any design flow. However, designs that are functionally validated and implemented at higher levels of abstraction are more suitable for an AS flow because they can be reused in many future products – even where design constraints and performance goals are quite different. Because, with AS, a single source specification can provide multiple implementations simply by rerunning the compiler with different constraints, a single piece of IP can provide a variety of implementations addressing high throughput, small area, or some optimal combination of the two. Furthermore, the optimal combination can be determined expressly within the context of your specific design.

In addition, the AS flow facilitates development and verification of IP because the IP designer can work at the functional level. And because a single specification can be targeted to a wider range of implementations, IP developed using AS is likely to find broader application.

Conclusion

Architectural synthesis is both a powerful new tool in your quest for the optimal system design solution, and a mighty weapon in the fight against design complexity. AS requires a new way of thinking about systems – not as separate hardware and software domains – but as an integrated whole, the boundary of which is extremely fluid. FPGAs, with the flexibility of embedded processors and the ability to transmit data at Gigabit rates, provide the power to drive new classes of systems. Architectural synthesis provides a way to harness that power and develop your designs in record time.

Spartan®-IIE with LVDS. It's the perfect fit.

The new Spartan®-IIE FPGA is the first in its class to provide advanced LVDS capabilities for applications like digital video, DVD players, LCDs, plasma displays, and scanners. Also, lightning-fast DSP algorithms enable more efficient designs for products such as cable modems, satellite dishes, and HDTV.

The Gateway to Consumer Digital Convergence

Combining audio, video, and data capabilities in one product is the challenge. With Spartan-IIE FPGAs, you'll get all the system-level integration of an ASIC, plus the time-to-market and reprogrammability benefits of an FPGA—all in a cost-optimized solution ranging from 50,000 to 300,000 system gates.

Industry-Leading Software . . . the Widest Range of IP

As with all Xilinx devices, the new Spartan-IIE family is fully supported by our ISE 4.1i software (compiling 100,000 gates per minute!), including ISE WebPACK™, free via the Internet. Designers also have access to the industry's widest range of IP cores, reference designs and third party support.

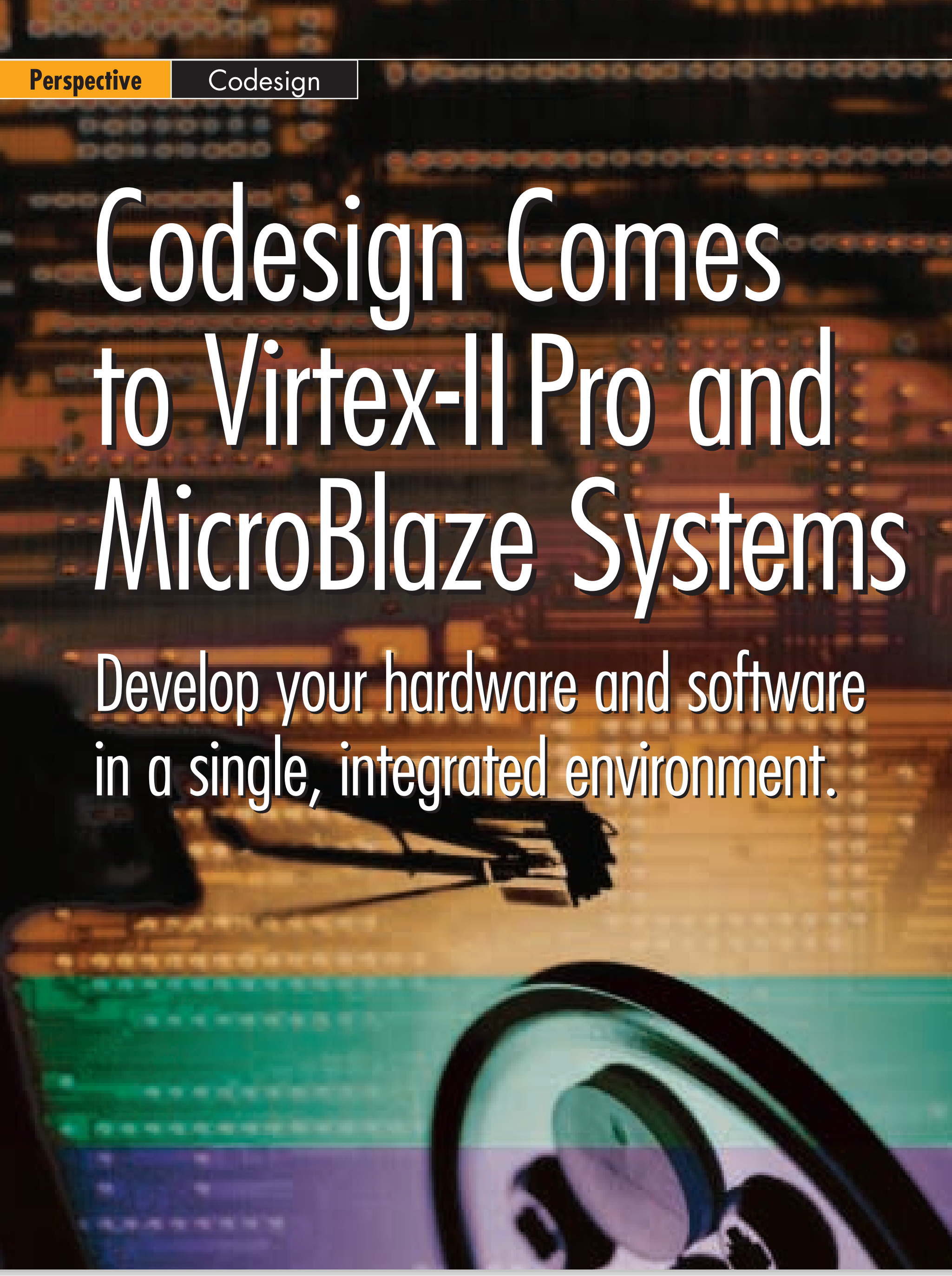
SPARTAN-IIE FPGA

Visit ***www.xilinx.com/spartan2e1*** today to see how Xilinx is changing the world of consumer digital convergence.

XILINX®
The Programmable Logic Company℠

www.xilinx.com/spartan2e1

FORTUNE® 2001
100 BEST COMPANIES TO WORK FOR



Codesign Comes to Virtex-II Pro and MicroBlaze Systems

Develop your hardware and software in a single, integrated environment.

by Chris Sullivan
Director of Strategic Alliances
Celoxica
chris.sullivan@celoxica.com

Virtex-II Pro™ FPGAs are powerful system-level devices, replacing microprocessors and ASICs in many new applications. This shift in design strategies necessitates a corresponding shift in the way programmable logic designs are created and deployed in electronic products. To efficiently manage your software and hardware design in these programmable systems, you must now move away from legacy ASIC design methods to a codesign methodology that gives you greater choice in the level of design abstraction.

Codesign

Codesign is a process in which you use similar methods, and sets of connected tools and languages, for both hardware and software design. Codesign helps shorten development time by enabling the concurrent development of hardware and software, and by allowing software to be developed on "virtual hardware platforms" before the final hardware is ready. In addition, a top-down approach enhances your ability to analyze and tackle system partitioning and verification by enabling you to explore the design space fully. This enables more informed consideration of hardware/software trade-offs and leads to better Quality of Design (QoD). Reducing the risks that arise from incorrect or changing specifications can help avoid the time-consuming and expensive optimization of an incorrect partition (which leads inevitably to a sub-optimal design) and increases your chances of first-time success.

Programmable Systems Require a Codesign Methodology

Historically, FPGA hardware was designed using techniques and languages borrowed from ASIC design methods – methods that are very different from those used to develop software or embedded systems. Up to now, there was a huge difference between these disciplines and their methodologies.

For example, current methods for embedded systems design require that hardware and software be specified and designed separately. Typically, C/C++ or a block-based methodology is used for the system specification. Once behavior has been fixed, the specification is then delegated to the (separate) hardware and software engineering teams, which code in different languages: HDLs (Hardware Description Languages) for the hardware, C/C++ for the software. While the system partition can be informed by profiling the specification or legacy software code, the partitioning is often decided in advance. And, because changes to the partition can necessitate extensive redesign elsewhere in the system (interfaces between the hardware and software, for example), that decision is adhered to as much as possible. The deficiencies of this methodology are clear:

- Lack of a unified hardware-software representation can lead to difficulties in verification of the entire system, and hence to incompatibilities across the hardware/software boundary.
- Defining a system partition in advance can lead to sub-optimal designs; incorrect partitioning requires costly refinement and is detrimental to QoD.
- Hardware partitions of the system specification or legacy software code require time-consuming (and sometimes error-prone) rewriting into HDL.

- Lack of a well-defined and flexible codesign methodology makes specification revision difficult and affects time to market.

While it is not yet possible to synthesize *efficient* hardware and software from a single language description, a codesign methodology that supports partitioning and co-verification, multiple languages, and tool interoperability is nevertheless invaluable when designing high-performance systems using Virtex II Pro FPGAs and MicroBlaze™ processors. Such a methodology makes it possible to:

- Prototype the system more easily and explore the design space better to identify the optimal design solution.
- Use generic hardware/software interfaces for system co-simulation and verification, using the software code as a testbench throughout.
- Implement changes to partitioning decisions – if required – much later in the design cycle.
- Target different hardware platforms more easily and even change the target platform later in the design cycle than would otherwise be possible.
- Drive system implementation from correct levels of abstraction.

The benefits of fusing separate design approaches into an effective and more "integrated software-compiled system design" flow that uses top-down design to tackle system partition, verification, and implementation are significant.

Working together, Celoxica and its strategic partners such as Wind River and Xilinx have developed a unique codesign flow and methodology (Figure 1) for Virtex-II Pro systems using MicroBlaze processors.

Software-Compiled System Design for Programmable Systems

Fundamental principles of the codesign methodology are:

- A top-down, idea-to-implementation flow
- A common higher-level language base for hardware and software design
- The distinction of processing fabric at correct levels of abstraction
- Interoperability with best-in-class hardware and embedded software tools
- Codesign API standards (for example, the DSM – Data Streaming Manager), which enable easy interfacing between software and hardware for partitioning, verification, and implementation.

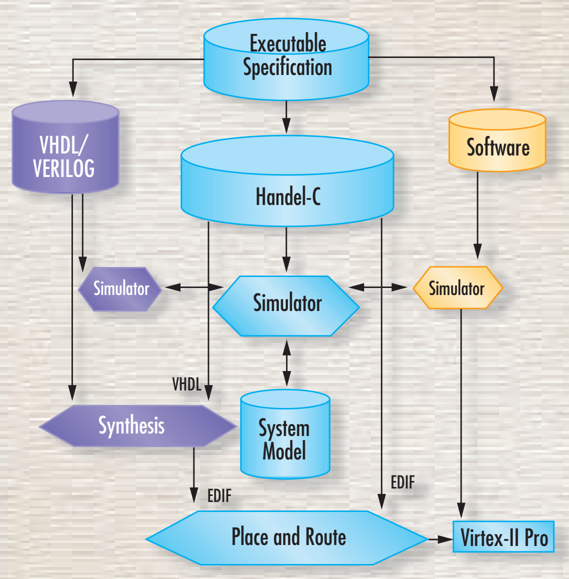


Figure 1 - Codesign flow for programmable systems with the flexibility for mixed language description interoperability

To make software-compiled system design possible, you need an environment that brings together the efficiencies of higher-level languages and the capabilities of powerful partition, verification, and design implementation.

DK Design Suite

The DK Design Suite enables you to enter system descriptions in higher-level programming languages, and to simulate and debug that code using a familiar, friendly integrated development environment (IDE). Block-based design and multiple languages are supported for simulation including C, C++, SystemC, HDLs, and Handel-C.

The package includes the Nexus-PDK co-verification environment, which also makes it possible to drive the entire functional verification process for the system with higher-level code.

Nexus PDK

Nexus-PDK is a powerful co-verification tool that allows you to simulate system functionality in multiple higher-level languages, and to continue to use these models through to design implementation by supporting co-simulation of software and hardware. Nexus communicates directly during simulation with popular third-party hardware RTL simulators and software ISS environments.

Handel-C

Handel-C, which is based on ANSI-C, has an added set of simple extensions for hardware development. These include:

- Flexible data widths
- Parallel processing
- Communication between parallel threads.

In addition, Handel-C uses a simple timing model that enables you to control pipelining without adding definitions for specific hardware. Handel-C also eliminates the need to code finite state machines exhaustively by providing the ability to describe serial and parallel execution flows.

Its familiar language has formal semantics for describing system functionality and complex algorithms that produce substantially shorter and more readable code than RTL-based representations. The level of design abstraction is above RTL (Register Transfer Level) but below the behavioral level, and everything that can be described in the language may be translated to hardware.

DSM

DSM (Figure 2) is a portable hardware-software codesign API that offers a simple and transparent interface for transferring multiple independent streams of data between hardware and software. DSM is independent of both bus/interconnects and operating systems. It consists of two parts: an OS-independent API for the FPGA application, and an API for ANSI-C or the software environment. In operation, each side opens a number of uni-directional ports; a "write to a port" on one side is then matched by a "read" on the other. In this way, multiple software applications can independently access multiple reconfigurable hardware resources using very few API calls.

In Figure 3 you can see how these solutions integrate with best-in-class embedded software tools from Wind River and Xilinx programmable systems to deliver a comprehensive software-compiled system design methodology.

The key elements of the methodology are:

- A minimal tool chain – comprising the Celoxica DK design suite, Wind River's XE (Xilinx Edition) embedded software tools, and Place and Route from Xilinx.
- A common language base – C and Handel-C, with the flexibility for interoperability with mixed language descriptions, such as HDLs and SystemC.
- API standards for common interfacing and platform abstraction – Celoxica PAL for platform abstraction, and Celoxica DSM for hardware/software integration.

Profiling and Partitioning

Profiling and partitioning are key to any codesign methodology and help identify optimal design methods early in the design cycle. In the software world, the profiler is mostly used as an analysis tool to examine the runtime behavior of a program. Profiler information helps you determine which sections of code are working efficiently and which are not. Profiling also gives you information about where the program spent its time, and which functions called which other functions while it was executing. In this way, profiling shows which pieces of the program are slower than expected and thus might be candidates for off-loading into hardware for coprocessor acceleration. It can also highlight which functions are being called more – or less often – than expected.

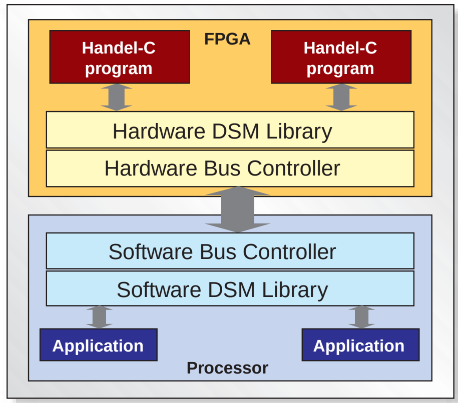


Figure 2 - DSM system overview

But profiling tools were developed to fine-tune software – making applications run better and identifying candidates for rewriting – not for system partitioning. Although profiling code is an extremely useful exercise for informing partitioning decisions, it should not be relied upon exclusively. For example, due to latency between the system boundary and interfaces, it makes sense to minimize dataflow between the hardware and software. And yet, software profiling tools do not explore dataflow over the hardware/software boundary. You can, however, deduce this dataflow through designer scrutiny of the code and by hardware/software coverification using API calls for run-time test.

To see how software-compiled system design can best be deployed for Virtex-II Pro FPGAs and MicroBlaze processors, let's use a simple design example within the context of codesign.

Codesign Methodology Design Example

In this example, we have a system that contains a GUI, an image compression engine, an encryption engine, and a control path through which we issue commands to the image compressor (Figure 4).

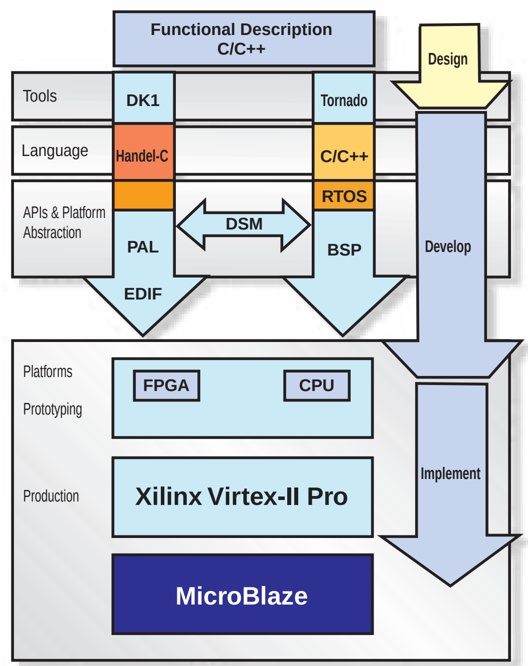


Figure 3 - Example HLL tool-chain

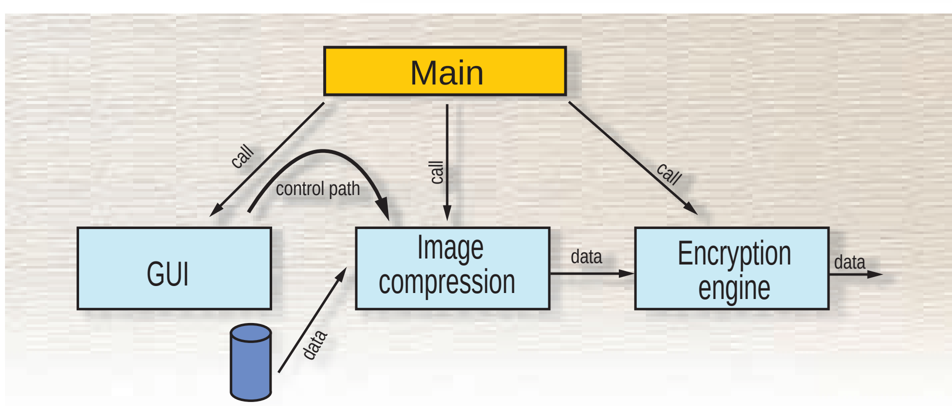


Figure 4 - Simple codesign example

1. First, we examine the system functionality against the project requirements, identify obvious system partitions, and also identify functions that will require further design investigation (such as those functions for which the optimum design partition is not immediately apparent).

   The GUI is an obvious candidate for software implementation; it is sequential and does not require processor-intensive resources. Likewise, the encryption engine is also a candidate for hardware implementation; it is parallel and integer-based. The partitioning of the compression function, however, is less clear and is targeted for profiling, iterative partition, and design exploration.

2. We move the compression function into software and obtain benchmarking information to provide a baseline for partition assessment. The software code can be used as a test bench throughout to support verification.

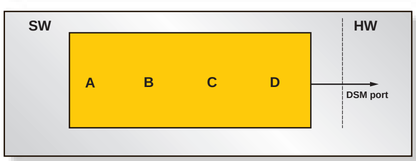


Figure 5 - DSM API port for hardware interface

3. With the function still in software, we use the DSM API to interface to the hardware component (Figures 5 and 6). We then begin to port blocks of the software to Handel-C for hardware prototyping, testing, and verifying at each stage. This process is relatively simple, because there is a common language base and, most importantly, a common level of abstraction for the software and the hardware. We also move the DSM port to enable the new partition to continue testing and verification at each stage (Figure 7).

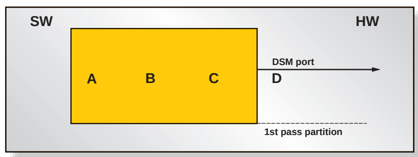


Figure 7 - DSM API port moved for new partition

4. Having completed the partition and debugging, we cosimulate to verify the effectiveness and efficiency of the partition, as measured against system constraints and design requirements.

5. We now enter what is effectively the partitioning cycle, in which we begin to reiterate and explore different partitions and design scenarios through testing and verification, using the simple procedure outlined in steps 3 and 4. This is an innovative process-driven approach to partitioning.

6. The partitioning cycle produces a number of partition alternatives. We now consider these alternatives, map them to our design requirements or system constraints (such as device size, target platform, bandwidth, and so on), and select the optimum partition for QoD.

7. We simulate and verify the partitioned system, using compiled C/C++ combined with the Handel-C compiled for the Nexus PDK simulator. For speed and efficiency, the cosimulation uses DSM Sim and PAL (Platform Abstraction Layer) Sim as virtual interconnects and virtual peripherals, respectively.

8. The system is cosimulated and verified at a cycle-accurate level, using Nexus PDK, combined either with an ISS (Instruction Set Simulator) or ModelSim running a Swift model of the target processor.

9. We recompile the system for the target platform and implement the design. The target platform is supported by DSM and by a PAL layer that provides a portable API for accessing on-board peripherals, such as RAM, video, and generic data I/O. Thus, the application written using PAL and DSM APIs can be ported to new platforms simply by recompiling. This supports design reuse and application portability, and helps address the issue of design obsolescence.

Hardware

```
// buffer to receive compressed data
ram DsmWord Buffer[256];

static unsigned DataCounter=0;

while(1) // loop forever
{
   do
   {
      // get output from SW
compression
      DsmRead (PortS2H, &Value);
      par
      {
         Buffer[DataCounter] = Value;
         DataCounter++;
      }

   }while(DataCounter!=0);


   // now encrypt the block of
data....
   EncryptData(Buffer);
}
```

Software

```
// buffer storing raw image
unsigned Image[1600][1200];

// buffer for compressed data (FIFO)
DsmWord CompData [256];

unsigned DataCounter,Count,ImageDone

do
{
// compress part of image (256 bytes output)
   CompressBlock(Image,CompData, ImageDone;

   DsmWrite (PortS2H, CompData, 256, &Count);

   if (Count!=256)
      printf("\n Error writing to HW");

}while(ImageDone==0);
   // loop till the end of the image
```

Figure 6 - Sample code showing DSM calls

Conclusion

According to Gary Smith, Dataquest's chief electronic design automation analyst, "Today the biggest challenge in EDA is to resolve the incompatibility of the hardware design methodology and the software design methodology." Software-compiled system design delivers an advanced methodology that offers significant advantages to hardware engineers, embedded software engineers, firmware engineers, and systems architects.

For more information see *www.celoxica.com* or contact *chris.sullivan@celoxica.com.*

[1] P. Garrault, Synthesis Tool Enhancements for Virtex Architectures, Xilinx, 2002.

[2] Hardware/Software Co-Design Group, Polis A Framework for hardware-software co-design of embedded systems, EECS, University of California, Berkeley.

Mentor Graphics Offers Seamless Integration for Virtex-II Pro Developers

Working with Xilinx, Mentor Graphics has enhanced its Seamless hardware/software co-verification solution specifically for developers using the Virtex-II Pro family of FPGAs with embedded IBM PowerPC 405 processor cores.

by Robert Kaye
Market Development Manager, SoC Verification Division
Mentor Graphics Corp.
robert_kaye@mentor.com

Perhaps you've been slaving in the lab for two weeks now, and you still can't get your first prototype running. More complex than previous projects, you're dealing with a new processor, a lot more code, and a substantial increase in logic content. Verifying the software on the target hardware requires that the board design be complete and that boards are fabricated and available. Errors uncovered in hardware on "finished" boards carry the risk of having to schedule a board respin, or compromising the software design to work around problems in the hardware.

What's the answer? A virtual system prototype – a prototype in which system designers can integrate their embedded software and hardware before committing to silicon. Working together, Xilinx and Mentor Graphics have developed a custom Seamless® hardware/software co-verification solution targeted specifically for use with the Virtex-II Pro™ family of FPGAs with embedded IBM® PowerPC™ 405 processor cores.

With both hardware and software readily changeable in a virtual prototype, you can perform comprehensive validation and analysis in a safe environment. And because a virtual system prototype incorporates both a logic simulator and debugging environment for the processors, it's possible to get full simultaneous control and visibility into the logic and internals of the processor.

Furthermore, many design problems exposed during system integration are attributable not to software or hardware but to the complex interaction between the two. Thus, you can gain substantial benefits from validating the design at the system level. Exercising the boot ROM code, hardware diagnostics, device drivers, and the RTOS (real-time operating system) will expose most hardware/software interface errors – eliminating hardware prototype iterations and significantly reducing system integration time.

Seamless Hardware/Software Co-Verification

The Seamless co-verification tool combines logic simulation environments used by hardware designers with debugging environments used by software engineers. Seamless co-verification controls the flow of data between the environments and synchronizes the simulations. The patented Seamless Coherent Memory Server allows you to switch dynamically between detailed hardware verification and high-speed software execution without requiring any changes to the design setup, or even halting the simulation.

A single processor system is illustrated in Figure 1. However, the Seamless solution also works in multi-processor environments. For example, the Virtex-II Pro FPGA may be used on a board (in this case, the system would be composed of one or more boards) with one or more standard CPUs, DSPs, or processors embedded into ASICs. There are currently more than 110 Seamless Processor Support Packages (PSPs) available for a comprehensive range of CPUs and DSPs.

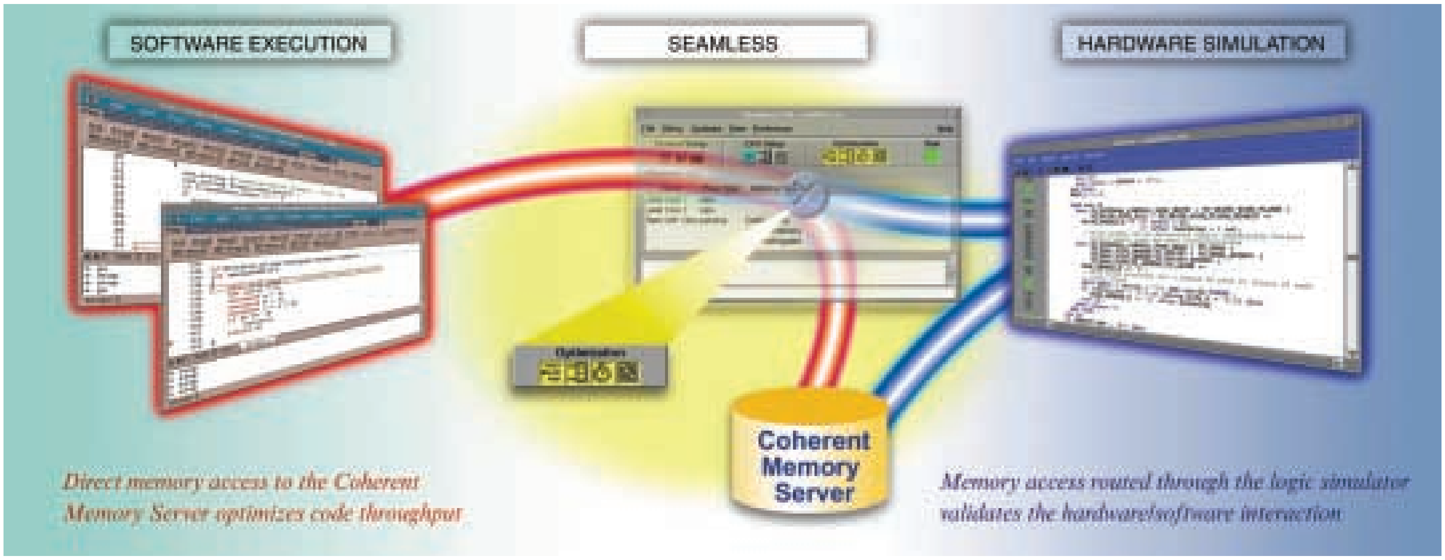


Figure 1 - The Seamless solution links hardware and software verification environments efficiently.

A Custom Solution for Virtex-II Pro FPGAs

The Seamless co-verification tool is a very flexible solution supporting a wide range of design styles, processor types, and memory system architectures. One caveat, however, is that system designers should make no assumptions about the processor used, the memories used, or how the processor interfaces to the rest of the design. Thus, there's a configuration process before you can start using the solution.

Although we cannot entirely eliminate this configuration process for Virtex-II Pro designers, we have made it simpler in several ways:

- We know that the PowerPC 405 core is the embedded processor.
- In Virtex-II Pro FPGAs, the PowerPC 405 core is interfaced to the logic fabric through a fixed block known as the "gasket." By expanding the boundary of the existing cycle-accurate Seamless PowerPC 405 model to include gasket logic, we simplify the task of bringing the Seamless tool into the design flow and raise the performance of the co-verification environment (Figure 2).
- Virtex-II Pro devices incorporate memory blocks known as BRAM (buffer random access memory). To use the Seamless optimization feature, memory models must be Seamless-aware. The solution for Virtex-II Pro FPGAs includes Seamless-ready models for Xilinx BRAM blocks (Figure 2).
- Virtex-II Pro design kits contain several reference designs. We at Mentor Graphics have created and verified Seamless-ready netlists of three of these designs (including creation of the relevant memory maps) to provide you with a jumping-off point for incorporating the Seamless solution into your particular design.

The Seamless solution works with the complete range of logic simulation solutions supported by Xilinx design kits, and has been verified with the software tool chains recommended and supported by Xilinx.

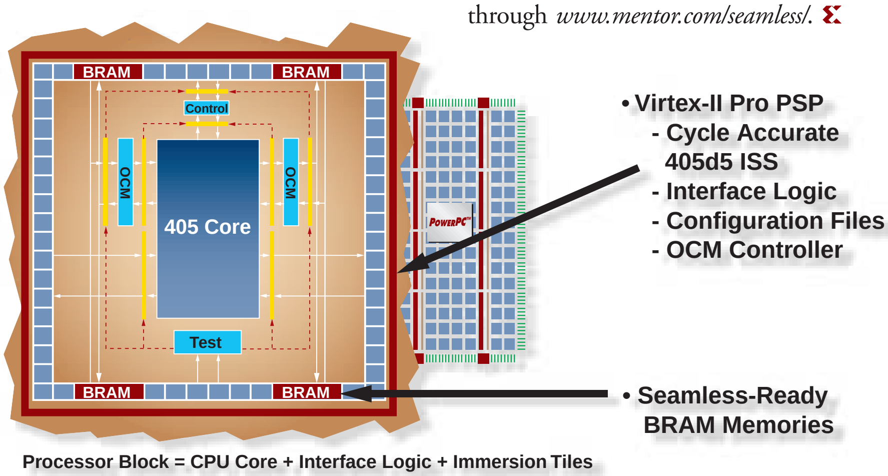


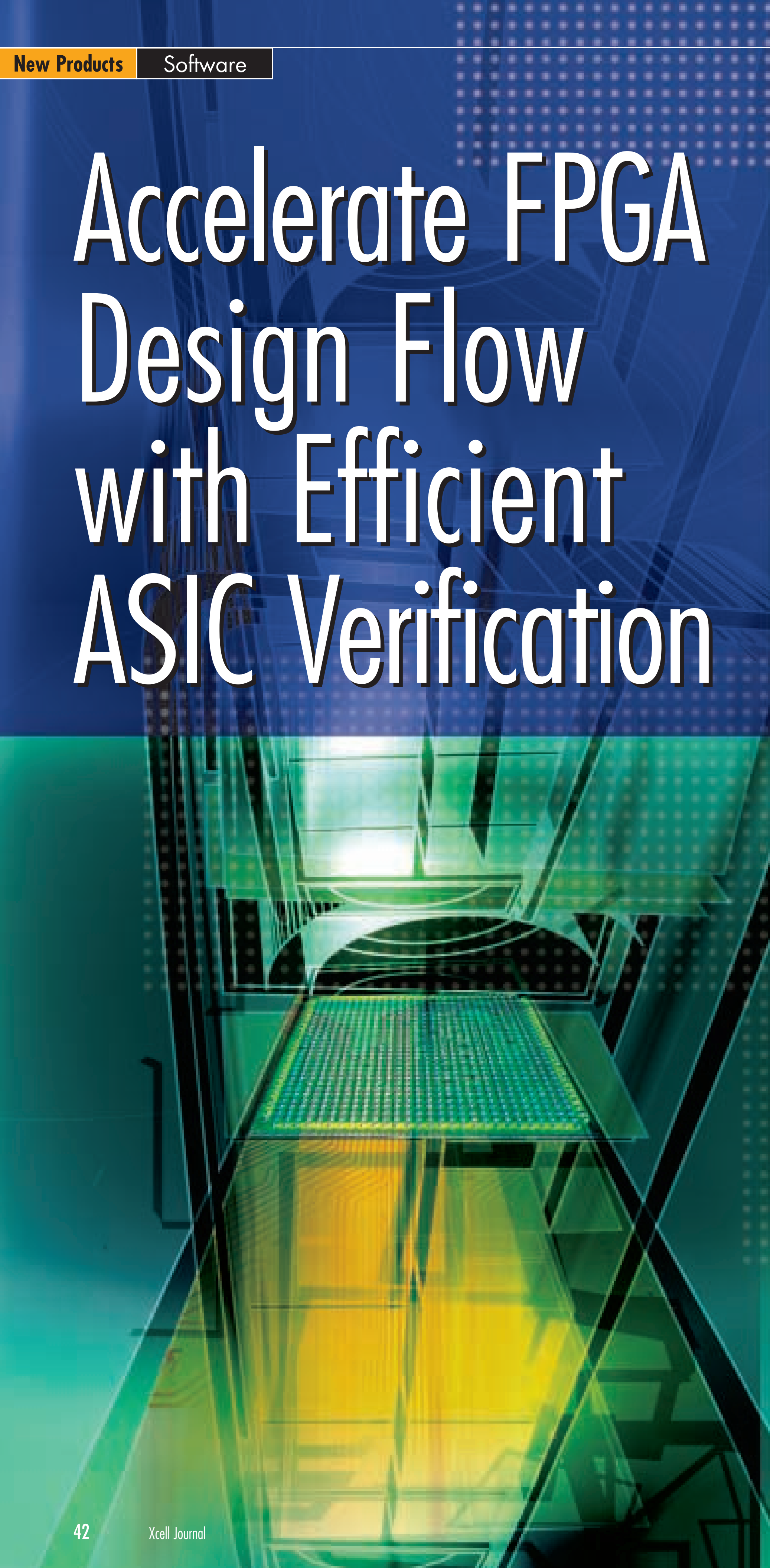
Figure 2 - The Seamless solution for Virtex-II Pro FPGAs includes a custom PowerPC 405 model and Seamless-ready BRAM memory models.

Conclusion

The Seamless design tool is an industry proven solution for co-verification of hardware and software across a wide range of embedded system applications and design styles.

The customized Seamless package for Virtex-II Pro FPGAs was presented at a live Web seminar co-hosted by Xilinx and Mentor Graphics on October 23, 2002. The seminar demonstrated and described how Seamless hardware/software co-verification specifically worked with Virtex-II Pro FPGAs. An archive of this Web seminar may be viewed at *www.mentor.com/seamless/seminars/xilinx/*.

Additionally, application notes and technical papers on how the Seamless solution has been applied in different environments are available through *www.mentor.com/seamless/*.



Accelerate FPGA Design Flow with Efficient ASIC Verification

The integration of Mentor Graphics' SpeedGate DSV tool with Sun Microsystems' Sun ONE Grid Engine software enables dramatically reduced synthesis run times.

by Al Benavides
Software Development Engineer
Mentor Graphics
al_benavides@mentor.com

Synthesis is one of many compute-intensive operations that can heavily affect the overall design process. To distribute these operations in parallel and accelerate the chip design flow, Mentor Graphics® SpeedGate DSV™ (Direct System Verification) software tool now has integrated support for Sun Microsystems' Sun™ ONE Grid Engine software.

SpeedGate DSV – also known as prototyping, rapid prototyping, or open system emulation – is an advanced methodology for verifying application-specific integrated circuit (ASIC) and system-on-chip (SoC) prototypes using off-the-shelf FPGAs in one or more custom or predefined printed circuit boards (PCBs). As a comprehensive and extensible solution for all aspects of prototype design flow, the SpeedGate DSV product includes partitioning and debugging support, with links to board creation and analysis tools. Currently, the SpeedGate DSV tool supports the Xilinx Virtex™, Virtex-E and Virtex-II families of FPGAs.

The SpeedGate DSV tool consists of an interactive design cockpit that launches partitioning and synthesis tools. It features a completely scriptable interface that plugs into any ASIC design environment – working hand-in-hand with emulation and gate-level simulation. The tool also includes patent-pending advanced partitioning technology that enables designers to maximize FPGA design prototypes. Furthermore, it fully supports the prototyping process within a team design environment, including sophisticated check-in/check-out features that track source code changes and manage version control.

Sun ONE Grid Engine by Sun Microsystems is a full-featured distributed resource management tool that controls very large numbers or groups of compute jobs. Compute jobs are submitted to the "Master Host," which matches job requirements to available resources for maximum throughput of the entire workload. This results in a nearly full utilization of all compute resources (systems, tool licenses, and so on). and an overall reduction in time-to-market, because engineers can focus on other design tasks while their jobs are queued and run automatically.

The SpeedGate DSV Flow Advantage

Today, ASIC verification consumes 30 to 70 percent of total ASIC design time. With costs for a 0.18-micron ASIC mask set exceeding $500,000, the financial impact of a silicon re-spin is substantial, if not prohibitive. Thus, budgetary and time-to-market pressures require a solution that reduces the verification effort, maintains a high level of accuracy, and delivers the product at or under budget.

The objective of Mentor Graphics' SpeedGate DSV tool is to convert an ASIC or SoC design to a functionally equivalent hardware prototype that can operate at a speed comparable to testing within the actual operating environment. Many observers consider such hardware-assisted verification the only emerging technology that can realistically impact the verification bottleneck. The availability of high-capacity, high-performance Xilinx FPGAs makes the SpeedGate DSV methodology a valid approach. Combined with other hardware such as bonded-out cores or memory, Xilinx FPGAs are interconnected on a PCB with other specialized hardware to duplicate the functionality of the ASIC or SoC at orders of magnitude faster than virtual simulations.

Multiple Distributed Processing of Compute-Intensive Operations

Unlike other prototyping development systems, the SpeedGate DSV flow allows design modules and processes to be synthesized independently of each other, in any order (Figure 1), rather than synthesizing the complete design as a single job. Consequently, this methodology allows synthesis jobs to be simultaneously spread among several workstations and/or servers, as well as licenses.

To utilize these distributed processing capabilities, the SpeedGate DSV software tool includes integrated support for Sun ONE Grid Engine software, which allows compute-intensive operations such as synthesis to be distributed across clusters of workstations and servers. This offers a number of additional benefits, such as job history tracking, simple resubmission of jobs, and job progress monitoring. Additionally, the SpeedGate DSV product's interface to Sun ONE Grid Engine software is easily expandable to other compute-intensive tasks, such as place and route. As a result, productivity is greatly increased, because the total time spent on compute-intensive tasks is significantly reduced.

Integrated Support, Job Grouping, and Automatic Prioritizing

The SpeedGate DSV tool interfaces to Sun ONE Grid Engine software via a Perl script named submit2grid. Developed by the SpeedGate DSV developers, submit2grid is included with all releases of SpeedGate DSV software, and can run from the command line or from the SpeedGate DSV GUI.

The submit2grid script takes an input file with executable commands (such as the .scr file exported for synthesis by the SpeedGate

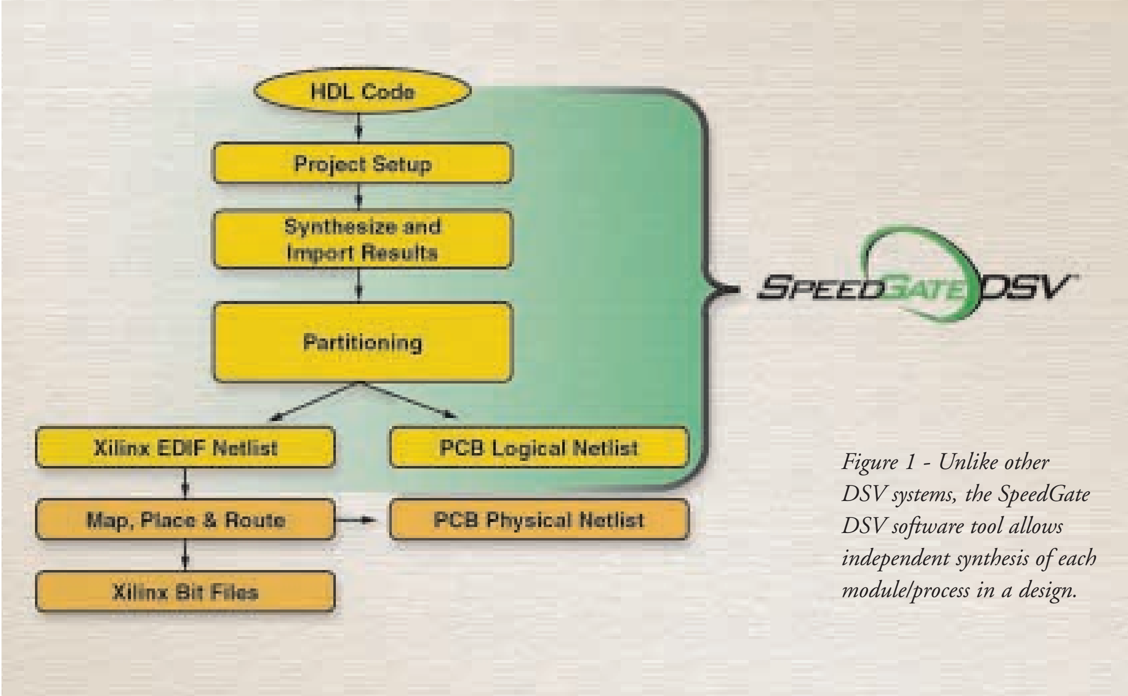


Figure 1 - Unlike other DSV systems, the SpeedGate DSV software tool allows independent synthesis of each module/process in a design.

METHOD	NUMBER OF JOBS	COMPLETION TIME (minutes:seconds)
One Sun Fire 280R Server (2x750MHz, 4GB RAM)	Submitted as one single top-level job	64:00
One Sun Blade 1000 Workstation (2x750MHz, 1GB RAM)	352 jobs serially run on one workstation	48:53
submit2grid (no grouping)	352 individual Sun ONE Grid Engine jobs	21:27
submit2grid -group 8	44 grouped Sun ONE Grid Engine jobs (eight jobs/group)	07:22
submit2grid -gen_opt 20 -group 9	40 Sun ONE Grid Engine jobs: three standalone jobs submitted first 37 grouped jobs (eight jobs/group)	04:26

Table 1: Benchmark Results/Performance Improvements

DSV tool) and then distributes the commands within that file among the execution hosts defined in a grid network using the Sun ONE Grid Engine software qsub command. submit2grid supports many arguments, including any valid qsub options, which allow for different submission conditions. In addition, submit2grid creates a log file that records such job submission details as submission start time, the name of the temp directory in which command files are stored, output log filename, ID number, and completion time.

A powerful feature of the submit2grid script is that it allows several jobs to be grouped into a single job. This feature, invoked with the -group flag, is particularly useful when a design contains many modules that synthesize individually very quickly (less than three seconds each). Submitting these fast-running jobs in groups improves the turnaround time for a design's complete synthesis because the number of submitted jobs is reduced; therefore, the job setup procedures for Sun ONE Grid Engine software are not constantly repeated over a short period of time.

SpeedGate DSV's submit2grid has a related flag, -gen_opt, which provides an advanced level of job grouping control. The -gen_opt flag creates a groups_options file that lists all jobs and specifies whether they are to be submitted individually or as part of a group. This groups_options file can be automatically generated or user-created and modified. When automatically generated, a job's previous run time history is compared to a user-defined time threshold to determine whether it should be grouped or not. If a job's previous run time exceeds this threshold, it will be individually submitted before grouped jobs. Prohibiting long-running jobs from being included in groups results in significant performance improvements for the synthesis turnaround time of a complete design.

Benchmark Results/Performance Improvements

For benchmarking purposes, the modules from Sun Microsystems' picoJava™ CPU design were synthesized in a relatively small Sun ONE Grid Engine cluster grid. The cluster grid comprised seven Sun Workstations configured to run a total of ten Sun ONE Grid Engine job slots with running ten simultaneous tool licenses.

Those results were then compared to results obtained by synthesizing the same design without distributed processing. Results are dependent on many factors and will most likely vary from run to run. Some of the basic factors that can influence the execution time of a job include compute grid design and set up; the number of CPUs allocated to the compute grid; the amount of physical memory available to the CPUs; the loading of grid resources at execution time; and the number of available tool licenses.

If SpeedGate DSV's flow did not allow for individual module synthesis, the entire design would have to be synthesized as a single job. The first row of Table 1 records how long a single job would take to synthesize with our benchmark design. The second row shows a small improvement when running individual module synthesis serially on one workstation, without taking advantage of distributed processing. The third and fourth rows illustrate the difference when taking advantage of the SpeedGate DSV software tool's distributed processing capabilities, while the fifth row shows the performance improvement obtained when using the submit2grid script with all optimizing options.

Using the SpeedGate DSV tool's distributed processing capabilities results in dramatic performance improvements. With the -group and -gen_opt options enabled, run times were almost 15 times faster than running the synthesis as a single top-level job, and more than 10 times faster than running serially on a single workstation.

Conclusion

The SpeedGate DSV tool's flexible synthesis methodology, which allows a design's modules to be synthesized separately and independently, results in great performance improvements when coupled with Sun ONE Grid Engine software. Faster module synthesis leads to an increase in productivity, a decrease in verification costs, and ultimately faster time-to-market because the prototyping flow is accelerated through distributed processing and automation. Combining this interface with the capacity, performance, and flexibility of the latest generation of Xilinx FPGAs creates a powerful environment for a prototyping flow.

For more Information, please visit *www.mentor.com/speedgatedsv/* or *www.sun.com/software/gridware/*.

Increase Your Design Confidence with Formality Equivalence Checker

Learn how to use Synopsys Formality Equivalence Checker with Xilinx ISE tools to verify complicated designs for Xilinx Platform FPGAs.

by George Mekhtarian
Technical Marketing Manager
Synopsys, Inc.
georgem@synopsys.com

Today's large, complex Platform FPGAs, such as the Xilinx Virtex™-II and Virtex-II Pro™ series, can exceed 10 million system gates and operate at speeds of 300 MHz or more. SoC (system-on-chip) designs targeting Xilinx Platform FPGAs are now subject to the same functional verification delays as large ASIC designs. Just as with ASICs, you must now employ a type of static verification technology known as equivalence checking (EC) to verify FPGA design logic and functionality.

Using the Formality® equivalence checker from Synopsys in a Xilinx Platform FPGA design flow allows you to verify equivalence quickly between RTL (Register Transfer Language) and the synthesized gate-level netlist – and between RTL and a post-Xilinx place-and-route (PAR) netlist as well. Formality EC increases confidence in functional integrity during design implementation, giving you the freedom to focus on debugging actual design problems.

How Equivalence Checking Works

EC is a branch of static verification that employs formal mathematical techniques to prove that two versions of a design are functionally equivalent. In the first stage of the process, both versions of the design are read into the equivalence-checking tool. During the read process, each design is automatically segmented into manageable sections called "logic cones." Logic cones (Figure 1) are groups of logic bordered by registers, ports, or black boxes (BB). The output border of a logic cone is referred to as a "compare point."

Next, the tool attempts to match, or "map," logic cones from the reference design to the corresponding logic cones within the implementation design. This is called "matching" (Figure 2). Both non-function (name-based) and function-based matching methods are deployed to map compare points.

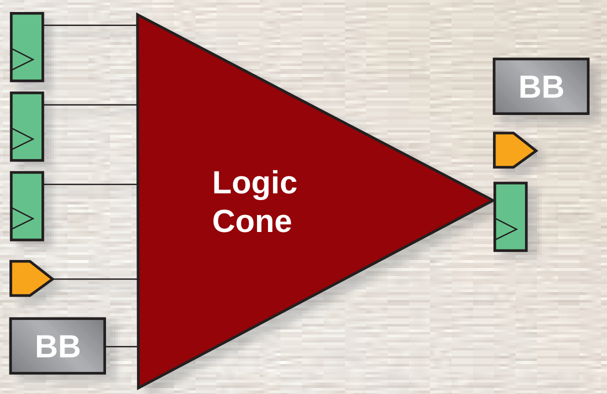


Figure 1 - Logic cone

Once the logic cones have been matched, the next step is to verify that the functionality of each matching cone is equivalent. Many solver (algorithm) technologies are available to prove the equivalence of logic cones: Formality EC uses SAT, BDD, Isomorphism, ATPG, and Arithmetic, among others. Once the verification step is completed, the tool produces a list of any compare points (logic cones) that are not equivalent. Formality EC also provides various debug and isolation capabilities to help isolate the implementation error.

Equivalence Checking in FPGAs

In an FPGA flow, verification challenges result from transformations during design implementation. Synthesis, place-and-route, and other tools in the design flow can cause many types of design transformations, such as combinatorial reductions, sequential optimizations (retiming), FSM re-encoding, register merging, or duplication, as well as other place-and-route optimizations.

If your EC tools are not set up to account for these transformations, verification becomes cumbersome. Formality EC accounts for the transformations performed by synthesis and Xilinx ISE (Integrated Software Environment) tools (Map and PAR) through use of the following files and utilities:

- **Verification libraries:** Formality-specific models for Unified Simulation (UNISIMS) components and post-PAR Simulation components (SIMPRIMS)
- **Constraint file(s):** to inform Formality EC of the synthesis tool's register-merging (if enabled) and Mapper optimizations relating to:
  - – registers that turned to constant
  - – ports that were optimized away
  - – ports whose direction changed
- **Netlist:** a Formality-compatible, gate-level netlist.

In traditional FPGA design flows, simulation is used to validate the functionality of the gate-level netlist produced by synthesis and PAR tools. In modern flows, simulation is replaced by equivalence checking (Figure 3).

RTL to Post-Synthesis Verification

You can use a number of synthesis tools to optimize designs during RTL operations. Xilinx supports the following synthesis tools for its Virtex and Spartan™ FPGAs:

- FPGA Express and FPGA Compiler II (FCII) from Synopsys
- SynplifyPro from Synplicity
- LeonardoSpectrum from Exemplar
- XST (Xilinx Synthesis Technology) from Xilinx.

Each synthesis tool employs its own combinatorial and sequential optimization, as well as retiming (if available) algorithms. Although the Xilinx/Formality flow as depicted in our model was validated using Synopsys FCII, the flow should work similarly with other synthesis tools.

Creating the Post-Synthesis Gate-Level Netlist

The Synopsys FCII post-synthesis gate-level netlist contains UNISIMS components and is in EDIF (Electronic Design Interchange Format). The EDIF netlist is fed into Xilinx ISE for mapping and PAR. The UNISIMS components are LUTs, flip-flops, I/O buffers, and other available resources in the targeted Xilinx architecture. Xilinx ISE provides the capability to generate a Verilog™ netlist at any stage in the implementation process.

We chose the Verilog post-synthesis netlist because Verilog netlists are commonplace and are easily read into Formality EC. We then created a Formality-compatible netlist using the following methodology:

- Read the design and the CORE Generator's™ EDIF netlists into ISE using NGDBUILD. This step transforms the EDIF netlist(s) into the Xilinx database format. The CORE Generator block will be covered in a later section.
- Create a Verilog netlist containing SIMPRIMS components using the NGD2VER program in ISE.
- Process this netlist using the xilinx2formality.pl Perl script to generate a Formality-compatible netlist.

The post-NGDBUILD netlist represents the result of two transformations: synthesis and NGDBUILD. Because the netlist contains non-synthesizable constructs and "defparam" statements that cannot be read directly into Formality EC, Xilinx and Synopsys developed the xilinx2formality.pl Perl script to process the post-NGDBUILD netlist into a usable format (Figure 3). Future improvements will enable Formality EC to read the Verilog netlist generated from the ISE environment directly.

UNISIMS and SIMPRIMS Libraries for Formality EC

Two special Xilinx verification libraries are needed for use with Formality EC:

- **UNISIMS:** The UNISIMS library contains the Xilinx primitives in RTL format. This library is required when the design contains Xilinx primitives, such as an instantiation of a DCM or block RAM.
- **SIMPRIMS:** The SIMPRIMS library contains the Xilinx primitives for back-annotated verification (Post-NGDBUILD, Post-MAP, Post-PAR).

These libraries must be read into their respective RTL and post-NGD containers within Formality EC during the design read stage. Xilinx provides specific unisims.fms and simprims.fms scripts to read the necessary models into Formality EC. Currently, the scripts read in the entire libraries. Synopsys is working with Xilinx to utilize Formality's read-library-on-demand feature – which will eliminate the need to read the entire UNISIMS and SIMPRIMS libraries and read only the components actually used in the design.

Reading CORGEN Models

Xilinx provides a comprehensive set of IP (intellectual property) blocks through the CORE Generator tool. These blocks, which range from simple shift registers and memories to complex Reed-Solomon encoder/decoder blocks, can be customized. The CORE Generator software generates all the necessary models for the customized IP blocks, including a behavioral model for simulation and an EDIF structural netlist with UNISIMS components. Together, these elements represent the optimum implementation of the IP block using the available resources on the targeted Xilinx FPGA architecture. You can instantiate these IP blocks as black boxes in your RTL code.

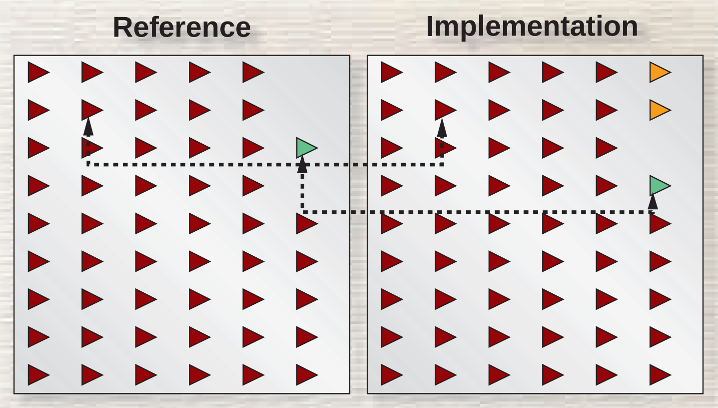


Figure 2 - Formality matches the cones from the reference design to their corresponding cones within the implementation design.

FCII generates an EDIF netlist containing the black boxes. NGDBUILD then uses the optimized structural EDIF representation of the blocks to fill the black boxes in the post-FCII EDIF netlist. The post-NGBUILD Verilog netlist, created using SIMPRIMS, contains the complete structural representation of the design, including the content of CORE Generator blocks. During RTL to post-NGDBUILD verification, Formality EC needs the functional model for a given IP block in the RTL to match it with the post-NGDBUILD netlist. For this, Xilinx provides core2formal, a Perl script that reads in the UNISIMS-based EDIF structural netlist for the IP block. This creates a Formality-compatible SIMPRIMS-based Verilog netlist. The SIMPRIMS-based netlist is the functional model that Formality EC uses to verify the CORE Generator blocks (Figure 3).

Performing the Verification

The RTL2postNGDBUILD equivalence-checking flow is easiest when FCII synthesizes the design without using the following optimization options: register-merging, max fanout control (register duplication), and register retiming. However, without these optimizations, QoR (Quality of Results) may be compromised. Therefore, handling these transformations in an equivalence-checking flow requires some additional consideration.

For the register-merging option (on by default), Synopsys developed the makeconstraints.sh script. The script reads the FCII-generated report, which details the list of merged registers, and then produces a Formality set_constraint command file. This command file is then read into Formality EC prior to verification.

Formality EC offers a special feature for handling max fanout control using the register duplication option (off by default): To handle the transformation automatically, enable the verification_merged_duplicated_registers variable in Formality EC prior to verification.

When a design is synthesized with retiming, verification becomes more difficult. Formality EC supports sequential optimizations (such as retiming) when localized or limited to a block, but FCII generally performs retiming on an entire design. To perform a successful verification with such optimizations, the command set_parameter–retimed must be used on all blocks that have undergone retiming. If you're planning to use Formality EC, use retiming sparingly in FCII.

RTL to Post-PAR Verification

Figure 3 illustrates the transformations that ISE applies to a synthesized netlist:

- **NGDBUILD:** Transforms the EDIF netlist(s) into Xilinx database format.
- **Map:** Packages the LUTs, flip-flops, SelectRAM, and other resources in the design into CLBs (configurable logic blocks), IOBs (input/output blocks), and so forth. Using the state-of-the-art Xilinx Mapper, you can apply certain transformations to the design, such as optimizing away constant registers, optimizing away ports that are no longer needed, and changing the direction of ports from bi-directional to output if warranted.
- **Place-and-Route (PAR):** PAR is the last step in implementing the design before creating the bitstream to program the Xilinx FPGA.

you must examine the Mapper's optimization of constant registers and some ports. Depending upon the target FPGA architecture and design constraints, the Xilinx Mapper uses special algorithms to identify:

- Registers that can be changed to a constant
- Ports that can be optimized away
- Bidirectional ports that can be changed to output only.

The Mapper performs these optimizations and records the result in the Mapper report.

These transformations must be accounted for during verification. The xilinx2formality.pl script reads the information relating to these optimizations from the placed-and-routed design database to produce a Formality constraint file. Reading this constraint file prior to verification enables Formality EC to account for these transformations.

Conclusion

Effective verification of today's large, complex FPGAs requires a static verification flow. Xilinx and Synopsys have created a solution that uses the Formality equivalence checker to provide a fast, thorough functional verification methodology. You can benefit from this flow today using existing implementation technology. Synopsys is currently developing an improved, streamlined verification flow to handle next-generation FPGA implementation technologies.

Xilinx provides a comprehensive FAQ, application notes, and updated information for the Xilinx/Formality EC flow. Go to *http://support.xilinx.com/company/search.htm* and search for "Formality."

Creating the Post-PAR Netlist

After PAR, a SIMPRIMS-based Verilog simulation netlist is created using NGD2VER, as shown in Figure 3. In the Xilinx design flow, this netlist, along with its accompanying SDF file, is used in functional and timing simulation to verify design integrity after Map and PAR. The same netlist, processed with the xilinx2formality.pl script, is read into Formality EC for functional verification.

Performing the Verification

Before the RTL to post-PAR verification with Formality EC can be completed successfully,

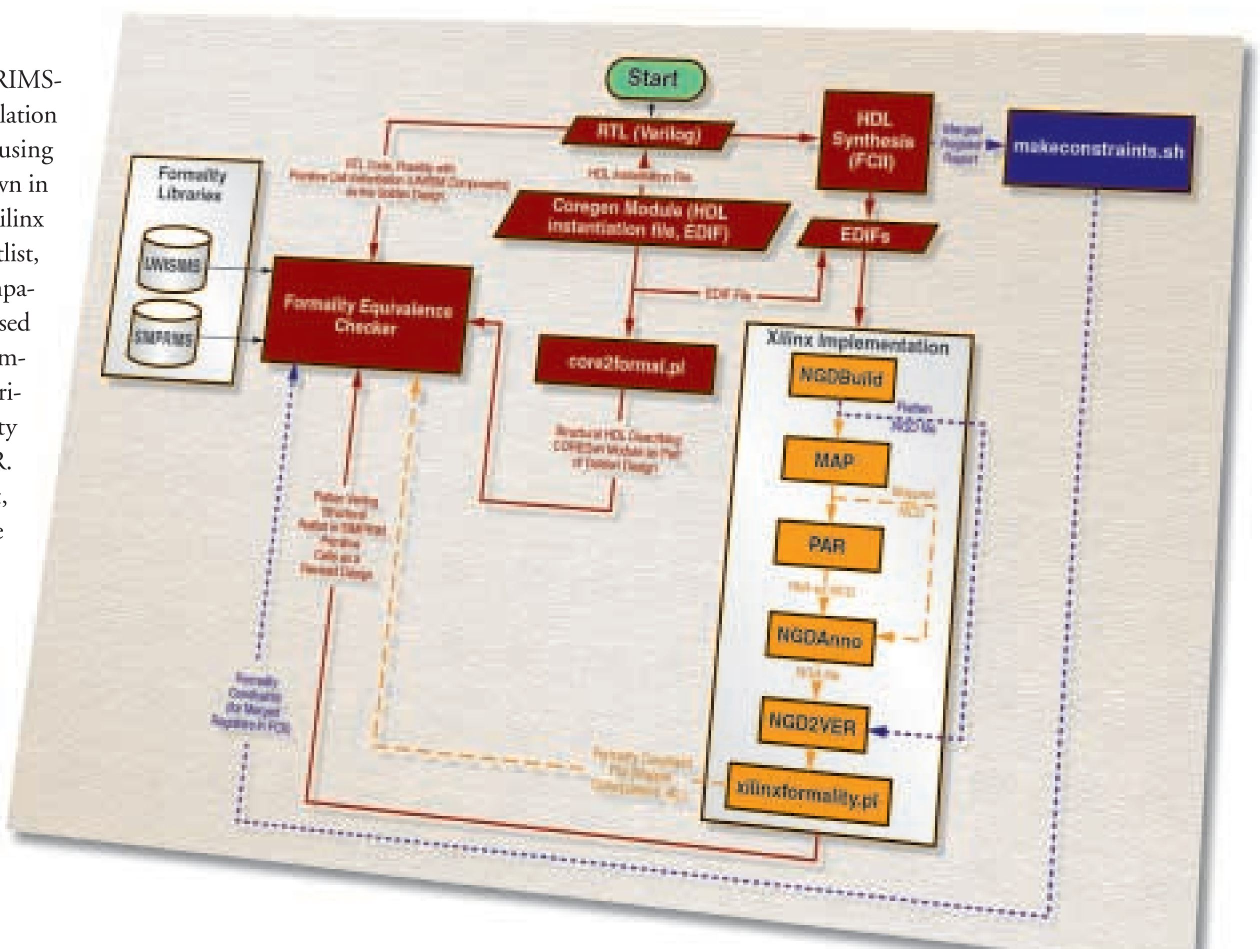
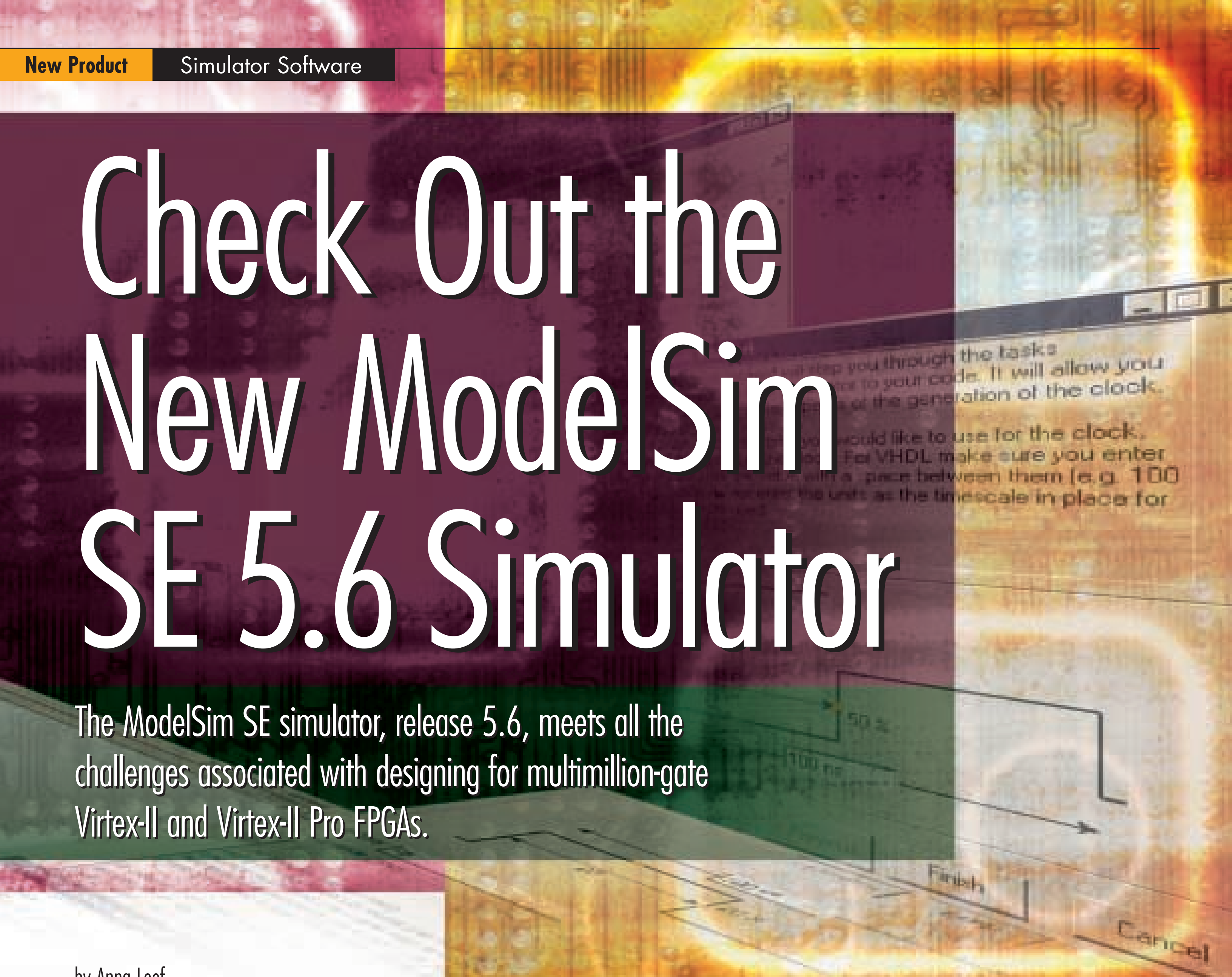


Figure 3 - Xilinx/Formality equivalence checking flow



Check Out the New ModelSim SE 5.6 Simulator

The ModelSim SE simulator, release 5.6, meets all the challenges associated with designing for multimillion-gate Virtex-II and Virtex-II Pro FPGAs.

by Anna Leef
Product Marketing Manager
Model Technology
annal@model.com

True to its origins, every feature of the newest member of the ModelSim® family from Model Technology is integrated into its Single Kernel Simulation (SKS) technology. SKS provides the highest capacity and performance, regardless of the languages or platform you choose.

- It excels in all types of design environments.
- Its performance is equal to the most demanding simulations.
- It provides VHDL, Verilog™, and mixed-language support.
- Its powerful debugging capabilities can solve the most difficult problems.

Multimillion-Gate FPGA Designs

Multimillion-gate designs require high performance for all simulations, as well as the capacity to handle the demands of gate-level timing simulation. Today, the ModelSim SE tool is used on designs exceeding 25 million gates. The ModelSim 5.6 tool offers a number of new performance-enhancing optimizations.

With 60% of the market, the 5.6 release accelerates the industry's leading VHDL simulator. The release also delivers ModelSim's third-generation Verilog global optimization technology and includes new optimizations for mixed-language designs. ModelSim VHDL has been updated with improved memory management, IEEE library performance optimizations, and other intelligent compiler advances that facilitate a broad range of designs. ModelSim's third-generation global optimization technology continues to improve Verilog RTL (Register Transfer Language) and gate-level performance across many design styles. For maximum Verilog performance, you should compile your top-level modules with "+opt." This turns on the optimizations. (As with all Verilog simulators, ModelSim's performance mode affords less visibility into the design than the debug mode, so debug your design first and then enable +opt for regression tests.)

Tuning Your Design For Performance

Larger designs mean more tests. Typically, billions of vectors are simulated against large designs, so any drag on simulator performance can dramatically increase the amount of time you spend on verification.

The ModelSim 5.6 release includes a new option that significantly improves simulation throughput. After a design is compiled, it must be loaded into the simulator. This process is called "elaboration." Elaboration, especially for large gate-level simulations with timing, can consume a significant part of the overall simulation run time.

ModelSim's new elaboration option loads the design into a reusable file so that multiple simulations can be run off the same file using different stimuli – eliminating the need to reload the design. As an example, a ModelSim customer had a 5 million-gate design. It took an hour to load the design and SDF (Standard Delay Format) file. His test suite contained thousands of tests. Adding an hour to each run would have created a drastic performance penalty, but with the new elaboration option, he only had to load the design once.

By analyzing your entire design flow, ModelSim SE's integrated Performance Analyzer can uncover bottlenecks such as the impact of testbench tools, .vcd file generation, or inefficient HDL coding styles – often identifying additional opportunities for better throughput. Measuring the performance impact of all areas of your environment gives you the power to make better technology decisions.

In addition, larger designs are also much more likely to include simulation models and testbenches written in languages other than VHDL or Verilog, which also can have a negative impact on performance. Many users unknowingly decrease performance by creating many events through the testbench interface. But with ModelSim, your productivity does not have to be hobbled by your testbench – or other tools. Many users have identified performance bottlenecks and have either modified their environment coding style or found replacement tools that can be easily integrated into the ModelSim tool.

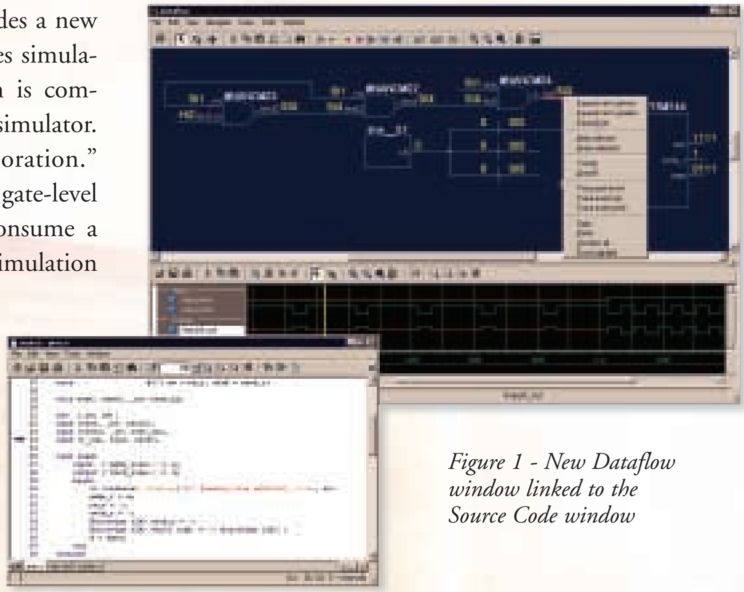


Figure 1 - New Dataflow window linked to the Source Code window

To better understand how to optimize the ModelSim simulator for performance, please refer to the performance application note at *www.model.com/resources/pdf/improving_performance.pdf*. This document provides performance flow details for Verilog, VHDL, and mixed-language simulations.

Best User Interface and Debug Tools

The complexity of multimillion-gate designs makes it no longer feasible to debug errors on the lab bench. What you need is an integrated debug environment with full access to the internal components of the design. The ModelSim SE 5.6 simulation tool delivers the industry's most tightly integrated and feature-rich solution for debugging. Source code debugging, waveform generation and comparison, an enhanced Dataflow window, and code coverage are some of the features already available in the ModelSim SE 5.6 release.

With the ModelSim 5.6 edition, a completely revamped Dataflow window enables you to view and debug your design graphically. The window depicts the physical connectivity of your design and lets you easily investigate unexpected values.

In addition, because all of the ModelSim windows are cross-linked (Figure 1), you can simply drag-and-drop design elements from the Structure or Signal windows to the Dataflow window. The visual trace engine then generates a graphical representation of that portion of the design. From there you can expand to any level by sprouting inputs or outputs.

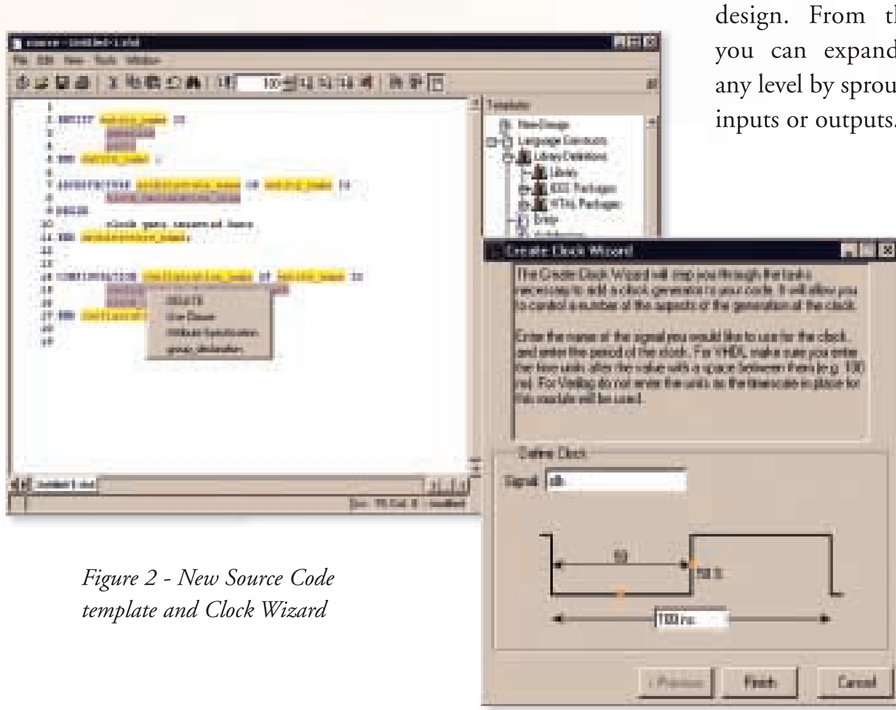


Figure 2 - New Source Code template and Clock Wizard

To identify a signal with an unknown value in the Dataflow window, simply place the cursor in the Wave window at the time of the unknown, then use ModelSim's new ChaseX™ feature to draw a path to the source of the unknown. The underlying HDL code will appear in the Source window.

To simplify design entry and editing, the Source window has new code templates and design wizards to help you create VHDL and Verilog code. All language constructs are available with a click of the mouse. Context-sensitive expansion of templates means you don't have to know which constructs go where. The design wizards walk you through building more complex HDL blocks, including parameterizable logic blocks, testbench stimuli, and new design objects. Advanced developers can use the code templates as an interactive language reference manual.

Saving simulation data is easier with waveform viewing and exporting. These features allow you to save simulation data for viewing or comparing, even while the simulation is still running.

Project management was also improved significantly in the ModelSim SE 5.6 release. The new version further enhances these improvements with a streamlined interface, an automatic compile-order function, and support for reusable design views that run different configurations of the simulation. Project management enables efficient debug, source modifications, recompiling, and resimulation without any scripting knowledge.

Conclusion

With fast performance, the most comprehensive set of integrated debug tools, and proven success on multimillion-gate designs, ModelSim SE 5.6 is a natural choice for Virtex™-II and Virtex-II Pro™ designs. To download an evaluation copy of ModelSim 5.6, go to *www.model.com/evaluations/default.asp.*

ModelSim SE 5.6 Easily Simulates 6 Million-Gate FPGA Design

Dillon Engineering, Inc., of Edina, Minnesota, recently developed a two-dimensional Fast Fourier Transform (FFT) for an image processing application. Among the many complex aspects of the project, the Dillon Engineering team had to contend with two different sizes of the same RTL (Register Transfer Language) design. One of the benchmarks required by Dillon's customer was a physical simulation of the whole design.

"We were able to simulate the smaller design with our existing tool, but we couldn't run the big, fast version because it exceeded the capabilities of the simulator we were using," said company President Tom Dillon.

Simulation Required for Large Designs

The larger of the two designs the Dillon team had to simulate was a 6 million-gate design with nearly 18 MB of external memory. The design targeted two Virtex-II XC2V6000 FPGAs. "This system had enormous processing requirements," said Dillon. "A huge amount of raw data had to undergo extensive processing to convert it into the final 2D FFT. The combination of 16-bit pixel data, a resolution of 2K x 2K pixels, and a required frame rate of 120 fps resulted in 480 megasamples of 16-bit data per second."

Existing Simulator Couldn't Handle the Design

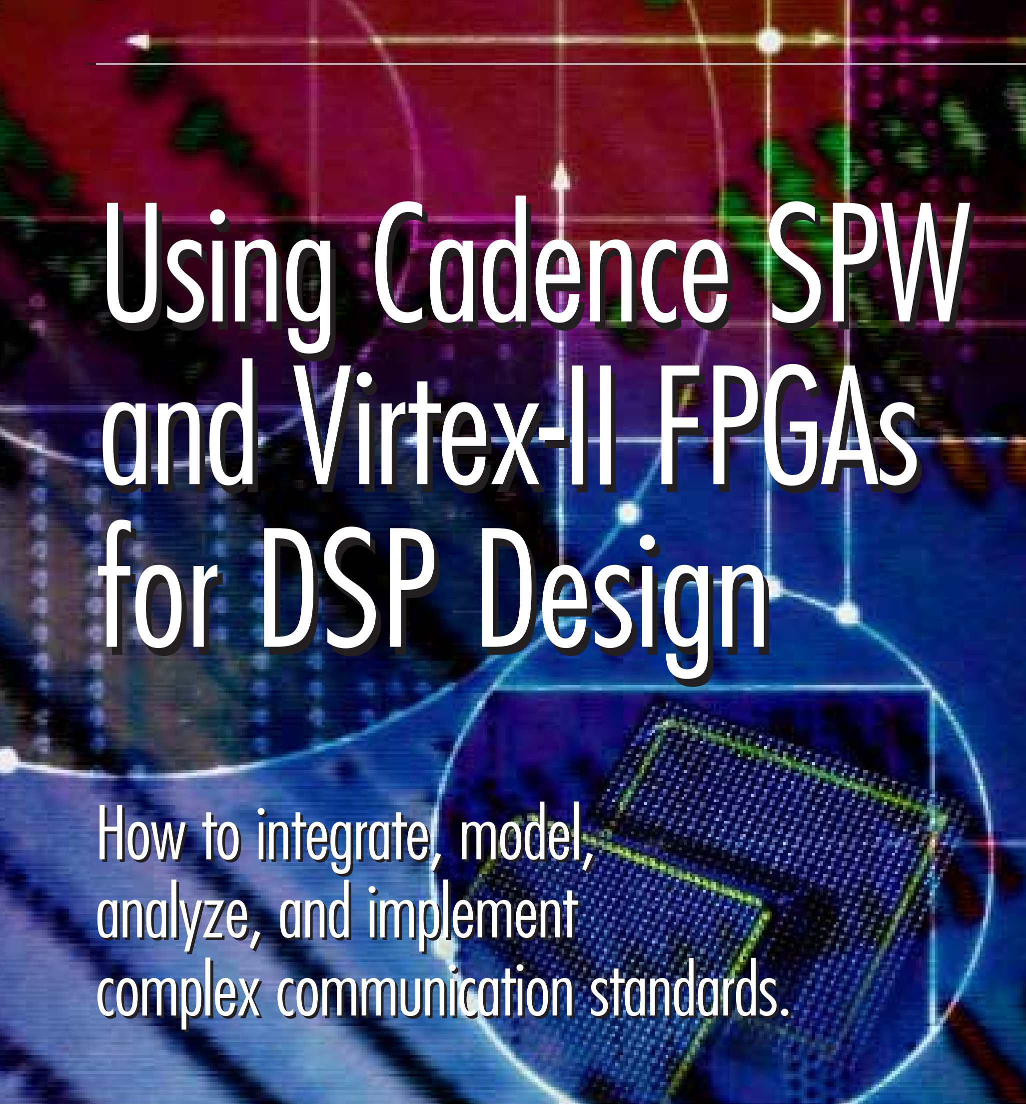
Because the 6 million-gate design kept crashing their existing simulation tool, the engineers decided to try a 30-day evaluation of the ModelSim simulation tool. "We knew we were in trouble if we stuck with our existing simulation tool. If you can't simulate the full design, you can't be sure it's going to fit and actually work. We had to produce simulation results of the full image in the specified number of clock cycles, so we didn't have a choice: we couldn't 'get by' with the smaller simulation," Dillon said.

ModelSim successfully completed the FFT simulation of the full design – meeting the required benchmark.

By upgrading to ModelSim, Dillon Engineering reaped a 30% increase in simulation performance on a million-gate FFT design. They were able to simulate an entire 6 million-gate design for the first time. And they had better language coverage than they had with their old tool.

Summary

"We are growing, and as we do, we are getting better and bigger projects. That means we need bigger and better tools," said Dillon. "As a custom design firm, we have to move up a class in tools to support the projects we want to bring in. The bigger the design, the more ModelSim stands out as the tool to use."



Using Cadence SPW and Virtex-II FPGAs for DSP Design

How to integrate, model, analyze, and implement complex communication standards.

by Michael R. Sturgill
Senior Technical Leader
Cadence Design Systems
sturgill@cadence.com

Today, communication designers face the daunting challenge of rapidly integrating multiple standards within their design to capture time-to-market opportunities. Specifically, these designs must support the seamless integration of WPAN, WLAN, and cellular radio standards into a single application, such as multimedia or digital video broadcast (DVB). In this article we will show how efficiently the Cadence® Signal Processing Worksystem (SPW) integrates, models, analyzes, and implements complex communication standards into a high-end Xilinx Virtex™-II FPGA.

Cadence Signal Processing Worksystem (SPW)

Based on years of proven technology, SPW offers a fully integrated solution for multi-ASIC/FPGA systems or SoCs/SoPCs (system-on-chips/system-on-programmable-chips), from algorithm design to implementation. Figure 1 depicts a simplified SPW flow.

SPW provides a unified design environment that accelerates the development of digital signal processing (DSP) systems, allowing hardware and system engineers to collaborate, share design libraries, and propagate testbenches at every level of design abstraction. In addition, SPW facilitates DSP design by offering the following:

- Integration of C/C++, MATLAB®, SystemC, Verilog-AMS, Verilog, and VHDL blocks into a single design, allowing multiple language flexibility
- Hierarchical design methodology via a graphical block diagram editor, promoting better design, reuse, and documentation
- Architectural convergence that combines datapath and control constructs into a single simulation environment, allowing you to capture and simulate the most advanced electronic systems
- Mix-and-match combinations of a variety of design styles and technologies, enabling your design teams to spend less time writing algorithms and more time optimizing designs
- Tight integration with SPW Hardware Design System (HDS), NC-Sim for RTL (Register Transfer Language) verification, Verilog-AMS (Analog Mixed Signal) for mixed analog and digital simulation, Synplify for logic synthesis, and Xilinx for core generation, place and route – minimizing overall design time.

Variable Interpolation Filter for DVB Applications

Figure 1 illustrates the concept-to-FPGA flow as an interpolation filter designed for use in DVB applications. The filter's output operates in the range of 4 to 48 times the input symbol rate (Rs), while Rs ranges from 1 to 45 megasamples per symbol (MSPS). The multiple interpolation rates available in this model allow the digital-to-analog (D/A) converter output of the DVB transmitter to operate in a relatively narrow frequency span (≈ 45-180 MHz, 4:1 ratio) while operating over a broad symbol rate range (≈ 1-45 MSPS, 45:1

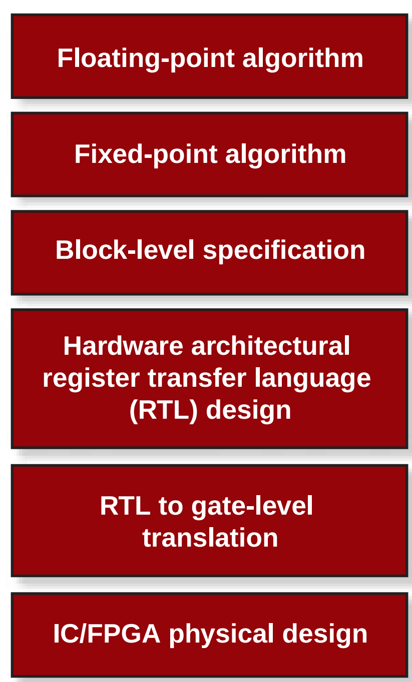


Figure 1 - SPW flow

Modulation Type	Filter Rolloff Factor	Interpolation Factor	Convolutional Encoder Rate
QPSK	35%	4, 8, 16, 32, 48	1/2, 2/3, 3/4, 5/6, 7/8
8PSK	35%	4, 8, 16, 32, 48	2/3, 5/6, 8/9
	25%	4, 8, 16, 32, 48	2/3, 5/6, 8/9
16QAM	35%	4, 8, 16, 32, 48	3/4, 7/8
	25%	4, 8, 16, 32, 48	3/4, 7/8

Table 1 – Interpolation Filter Operating Modes

ratio). This significantly simplifies the design of the analog anti-aliasing filters following the D/A converter.

Operating Modes

The interpolation filter module operates in the modes detailed in Table 1, which also defines the modulation types, rolloff factors, and convolutional encoder rates. The final output frequencies are a function of the Xilinx technology and speed grade chosen, and have been designed to operate at up to 180 MHz. This corresponds to a maximum symbol rate of 45 MSPS.

Theory of Operation

The module operates over a broad range of interpolation factors for each of the modulation types and rolloff factors described above. Figure 2 illustrates the block diagram of this module and the distinct filters used to form the multiple interpolation rates. Combinations of interpolate x4, x3, and x2 make up the specified 4x to 48x range.

The initial interpolate x4 stages for the QPSK (Quadrature Phase-Shift Keying), 8 PSK, and 16 QAM (Quadrature Amplitude Moderation) modes apply the square root raised cosine (SRRC) masks defined in Ref. [1]. These masks also have an x/sinx pre-distortion applied to them. This pre-distortion compensates for the sinx/x droop that occurs when the digital signal is processed through a D/A converter. The output of the D/A converter is then spectrally flat. Filt1, Filt2, Filt3, and Filt4, which are spectrally flat over the passband range of the shaping SRRC filters, are used to interpolate the signal and attenuate the out-of-band images that occur as a result of the interpolation process.

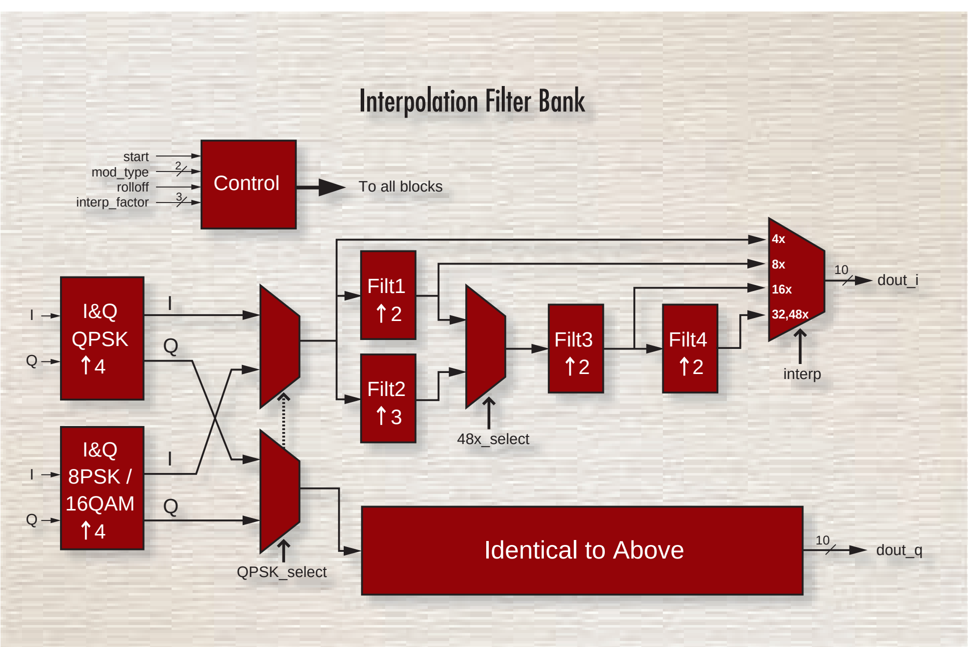


Figure 2 -Interpolation Filter Bank

Fixed Point Architectural (Implementation) Model

There are multiple ways to model systems within SPW. Here, we will focus on the fixed point architectural (implementation) model.

The SPW simulation model, or implementation testbench, depicted in Figure 3 consists of the design under test (DUT), input stimulus data, multirate input controls, parameterizable mode selection and data rate control, and output response capture for visualization and post-processing. All of the SPW building blocks that comprise the DUT are synthesizable. Several blocks in the DUT also have Virtex-II core component instantiations as well.

Implementation Testbench

The simplistic illustration (Figure 3) of this testbench masks the scope of its capabilities. Table 1 lists the operating modes of the design. Careful examination of this table yields roughly 40 different operating modes that must be tested. This testing is achieved by using parameterization at the testbench level.

Figure 3 is color-coded for easy reference to the different parts of the testbench. Together, these blocks make up the testbench. The descriptions are as follows:

- **DUT –** Design under test. Figure 2 shows the modules contained within this block.
- **Simulation Parameters –** These settings allow the DUT to be tested over your required range.
- **Operating Modes –** The top four entries are pointers to the coefficient files used for the QPSK filters. The remaining three enumerated parameters allow you to select the interpolation factor, modulation type, and the filter's rolloff factor.
- **Dynamic Control –** These include the system clock, power on reset signal, and the data input strobe signal.
- **Static Control –** These are constant blocks whose values are computed from the simulation parameters. They provide a convenient way to mimic the control registers in the final design.

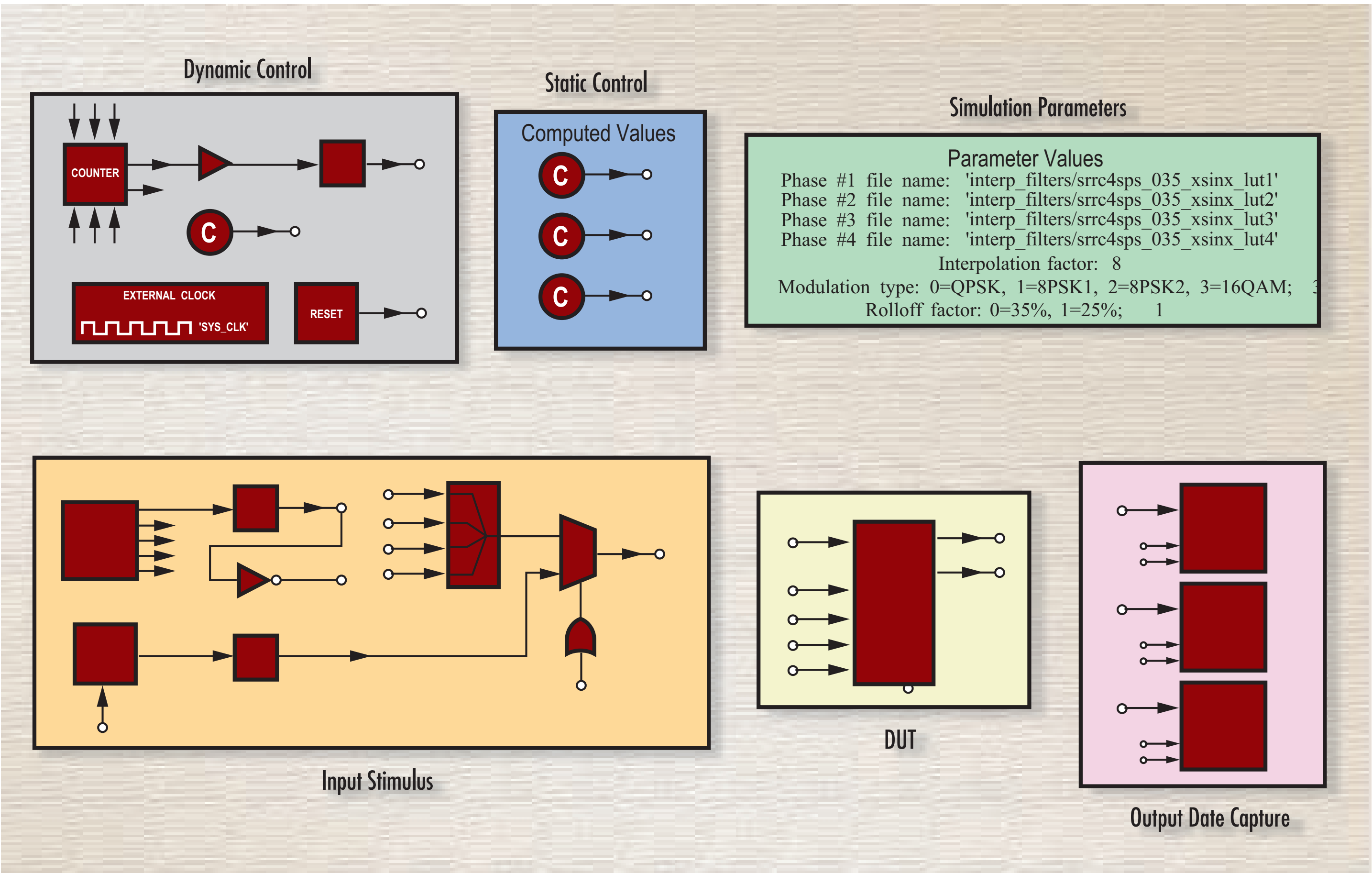


Figure 3 - Testbench for the interpolation filter design

- **Input Stimulus –** There are two possible input selections: The first is a signal source that contains quantized data of a swept sinusoidal waveform. The second input is a random noise generator set to output a uniform white random variable in the range [0, 15]. This allows the filter to operate over all possible transitions of a constellation.
- **Output Data Capture –** The output data capture comprises a set of signal sinks that capture data and write it to a file for post-simulation analysis. You can also use interactive instrumentation to view other aspects of the system, such as eye diagrams, constellations, and so forth.

Implementing the Design

Xilinx Virtex-II devices are an ideal hardware solution for the stringent timing requirements of this design. The Virtex-II devices have dedicated on-board resources specifically meant for high-speed DSP designs, and the design makes efficient use of the hardware multipliers and numerous block RAMs.

SPW has a direct link to the Xilinx CORE Generator™ tool, allowing you to specify the Xilinx core to be imported. Once the core is defined, an SPW library block is created, which is then available for instantiation into all designs. This is the method we used to instantiate the on-board multipliers and the block RAMs in our model.

Using SPW eliminates the need to create Xilinx cores for every block in the design because it allows you to import VHDL and Verilog in combination with Xilinx cores and blocks from SPW's hardware design system. Figure 4 shows the top-level hierarchy for the entire interpolation filter module.

After the design has been captured, simulated, and validated against the behavioral model, it is time to take it to the hardware. The steps involved in this process are as follows:

1. Generate the RTL (VHDL or Verilog) for the design under test.
2. Synthesize the design, targeting the Virtex-II device.
   (Note – If the target synthesizer has the capability of outputting an RTL netlist, this netlist can also be simulated in SPW with the same testbench used for the DUT.)
3. Place-and-route the design using the Xilinx tools. Here again, the netlist created by the Xilinx tools may be simulated in the same SPW testbench.
4. The bit file created is then downloaded to the board, where the design may be run in real time.

The full design runs at data rates of greater than 180 MHz on a Virtex-II XC2V3000-6-FG676 device.

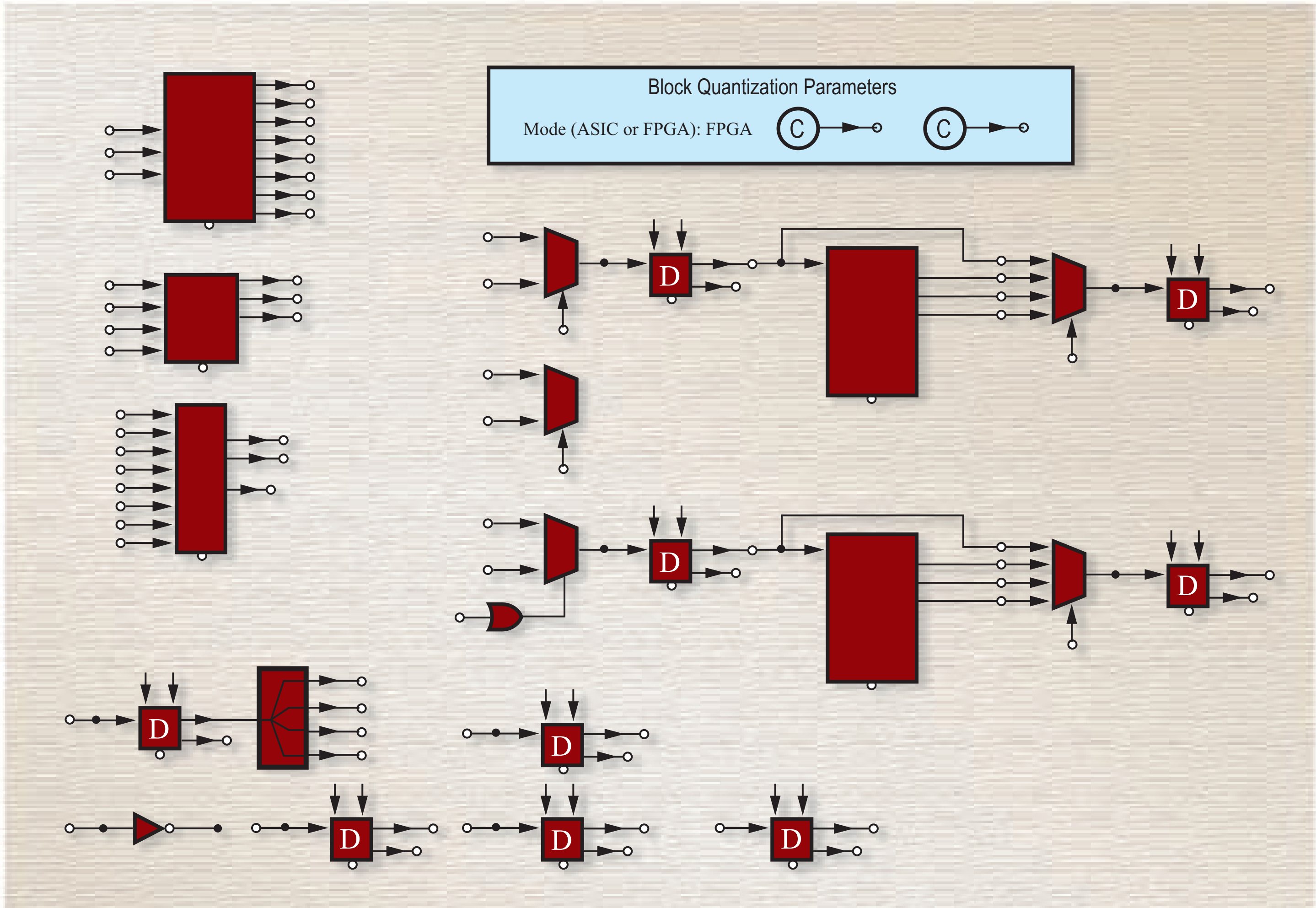


Figure 4 - Interpolation Filter Design (DUT)

Downloading to the Nallatech Development Board

The Nallatech board, which is part of the Xilinx XtremeDSP™ Development Kit, comes with PC software for FPGA configuration and control. The board is populated with a Virtex-II XC2V3000 FPGA for general use, as well as an additional FPGA dedicated to the many clocking schemes available. Connection to the PC is via PCI or USB, and installation is a snap.

We edited the constraint files included with the kit to match the I/O and clocking requirements of the design, and used the included software to download the clock and interpolation filter bit files. The software also includes an interface to the hardware, which enabled us to design a simple bus interface into the interpolation filter to allow configuration changes via the PC software. All modes defined in Table 1 are programmable via this interface. The filter's inputs may also be configured to use either the on-board A/D converters or an internally generated noise source.

Conclusion

Cadence SPW, used in combination with the Xilinx Virtex-II FPGA, creates a powerful and robust solution for meeting the demands of today's DSP designers. By providing a smooth path from system-level design and verification to implementation, SPW offers an effective bridge from system concept to hardware realization. The Virtex-II devices contain the requisite components needed in DSP design, and their speed and density allow entire systems to run in real time.

In this article, we have barely touched on the breadth of capabilities of SPW coupled with Virtex-II devices. We hope, however, that we have given you an indication of the capabilities available. In the past, high-speed designs, such as the one depicted in our model, were feasible only in ASICs, but with Xilinx pushing the speed and density envelope, they are now possible in a reprogrammable device.

For more information on Cadence SPW and NC simulators, visit *www.cadence.com/products.*

References

[1] EN 301 210 v1.1.1 (1999-03) "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for Digital Satellite News Gathering (DSNG) and other contribution applications by satellite," European Standard (Telecommunications series), 1999-03.

[2] M. Sturgill, "Square Root Raised Cosine Transmit FIR Filter For DVB Applications," Cadence Design Systems Publication, 2000.

How to Manage Power Consumption in Advanced FPGAs

In this article, we discuss various sources of power consumption and techniques that can be used for power management in advanced FPGA devices. Then we describe how various features of Synplicity's Amplify Physical Optimizer software can be used to realize the power management techniques.

by Sanjay Thatte
Technical Marketing Manager
Synplicity Inc.
sanjay@synplicity.com

and John Blaine
Customer Application Engineer
Xilinx Inc.
john.blaine@xilinx.com

Current technology trends point to a rapid growth in FPGA device sizes with designs operating at higher frequencies. These design characteristics give rise to a number of complex issues. As designers, you must simultaneously:

- Address the stringent goals of meeting fast timing performance
- Fit the design into a cost-effective device
- Meet aggressive product schedules.

Additionally, high operating frequency and a high percentage of device utilization can increase design power consumption and junction temperature substantially. As the temperature rises past typical device ratings, performance and reliability are degraded. Device power consumption can reach a level that causes the maximum rated junction temperature to be exceeded, resulting in thermal destruction of the FPGA device.

Factors such as ambient temperature, airflow, and heat sinks, which can prevent device overheating, may be beyond your control. Industrial parts with extended temperature ratings are an option, but these parts are expensive and have limited package selection.

In addition to timing, area, and time to market, it is imperative that you proactively address the issue of power consumption and thermal stability.

An FPGA device, used in battery-powered applications or even high-performance applications where heat dissipation is a concern, can benefit significantly by applying some basic power management techniques.

Power Consumption

Power consumption in digital CMOS circuits arises from:

- Leakage current
- Transient short-circuit current between supply rails during transistor switching
- Charging/discharging of parasitic capacitances during normal internal logic state changes
- Charging/discharging of parasitic capacitances due to variation in input arrival times
- Charging/discharging of external load capacitances.

Neither the leakage nor the transient current can be optimized by design implementation. Therefore, to minimize power consumption, you must focus on optimizing the last three sources of power consumption involving capacitance.

The general formula for calculating power consumption for a design is based on the operating voltage, sum capacitance of all interconnect and logic resources, and the frequency of transition at the nodes.

$$P = \text{sum}(C * V^2 * f)$$

P = total power consumption
V = operating voltage
C = net capacitance
f = transition frequency

The operating voltage and external load capacitance are typically determined by system design requirements. To minimize power consumption for an FPGA device, the internal net capacitance and the toggle frequency must be reduced.

Power Management Techniques

Techniques used to minimize power consumption attempt to reduce the number of switching signals and the capacitance on the nets that are switching frequently. Some of these power minimization techniques are to:

- Minimize the number of clock buffers switching and the clock network capacitance
- Minimize capacitance on high frequency logic
- Isolate high activity logic to reduce interconnect length
- Isolate memory with high frequency access
- Minimize unnecessary switching and eliminate glitches.

Power management involves the application of advanced tools at the beginning of the design cycle to address the complex issues of interconnect. Fortunately, minimizing interconnect capacitance on critical nets helps you to reach your timing performance goals as well as reduce power consumption. Standard logic synthesis tools are not equipped to help you proactively manage the interconnect capacitance. As a designer, you must use a more advanced physical synthesis tool to achieve these goals.

Synplicity's Amplify® Physical Optimizer™ tool is the only market-proven FPGA physical synthesis software solution available today. More than 130 companies are already using the Amplify tool to manage the interconnect-related issues effectively and to reach aggressive timing performance goals.

Amplify Physical Synthesis

The Amplify tool provides an intuitive interface for creating regions on an FPGA device and then assigning the desired logic to those regions. The tool then uses physical constraints that incorporate your knowledge of the design's timing and power requirements to perform physical synthesis and to create a highly optimized design netlist.

The following sections briefly describe a number of Amplify's advanced features that can help you effectively manage timing and power goals. [Editor's note: For a more in-depth description of Amplify tool capabilities, go to Xcell Online at *www.xilinx.com/publications/xcellonline/*.]

1. HDL Coding Style

The Amplify tool uses HDL code with design constraints, such as timing and physical constraints, to perform advanced physical synthesis. You can significantly influence device power consumption through careful HDL coding. With proper coding structures, logic can be turned off when not needed.

2. Gated Clock Support

The Virtex™-II family of FPGAs makes advanced clocking schemes available to designers. Using Amplify software, you can take advantage of primitives like BUFGMUX for switching from a high-frequency clock to a low-frequency clock,

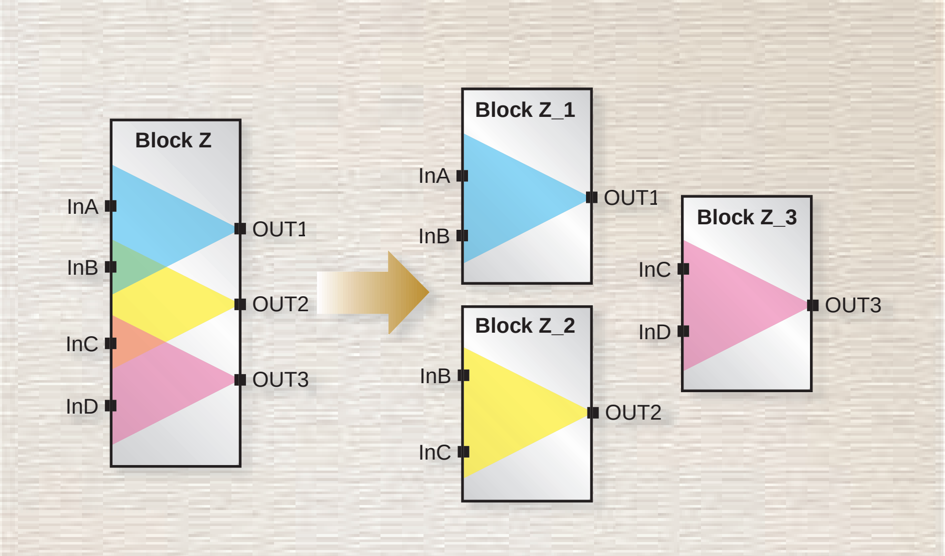


Figure 1 - Zippering

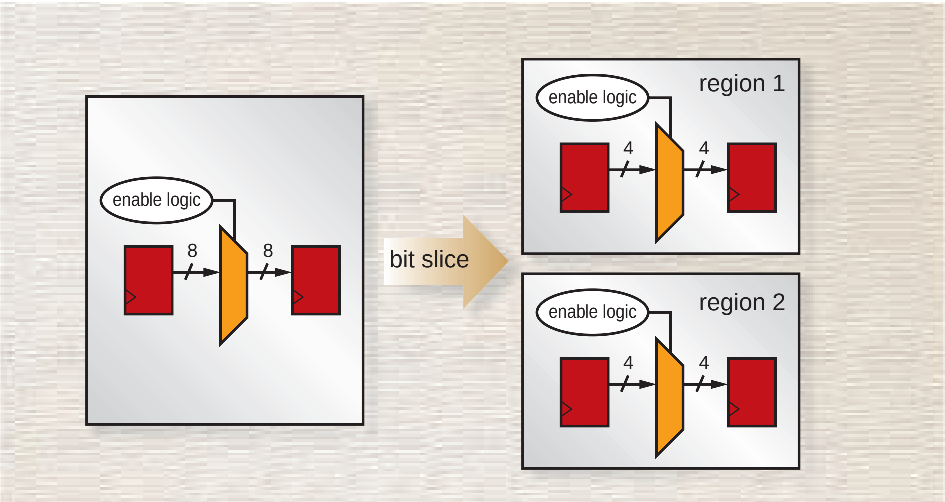


Figure 2 - Bit-slicing critical paths

and BUFGCE for dynamically driving a clock tree only when the corresponding logic is used.

3. Retiming and Pipelining

Typically, large logic blocks have long and active critical paths The Amplify tool automatically rebalances long critical paths by moving registers across logic boundaries to reduce the path length, net capacitances, and variance in delay paths to minimize glitch power.

4. FSM Encoding

The Amplify tool performs powerful state machine encoding and optimizations automatically. The FSM Explorer function can make use of user-specified constraints to choose the optimal encoding for state machines in the design.

5. Pad Type Selection Support

A large output load can increase power consumption. The Amplify software allows you to specify the pad type used when driving these loads. You can select a slower pad with the xc_pad_type attribute to reduce power consumption.

6. Zippering

Large functional blocks typically tend to be spread out over the FPGA device during placement and routing, causing some nets to have large routing capacitances and unnecessary power consumption. The Amplify tool provides a powerful netlist restructuring capability (Figure 1) to manage such large functional blocks in a design. You don't have to make any changes to the RTL code, and all netlist restructuring is done by the Amplify tool.

7. Bit-Slicing

When a large bus is routed, groups of bus bits must be clustered together to ensure similar timing. The Amplify tool's easy to use graphical interface (Figure 2) allows you to specify both the number of slices and the number of bits per slice. Combining bit-slicing with RTL floorplanning helps you get more uniform delays on the inputs to logic blocks and minimize glitch power. Also, by gaining finer control over bus placement and routing, you can control the capacitances of associated nets and minimize power consumption.

8. RTL Floorplanning

The Amplify Physical Optimizer software provides a user-friendly graphical interface (Figure 3) for creating physical constraints interactively. Working on a display of the device footprint, you can create rectangular regions of desired size at selected locations on the device. You can then assign entire modules or logic on selected critical paths. Through this process, you can easily localize critical paths to restrict the length of critical nets. Controlling the net length prevents incurring large routing capacitances and in turn, excessive delays and power consumption on these nets.

Because the floorplan is created before the design is synthesized, the Amplify tool makes use of the physical constraint information to better optimize the netlist. This netlist, created through Amplify's physical synthesis function, can be tailored specifically to your timing and physical constraints. The Amplify software also synthesizes the gate-level floorplanning it derives from the specified physical constraints, and it forward-annotates the information to the place-and-route tools.

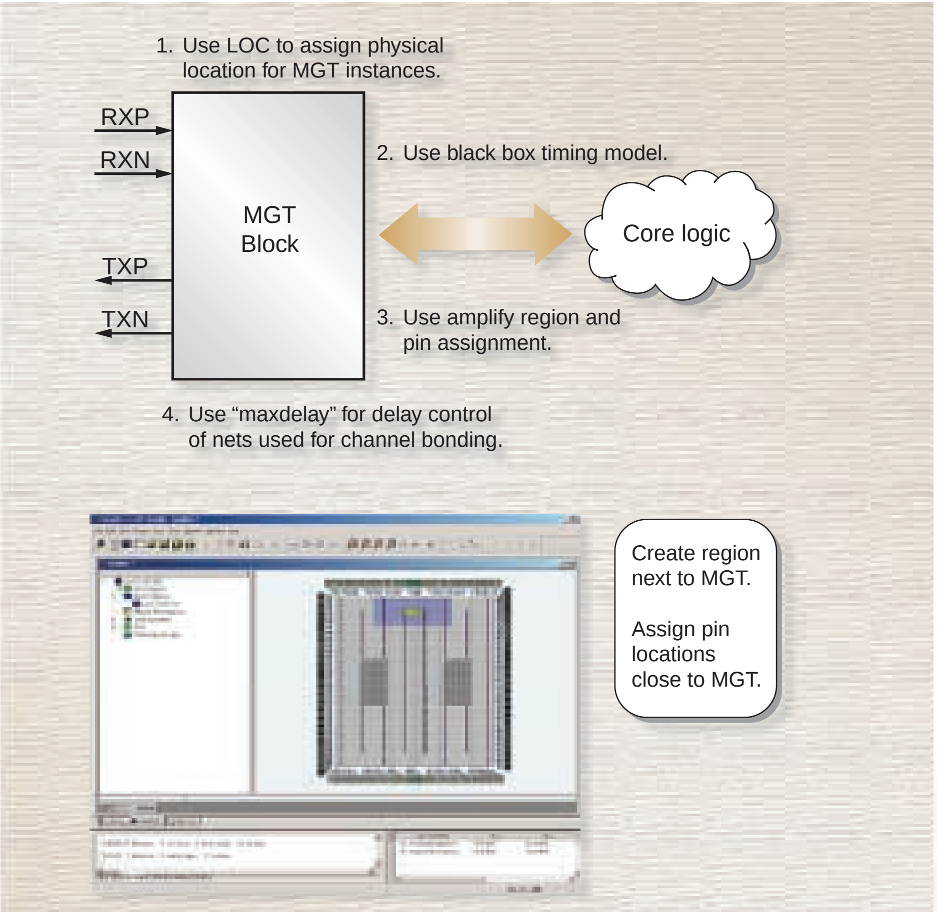


Figure 3 - Amplify RTL floorplanning with MGTs (multi-gigabit transceivers) in Virtex-II Pro™ FPGAs

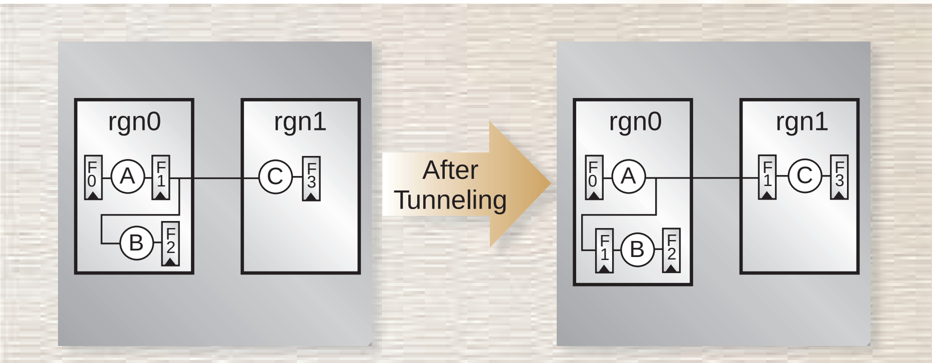


Figure 4 - Tunneling

Amplify allows you to perform clock domain floorplanning easily. You can select all the registers driven by a high-frequency clock net and assign them to a region that follows clock tree boundaries. Xilinx P&R software disconnects unused clock tree branches. Reduction in number of switching clock buffers and clock net capacitance reduces power consumption. The Amplify interface allows selection and assignment of RAM resources to isolate high-access memory, BlockMULT to isolate high-activity logic, and I/O pins to minimize external loading.

9. Tunneling

After you create regions and assign logic to those regions, the Amplify tool performs some intelligent optimizations to make sure there is no excessive region-to-region routing. Whenever a register drives logic into another region, an unnecessary routing penalty is incurred. The Amplify tool automatically replicates and moves a copy of the register to the region where the logic is being driven (Figure 4), keeping the net capacitance low and minimizing power consumption.

10. Replication

The Amplify tool performs various physical optimizations to control net capacitances. For nets with large fanouts, the Amplify software automatically replicates the driving cells to reduce the fanout and power consumption. Reduction in net capacitance can help control power consumption. The Amplify tool also provides the ability to perform manual replication.

Conclusion

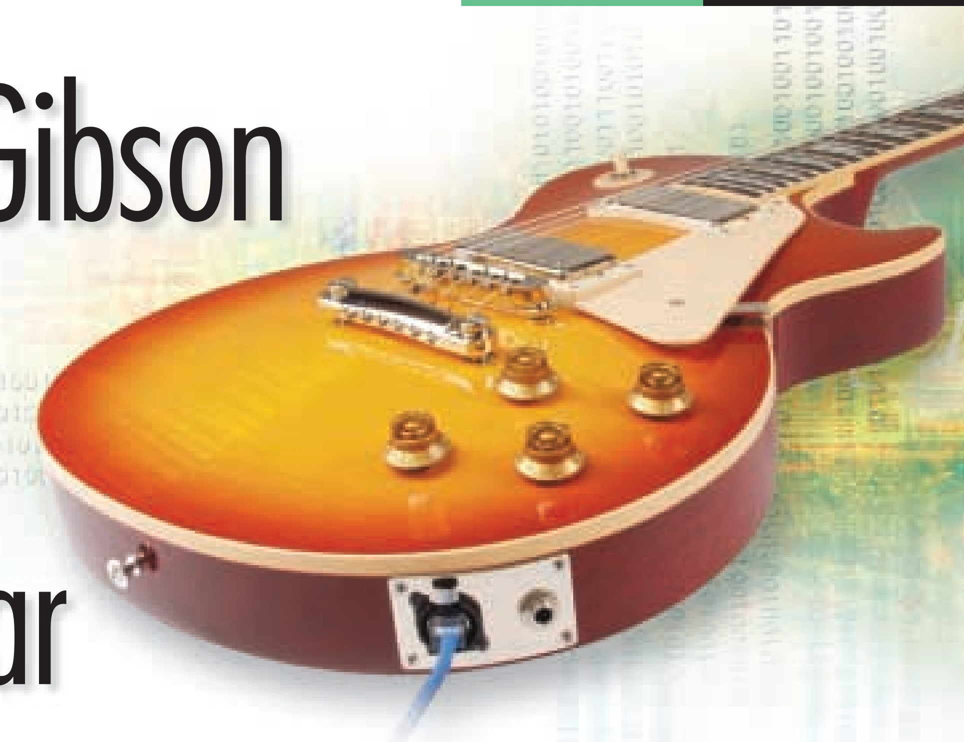
Amplify Physical Optimizer software from Synplicity is becoming a must-have tool for FPGA design. The Amplify tool helps you resolve interconnect related issues early in the design cycle and enables you to manage power consumption without sacrificing timing performance in advanced FPGA designs.

Synplicity offers various channels to help and support you in using Amplify software. The Amplify software installation includes extensive help documentation and tutorials. Amplify training is also available through an online, self-paced course – and through a one-day laboratory session that will give you a detailed, hands-on understanding of the full potential of Amplify software. To request Amplify training, send an e-mail message to *training@synplicity.com*.

To get the link for downloading the latest Amplify installation, contact your local Synplicity sales office. You must also send an e-mail request to *license@synplicity.com* to obtain an evaluation license.

For help with any questions about Amplify product usage, please contact Synplicity at 408-215-6000 or send an e-mail to *support@synplicity.com*.

Xilinx and Gibson Deliver the First True Digital Guitar



A music industry giant teams with Xilinx to create MaGIC.

by Xilinx Staff

Gibson Guitar and Xilinx recently announced a collaboration that resulted in the industry's first electric guitar to deliver true digital sound. Through their internally derived MaGIC (Media-accelerated Global Information Carrier) digital transfer protocol, Gibson developed a way to take the traditional analog output from the guitar and convert it into a digital signal, providing real-time high-fidelity digital audio to benefit both production and live performances. Gibson credits the reprogrammable Xilinx Spartan™-IIE FPGA as the enabling critical component in its groundbreaking guitar, and plans to use FPGA chips in a variety of MaGIC-enabled applications. The Xilinx Spartan-IIE FPGAs are the world's lowest cost programmable devices available today.

Gibson will offer MaGIC in every Gibson electric guitar within the next 12-18 months. MaGIC applies the digital technology invented for computer network products to audio networks. This requires adaptability to the MaGIC standard, made possible by using a programmable-versus-fixed logic solution.

The programmability of Xilinx FPGAs also provides Gibson with the ability to achieve its vision of licensing its technology to other music and consumer product manufacturers for future product development. Gibson hopes to achieve this vision by licensing MaGIC free of charge so that it will be embraced as the standard not just in the music industry, but in home networking, home automation, and medical imaging markets as well.

"Multiple uses of MaGIC would not have been financially or technically possible using traditional ASIC fixed logic. An ASIC platform would have required the design to be re-spun each time a change was made. The programmable nature of Xilinx FPGAs not only provided a flexible, high-performance design platform for Gibson, it also provided the low-cost silicon solution we needed to make it happen."

Henry Juszkiewicz – Gibson Chairman and CEO

About MaGIC

Despite dramatic advances in recent history, real-time high-fidelity digital audio has yet to penetrate both production and live performances. Increasing demand has motivated the effort to apply modern network technology toward producing superior quality real-time audio devices at low prices. MaGIC uses state-of-the-art technology to provide as many as 32 channels of 32-bit bidirectional high-fidelity audio with sample rates up to 192 KHz. Data and control can be transported 30 to 30,000 times faster than MIDI (musical instrument digital interface).

About Xilinx Spartan-IIE FPGAs

Since introducing the low-cost Spartan family more than four years ago, Xilinx has delivered four generations of devices, offering a low cost, programmable alternative to ASICs without NRE costs. The Spartan-IIE family is delivering the lowest system cost solution in the industry, and is the only true ASIC alternative FPGA solution available. For more information on Spartan-IIE FPGAs, visit *www.xilinx.com* and search Products.

About Gibson Guitar

Gibson, founded in 1894, continues to be one of the most highly respected names in musical instruments. Gibson guitars are fully created and assembled in the U.S. Headquartered in Nashville, Tennessee, Gibson Musical Instruments currently encompasses a large family of companies that make and sell the world's finest guitars, basses, banjos, mandolins, drums, keyboards, amplifiers, strings, and accessories. For more information on Gibson, please visit *www.gibson.com.*

Take the HP Route to Next-Generation Linux Workstations

A promising new architecture – developed jointly by Hewlett-Packard and Intel – offers dramatic performance gains using the Linux operating system.

by Sam Giovinazzi
North American EDA Segment Manager
Hewlett-Packard Corp.
Sam_Giovinazzi@hp.com

Performance-hungry workstation users have been eagerly awaiting the arrival of next-generation systems based on Intel® Itanium™ 2 processors. These systems, which use the EPIC (explicitly parallel instruction computing) philosophy, offer the promise of delivering dramatic performance and capacity gains over systems based on the earlier CISC (complex instruction set computing) and RISC (reduced instruction set computing) architectures.

Workstations based on Intel Itanium 2 processors are now available. These workstations are particularly well suited to complex analyses involving large data sets, including EDA (electronic design automation) simulation and verification.

Codeveloped by HP and Intel, the Intel Itanium architecture is emerging as a potential new standard for technical computing. Built-in design features provide exceptional performance for the complex computations of technical applications, 64-bit addressing, and the flexibility to support multiple operating systems – UNIX, Linux, and Windows.

The Intel Itanium architecture incorporates both hardware and software advances focused on enabling, enhancing, expressing, and exploiting parallelism by the hardware and software compilers. Some performance-enhancing aspects of the design philosophy include:

- Predication
- Speculation
- Software pipelining
- Rotating registers and other processing efficiencies
- Hardware enhancements, such as larger integer and floating point units.

The Intel Itanium architecture performs more instructions per machine cycle than conventional CISC or RISC architectures. This advance will yield outstanding performance gains well into the future, as competing architectures reach the point of diminishing returns.

Given its distinct advantages for technical computing, the Intel Itanium architecture has a great deal of momentum. More Itanium 2-based applications are becoming available, and many more are planned for release in the months ahead. Many forward-looking organizations are deploying or are actively evaluating Itanium 2-based workstations for scientific research and advanced engineering and design work. Among these organizations is the U.S. Department of Energy's Pacific Northwest National Laboratory, which is deploying an Itanium 2-based HP supercomputer running Linux.

At the same time, the Linux operating system has tremendous market momentum. Organizations running high-end computing applications, particularly EDA applications, are looking to Linux for the advantages of open-source code, including the ability to break away from the limitations of proprietary operating systems.

HP has brought together the 64-bit Linux operating system and Intel Itanium 2 processors in new workstations that meet the need for high-end computing on Linux. The result is a powerful computing combination particularly helpful for designers, engineers, and scientists, who require big memory and faster processing.

If this description fits your organization, the issue isn't so much a question of where you are going but how you are going to get there. The good news is, if you are planning to move to the Linux operating system and Itanium-based systems, the transition doesn't have to be turbulent. With the right strategy, your migration to 64-bit Linux on Itanium 2-based workstations can take place smoothly, in a manner that allows the OS transition to occur independently of the hardware transition.

Making the Transition

If the applications you need aren't yet available on 64-bit Linux and Itanium 2-based systems, you can begin your transition by using the best of what's available today – including IA-32 (Intel 32-bit architecture – Pentium 4, Xeon) and PA-RISC (precision architecture – reduced instruction set computing) systems.

Running Linux on IA-32 systems makes sense for work that is completed early in the design process and for smaller design tasks that don't require a large amount of memory. A 32-bit system can typically support up to 4 GB of memory. Moving small design tasks to 32-bit Linux leaves you positioned to make a smoother transition to 64-bit Linux on Itanium 2-based systems because you're staying on Linux all the way.

In general, IA-32 applications can be run unmodified on Itanium 2-based workstations. This is also true of IA-32 Linux applications, but with caveats: The 32-bit Linux applications will tend to run slowly on Itanium 2-based systems, and they can't take advantage of the extended capacity of a 64-bit architecture. Nevertheless, in transitioning to Itanium 2-based systems, it may be useful to deploy your IA-32 Linux applications on your Itanium 2-based systems first and later modify them to take advantage of the Intel Itanium architecture.

To get the full benefits of the Intel Itanium architecture, your IA-32 Linux applications should be compiled natively for Intel Itanium 2 systems. This is a two-step process. The 32-bit applications must first be converted to 64-bit and then recompiled for the Intel Itanium architecture. The 32-bit to 64-bit conversion process will typically include a significant amount of programming work, including code changes to address programming practices that worked on a 32-bit architecture but won't work on a 64-bit architecture. Once the code is 64-bit ready, IA-32 software can then be recompiled for the Intel Itanium architecture. Extracting maximum performance on Itanium 2-based systems is made easier by advanced compilers, which are designed to take maximum advantage of the Intel Itanium architecture.

Once your Linux applications are 64-bit ready, they should be capable of being compiled to run on either IA-32 or Itanium 2-based systems. This means there is no need to maintain separate 32-bit and 64-bit source code streams, because the same source code should work for both architectures. This principle has been tested widely in actual implementations. Today, Linux distributions include thousands of open-source packages that have a single set of source code for applications to be built and run on IA-32-based and Itanium-based workstations, as well as other architectures.

The HP-UX Gateway to Itanium-Based Systems

A good deal of higher-end design and engineering work requires far more memory than an IA-32 system can address. If you've run EDA simulation and verification applications on IA-32 systems, chances are you've run up against the architecture's typical 4 GB memory limitation. For work requiring more than 4 MB, heavier work with large data sets is ideally suited for PA-RISC systems and their higher memory capacities. A dual-processor HP-UX (Unix) PA-RISC workstation can hold up to 16 GB of memory.

The ability of the Intel Itanium architecture to work with multiple operating systems makes for an easy transition from HP-UX to HP's Itanium 2-based workstations. The Intel Itanium architecture already supports HP-UX, so applications running on 64-bit HP-UX 11 systems can easily be migrated to the Intel Itanium 2 platform.

If you are a current PA-RISC customer, you may already be using an operating system and hardware that is ready for the Intel Itanium 2 processor. The 64-bit HP-UX 11 operating system, designed to serve as a gateway to the Intel Itanium architecture, offers binary compatibility with Itanium-based systems. This makes it relatively straightforward to move an HP-UX 11 application from a PA-RISC workstation to an HP Itanium 2-based workstation.

HP-UX, used in concert with the Intel Itanium architecture, has an emulation mode that allows it to execute PA-RISC binaries – which means that HP-UX applications don't necessarily have to be recompiled to run on Itanium 2-based systems. Performance is better, however, if PA-RISC applications are recompiled for the Intel Itanium architecture. So, if top performance is essential, you will want to take this extra step. The process is fairly straightforward with HP-UX applications because you don't have to convert the source code to be 64-bit compliant – HP-UX supports both 32-bit and 64-bit programming models, which means your 32-bit applications are already 64-bit ready.

Migrating on Your Schedule

Once your HP-UX applications are running on the new Itanium 2-based system, you can either remain on HP-UX or, if your computing strategy calls for moving to Linux, you can migrate your applications to Linux when the time is right for your organization.

Because Linux for Itanium 2-based systems supports only 64-bit applications, migrating HP-UX applications to 64-bit Linux is more involved than the relatively easy task of moving HP-UX applications to Itanium 2-based systems. This means that (just as with 32-bit Linux applications) 32-bit HP-UX applications running on Itanium 2-based systems must first be converted to 64-bit, and then recompiled for Linux. But if this is your strategic direction, there's no urgency to make this transition. HP-UX applications will continue to give you all the benefits of Itanium-based systems until you are ready to port your applications to Linux.

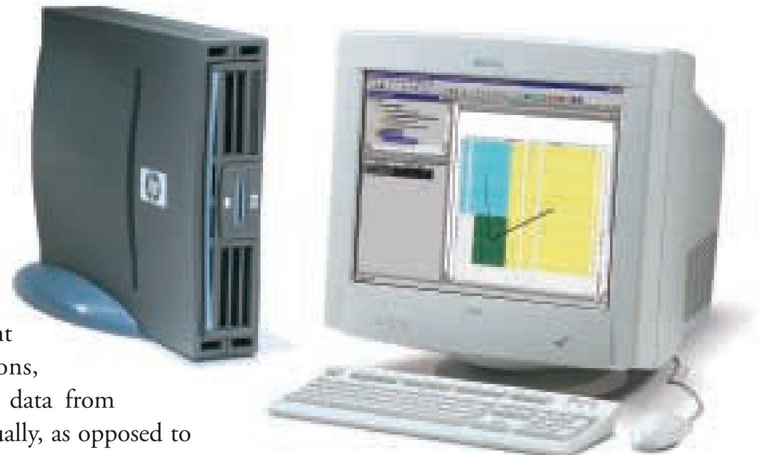
Further, HP-UX on Itanium 2-based systems supports a Linux ABI (application binary interface) that will allow you to run Linux Itanium applications under HP-UX – yet another path to the future.

You can follow any of these paths to transition your operating system to Linux, independent of your hardware transition to the Intel Itanium architecture. This gives you the best of all worlds – the use of Linux on lower-cost IA-32 systems for as long as they make sense; the proven performance of HP-UX for demanding analysis, engineering, and design work; and a clear path to Itanium-based systems. When you are satisfied that the applications you need are available on Itanium, you can begin your hardware transition.

Is HP the Right Choice For You?

HP's Itanium-based workstations take maximum advantage of the Intel Itanium architecture. In particular, the HP Chipset zx1 greatly extends the gains made possible by the Intel Itanium architecture. This high-bandwidth, low-latency chipset enables the Intel Itanium 2 processor better than any other system.

The HP Chipset zx1 is at the heart of the HP Workstation zx6000, the performance leader among 64-bit workstations. This one-way or two-way Itanium 2-based workstation surpasses all systems for floating point performance and surpasses all other 64-bit systems for integer performance. Its low latency is extremely important for EDA applications, which tend to access data from main memory continually, as opposed to using cache memory and data made ready by pre-fetch and branch prediction.



On the floating-point engine measure, the HP Workstation zx6000 achieved the world's fastest SPECfpbase_2000 result of 1,356, according to the Standard Performance Evaluation Corporation (SPEC). This score for the HP workstation is 13 percent higher than IBM's most powerful CPU, the Power4 at 1.3 GHz, with a SPECfp_base2000 score of 1,202. The HP workstation is also 1.9 times faster than the Sun Blade 2000 (UltraSPARC III 1050 MHz copper), with a SPECfp_base2000 score of 701. (For more detailed benchmark information, see *www.spec.org*.)

The HP Workstation zx6000 delivers the pinnacle of workstation 64-bit performance for scientists, engineers, designers, and others running memory-hungry applications. It can be equipped with up to two 1 GHz Intel Itanium 2 processors loaded with 3 MB of on-chip L3 cache and as much as 12 GB of RAM, increasing to 24 GB when 2 GB DIMMs become available.

At the same time, the HP zx6000 is flexible. In addition to providing a choice of 64-bit Linux, HP-UX, or Windows, it can be deployed as part of a racked computing solution or as a single-user system. Moreover, its use model can change over time. An HP zx6000 might be deployed initially in a cluster node running Linux and later redeployed at the deskside running Windows. In racked implementations, the HP zx6000 offers extraordinary compute density – 20 workstations can be placed in a 2-meter rack for an astounding 160 GFLOPS (GigaFLOPS) of potential power.

Conclusion

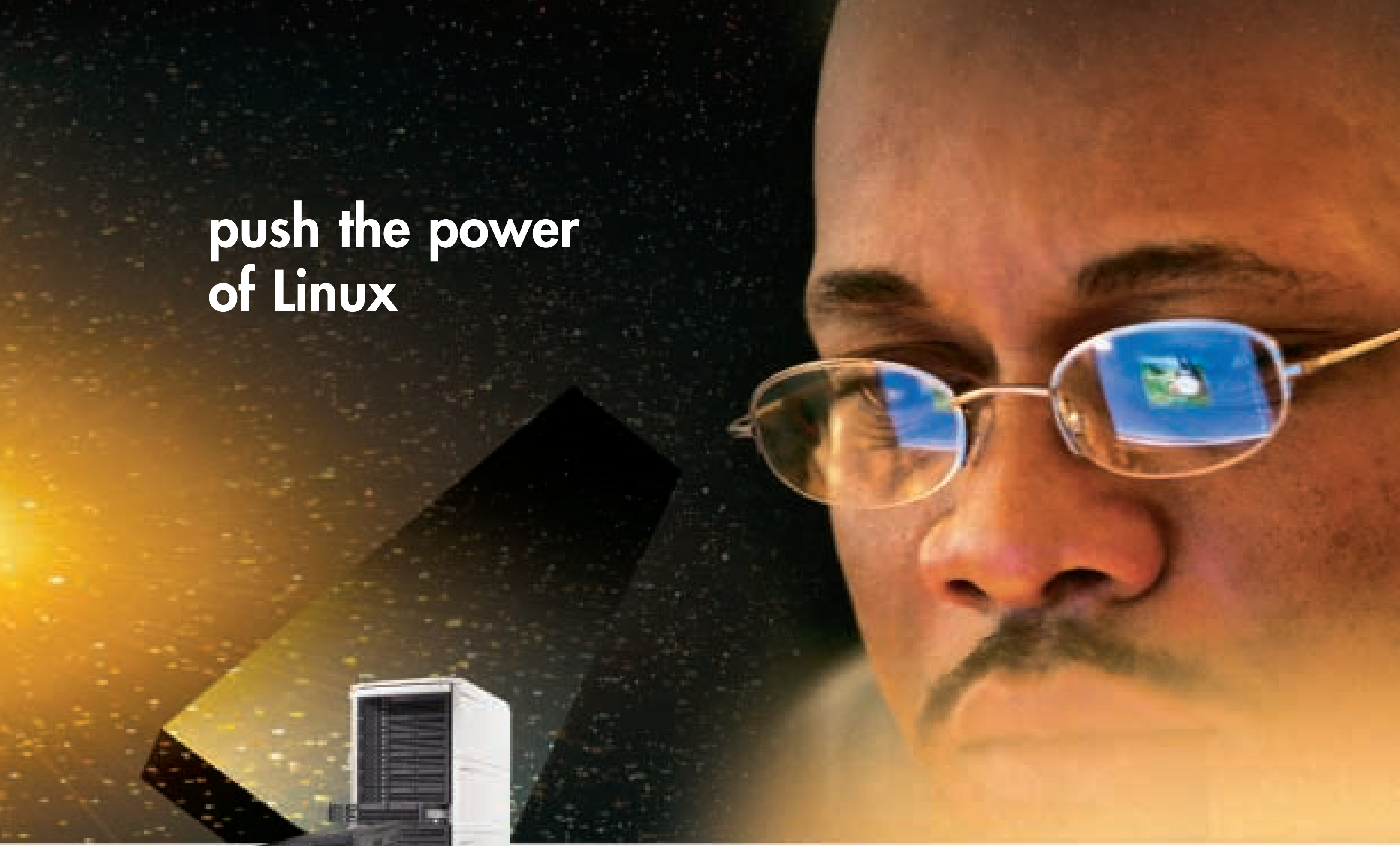
Factors like those mentioned above make the HP zx6000 workstation a powerful choice for electronic design, computer-aided engineering, scientific research, life sciences, and digital content creation and rendering. It also provides an ideal software development platform for Symmetric MultiProcessor-capable code.

These same factors help make HP an optimal choice for organizations transitioning to Itanium 2-based workstations and the Linux operating system for EDA work. With support for 64-bit Linux, the fastest floating-point performance, and the lowest-cost big-memory solution, HP's Itanium 2-based workstations offer clear advantages for EDA customers.

If you are constrained by memory limits or you need exceptional price performance for 64-bit computing on Linux, HP has a solution designed for you – and a clear strategy for getting you there. And to enable a smooth transition, HP offers a full suite of services spanning your planning, porting and migration, support, and education needs.

To discuss your specific needs and transition issues, contact your HP sales representative. To find an HP sales representative online, visit *www.hp.com/go/workstationrep*.

To learn more about HP Workstations, including the Itanium 2-based systems, visit *www.hp.com/go/workstations*.



the new performance leader for Linux has arrived

the world's fastest floating point performance
the world's lowest-cost, big-memory solution
the capacity of 64-bit Linux

Don't think it's all down the road. It's here today in the HP Workstation zx6000, the new performance leader for Linux. The HP zx6000 features dual Intel® Itanium® 2 processors, the performance-leading HP Chipset zx1, up to 12GB of memory and other advanced features—such as built-in Gig-E and enablement of 64-bit Red Hat Linux Advanced Workstation 2.1, HP-UX or Windows®.

If you're looking to really push the power of Linux, the HP Workstation zx6000 and the single-processor HP Workstation zx2000 are ready to work with you. They break through today's computing barriers—and do it at price points everyone can afford.

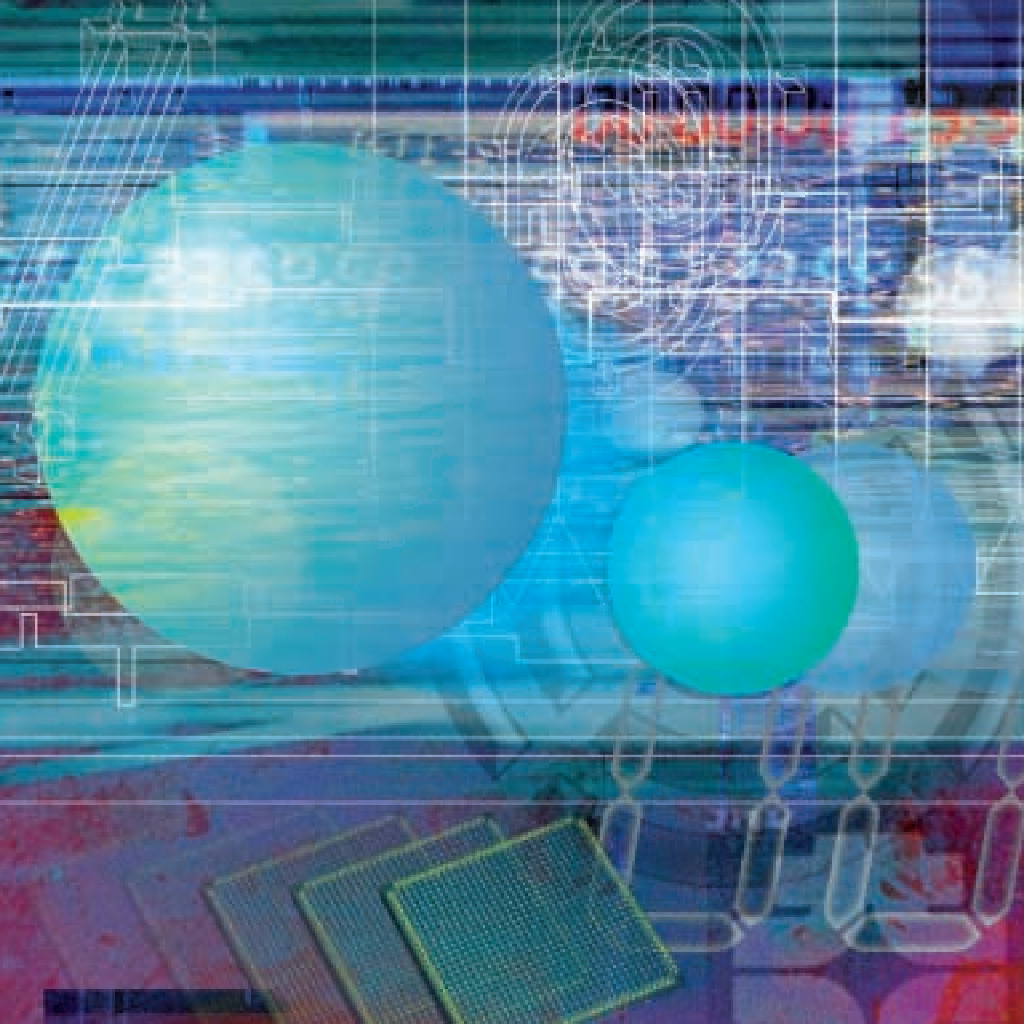
To check out the performance leadership of the HP Workstation zx6000, visit www.spec.org.

www.hp.com/go/workstations

hp® invent

500+ Xilinx FPGAs Search for Elusive Higgs Boson at 1.5 Terabytes per Second

With an array of more than 500 Virtex and Spartan FPGAs processing 1.5 terabytes of real-time data per second, scientists at the Fermi National Accelerator Laboratory hope to track down the last subatomic particle – the Higgs boson.



WE FIND OURSELVES IN A BEWILDERING WORLD. WE WANT TO MAKE SENSE OF WHAT WE SEE AROUND US AND ASK: WHAT IS THE NATURE OF THE UNIVERSE?

– STEPHEN W. HAWKING, LUCASIAN PROFESSOR OF MATHEMATICS AT CAMBRIDGE UNIVERSITY

by Mark Havener
Science Writer, Software Consultant
havener@inreach.com

Nick Hartl
Gold FAE, Avnet Design Services
nick.hartl@avnet.com

Inside the four-mile long Tevatron, the world's most powerful particle accelerator, protons and antiprotons collide at nearly the speed of light, creating bursts of energy and showers of millions of subatomic particles. If theoretical predictions are correct, over the next five years a million billion collisions (10^{15}) will produce only 120 events with the characteristic pattern most easily recognizable as evidence of the existence of Higgs boson.

Discovery of the Higgs boson will verify the "Standard Model" theory that is the foundation of modern particle physics. Finding a Higgs boson needle in this haystack of particles, however, requires a digital signal processing (DSP) system capable of gathering and processing 1.5 terabytes of data per second.

The scientists at Fermilab (*www.fnal.gov*) in Batavia, Illinois, investigated multiple ways of capturing this enormous data flow before finally settling on an array of more than 500 Xilinx Virtex™ and Spartan™ FPGAs. The Xilinx components are assembled into a "trigger" – a homemade, massively parallel supercomputer programmed as a multi-level pattern recognition filter. Tracks left behind charged particles are examined, and complex algorithms recognize and discard known patterns. Data about unknown particles are passed on and stored for later analyses. These data could prove that the Higgs boson exists. If it does, the proof will not only extend our understanding of the universe, but it may also earn a Nobel Prize for the physicists at Fermilab.

The Standard Model and the Higgs Boson

Modern theoretical physics describes the world as composed of twelve fundamental matter particles in three generations. First-generation particles are stable and can easily be found in nature, while second- and third-generation particles are extremely unstable and exist for only a tiny fraction of a second before decaying into other particles. Force-carrying particles interact with the matter particles. These particles and their interactions make up the Standard Model of Fundamental Particles and Interactions (Figure 1). (For more information on the Standard Model, see "The Building Blocks of Matter," *www.fnal.gov/pub/inquiring/matter/madeof/*, and "The Particle Adventure: Fundamentals of Matter and Force," *particleadventure.org/particleadventure/*.)

The four known fundamental force-carrying particles are photons, W and Z bosons, and gluons. First-generation matter particles include up quarks, down quarks, and electrons. Two down quarks and one up quark form a neutron; two up quarks and one down quark form a proton. Protons, neutrons, and electrons combine to form atoms, atoms combine to form molecules, and molecules combine to form The World As We Know It. All of this is supported by physical evidence gathered from experiments.

Experimental measurements show that most fundamental particles have a very small mass, typically 1 giga electron volt (GeV/c^2) or less. An electron has a mass of 0.511 mega electron volt (MeV/c^2), or 9.11 x 10^{-31} kilogram. The photon mass is theoretically zero. Indeed, experimental evidence indicates that the photon mass can't be greater than 10^{-33} GeV/c^2, in excellent agreement with theoretical prediction. However, the W boson and Z boson masses have been measured as 80.4 GeV/c^2 and 91.187 GeV/c^2, respectively.

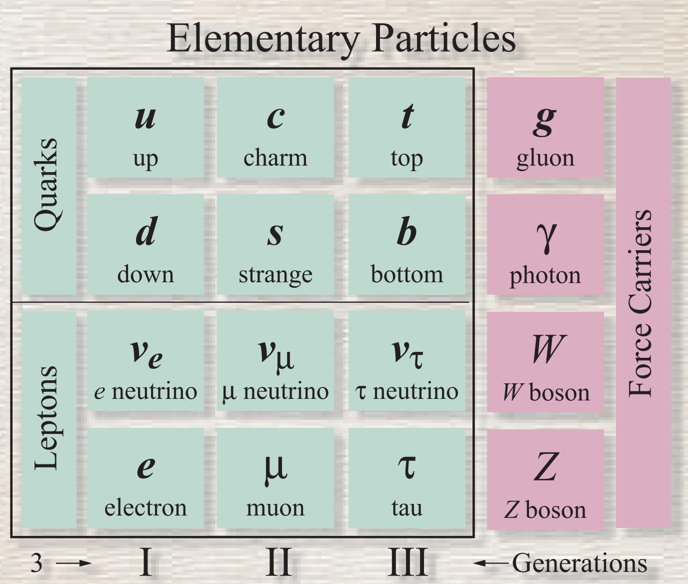


Figure 1 - The Standard Model of Fundamental Particles

Courtesy of Fermilab

The large masses of these bosons, and their observed interactions with known elementary particles, create a curious inconsistency in the mathematical equations that describe the behavior of matter and force. The equations predict the probability of two very high-energy particles colliding is greater than one. It's like knowing you'll always win the lottery, and you won't even have to buy a ticket. This would be nice, but it's impossible.

One way to resolve this theoretical dilemma is to introduce additional particles. In 1964, British physicist Peter Higgs postulated the existence of an invisible field that permeates the universe and is responsible for endowing all matter with mass. (To find out more on the Higgs theory, see "The Higgs Boson," *www.jlab.org/~cecire/higgs.html*.) According to theory, when a subatomic particle, such as an electron or quark, moves through the Higgs field, the particle acquires mass. The existence of a fundamental force-carrying particle – the Higgs boson – supports the simplest theory that would explain the large masses of the W and Z bosons. The Higgs boson exists as both a field and particle, because matter and force exist as both fields and particles, according to Quantum Theory.

In 1971, Glashow, Salam, and Weinberg included an *ad hoc* Higgs mechanism in calculations that predicted the massive W and Z bosons. These predictions were beautifully confirmed a decade later with their discovery by Carlo Rubbia's group of experimenters at CERN (European Organization for Nuclear Research, *public.web.cern.ch/Public/*). The prediction and discovery led to the award of three Nobel Prizes.

If the Standard Model is correct, high-energy collisions in the Tevatron will produce Higgs bosons. Each Higgs boson will exist for only a fraction of a second before decaying, but measurement of the angle and velocity of the resulting decay particles will provide proof of its existence. Discovery of the Higgs boson will provide additional confirmation of the Standard Model and expand physicists' understanding of mass.

Subatomic Collisions

When protons and antiprotons collide, force particles and unstable second- and third-generation matter particles are created. Because these particles are so short-lived, the only evidence of their existence is the tracks they leave as they decay, as well as the tracks left by the other particles created in the decay process. The Fermilab

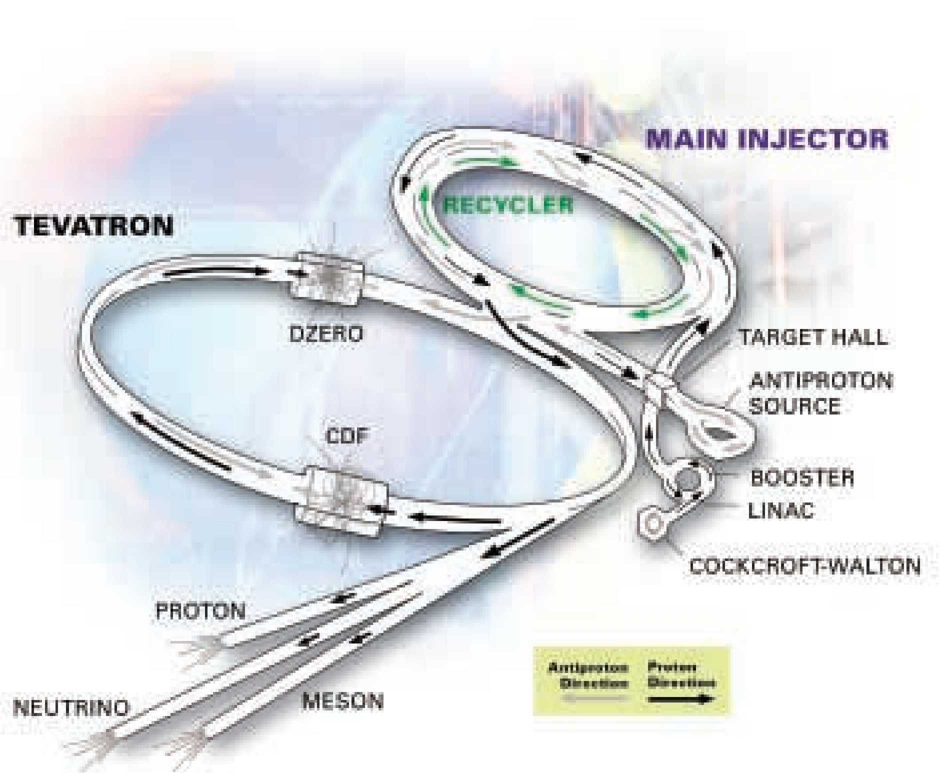


Figure 2 - The Fermilab Accelerator Chain *Courtesy of Fermilab.*

physicists use the detectors inside the Tevatron to observe these tracks and the FPGA array to analyze them for proof of the existence of these ephemeral particles.

To get a sense of how difficult this is, imagine a child's Hula Hoop® toy suspended at the 50-yard line of a 100-yard American football field with a machine gun in each end zone. The machine guns aim for the center of the Hula Hoop and fire as fast as they can. Sometimes the bullets collide, sometimes they don't. Some collisions are head-on, and some are indirect glancing blows. Your job is to measure the direction and velocity of the bullet fragments after each collision and then use that data to analyze and re-create the collision.

Just as the imaginary machine-gun bullet collisions are not all identical, the proton and antiproton collisions inside the Tevatron are not all identical. Some collisions are direct, some glancing. The protons and antiprotons travel at slightly different velocities and orientations. The types of particles created – and the directions and velocities in which they are scattered – depend on many factors, with each collision producing a different and distinct "signatures" of particles.

Out of the 10^{15} proton-antiproton collisions expected in Run II of the Tevatron, a very small fraction will result in a top quark and top antiquark meeting head-on. In theory, quark-antiquark collisions can produce a Higgs boson in three distinct ways, each with a distinct particle signature. The current estimate is that all these collisions will produce less than 20,000 Higgs particles – one Higgs boson for every 50 billion collisions. And only 120 of the collisions will yield the characteristic pattern most easily recognizable as Higgs production.

The Tevatron

The Tevatron is the world's most powerful particle accelerator (Figure 2). It uses oscillating magnetic fields to push protons and antiprotons in opposite directions, reaching nearly the speed of light on the four-mile circular path before colliding in one of the two detectors (CDF and DZero).

The process begins with the ionization of hydrogen atoms, creating two electrons and one proton. These particles are accelerated to an energy of 400 MeV and passed through a carbon foil filter that removes the electrons. The protons are further accelerated to 8 GeV and sent into the Main Injector, where they are yet further accelerated to 120 GeV before some of them are siphoned off and crashed into a fixed nickel target. These collisions produce secondary particles, most of which are ignored and discarded – with the exception of the antiprotons, which are collected and sent back to the Main Injector. About half the size of the Main Accelerator, the Main Injector increases the energy of both the protons and antiprotons to 150 GeV before injecting them into the Main Accelerator.

Inside the Main Accelerator, protons and antiprotons are accelerated with powerful electromagnetic fields, using harmonic oscillation at gigahertz frequencies. As the particles reach higher speeds, additional magnetic force is used to bend the beams into a circular path. Protons travel clockwise and antiprotons counterclockwise, faster and faster, to within 200 miles an hour of the speed of light. At this speed, the energy of the particles approaches a thousand billion electron volts, or one tera electron volt (1 TeV) – and this is where the Tevatron gets its name. The beams are slightly offset from each other, crossing at two points. High energy collisions occur at these intersections, where the detectors are located.

The DZero Detector (*www-d0.fnal.gov*) uses Silicon Microstrip Tracker (SMT) and Central Fiber Tracker (CFT) subdetectors to record tracks of charged particles produced in the collisions. The CFT is made of scintillating fibers mounted on eight concentric cylinders. A charged particle passing through the fiber produces a tiny amount of light that is converted into an electrical pulse by visible light photon counters. These are small silicon devices with an array of eight photo sensitive areas, each 1 mm in diameter. Recorded electric signals make it possible to reconstruct an accurate three-dimensional image of the particle's path.

The DZero Upgrade

The Tevatron collider began operating in 1983, with continuing improvements and additions over the next 14 years. The original DZero (a.k.a. DØ) Detector was commissioned on Valentine's Day 1992. By any measure, Run I was a successful experiment, monitoring a few trillion collisions and culminating with the discovery of the top quark in 1995.

In trillions of collisions, the physicists observed only 90 top quark events – events with a signature similar to what the Standard Model would predict if a top and an antitop quark were produced in the collision. Higgs candidates are even more elusive than top quarks, with an expected production of one Higgs boson in every 50 billion collisions. To find a Higgs, many more collisions will be needed. In 1997, the Tevatron was shut down for final installation of the Main Injector and Antiproton Recycler to increase the luminosity of the beams, and for improvements to the particle detectors to monitor the additional collisions.

The improved DZero Detector (Figure 3) was initially designed using commercially available DSP components. Computer models were built, with simulations

Figure 3 - DZero, side view

Courtesy of Fermilab

designed and executed to verify the design's functionality. Running the simulator revealed that the DSP design was wholly inadequate to handle the number of collisions that the increased luminosity would produce. A new approach was needed ... and the final design of the new DZero Trigger relies heavily on Xilinx FPGAs.

The DZero Trigger

The DZero Trigger is a multi-level pattern recognition filter that processes nearly a trillion signals every second. Its job is to identify the collisions that are most likely to produce Higgs particles, and save that data for later detailed analyses. When a charged particle passes through the CFT, the light from the fibers is first converted to an electric signal. Next, the digital signal is sent to a DZero Trigger subsystem called the Central Track Trigger (CTT), which progressively filters signals (Figure 4 - *d0server1.fnal.gov/projects/VHDL/General/ctt-diagram.pdf*). Each collision creates multiple particles and each particle creates multiple signals. From the resulting signals, it is possible to reconstruct the paths of the particles involved in the collision. Complex algorithms in the DZero Trigger identify and separate "interesting" signals from "uninteresting" ones.

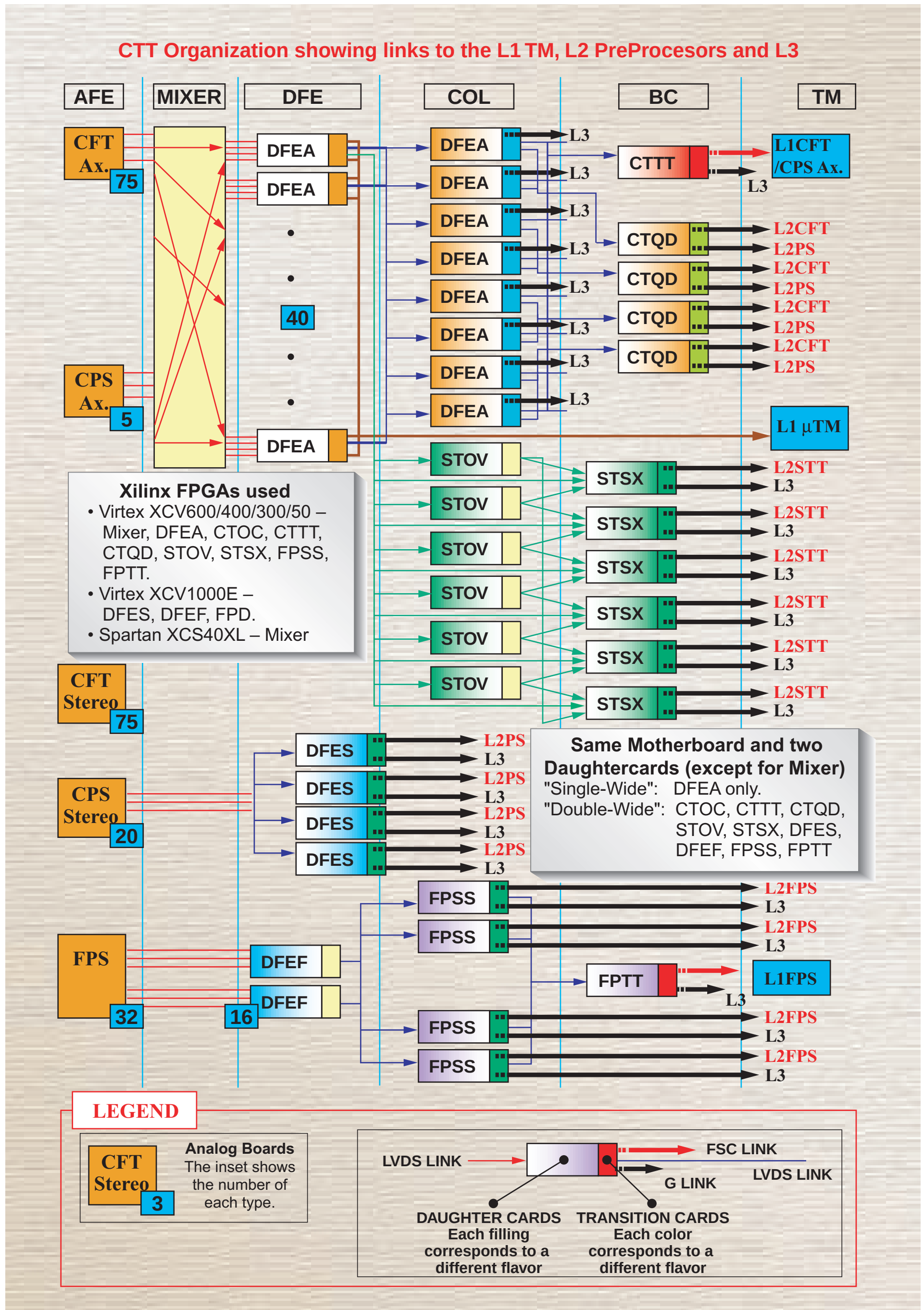


Figure 4 - The DZero CTT FPGA Array

Courtesy of Fermilab

Each second the DZero Trigger must look at 7 million collisions and decide in real time which ones to save. Only a fraction of the collisions can be saved, so recognizing and saving the ones that are most likely to show important events (like the production of a Higgs boson) is the key to success.

The trigger has three separate levels called L1, L2, and L3. Each level has a progressively finer filter. The data output rate at each level is lower than the data input rate. The difference between the rates determines how much data is rejected.

The DZero FPGA Array

The complete trigger consists of 582 Xilinx FPGAs ranging from Spartan FPGAs to Virtex 300s to Virtex-E 1000s. The 582 FPGAs are assembled into 21 unique designs that are repeated to make multiple data channels. The common footprint of the Virtex family allowed a design utilizing a single printed circuit board that could be populated with different chips. This was a great advantage, because the single common board could be customized by placing different numbers and sizes of FPGAs on it to create the various subsystems used in the trigger.

Once the hardware design was completed, the next major challenge was programming the chips. The function of the DZero Detector and the data it produces had to be fully understood and incorporated into an algorithm that would save the correct data. The most difficult task was creating an algorithm that operated in the minimum amount of time. Because data cannot be discarded until the system reaches a save/don't save decision, and because there is a finite data buffer, it is important that the calculations be completed before the buffer is overwritten. Completing this task in the limited time available proved to be very challenging.

Choosing the Right FPGAs

After careful consideration, the DZero team chose Xilinx FPGAs. As Jamison Olsen, principal EE on the project, explained: "The common footprints used for the Virtex family allowed us to lay out one board that could be populated by a variety of different size chips with no change to the board. This was a great advantage, because we could design a common printed circuit card that we could customize by placing different numbers and sizes of FPGA to create the various subsystems used in the trigger."

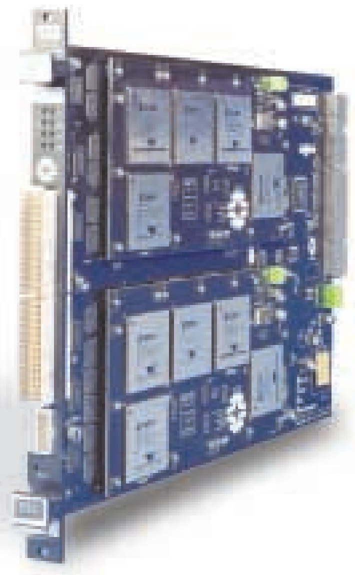


Figure 5 - Single-wide daughtercard

Courtesy of Fermilab

Olsen continued, "Other considerations that led us to Xilinx were very fast fitting of the devices, a good price-to-performance ratio, and several Virtex features, including the flexible RAM architectures."

Common Footprint

The DZero Trigger system architecture uses a base carrier card to take care of backplane I/O and to carry one or two daughtercards with the Xilinx FPGAs that do the actual work (Figures 5 and 6). This architecture allowed the team to build a variety of subsystems on common hardware. The three different tiers of processing within CTT have different processing requirements, so the CTT was built with the appropriate components at each level. Because the Xilinx components all share a common footprint, the base printed circuit boards can all be identical, providing both an initial cost savings and a much more efficient store of replacement parts.

The common footprint also provides the ability to boost performance by reconfiguring with more powerful devices as they become available during Run II.

Memory

The Xilinx components have more on-board RAM than competing devices. The Virtex-E FPGAs have as much as 1 Mb of internal configurable distributed RAM and up to 832 Kb of synchronous internal block RAM. Data cannot be discarded until the trigger system reaches a decision whether or not to save it, so the generous RAM provides a buffer to store the data while the system completes its calculations. Even with this much RAM, completing the calculations in as few clock cycles as possible before the buffer was overridden was a challenging task.

The Virtex flexible RAM architecture also came into play. The hierarchical memory system LUTs are configurable as 16-bit RAM, 32-bit RAM, 16-bit dual ported RAM, or 16-bit shift register, with fast interfaces to external high-performance RAMs.

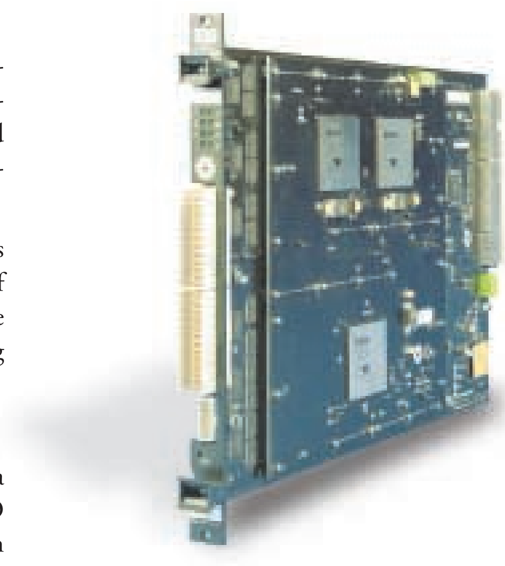


Figure 6 - Double-wide daughtercard

Courtesy of Fermilab

Performance and Bandwidth

In the development phase, the scientists decided the initial DSP design was unacceptably slow. They selected Xilinx components because the FPGAs were faster. The Virtex FPGAs operated at system speeds as fast as 200 MHz, and the Virtex-E parts achieved more than 311 MHz. The DZero processing is extremely I/O bound. With more than 1.5 trillion events per second, the amount of data flowing into the array is staggering. In the Virtex-E family, I/O performance in each component is 622 Mb/s using source-synchronous data transmission architectures.

Configurability

Xilinx FPGAs provided design flexibility that allowed the scientists to connect all the data paths early in the design – and figure out what to do with the data later. The physicists were able to implement their original algorithms and begin the experiments – and if necessary, they will be able to reconfigure and upgrade the FPGAs during the course of Run II.

Tool Set

The high-level software tools available allowed the DZero team to go from knowing nothing about programming FPGAs to building some of the most sophisticated DSP devices in the world. Their learning curve included understanding a new computer language and mastering all the new tools that go with it. In less than a year, the physicists and engineers were able to program in VHDL, adopt the tools, and use them at the level of very experienced digital designers.

"While our people are very talented, I think the fact that we were able to learn and master the new language, tools, and art of high-level digital design so quickly also speaks very well about the ease of use of Xilinx development tools," said Levan Babukhadia, who led the team in developing the VHDL firmware. The DZero team used various releases of Xilinx ISE, as well as Aldec Active-HDL™, Synopsys FPGA Express™ (Xilinx Edition), and Synplicity software.

Vendor Support

Avnet Design Services was able to provide all necessary training. Nick Hartl, an ADS Gold FAE, taught part of an intense five-day introduction to VHDL and Active HDL. He also arranged for two days of Aldec instruction. For many of the physicists, this was their first exposure to digital design. Additionally, Hartl worked with Fermilab on product selection, and he provided consulting and information on core integration, design optimization, and system-level architecture choices.

Conclusion

The DZero team has built an ultra high-bandwidth real-time supercomputer out of off-the-shelf Xilinx components to search for the Higgs boson. As powerful as this system is, it still is not able to monitor every collision and record every event. Within two years, the Fermilab Tevatron is going to ramp up its luminosity to a higher level. The ramp up will require refinements to the track finding and other algorithms – and much more powerful FPGAs.

Certain parts of the algorithms are easiest to implement in software, yet the team cannot afford to give up the raw power of parallel processing in FPGA hardware. The natural next step appears to be to marry the software and hardware by migrating to Xilinx Platform FPGAs, such as the Xilinx Virtex-II Pro™ series. Virtex-II Pro Platform FPGAs offer as many as four embedded IBM PowerPC™ 405 cores – and as many as 10 million system gates.

Meanwhile, the Fermilab scientists will continue to refine and improve their equipment, looking not only for the Higgs boson, but also searching for supersymmetry, extra dimensions, and other new phenomena. Babukhadia concluded, "We are on the way to exciting physics, with the first results coming soon, and exciting years ahead!"

The ultimate goal of the DZero team might best be described with another quote from Stephen Hawking: ***Ever since the dawn of civilization, people have not been content to see events as unconnected and inexplicable. They have craved an understanding of the underlying order in the world. Today we still yearn to know why we are here and where we came from. Humanity's deepest desire for knowledge is justification enough for our continuing quest. And our goal is nothing less than a complete description of the universe we live in.***

Glossary

- **Energy:** Since energies in the world of elementary particles are so tiny compared to our everyday, macroscopic experience, they are typically given in units of electron volts (eV). One eV is the amount of energy one electron would acquire having passed through a +1 Volt potential difference. Or perhaps in more familiar energy units of food ratings, it is equal to about 3.8×10^{-23} (food) calories. Because particle accelerators collide beams of particles of very high energies, these energies are usually given in billions of electron volts, or GeV.
  - MeV - million electron volts (mega electron volts)
  - GeV - billion electron volts (giga electron volts)
  - TeV - trillion electron volts (tera electron volts)
- **Mass:** Owing to Einstein's celebrated relation $E=mc^2$, describing the equivalence of mass and energy, mass of fundamental particles is typically given in units of energy. A convenient unit turns out to be GeV/c^2, or billions of electron volts divided by the speed of light (in a vacuum) squared. For example, in these units the proton mass is approximately 1 GeV/c^2 or, equivalently, about 1.78×10^{-27} kilograms. With the speed of light further set to unity, mass is often given simply in units of GeV.
- **Luminosity:** This is the "brightness" of the particle beam. Measured in particles per square centimeter per second, luminosity determines how many collisions can occur. The higher the luminosity, the higher the collision rate.

Develop Professional Digital Video Applications

Xilinx FPGAs, tools, and design support will enable you to be first to market and best in market.

by John F. Snow
Staff Applications Engineer
Xilinx, Inc.
john.snow@xilinx.com

Digital video is rapidly replacing the traditional analog video signal throughout the video broadcast chain. From the content source in the studio or remote news site, through the editing, storage, and transmission processes, to the set-top boxes and digital television sets in consumers' homes, digital video is now firmly established throughout the broadcast industry.

Many digital video standards remain subject to change and refinement, however. Others are still going through the standardization process. In this uncertain period, it is both difficult and expensive for suppliers of professional digital video equipment to stay abreast of new developments. Digital video equipment designs require flexibility to meet current standards and to adapt to emerging standards, even after deployment in the field. Xilinx FPGAs and Internet Reconfigurable Logic (IRL™) technology provide this flexibility, allowing you to update your equipment rapidly as standards evolve.

A variety of resources are available to help developers use Xilinx FPGAs in professional digital video applications. This article describes some of the tools and design support available from Xilinx to help you gain a competitive edge in the digital video equipment market.

Hardware Aids

The Xilinx MicroBlaze™ and Multimedia Development Board is available from Xilinx distributors to help you develop and test digital video algorithms, including video format conversion, video compression, and image processing. The board accepts an analog composite video input, decodes it into digital component video, and processes the digital video in a Virtex™-II FPGA. The processed video is then converted back to analog and is available on the board's outputs as composite, S-video, RGB, or SVGA video.

Reference Designs and App Notes

Xilinx has created a number of reference designs and application notes for professional digital video applications. Many of these reference designs are tailored to run on the MicroBlaze and Multimedia Development Board, but are also applicable to real-world digital video applications. The reference designs and application notes focus on three different areas of professional digital video applications: scan-line processing, serial digital interface, and video compression.

Scan Line Processing

The scan line processing section of the MicroBlaze and Multimedia Development Board is illustrated in the block diagram shown in Figure 1; relevant "XAPP" application notes are also displayed.

This scan line processing section accepts NTSC (National Television Standards Committee) or PAL (phase alternating line) composite analog video and converts

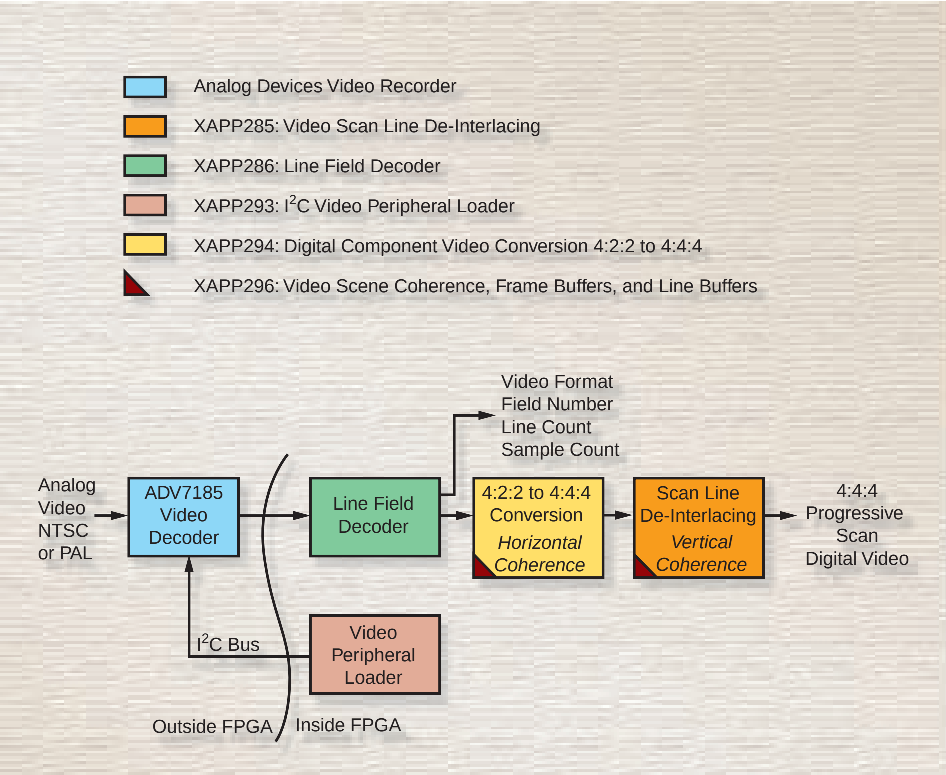


Figure 1 - Scan line processing section

it to progressive-scan, 4:4:4 component digital video in a series of steps. An Analog Devices ADV7185 video decoder converts the analog video to digital video. The other conversion steps are implemented in the Virtex-II FPGA.

The line field decoder on the development board examines the digital video and determines the video format (NTSC or PAL). It also synchronizes to the digital video stream, providing the current video line and sample counts to the other video processing blocks.

The ADV7185 video decoder generates interlaced component digital video having chroma components at half the horizontal resolution of the luma component. This is 4:2:2 component video. The 4:2:2 to 4:4:4 conversion block converts the video from the decoder to 4:4:4 component video having equal resolution of chroma and luma components. As a last step, the video is de-interlaced to create a progressive-scan video signal.

The video peripheral loader initializes the video encoder and decoder chips on the development board using the I²C bus.

Future application notes will describe the processing of the progressive 4:4:4 video generated by the scan line processing section in a frame buffer environment. Some of these frame-oriented functions are 2D image scaling, image enhancement, and noise removal.

Serial Digital Interface

The ANSI/SMPTE 259M-1997 standard specifies how to transport digital video serially over video coax cable. This standard, commonly called SDI (serial digital interface), is now widely used to distribute digital video throughout television studios and video production centers over the video coax cable previously used to transport analog video.

Figure 2 shows a block diagram of a typical SDI video link, along with a list of pertinent application notes. Ancillary data, such as digital audio, is inserted into the inactive portions of the digital video stream. Error detection handling (EDH) packets are calculated and inserted. The digital video is then encoded, serialized, and transmitted through the coax cable. At the receiving end, the data and clock are recovered from the serial bitstream and the bitstream is decoded, framed, and de-serialized. Finally, a processor implements error detection and extracts the ancillary data from the digital video data.

The digital video test pattern generator creates pathological test cases designed to stress the equalization and clock-and-data recovery units in an SDI receiver. The XAPP248 application note on SDI also includes reference designs to generate industry standard color bar video test patterns.

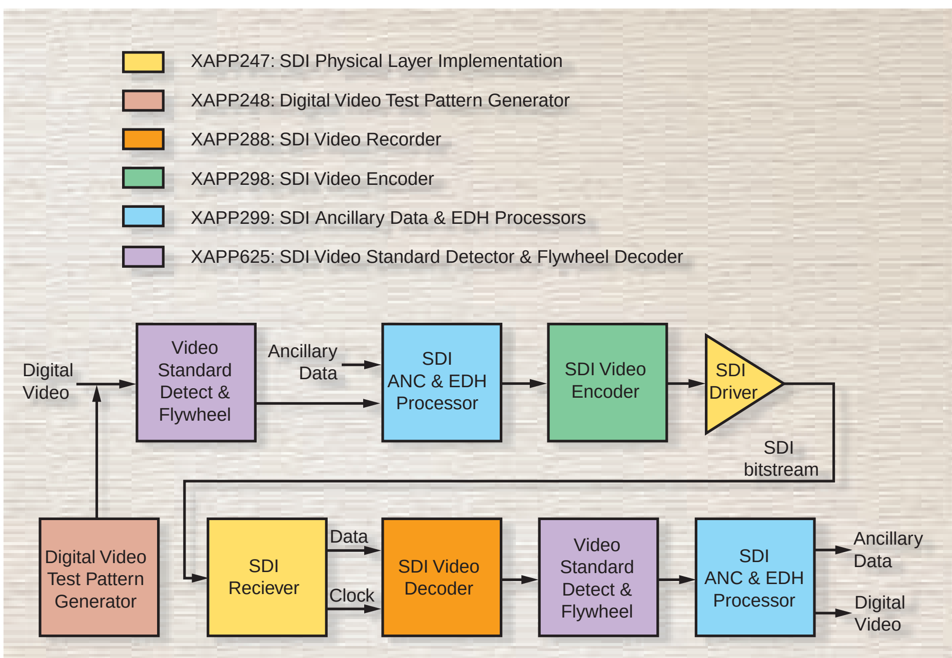


Figure 2 - Serial digital interface section

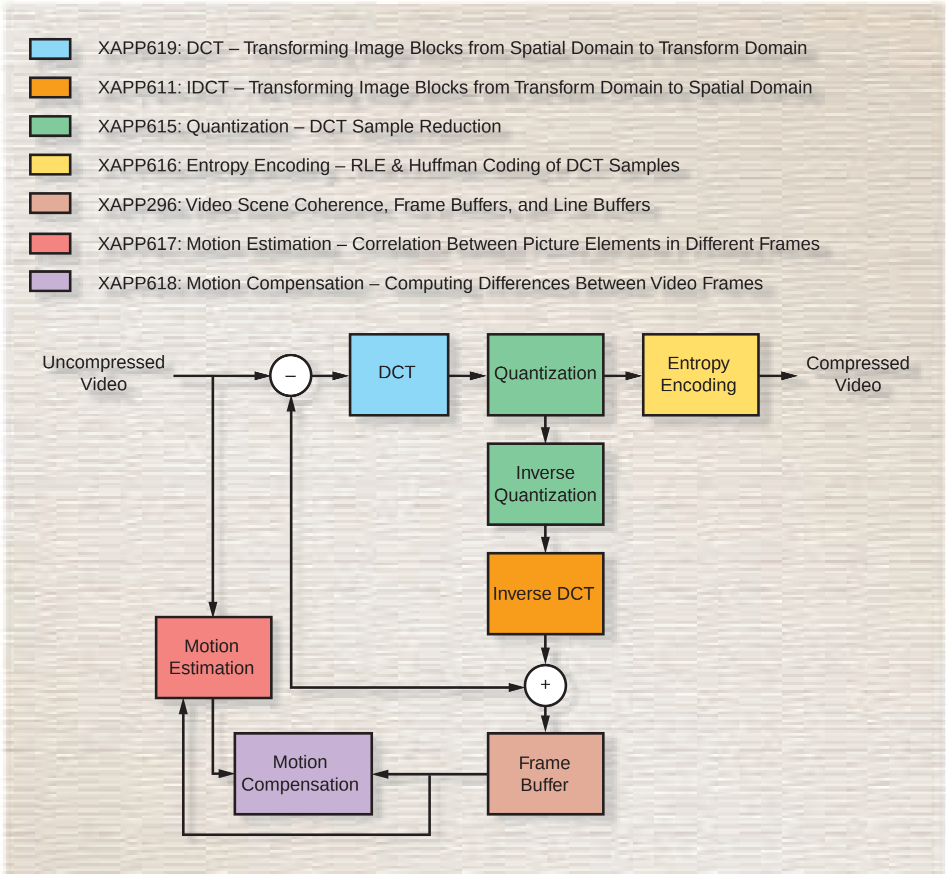


Figure 3 - Typical video compression system

Video Compression

Digital video compression and decompression are an integral part of most professional digital video systems. As shown in Figure 3, a set of application notes is available that describes some of the fundamental building blocks used in many video compression standards, including MPEG-2. Figure 3 also shows how these functions are used in a typical video compression system.

The discrete cosine transform (DCT) function reduces an image block into spatial frequency components. This transformation sorts the information in the image block, separating the higher frequency components from the lower frequency components. With the image de-composed in this manner, it is possible for the compression scheme to take advantage of the human visual system's lower sensitivity to the higher frequency components of the image.

After transformation by the DCT, quantization compresses the higher frequency components of the image more than the lower frequency components. The lower frequency components are quantized in small steps while the high frequency components are quantized in larger steps or are altogether discarded and converted to zeros.

Entropy encoding further compresses the quantized data by run-length encoding into short codewords. Variable length coding is also used to assign shorter codewords to commonly occurring data sequences and longer codewords to infrequent data sequences.

Additional significant compression of video images is achieved by taking advantage of the temporal coherence of the image. In most video images, a frame of video usually only has minor differences from the previous frame. Many video compression schemes take advantage of temporal coherence by periodically transmitting a full reference frame and then sending only the arithmetic differences for successive frames. Motion estimation identifies portions of the image that have moved from the previous video frame. Motion compensation generates the arithmetic differences between the frames based on the motion vectors found by motion estimation.

A good reference book on video compression is *Image and Video Compression: Algorithms and Architectures – Second Edition*, by Vasudev Bhaskaran and Konstantinos Konstantinides (1997, Kluwer Academic Publishers, ISBN:0792399528).

Conclusion

Xilinx has development tools and technical support to assist you in using a Xilinx FPGA as the video processing engine of a digital video application. Digital video reference designs and application notes from Xilinx provide the building blocks for profession digital video applications. Because the reference designs are supplied with complete source code, they can be combined and customized to suit the requirements of your specific application.

Using Xilinx components and reference designs will allow you to:

- Integrate a variety of video functions in one FPGA device as opposed to implementation in several separate ASICs.
- Customize video functions that previously were inaccessible inside ASICs.
- Update field equipment to new video standards by reconfiguring the FPGA via Xilinx IRL technology.
- Reduce development costs and shorten design cycles.

The application notes, reference designs, and FPGA product information are all available on the Xilinx website at *www.xilinx.com*. The MicroBlaze and Multimedia Development Board is available through Xilinx sales representatives and distributors, who can be found at *www.xilinx.com/company/contact.htm*.

XtremePMC

ADM-XPL

XILINX
RECONFIGURABLE COMPUTER

Benefits

Alpha Data's ADM-XRC family of PCI Mezzanine Cards make it easy for you to enjoy the benefits of Platform FPGA solutions. With up to 8 million gates, embedded PowerPC and flexible I/O options, the ADM-XRC family makes FPGA development a breeze.

A common software API makes it easy to take host applications from development environment to embedded solution. Complement this with ready-made FPGA applications providing hooks to the latest bring-up tools and you'll be up and running faster, accelerating productivity.

Using the ADM-XRC family speeds deployment of Platform FPGAs and takes the pain out of the development process enabling you to concentrate on what matters – the solution!

ADM-XRC family: the best development and run-time FPGA platform you can buy.

Features

- Industry standard Xilinx Virtex-II Pro based PMC
- XPL-DEV package available including Wind River visionPROBE/ICE-II, SingleStep and visionWARE
- High performance bus mastering 66/64 PCI interface
- Flexible front panel I/O options using Alpha Data XRM modules
- Interfaces include MGT, RapidIO, FPDP and LVDS
- ZBT SRAM, DDR SDRAM and flash memory
- Programmable clock generators
- Battery backup for DES bitstream encryption
- Adapters available for PCI, CompactPCI and VME
- Drivers for Windows, VxWorks and Linux
- Platform neutral API for easy migration
- Template designs included in Verilog, VHDL and Handel-C
- Support for Xilinx PAVE and ChipScope

ALPHA DATA

58 Timber Bush Edinburgh EH6 6QH UK
t: +44 131 555 0303 f: +44 131 555 0728
e: info@alphadata.co.uk w: www.alpha-data.com

226 Airport Parkway Suite 470 San Jose CA 95110
t: (408) 467 5076 f: (408) 436 5524
e: sales@alpha-data.com w: www.alpha-data.com

Create Real-World Designs with MicroBlaze Development Kits

Four new MicroBlaze Development Kits from Avnet Design Services provide processing power, memory, and high-speed I/Os inside FPGAs.

by Warren Miller
VP of Marketing
Avnet Design Services
warren.miller@avnet.com

With the introduction of the Xilinx MicroBlaze™ soft processor core, high-performance processing power has moved inside the FPGA itself – bringing new classes of applications and architectures within your reach. FPGAs have grown sufficiently in capacity and functionality to support complete platforms on a single chip. In addition to MicroBlaze processors, more memory and high speed I/Os can now be implemented on a single FPGA.

Designs such as sequential data processing algorithms, which previously involved large and complex VHDL or Verilog™ code, can now be implemented in a standard high-level language, such as C. In many cases, this results in quicker design time and lower gate counts.

Kits Are Feature Rich

Avnet Design Services has created a suite of MicroBlaze Development Kits that speed development of applications based on the MicroBlaze soft processor core. The kits include:

- MicroBlaze Development Environment
- Full-featured hardware development boards based on:
  - Virtex™-II FPGAs
  - Virtex-E FPGAs
  - Spartan™-IIE FPGAs
- A set of additional IP cores for popular MicroBlaze-compatible peripherals and memories.

The kits offer a complete set of hardware, software, and IP that will enable you to start building real-world applications in your target FPGA device without the need to create prototypes (Table 1).

Virtex-II Kit

The Virtex-II based development kit starts with a PCI/PCI-X form factor board (Figure 1) and contains 128 MB of 133

Part Number	Description	Price
ADS-XLX-MB-DEV1500	Virtex-II development board with XC2V1500, Communications/Memory board, and MicroBlaze IP Core License	$1,400
ADS-V2-MB-DEV4000XP	same as above with XC2V4000	$3,000
ADS-V2-MB-DEV6000XP	same as above with XC2V6000	$6,400
ADS-SP2E-MB-EVL	Spartan-IIE evaluation board with XC2S200E, Communications/Memory board, and MicroBlaze IP Core License	$650
ADS-VE-MB-DEV	Virtex-E development board with XCV1000E, Communications/Memory board, and MicroBlaze IP Core License	$1,650

Table 1 - MicroBlaze Development Kits – price and availability

MHz Micron DDR (double data rate) SDRAM in a SODIMM (small outline dual in-line memory module) format, and 8 MB of flash memory. It has I/Os for JTAG, RS-232, and Xilinx System ACE™ MPM (message passing memory) connectors.

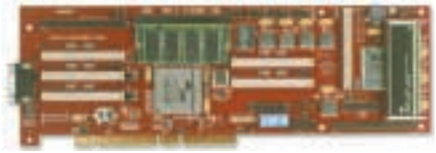


Figure 1 - Virtex-II development board

Virtex-II devices available on the kit include the XC2V1500, XC2V4000, or the XC2V6000 FPGAs – making these kits appropriate for even your most complex designs. Additionally, as many as 541 user-accessible I/O pins are available for expansion.

Expansion Board

The MicroBlaze development kit also comes bundled with the communications/memory expansion card (Figure 2). This card includes 64 MB of Micron SDRAM, 16 MB of Micron flash memory, 1 MB of high-speed Cypress SRAM, a 10/100/1000 National Ethernet PHY, a Cypress USB 2.0 transceiver, irDA, mouse, keyboard, and PCMCIA slot.

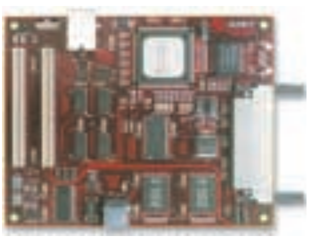


Figure 2- Communications/ memory expansion board

Virtex-E Kit

The Virtex-E based development kit starts with a PCI form factor board (Figure 3) and contains a XCV1000E-6FG1156 Virtex-E FPGA, 64 MB SDRAM, 32 MB flash memory, PC card interface, video RAM DAC, USB 2.0 PHY, CAN bus, audio DAC, video decoder, 10/100 Ethernet, and PCI and PMC interfaces. The wide range of hardware native to the board makes this kit an excellent development platform for a variety of host, end-point, or bridging applications in the networking, audio, video, industrial control, and consumer markets. The native hardware can be augmented with expansion cards if needed.

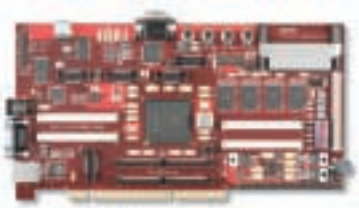


Figure 3- Virtex-E development board

Spartan-IIE Kit

The Spartan-IIE–based evaluation kit starts with a low-cost expansion board (Figure 4) and contains a XC2S200E-6FT256C Spartan-IIE FPGA with LVDS I/O, multimedia audio codec, and LCD interface. The kit features a variety of push buttons, LEDs, and four high-capacity expansion connectors. This MicroBlaze development kit comes bundled with the same communications/memory expansion card described above. This development kit has been optimized for low-cost applications, such as consumer and industrial control, and provides all the hardware required to develop complete applications.

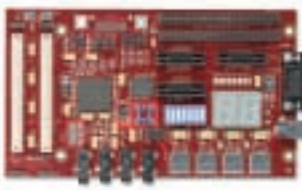


Figure 4- Spartan-IIE evaluation board

Additional expansion boards are available from Avnet Design Services, making it easy to configure just the right set of hardware for specific applications (Table 2).

All MicroBlaze Development Kits ship with the complete software development tools for MicroBlaze microprocessor development, a MicroBlaze IP core license, and detailed design documentation (including layout and bill of materials) sufficient to easily create customized designs for specific applications and hardware form factors.

Visit *www.ads.avnet.com* to get all the details on our expanding suite of development kits and reference designs, or contact your local Avnet FAE.

Expansion Cards for Avnet Design Services Development Kits

Communications/Memory Module: 10/100/1000 Enet, 16 MB flash, 1 MB SRAM, 64 MB SDRAM, IrDA, PC Card, USB 2.0 on keyboard/mouse port.

Motorola 857T Processor Module: MPC857T PowerQUICCTM processor, 10/100 Enet, USB 1.1, RS-232, 16 MB flash, 64 MB SDRAM, 1 MB SRAM, 4 Kb EEPROM, Linux-based embedded OS and board support package.

RLDRAM Memory Module: 200 MHz memory controller card with DDR access to Infineon and Micron RLDRAM devices in a Virtex-II FPGA.

USB 2.0 to SCSI Module: Spartan-IIE based USB 2.0 to SCSI interface is plug-and-play compatible with Windows 2000.

Xilinx IRLTM PMC Platform: Virtex-based IRL platform using the PAVE Framework. PMC connector is compatible with other development boards.

CoolRunner™-II Evaluation Board: Expansion card features the XC2C256, serial A/D converter and user interface.

Spartan-IIE Evaluation Board: Features XC2S150-5PQ208 FPGA, digital thermometer, and user interface.

Virtex-II Evaluation Board: Features XC2V1000 FPGA, digital thermometer, and user interface.

Virtex-E Evaluation Board: Features XCV100E-6PQ240C FPGA, infrared transceiver, digital thermometer, and user interface.

Breakout Module: Expansion headers to create customer connections to a variety of external signals – uses 6 MICTOR connectors and four 50-pin headers.

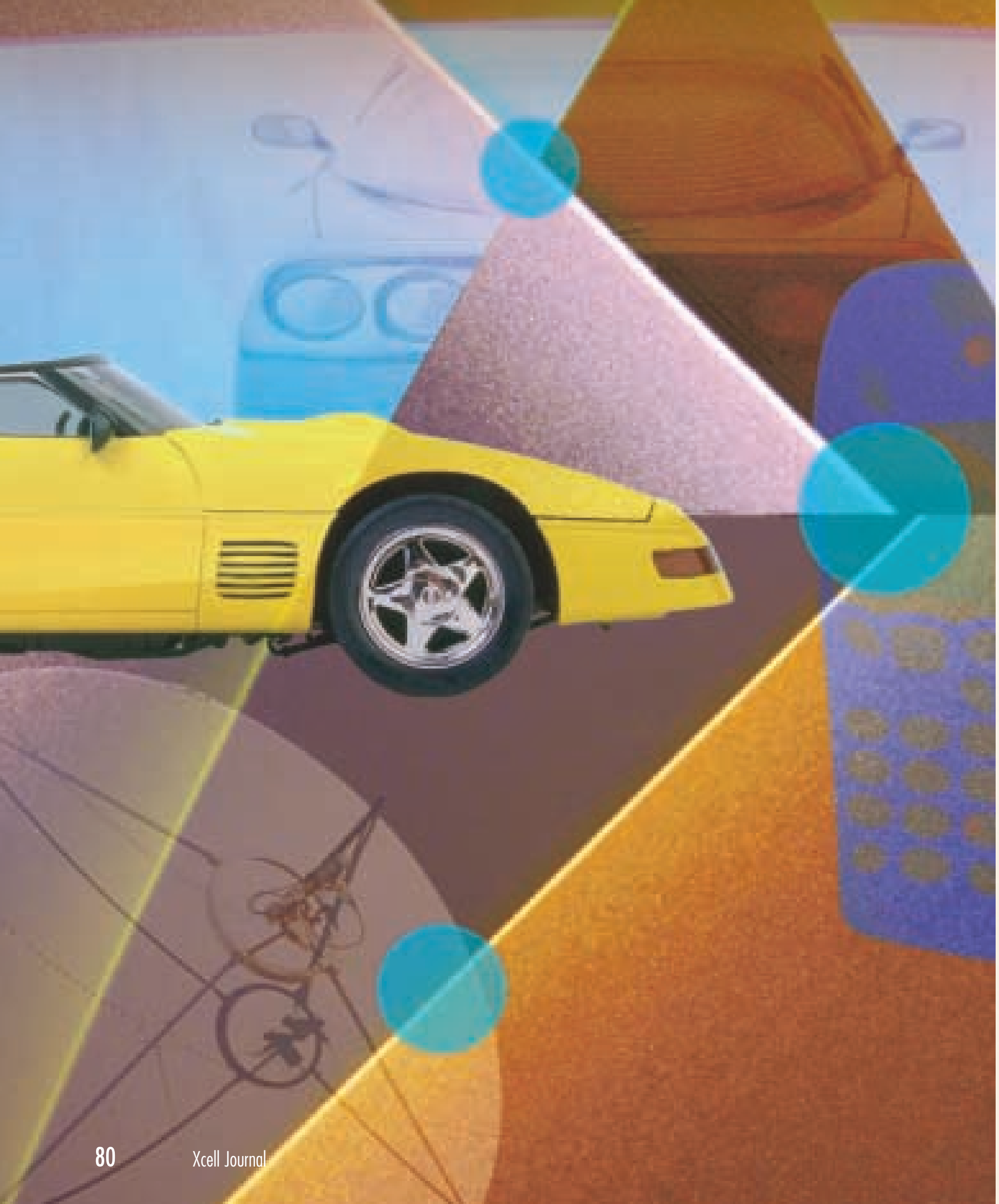
User Prototyping Module: Features a .1" grid prototype area and surface mount footprints.

Adapter Module: Connects to on-board connectors for easy interface.

Table 2 - Expansion boards for use with Avnet Design Services MicroBlaze Development Kits

Telematics Drives the New Automotive Business Model

The emerging technology of telematics heralds the convergence of two-way mobile telecommunications with in-car infotainment services.



by Karen Parnell
Product Marketing Manager, Automotive
Xilinx, Inc.
karen.parnell@xilinx.com

Historically, the business model of the automotive industry has been one of large corporations, long time scales from conception to production – and far from the leading edge of electronics systems.

Those days are gone.

Now, both the business model and design environment of the automotive industry are experiencing rapid change and growth. Telematics – the convergence of mobile telecommunications and information processing in cars – is driving much of the change. Some companies have embraced the telematics concept and are striving to be first to market with new in-car "killer applications." For example, Viasat (a Magneti Marelli and Telecom Italia joint venture) has produced a prototype version of an "Internet car." Other ventures include "WirelessCar" (Volvo, Ericsson, and Telia) and "OnStar" (General Motors). These companies recognize that we are on the brink of an in-car revolution that is bigger than the car manufacturers, bigger than the telecommunications manufacturers, and bigger than the service providers.

For me, as a consumer, telematics means not getting stuck in traffic. A telematics system tells me where I am, where the traffic jams are, and where I must drive to get where I'm going on time. A telematics in-car system knows who I am and automatically adjusts my seat, my steering wheel, and my mirrors the way I like them. The system automatically detects and synchronizes my personal digital assistant (PDA) and my mobile phone with my on-board personal computer when I enter the car. With telematics, I can dial my PDA or mobile phone list using voice recognition while keeping my eyes on the road and my PDA and mobile phone in my handbag. To preserve security, the system automatically erases the call data when I leave the car.

I want all of these information exchange functions and services, but I want only one

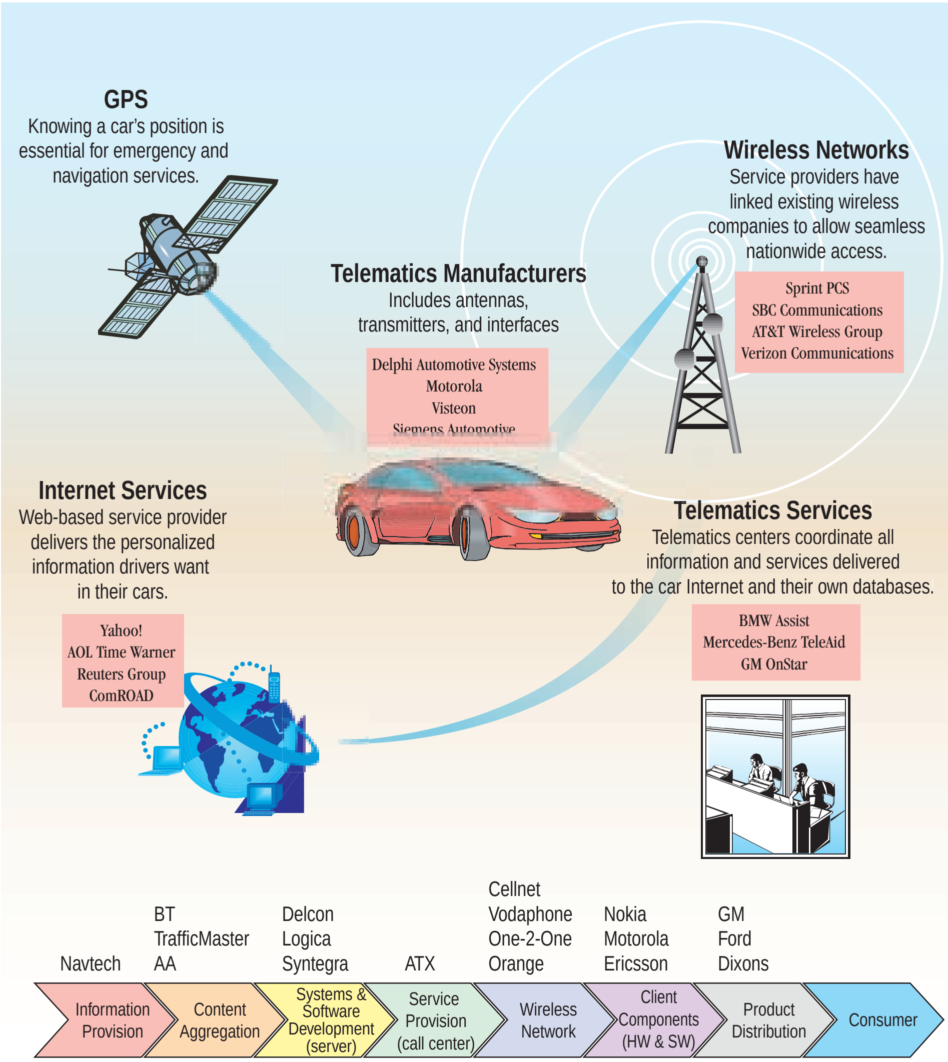


Figure 1 - Telematics value chain

point of contact – typically through the car company or the mobile phone service provider. For telematics-based products and services to succeed, automotive manufacturers must partner with the leaders in other fields. This new business model leads to more consolidated offerings. Examples of successful partnerships are:

- In 1998, Citroen and Trafficmaster™ announced the factory installation of Trafficmaster Oracle in all Xantia models.
- Webraska, the worldwide provider of wireless navigation services and technologies, has signed a contract with Borg Instruments, a first-tier automotive supplier for innovative electronics, to provide off-board navigation services.

Telematics Offers a New Value Chain

Car radios are mutating into a variety of products with increased communications and entertainment functionality, starting with the digital convergence of the audio and navigation functions into one unit. Going forward, we will see further convergence with gaming consoles, PDA-type functionality, and Internet connectivity.

With mobile phone or set-top boxes, we can offset higher hardware costs through server-based applications (for example, off-board navigation) as wireless data transfer rates increase. But the costs depend on the relative cost of in-vehicle hardware versus the airtime charge per byte. In this model, the mobile phone manufacturers work with the network providers to offset the cost of the hardware.

We can now add to this value chain the provision of a vehicle emergency messaging system (VEMS) (Rescu in the U.S.). This could mean, for example, using the services of ATX Technologies, Sprint networks, and Motorola (and Visteon) hardware.

The next step in the telematics value chain is adding fleet management and wireless application protocol (WAP) or third-generation (3G) wireless Internet access. At this point, we realize that no one really "owns" the customer. Figure 1 shows the full telematics value chain.

Conclusion

The automotive industry is facing one of the most exciting and challenging times in its history. New design practices, schedules akin to those of the consumer electronics market, and the Internet connectivity challenges of mobile communications products, all converge into one system that has restricted space and is often exposed to harsh environments. It has been said, "If you can design a reliable, full-functioning system within the cost constraints of the automotive industry, you can design anything."

As we reported in "You Can Take It With You: On the Road with Xilinx" in the Summer 2002 edition of Xcell Journal, Xilinx has developed a new "IQ" grade of industrial FPGAs and CPLDs with an extended temperature operating range specifically designed for telematics applications.

To learn more about automotive telematics, please visit the following websites:

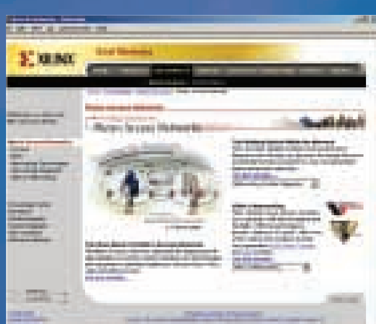
- *www.atxtechnologies.com*
- *www.magnetimarelli.net/eng/inf_d.html*
- *www.navtech.com*
- *www.onstar.com*
- *www.trafficmaster.co.uk*
- *www.webraska.com*
- *www.wirelesscar.com*
- *www.xilinx.com/automotive*

Designing Next-Generation MAN Products with Xilinx

Xilinx enables designers to meet the challenges of building metro and edge access products.

by Diane Katsuyoshi
Marketing Manager, Strategic Solutions
Xilinx, Inc.
diane.katsuyoshi@xilinx.com

With over four million visitors since its introduction, the eSP web portal has quickly become an invaluable resource for the engineering community. A rich source of information, the eSP site includes technology tutorials, market overviews, system block diagrams, in-depth presentations on product applications, and comprehensive glossaries. Since its inception, the portal has covered the emerging markets of home networking, wireless, and digital video technologies by providing comprehensive solutions that accelerate product development and time-to-market. The site has now been expanded with a new segment targeted at metro and edge access networks.



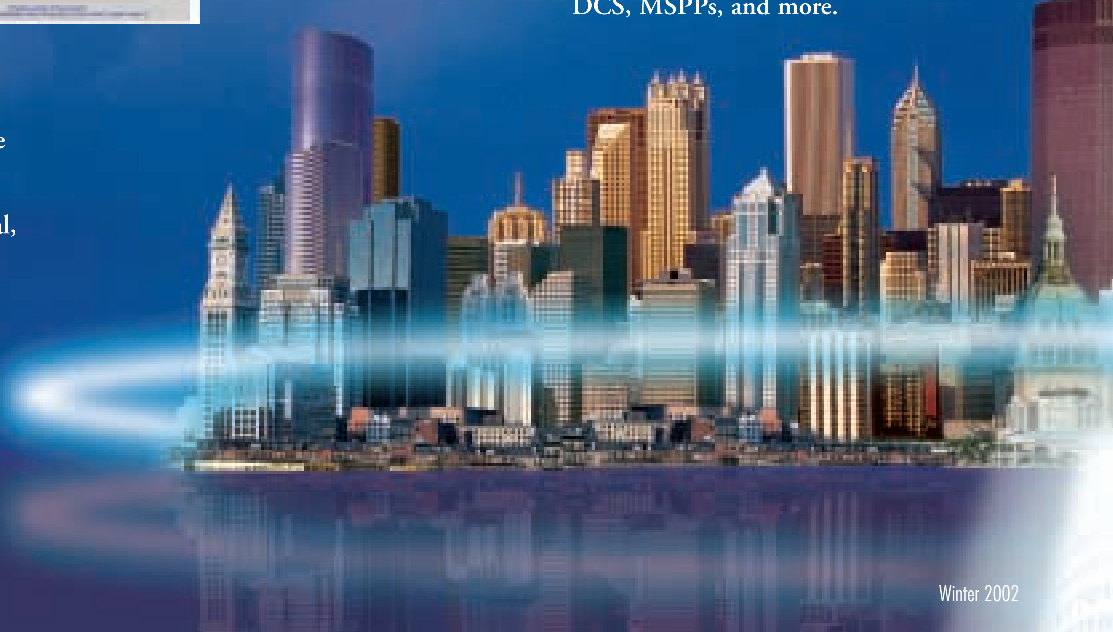
This article discusses the eSP web portal, as well as giving details on the recent Metro-Optical Networking Forum.

eSP Introduces Metro Access Networks Segment

Metro Access Networking (MAN), the industry's only online resource dedicated to addressing the challenges of designing products for the MAN market, is the latest segment to be added to the eSP portal.

The key components of the MAN portal include:

- Building Blocks within the Network – This section provides detailed system solutions and block diagrams from Xilinx to accelerate time-to-market in the areas of the 10 GE (gigabit Ethernet) router, GE switch, backplane switch fabric, and much more.
- Xilinx in Networking – Xilinx offers comprehensive design solutions for the entire line card, control card and backplane including, PHY, Framer/MAC, network processor, memory interface, backplane interface and system interface. This section also provides detailed information on IP, silicon solutions and system design details.
- Comprehensive Resource for MAN Technologies – Provides detailed descriptions of the myriad of MAN, wide-area networks (WANs), local area networks (LANs), and access technologies, along with complete presentations and tutorials on the Xilinx fit-for-each-optical-networking technology, from SONET/SDH and RPR to 10 GE and ATM.
- MAN Products – This section provides details on how Xilinx solutions provide value in MAN products including ADM, DCS, MSPPs, and more.



In conjunction with the eSP MAN segment launch, Xilinx also held an industry forum to bring together MAN leaders to discuss issues in this dynamic market.

Metro-Optical Networking Forum

Xilinx, along with Reed Electronics Group, Avnet Design Services and Cilicon, an Avnet Company, recently hosted the Metro-Optical Networking Forum. This was truly a premier gathering of key industry leaders and visionaries who addressed the technologies and challenges of developing and deploying successful products for the MAN market.

This highly successful event brought together over 700 design engineers, system architects and executives to hear first-hand from executives from the top MAN box builders, standards committees and semiconductor suppliers about the future of the MAN market in terms of its technologies, solutions and applications. Here's an overview of the discussion.

Executives debate whether RPR or MEF will be the leading metro standard.

Vint Cerf explains the metro from a service provider perspective.

The event gave engineers, system architects and technology executives a good top-to-bottom understanding of the current MAN market and what the future holds for companies designing products for the MAN. Many of the industry experts from companies, including Allegro Networks, Lantern Communications™, Luminous Networks™ and more, agreed industry standards are still in flux and the MAN bandwidth bottleneck must be resolved by upgrading the infrastructure in terms of providing performance and flexibility at a low cost. Those who presented believe programmable logic plays an integral part in making sure that their MAN products are flexible enough to adapt to the vastly changing networking technologies and standards.

The key highlights of the event included:

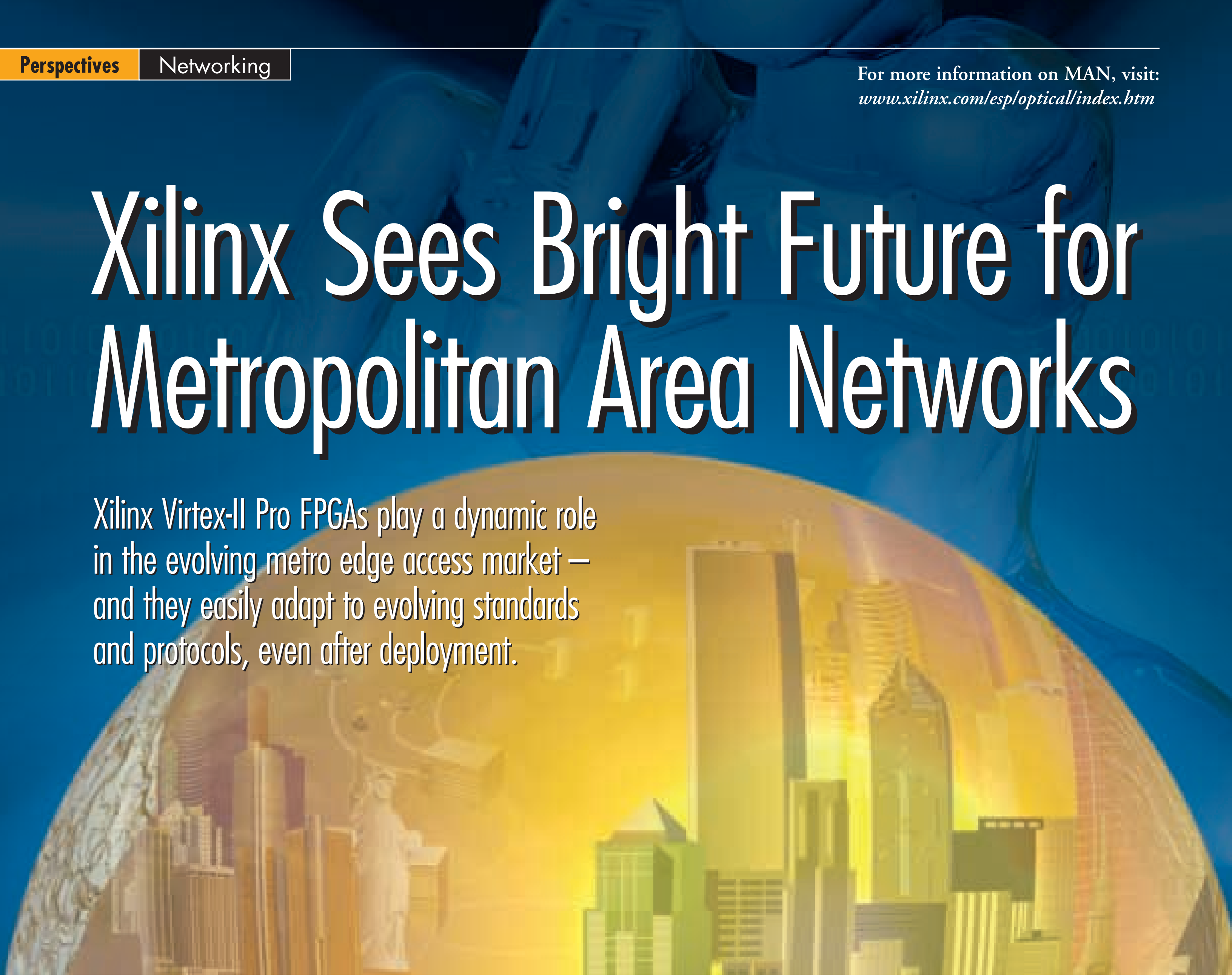
- Panel discussion – Executives debated whether RPR or Metro Ethernet would prevail as the standard for the MAN industry.
- Keynote Presentation – Vinton G. Cerf, Sr. Vice President of Architecture and Technology of WorldCom, widely known as one of the "Fathers of the Internet," shared an exciting and interactive view on the future of the MAN from a service provider perspective with the audience.
- Exhibits and Live Demonstrations – A key component of the event was to showcase various solutions available from many top IP, reference board, ASSP, and semiconductor companies in the MAN industry.

Attendees viewing the MAN solutions offered by participating companies.

For More Information

For information on the Metro Access Networks portal on eSP, visit *www.xilinx.com/esp/optical.* You can download the material presented at the Metro-Optical Networking Forum. Visit *www.xilinx.com/esp/knowledge_center/events/monf.htm.*

For more information on MAN, visit:
www.xilinx.com/esp/optical/index.htm



Xilinx Sees Bright Future for Metropolitan Area Networks

Xilinx Virtex-II Pro FPGAs play a dynamic role in the evolving metro edge access market – and they easily adapt to evolving standards and protocols, even after deployment.

by Robert Bielby, Senior Director
Strategic Solutions Marketing
Xilinx, Inc.
robert.bielby@xilinx.com

The dynamics of the metropolitan area network (MAN) are undergoing a fundamental transformation. The explosion of bandwidth in local area networks (LANs), the deployment of Gigabit Ethernet, and the growth of dense wave division multiplexing (DWDM) in long-haul, wide area networks (WANs) have all served to fuel the demand for networks capable of servicing significantly more data traffic.

All Roads Lead to the MAN

Today, most LAN and WAN traffic converges at the MAN – a transport technology comprising a series of fiber-optic rings that typically encircle major metropolitan areas. Within the MAN, Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) is the principal networking protocol. Initially deployed for voice traffic, where a typical line, such as T1 (1.54 Mbps), was sufficient to transport multiple voice channels, SONET is very inefficient when it comes to handling IP-based traffic. In addition, SONET is not highly scalable. So, while corporate LANs are moving to 10 Gigabit Ethernet, and WANs are moving to line speeds up to 40 Gbps, the interface between the two networks can easily be 1,000 times slower than the slowest technology in the network.

According to industry estimates, 80% of today's telecommunications traffic consists of data. Even though this percentage is expected to increase further, service providers are still focusing on legacy voice services, because they provide the revenue base that will allow carriers to build out their new service models. To increase the efficiency and effectiveness of their investments, service providers are turning to the MAN.

In addition to their efforts to manage existing data traffic more effectively in the MAN, carriers are adding new services – voice and video over IP (Internet Protocol), virtual private networks (VPNs), 3G (3rd generation) wireless access, wholesale Ethernet delivery, and transparent LAN services – to create new revenue sources.

Another benefit of this increased revenue-generating traffic is that it allows carriers to take advantage of the currently overbuilt Internet backbone, helping to offset the heavy investments carriers already have in WANs.

Although there remains much discussion and indecision regarding the eventual MAN technology winner, Resilient Packet Ring (RPR) and Metro Ethernet Forum (MEF) are the prime candidates for providing higher performance and more efficient data transport within the MAN. Besides being able to handle voice and data traffic more efficiently, technologies such as RPR can also significantly reduce a service provider's costs by realizing the benefits of a single, converged network. These benefits are far-reaching in their ability to reduce significantly the cost of operations, accounting, management, and provisioning (OAMP) – which typically constitute up to 49% of a service provider's network costs. In short, many factors make it clear that the high-growth area in the telecommunications market is centered squarely in the MAN.

Requirements of the New MAN

For the MAN market to take off, the equipment must do two things – provision and billing for services. This is tougher than it sounds because the metro edge will be a primary point where multiple traffic types – with varying traffic requirements – will converge. To provide basic provisioning and billing, successful interface equipment at the metro edge must meet all the following requirements:

- Deliver provisioning and bandwidth consistent with Service Level Agreement (SLA) policies, regardless of subscriber location on the network.
- Feature a highly scalable architecture capable of servicing thousands of endpoints while supporting a broad range of applications.
- Provide reliability on the order of 99.999% uptime with support for redundant hardware and ring topologies, fiber protection, and restoration capabilities.
- Support services requiring deterministic and predictable performance, such as real-time voice and video applications. These services should deliver minimal latency and jitter.
- Converge voice and data services seamlessly.
- Be optimized for a ring topology and incorporate service protection.
- Be agile and flexible enough to support a wide range of services.
- Be cost-effective to operate.

These requirements place extreme demands on areas such as packet processing, network traffic management, and backplane technologies. Developing a product that delivers this broad range of features and capabilities – while remaining flexible enough to accommodate a variety of traffic types, specification changes, and/or enhancements – requires high-performance, leading-edge technologies.

MANaging Traffic

Networking solutions for the MAN must be able to cost-effectively support a high density of customers using multiple traffic types. Moreover, the density of aggregated traffic leading into a single network area requires that traffic management provide highly effective throughput, while supporting such services as multicasting.

Source address filtering, which helps reduce traffic congestion by identifying traffic that can be "touched" (as opposed to traffic that should not be), requires operation at layers 2 and 3 of the Open System Interconnection (OSI) model. At line rates of 10 gigabits per second (OC-192), filtering poses significant processing challenges for most silicon technologies. Furthermore, while traffic management places extreme packet processing requirements on the data plane, traffic management also requires the ability to prioritize traffic on the control plane.

Because differentiated services are critical to revenue generation, advanced traffic management must be applied to both on-ramp and off-ramp access points. This is the only way to ensure that customers receive the services and bandwidth that they are paying for. Conversely, traffic management must also make sure that customers aren't receiving more bandwidth than they are paying for.

This traffic contract is typically enforced at the on-ramp, or ingress side, of the network. Enforcement is usually based on "leaky-bucket" policing algorithms that drop arriving packets that are outside the scope of the provisioned service contract.

Typically, carriers use weighted fair queuing scheduling in conjunction with shaping to ensure that bandwidth guarantees are supported. To deliver the low-jitter services required to support voice and video traffic, weighted fair queuing scheduling is usually applied on a per-flow basis.

Placing these system requirements in the context of a router that supports OC-768 (40 Gbps) line speeds requires enqueue/dequeue packet processing at rates of greater than 100M packets per second (PPS), with peak scheduling decisions of up to 100M PPS. To be competitive, the router must support in excess of 100K unique flows, with each flow spanning a wide range of granularity, from 64 Kbps to 40 Gbps.

> **ALTHOUGH THERE REMAINS MUCH DISCUSSION AND INDECISION REGARDING THE EVENTUAL MAN TECHNOLOGY WINNER, RESILIENT PACKET RING (RPR) AND METRO ETHERNET FORUM (MEF) ARE THE PRIME CANDIDATES FOR PROVIDING HIGHER PERFORMANCE AND MORE EFFICIENT DATA TRANSPORT WITH IN THE MAN.**

Suffice to say, supporting this level of processing performance imposes significant demands on the basic characteristics of semiconductor technologies. To make matters worse – as mentioned earlier – two principal Layer 2 technologies are vying for acceptance as the standard for MAN applications: Resilient Packet Ring and optical Metro Ethernet Forum – and both are still being defined.

With RPR, for example, there is still considerable consternation regarding traffic fairness – which details how ring traffic is added and dropped. This lack of agreement has caused a fundamental split across the industry, spawning three derivatives of the RPR specification.

RPR, officially referred to as 802.17, is expected to become a formal standard in 2003, but early versions of the RPR specification have already been shipped to carriers. Long-term compatibility between equipment shipped today and the final 802.17 specification is far from certain. Indeed, compatibility is most likely impossible – unless that equipment has been implemented in a flexible, high-performance fabric.

The Virtex-II Pro "Killer App"

On March 4, 2002, Xilinx redefined the programmable logic landscape – again. The new Virtex-II Pro™ FPGAs herald an astonishing breakthrough in system-level solutions. With as many as four IBM PowerPC™ 405 processors immersed in the industry's leading FPGA fabric, Xilinx/Conexant's high-speed serial I/O technology, and Wind River System's cutting-edge embedded design tools, Xilinx delivers a complete development platform of infinite possibilities. The inherent system-level performance and feature sets are a perfect match for the performance demands and diverse requirements of equipment for the emerging MAN.

Each PowerPC core runs at 300+ MHz, delivering 420 Dhrystone MIPS, and is supported by IBM CoreConnect™ bus technology. The unique Xilinx IP-Immersion architecture makes it easy to harness the power of high-performance processors, and to integrate soft IP (intellectual property) easily into the industry's highest-performance programmable logic.

The Xilinx XtremeDSP solution is the world's fastest programmable DSP solution. With up to 556 embedded 18 x 18 multipliers, 10 Mb of embedded block RAM, an extensive library of DSP algorithms, and tools that include System Generator for DSP, Xilinx ISE, and Cadence SPW, XtremeDSP is the industry's premier programmable solution for enabling tera-MACs per second applications. This level of high-performance DSP is critical to supporting the computation of packet transmit schedules.

The first programmable devices to combine embedded processors along with 3.125 Gbps serial transceivers, the Virtex-II Pro series addresses all existing connectivity requirements as well as those associated with emerging high-speed interface standards. Xilinx Rocket I/O™ transceivers offer a complete serial interface solution, supporting 10 Gigabit Ethernet with XAUI, 3GIO, SerialATA, and a host of other protocol technologies. Xilinx SelectI/O™-Ultra supports 840 Mbps, LVDS, and high-speed single-ended standards such as XSBI and SFI-4.

In a single off-the-shelf programmable device, systems architects can take advantage of microprocessors, the highest density of on-chip memory, multi-gigabit serial transceivers, digital clock managers, on-chip termination, and more. This will result in a dramatic simplification of board layout, a reduced bill of materials, and unbeatable time to market.

Additionally, systems designers can partition and repartition their systems between hardware and software at any time during the development cycle – and even after the product ships. The overall system can thus be optimized – guaranteeing that performance targets are achieved in the most cost-efficient manner – and hardware and software can be debugged and observed simultaneously at speed. This capability is critical, as traffic types typically are not known until the product is out in the market and because, even when possible, evaluating "corner cases" is either impossible or too time-intensive.

Optimized for the PowerPC, Wind River's industry-proven embedded tools are the premier support for real-time microprocessor and logic designs. The Virtex-II Pro FPGA is driven by the lightning-fast Xilinx ISE 5.1i software, the most comprehensive, easy-to-use development system available.

Conclusion

Clearly, the metropolitan area will be next arena of growth in the telecommunications market. While many factors contribute to this potential for growth, factors such as the current lack of standardization are impeding broad-based deployment. Bridging the bandwidth gap between the LAN and the WAN – while simultaneously supporting a host of new applications and corresponding traffic patterns – will place unprecedented demands on semiconductor technologies.

The introduction of the Virtex-II Pro FPGA family heralds a new era of programmable solutions that will provide the performance, features, capabilities, and flexibility to address the extreme demands of the metro market. In short, these FPGAs promise to be a key technology in catalyzing the growth of this new market.

Speed Your Time to Market with a Dedicated Xilinx Engineer

Improve your design productivity and accelerate your time to market with a dedicated application engineer from Xilinx Titanium Technical Service.

by Jack Dunnigan
Titanium Marketing Manager
Xilinx, Inc.
jack.dunnigan@xilinx.com

If you need extra expertise to meet design performance specifications or assistance in getting the most out of Xilinx programmable logic devices and software, Titanium Technical Service could be what you need. With Titanium Technical Service from the Xilinx Global Services Division, you get a dedicated application engineer on a contract basis – at your site, at Xilinx, or both.

It's All About Efficiency

The mission of Titanium Technical Service is to improve your efficiency by helping you achieve your design goals and meet – or beat – production deadlines.

Benefits

- Competitive advantage – Faster time to market, increased design productivity
- Assurance – Direct access to a dedicated application engineer to address your individual needs
- Flexibility – Dedicated application engineers can work on-site or provide services from their Xilinx offices.

Get It Right the First Time

Xilinx Titanium Technical Service application engineers have in-depth application knowledge that few people in the digital design world possess. Our Titanium application engineers are an integral part of Xilinx and have working relationships with all of the technical resources within the company. All of the Titanium engineers have direct escalation paths to resolve issues, allowing you to get your design completed faster. This escalation path is difficult to match by any other premium service provider.

With today's complexity of designs, and design possibilities, it is critical that you get it right the first time. Titanium Technical Service application engineers are especially adept at ensuring you start your designs the right way. Our engineers provide design flow methodology coaching to make sure you take the most efficient approach possible.

Meet Your Goals and Deadlines

One of the toughest challenges designers face is when they need extra performance to meet design goals, and they are already at the end of the design cycle. Fortunately, our engineers have in-depth expert knowledge of Xilinx back-end tools. With this knowledge, we can take your design and squeeze out all of the performance possible and/or ensure that the design stays within the specified product size. Titanium Technical Service engineers use all of the latest floorplanning, timing analysis, and HDL code optimization techniques to achieve the needed results.

Design style and techniques can have a significant impact on performance and size. Titanium application engineers' skill in tracing these issues back to your design is our most powerful service. Our engineers have encountered many tough situations, and their technical knowledge and experience really pay off in the end. Tweaking a state machine, or using a different multiplier to achieve needed design results, is all in a day's work for a Titanium Technical Service engineer.

You Have Control

A Titanium Technical Service application engineer can work at your site, at Xilinx, or a mix of both. This flexibility allows our engineers to fully understand your needs and requirements. Furthermore, Titanium engineers have the ability to leverage our factory resources to resolve problems and accelerate production.

Our contract method gives you control over your Titanium-related expenses. There are specific start and end dates written into the contract. Your Titanium Technical Service application engineer and account manager can provide you with regular status reports. These reports serve as a useful tool to determine the progress of Titanium Technical Service in meeting your needs.

For more information about the range of Titanium Technical Services, including purchasing and contact information, please go to our website at *http://support.xilinx.com/support/services/titanium.htm.*

Get Top Priority Support with Platinum Technical Service

Time is money – and you'll gain time and money by signing up for Platinum Technical Service.

by Bill Okubo
Marketing Manager
Product Solutions Marketing
Xilinx, Inc.
bill.okubo@xilinx.com

In the rush to get your product to market, the last thing your designers need when they have a technical question is an earful of elevator music while they wait on hold. That won't happen when you sign up for Platinum Technical Service from the Xilinx Global Services Division. Your designers will get a dedicated toll-free number that puts them in direct contact with our senior application engineers so they can get the answers they need without having to wait.

With Xilinx Platinum Technical Service, your designers' calls get top priority. Furthermore, Platinum Technical Service calls are answered by skilled senior application engineers with a track record of successfully solving just about any complex problem your designers are likely to face. Platinum Technical Service has twice as many engineers for the same volume of customers as our standard Gold-level of service. Platinum Technical Service not only provides faster help, but we also deliver proactive status updates until your case is resolved.

How serious are we about fast problem resolution? With Platinum Technical Service you'll have a 65% shorter wait time, which means our senior application engineers waste no time getting started on a solution for your technical issue.

We make it easy to reach us either by calling or sending us e-mail:

- **North America:** Monday – Friday, 7 a.m. to 5 p.m. Pacific Standard Time.
  - Dedicated toll-free number is available in North America only.
  - In North America, hours of availability on Thursdays are 7 a.m. – 4 p.m. PST.
  - Hours of availability exclude published Xilinx holidays.
- **Europe:** Monday – Friday, 9 a.m. to 5:30 p.m. Greenwich Mean Time.
  - Local dedicated numbers are available across Europe.
  - In Europe, Platinum Technical Service customers have a zero wait time if they contact us by phone and a 1-2 hour reply if they use e-mail.

Regardless of where you are located, you can also pose your question online anytime, day or night, through our acclaimed website, *support.xilinx.com*. If you have an online technical question after hours, it will be addressed as soon as possible on the next business day.

In addition to a dedicated toll-free number and access to senior application engineers, Platinum Technical Service entitles you to 10 education credits. You can apply your designers' education credits to a two-day public class led by instructors who are experienced designers themselves, or your design team may take any of our 70 different Live e-Learning modules. For a complete list of available Education Services courses, go to *support.xilinx.com* and select the education tab.

Sign up for Platinum Technical Service right away and give your design team top priority status. Call us at 1-800-888-FPGA (3742), e-mail us at *fpga@xilinx.com*, or find the Xilinx sales office nearest you at *www.xilinx.com/company/sales/offices.htm*.

Features	Platinum	Gold
Senior Applications Engineers	✓	Not Included
Dedicated Toll-Free Number	✓	Not Included
Priority Case Resolution	✓	Not Included
Proactive Status Updates	✓	Not Included
Ten Education Credits	✓	Not Included
Electronic Newsletter	✓	Not Included
Formal Escalation Process	✓	Not Included
Service Packs and Software Updates	✓	✓
Application Engineers/Customer Ratio	2X Gold	Standard

Figure 1 - Platinum Technical Service feature comparison

Get It All with XPA – Xilinx Productivity Advantage

Software, IP cores, Education and Support Services are all rolled up in one convenient solution.

by Bill Okubo
Marketing Manager
Product Solutions Marketing
Xilinx, Inc.
bill.okubo@xilinx.com

The Xilinx Productivity Advantage (XPA) Program delivers everything you need to improve your designs – software, Education and Support Services, and IP cores – in one convenient package. You and your designers get everything you need when you need it, at a better value than when ordered separately.

The advantages of the XPA Program include:

- One purchase order delivers a complete package of software, services, and IP cores.
- The purchasing process is accelerated by reducing paperwork.
- The packaged solution gives you best-value pricing.

XPA Seat – A New Pre-Packaged Solution

The new XPA Seat is a single-unit package that delivers a pre-determined quantity of software design tools, training, and premium "hotline" support. It provides a pre-packaged solution to individuals who have an immediate need for tools and services. As these individuals begin new FPGA designs, they may choose to purchase the XPA Seat on an as-needed basis. The XPA Seat increases productivity with its tools and services, and it reduces the paperwork necessary to order.

Individual XPA Seat part numbers correspond to the software – either ISE Alliance Series™ or ISE Foundation™ software:

XPA Seat – ISE Alliance Series tools:

- Part # DS-ISE-ALI-XPA
- One seat ISE Alliance Series software
- 10 training credits
- One seat Platinum Technical Service.

XPA Seat – ISE Foundation tools:

- Part # DS-ISE-FND-XPA
- One seat ISE Foundation software
- 10 training credits
- One seat Platinum Technical Service.

Previously, the XPA Program was only offered as a custom solution of software, education and support services, and IP cores. The custom XPA solution was tailored to the customer's organization and specific design requirements. This made-to-order package of custom tools, training, and support package is well-suited to large design organizations that wanted to equip an entire design team with the same tools and services.

By contrast, the XPA Seat was developed to provide a pre-packed offering for individual designers.

How to Order An XPA Seat

The XPA Seat for individuals is easy to order. Just contact your Xilinx distributor to order the XPA Seat.

For specialized assistance with a custom XPA package, contact your regional Xilinx sales representative.

Visit *www.support.xilinx.com/support/gsd/xpa_program.htm* to see a complete list of Xilinx distributors and sales representatives.

High-Performance DSP Workshop – For University Professors

University courses in DSP design get a head start.

by Jeff Weintraub
Technical Marketing Engineer
Xilinx, Inc.
jeff.weintraub@xilinx.com

The Xilinx System Generator for DSP is a significant advancement, allowing you to quickly model and simulate DSP algorithms in a graphical environment. University professors are now using this software to teach engineering courses that focus on high-performance DSP design techniques. For courses with labs, students can quickly and easily implement a completed DSP model, in an FPGA, at the computer desktop. This software, combined with Xilinx FPGAs, makes an ideal environment for learning how to create high-performance DSP systems.

To help professors integrate the System Generator for DSP into the engineering curriculum, Xilinx offers workshops on Digital Signal Processing with FPGAs through the Xilinx University Program (XUP). A series of hands-on labs allow professors to step through the process of creating an audio FIR filter and implementing it in hardware, learning how FPGAs are used to create high-performance DSP designs through parallelism.

During the summer of 2002, XUP presented two of these workshops, one at Xilinx headquarters in San Jose, California, and one at Imperial College London. A total of 69 university professors attended from 42 institutions and 13 countries.

"THE DSP DESIGN FLOW WORKSHOP IS AN EXCELLENT TRAINING FOR ANY UNIVERSITY PROFESSOR TO BRING TOGETHER A LARGE VARIETY OF TOOLS SUCH AS MATLAB, SYSTEM GENERATOR FOR DSP, ISE, SYNPLICITY, AND MODELSIM. I THINK ALL THE DELEGATES THOUGHT IT WAS AN EXCELLENT COURSE."

– PETER CHEUNG - IMPERIAL COLLEGE - ENGLAND

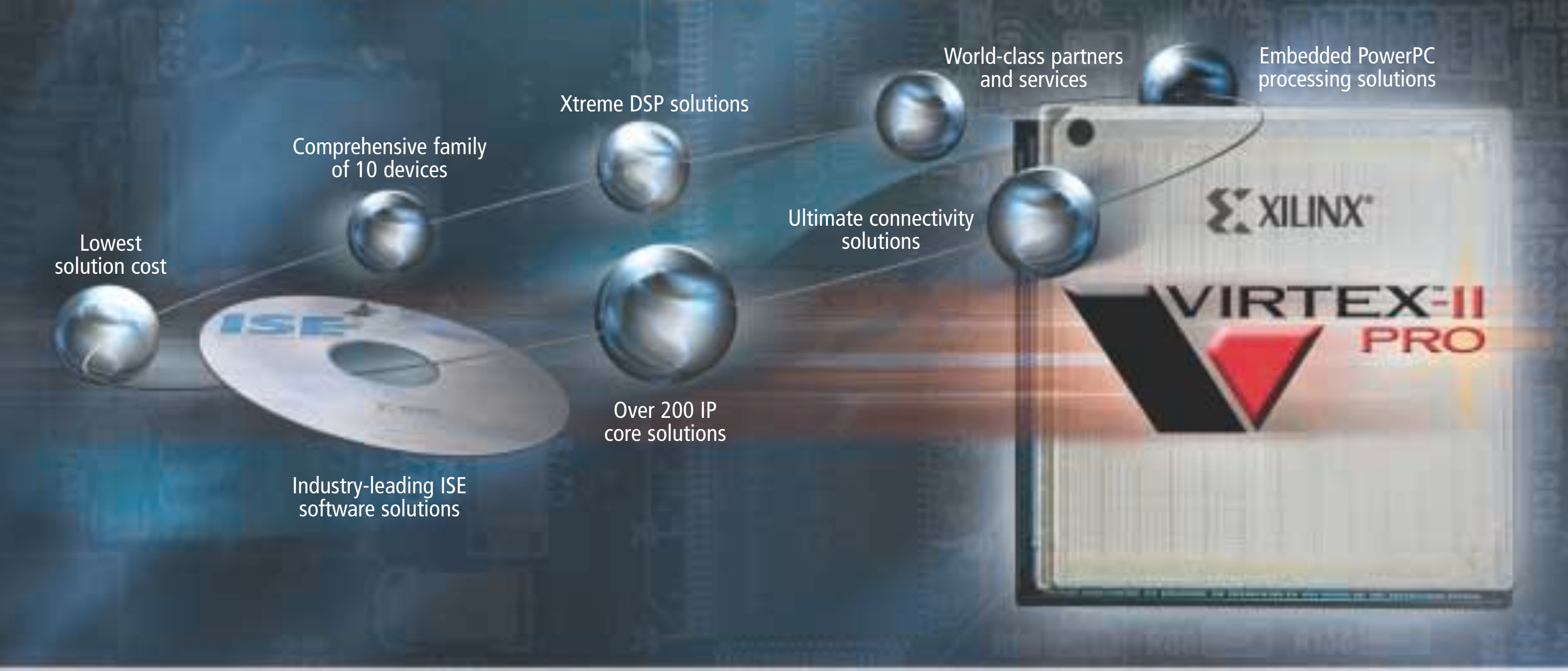
"I WAS REALLY IMPRESSED WITH THE QUALITY OF THE DSP DESIGN FLOW WORKSHOP. SYSTEM GENERATOR FOR DSP IS A VERY ATTRACTIVE TOOL FOR CONCEPTUALIZING AND IMPLEMENTING HIGH-LEVEL ALGORITHMS TARGETING FPGAS. THE EASY-TO-USE TOOL ENABLED ME TO MANIPULATE DATA FLOW PATHS USING HIGH-LEVEL BLOCK FUNCTIONS."

– FRANK POPPEN - OFFIS (OLDENBURGER FORSCHUNGS UND ENTWICKLUNGSINSTITUT FUER INFORMATIKWERKZEUGE UND SYSTEME) RESEARCH INSTITUTE - GERMANY

Conclusion

The Xilinx System Generator for DSP is finding many uses in industry and academia – there is no faster or easier way to develop DSP designs. For more information on the Digital Signal Processing with FPGAs workshops, go to: *www.xilinx.com/univ.*

WHAT WOULD YOU LIKE TO BUILD TODAY?



The Virtex-II Pro™ FPGAs provide the highest logic performance, density, and memory capacity in the industry. Plus there are up to four IBM PowerPC™ processors and up to 24 Rocket I/O™ transceivers included at no additional charge. Supported by the industry-leading ISE software and over 200 IP cores, Xilinx delivers more value than ever.

COMPLETE SOLUTIONS FOR HIGH-PERFORMANCE LOGIC

Logic designers can take advantage of the superior density and performance of the Virtex-II Pro family. It's a complete solution, with 10 family members ranging from 3K up to 125K logic cells and with 400+MHz clock rates – better than any competitive device. You can design with a Virtex-II Pro FPGA, shipping today in 0.13 micron process technology, and boost your design performance. Plus you can access the 200+ IP cores that support Virtex-II Pro today.

Virtex-II Pro Device	2VP2	2VP4	2VP7	2VP20	2VP30	2VP40	2VP50	2VP70	2VP100	2VP125
Logic Cells	3,168	6,768	11,088	20,880	30,816	43,632	53,136	74,448	99,216	125,136
BRAM (Kbits)	216	504	792	1,584	2,448	3,456	4,176	5,904	7,992	10,008
IBM PowerPC Processors	0	1	1	2	2	2	2	2	2	4
Rocket I/O Transceivers	4	4	8	8	8	12	16	20	20	24
DCMs	4	4	4	8	8	8	8	8	12	12
Max User IO	204	348	396	564	692	804	852	996	1164	1200

DRIVING DOWN SOLUTION COST

Our ISE tools speed you through design and debug. System integration capabilities reduce your overall bill of materials, and our software extracts maximum performance and density out of the silicon for the lowest production cost. And with 300mm wafer technology and Virtex-II Pro EasyPath solutions for cost reduction, we ensure you'll always have a system cost advantage.

HIGH-PERFORMANCE SYSTEM SOLUTIONS

Virtex-II Pro FPGAs extend performance and integration into the system realm with TeraMAC DSP performance, over 2000 D-MIPS of PowerPC processing power, and up to 24 3.125 Gbps Rocket I/O serial transceivers. And our SelectI/O™ Ultra delivers 840 Mbps LVDS performance, all with the world's leading FPGA logic fabric.

INDUSTRY-LEADING TOOLS

Driving the Virtex-II Pro FPGA is Xilinx's ISE 5.1i software. ISE 5.1i includes incremental design, a macro builder, our intuitive Architecture Wizard, the ChipScope Pro debug environment, and compile times up to 6x faster than our nearest competitor, making it the industry's fastest and most productive tool set.

Visit *www.xilinx.com/virtex2pro* today and start building with the best.

XILINX®
The Programmable Logic Company℠

www.xilinx.com/virtex2pro

FORTUNE® 2002
100 BEST COMPANIES TO WORK FOR

Xilinx Virtex FPGA Product Selection Matrix

VIRTEX-II PRO | VIRTEX-II

Pins	Body Size	Virtex-II Pro (1.5V) XC2VP2	XC2VP4	XC2VP7	XC2VP20	XC2VP30	XC2VP40	XC2VP50	XC2VP70	XC2VP100	XC2VP125	Virtex-II (1.5V) XC2V40	XC2V80	XC2V250	XC2V500	XC2V1000	XC2V1500	XC2V2000	XC2V3000	XC2V4000	XC2V6000	XC2V8000
	I/O's	204	348	396	564	692	804	852	996	1164	1200	88	120	200	264	432	528	624	720	912	1104	1296
Chip Scale Packages — wire-bond chip-scale BGA (0.8 mm ball spacing)																						
144	12 x 12 mm											88	92	92								
BGA Packages (BG) — wire-bond standard BGA (1.27 mm ball spacing)																						
575	31 x 31 mm															328	392	408				
728	35 x 35 mm																		516			
FGA Packages (FG) — wire-bond fine-pitch BGA (1.0 mm ball spacing)																						
256	17 x 17 mm	140	140									88	120	172	172	172						
456	23 x 23 mm	156	248	248										200	264	324						
676	27 x 27 mm																392	456	484			
FFA Packages (FF) — flip-chip fine-pitch BGA (1.0 mm ball spacing)																						
672	27 x 27 mm	204	348	396																		
896	31 x 31 mm			396	556	556										432	528	624				
1152	35 x 35 mm				564	644	692	692											720	824	824	824
1148*	35 x 35 mm						804	812														
1517	40 x 40 mm						804	852	964											912	1104	1108
1704	42.5 x 42.5 mm								996	1040	1040											
1696*	42.5 x 42.5 mm									1164	1200											
BFA Packages (BF) — flip-chip fine-pitch BGA (1.27 mm ball spacing)																						
957	40 x 40 mm																	624	684	684	684	

Note: Within the same family, all devices in a particular package are pin-out (footprint) compatible.
Virtex-II packages FG456 and FG676 are also footprint compatible.
Virtex-II packages FF896 and FF1152 are also footprint compatible.
* The FF1148 and FF1696 packages support higher number of user I/O and zero RocketIO™ multi-gigabit transceivers.
Important: Verify all Data with Device Data Sheet (http://www.xilinx.com/partinfo/databook.htm)

Numbers indicated in the matrix are the maximum number of user I/O's for that package and device combination, I/Os for RocketIO MGTs are not included in this table.

VIRTEX-II PRO PACKAGE CONFIGURATIONS WITH AVAILABLE ROCKETIO TRANSCEIVER BLOCKS

Package	Transceiver Blocks									
FG256	4	4								
FG456	4	4	8							
FF672	4	4	8							
FF896			8	8	8					
FF1152				8	8	12	16			
FF1148						0*	0*			
FF1517						12	16	16		
FF1704								20	20	24
FF1696									0*	0*

Note: * FF1148 and FF1696 packages support higher number of user I/O and zero RocketIO multi-gigabit transceivers

Platform FPGAs

		CLB Resources				Memory Resources			DSP	Clock Resources		I/O Features				Speed						
Device	System Gates (see note 1)	CLB Array (Row X Col)	Number of Slices	Logic Cells (see note 4)	CLB Flip-Flops	Max. Distributed RAM Bits(kbits)	# 18 kbits Block RAM	Total Block RAM (kbits)	# 18x18 Dedicated Multipliers	DCM Frequency (min/max)	# DCM Blocks (see note 2)	Digitally Controlled Impedance	Maximum Differential I/O Pairs	Max. I/O	I/O Standards	Commercial Speed Grades (slowest to fastest)	Industrial Speed Grades (slowest to fastest)	Serial PROM Family	System ACE	Config. Memory (Bits)	RocketIO™ Transceiver Blocks	PowerPC Processor Blocks
Virtex-II Pro Family — 1.5 Volt																**.13um Nine Layer Copper Process**						
XC2VP2	*	16 x 22	1,408	3,168	2,816	44	12	216	12	24/420	4	YES	100	204	LDT-25, LVDS-25,	-5 -6 -7	-5 -6	ISP/OTP	ISP	1.31M	4	0
XC2VP4	*	40 x 22	3,008	6,768	6,016	94	28	504	28	24/420	4	YES	172	348	LVDSEXT-25, BLVDS-25,	-5 -6 -7	-5 -6			3.01M	4	1
XC2VP7	*	40 x 34	4,928	11,088	9,856	154	44	792	44	24/420	4	YES	196	396	ULVDS-25, LVTECL-25,	-5 -6 -7	-5 -6			4.49M	8	1
XC2VP20	*	56 x 46	9,280	20,880	18,560	290	88	1,584	88	24/420	8	YES	276	564	LVCMOS25, LVCMOS18,	-5 -6 -7	-5 -6			8.21M	8	2
XC2VP30[3]	*	80 x 46	13,696	30,816	27,392	428	136	2,448	136	24/420	8	YES	372	644	LVCMOS15, PCI33, PCI66,	-5 -6 -7	-5 -6			11.36M	8	2
XC2VP40[3]	*	88 x 58	19,392	43,632	38,784	606	192	3,456	192	24/420	8	YES	396	804	GTL, GTL+, HSTL I (1.5V,1.8V),	-5 -6 -7	-5 -6			15.56M	0** or 12	2
XC2VP50[3]	*	88 x 70	23,616	53,136	47,232	738	232	4,176	232	24/420	8	YES	420	852	HSTL II (1.5V,1.8V),	-5 -6 -7	-5 -6			19.02M	0** or 16	2
XC2VP70[3]	*	104 x 82	33,088	74,448	66,176	1,034	328	5,904	328	24/420	8	YES	492	996	HSTL III (1.5V,1.8V),	-5 -6 -7	-5 -6			25.60M	16 or 20	2
XC2VP100[3]	*	120 x 94	44,096	99,216	88,192	1,378	444	7,992	444	24/420	12	YES	572	1,164	HSTL IV (1.5V,1.8V), SSTL2I,	-5 -6 -7	-5 -6			33.65M	0** or 20	2
XC2VP125[3]	*	136 x 106	55,616	125,136	111,232	1,738	556	10,008	556	24/420	12	YES	644	1,200	SSTL2II, SSTL18 I, SSTL18 II	-5 -6 -7	-5 -6			42.78M	0**,20, or 24	4
Virtex-II Family — 1.5 Volt																**.15um Eight Layer Metal Process**						
XC2V40	40K	8 x 8	256	576	512	8	4	72	4	24/420	4	YES	44	88	LDT-25, LVPECL-33,	-4 -5 -6	-4 -5	ISP/OTP	ISP	0.4M		
XC2V80	80K	16 x8	512	1,152	1,024	16	8	144	8	24/420	4	YES	60	120	LVDS-33, LVDS-25,	-4 -5 -6	-4 -5			0.6M		
XC2V250	250K	24 x16	1,536	3,456	3,072	48	24	432	24	24/420	8	YES	100	200	LVDSEXT-33, LVDSEXT-25,	-4 -5 -6	-4 -5			1.7M		
XC2V500	500K	32 x 24	3,072	6,912	6,144	96	32	576	32	24/420	8	YES	132	264	BLVDS-25, ULVDS-25,	-4 -5 -6	-4 -5			2.8M		
XC2V1000	1M	40 x 32	5,120	11,520	10,240	160	40	720	40	24/420	8	YES	216	432	LVTTL, LVCMOS33,	-4 -5 -6	-4 -5			4.1M		
XC2V1500	1.5M	48 x 40	7,680	17,280	15,360	240	48	864	48	24/420	8	YES	264	528	LVCMOS25, LVCMOS18,	-4 -5 -6	-4 -5			5.7M		
XC2V2000	2M	56 x 48	10,752	24,192	21,504	336	56	1,008	56	24/420	8	YES	312	624	LVCMOS15, PCI33, PCI66,	-4 -5 -6	-4 -5			7.5M		
XC2V3000[3]	3M	64 x 56	14,336	32,256	28,672	448	96	1,728	96	24/420	12	YES	360	720	PCI-X, GTL, GTL+, HSTL I,	-4 -5 -6	-4 -5			10.5M		
XC2V4000[3]	4M	80 x 72	23,040	51,840	46,080	720	120	2,160	120	24/420	12	YES	456	912	HSTL II, HSTL III, HSTL IV,	-4 -5 -6	-4 -5			15.7M		
XC2V6000[3]	6M	96 x 88	33,792	76,032	67,584	1,056	144	2,592	144	24/420	12	YES	552	1,104	SSTL2I, SSTL2II, SSTL3 I,	-4 -5 -6	-4 -5			21.9M		
XC2V8000[3]	8M	112 x 104	46,592	104,832	93,184	1,456	168	3,024	168	24/420	12	YES	554	1,108	SSTL3 II, AGP, AGP-2X	-4 -5				29.1M		

Note: 1. System Gates include 20-30% of CLBs used as RAM
2. DCM – Digital Clock Management
3. Available as Virtex-II Series EasyPath Solutions – the low risk cost-reduction path for volume production with Virtex-II and Virtex-II Pro FPGAs.
4. Logic cell = (1) 4 Input (LUT) Look Up Table + Flip Flop + Carry Logic.
* System gate count not meaningful for Virtex-II Pro devices with immersed special blocks such as PowerPC processors and multi-gigabit transceivers.
** The FF1148 and FF1696 packages support higher number of user I/O and zero RocketIO multi-gigabit transceivers.
Important: Verify all Data with Device Data Sheet (http://www.xilinx.com/partinfo/databook.htm)

Xilinx Spartan FPGAs

PRODUCT SELECTION MATRIX

Device	System Gates (see note 1)	CLB Resources: CLB Array (Row X Col)	Number of Slices	Logic Cells (see note 2)	CLB Flip-Flops	Max. Distributed RAM Bits	BLK RAM: # Block RAM	Block RAM (kbits)	CLK Resources: # Dedicated Multipliers	DLL Frequency (min/max)	# DLL's	Frequency Synthesis	Phase Shift	I/O Features: Digitally Controlled Impedance	Number of Differential I/O Pairs	Max. I/O	I/O Standards	Speed: Commercial Speed Grades (slowest to fastest)	Industrial Speed Grades (slowest to fastest)	Serial PROM Family		Config. Memory (Bits)
Spartan-IIE Family — 1.8 Volt																		.18/.15um Six Layer Metal Process				
XC2S50E	50K	16 x 24	768	1,728	1,536	24K	8	32K	NA	25/320	4	YES	YES	NA	83	182	LVTTL,LVCMOS2,	-6 -7	-6	ISP	OTP	0.6M
XC2S100E	100K	20 x 30	1,200	2,700	2,400	37.5K	10	40K	NA	25/320	4	YES	YES	NA	86	202	LVCMOS18, PCI33, PCI66,	-6 -7	-6			0.9M
XC2S150E	150K	24 x 36	1,728	3,888	3,456	54K	12	48K	NA	25/320	4	YES	YES	NA	114	265	GTL, GTL+, HSTL I, HSTL III,	-6 -7	-6			1.1M
XC2S200E	200K	28 x 42	2,352	5,292	4,704	73.5K	14	56K	NA	25/320	4	YES	YES	NA	120	289	HSTL IV, SSTL3 I, SSTL3 II,	-6 -7	-6			1.4M
XC2S300E	300K	32 x 48	3,072	6,912	6,144	96K	16	64K	NA	25/320	4	YES	YES	NA	120	329	SSTL2 I, SSTL2 II, AGP-2X,	-6 -7	-6			1.9M
XC2S400E	400K	40 x 60	4,800	10,800	9,600	153.6K	40	160K	NA	25/320	4	YES	YES	NA	172	410	CTT, LVDS, BLVDS, LVPECL	-6 -7	-6			2.7M
XC2S600E	600K	48 x 72	6,912	15,552	13,824	221.2K	72	288K	NA	25/320	4	YES	YES	NA	205	514		-6 -7	-6			4.0M
Spartan-II Family — 2.5 Volt																		.22/.18um Six Layer Metal Process				
XC2S15	15K	8 x 12	192	432	384	6K	4	16K	NA	25/200	4	YES	YES	NA	NA	86	LVTTL, LVCMOS2,	-5 -6	-5	ISP	OTP	0.2M
XC2S30	30K	12 x 18	432	972	864	13.5K	6	24K	NA	25/200	4	YES	YES	NA	NA	132	PCI33 (3.3V & 5V),	-5 -6	-5			0.4M
XC2S50	50K	16 x 24	768	1,728	1,536	24K	8	32K	NA	25/200	4	YES	YES	NA	NA	176	PCI66 (3.3V), GTL, GTL+,	-5 -6	-5			0.6M
XC2S100	100K	20 x 30	1,200	2,700	2,400	37.5K	10	40K	NA	25/200	4	YES	YES	NA	NA	196	HSTL I, HSTL III, HSTL IV,	-5 -6	-5			0.8M
XC2S150	150K	24 x 36	1,728	3,888	3,456	54K	12	48K	NA	25/200	4	YES	YES	NA	NA	260	SSTL3 I, SSTL3 II, SSTL2 I,	-5 -6	-5			1.1M
XC2S200	200K	28 x 42	2,352	5,292	4,704	73.5K	14	56K	NA	25/200	4	YES	YES	NA	NA	284	SSTL2 II, AGP-2X, CTT	-5 -6	-5			1.4M
Spartan-XL Family — 3.3 Volt																						
XCS05XL	5K	10 x 10	100	238	200	3.2K	NA	NA	NA	NA	NA	NA	NA	NA	NA	77	TTL, LVTTL, CMOS,	-4 -5	-4	ISP	OTP	0.05M
XCS10XL	10K	14 x 14	196	466	392	6.3K	NA	NA	NA	NA	NA	NA	NA	NA	NA	112	LVMOS, PCI	-4 -5	-4			0.09M
XCS20XL	20K	20 x 20	400	950	800	12.8K	NA	NA	NA	NA	NA	NA	NA	NA	NA	160		-4 -5	-4			0.18M
XCS30XL	30K	24 x 24	576	1,368	1,152	18.4K	NA	NA	NA	NA	NA	NA	NA	NA	NA	192		-4 -5	-4			0.25M
XCS40XL	40K	28 x 28	784	1,862	1,568	25.1K	NA	NA	NA	NA	NA	NA	NA	NA	NA	224		-4 -5	-4			0.33M

PACKAGE OPTIONS AND USER I/O

			Spartan-IIE (1.8V)							Spartan-II (2.5V)						Spartan-XL (3.3V)				
			XC2S50E	XC2S100E	XC2S150E	XC2S200E	XC2S300E	XC2S400E	XC2S600E	XC2S15	XC2S30	XC2S50	XC2S100	XC2S150	XC2S200	XCS05XL	XCS10XL	XCS20XL	XCS30XL	XCS40XL
Pins	Body Size	I/O's	182	202	265	289	329	410	514	86	132	176	196	260	284	77	112	160	192	224
PLCC Packages																				
84	30 x 30 mm															61	61			
PQFP Packages (PQ)																				
208	28 x 28 mm		146	146	146	146	146				132	140	140	140	140			160	169	169
240	32 x 32 mm																		192	192
VQFP Packages (VQ)																				
100	14 x 14 mm									60	60					77	77	77	77	
TQFP Packages (TQ)																				
144	20 x 20 mm		102	102						86	92	92	92				112	113	113	
Chip Scale Packages — wire-bond chip-scale BGA (0.8 mm ball spacing)																				
144	12 x 12 mm									86	92						112	113		
280	16 x 16 mm																		192	224
FGA Packages (FT) — wire-bond fine-pitch thin BGA (1.0 mm ball spacing)																				
256	17 x 17 mm		182	182	182	182	182	182												
FGA Packages (FG) — wire-bond fine-pitch BGA (1.0 mm ball spacing)																				
256	17 x 17 mm											176	176	176	176					
456	23 x 23 mm			202	265	289	329	329	329				196	260	284					
676	27 x 27 mm							410	514											
BGA Packages																				
256	27 x 27 mm																		192	205

Note: 1. System Gates include 20-30% of CLBs used as RAM
2. Logic Cell is defined as a 4 input LUT and a register

Important: Verify all Data with Device Data Sheet (http://www.xilinx.com/spartan)

Numbers indicated in the matrix are the maximum number of user I/O's for that package and device combination.

Automotive products are highlighted:
-40C to +125C junction temperature for FPGAs

Xilinx IQ Solutions for Automotive Intelligence

Xilinx IQ Solutions

Part Number	Speed	Package	Voltage	Description
XC9500XL CPLD				
XC9536XL	10 ns/100 MHz	VQ44, VQ64	3.3V	36 Macrocells (800 Gates), ISP, JTAG, Bus Hold & I/P Hysteresis
XC9572XL	10 ns/100 MHz	VQ64, TQ100	3.3V	72 Macrocells (1,600 Gates), ISP, JTAG, Bus Hold & I/P Hysteresis
CoolRunner XPLA3				
XCR3032XL	10 ns/100 MHz	VQ44	3.3V	32 Macrocells (800 Gates), Low Power, Slew Rate Control, ISP & JTAG
XCR3064XL	10 ns/100 MHz	VQ44, VQ100	3.3V	64 Macrocells (1,600 Gates), Low Power, Slew Rate Control, ISP & JTAG
XCR3128XL	10 ns/100 MHz	VQ100, TQ144	3.3V	128 Macrocells (3,200 Gates), Low Power, Slew Rate Control, ISP & JTAG
XCR3256XL	10 ns/100 MHz	TQ144, PQ208	3.3V	256 Macrocells (6,400 Gates), Low Power, Slew Rate Control, ISP & JTAG
XCR3384XL	10 ns/100 MHz	PQ208	3.3V	384 Macrocells (9,600 Gates), Low Power, Slew Rate Control, ISP & JTAG
XCR3512XL	10 ns/100 MHz	PQ208	3.3V	512 Macrocells (12,800 Gates), Low Power, Slew Rate Control, ISP & JTAG
CoolRunner-II				
XC2C32	6 ns/145 MHz	VQ44	1.8V	32 Macrocells (800 Gates), 6 I/O Standards, Slew Rate Control, Clock Doubler, Bus Hold, I/P Hysteresis. Ultra low power.
XC2C64	7.5 ns/127 MHz	VQ44, VQ100	1.8V	64 Macrocells (1,600 Gates), 6 I/O Standards, Slew Rate Control, Clock Doubler, Bus Hold, I/P Hysteresis. Ultra low power.
XC2C128	7.5 ns/127 MHz	VQ44, VQ100	1.8V	128 Macrocells (3,200 Gates), 9 I/O Standards, Slew Rate Control, Clock Doubler, Clcok Divider, CoolClock, DataGate, Bus Hold, I/P Hysteresis. Ultra low power.
XC2C256	7.5 ns/127 MHz	VQ100, TQ144	1.8V	256 Macrocells (6,400 Gates), 9 I/O Standards, Slew Rate Control, Clock Doubler, Clcok Divider, CoolClock, DataGate, Bus Hold, I/P Hysteresis. Ultra low power.
XC2C384	10 ns/100 MHz	TQ144, PQ208	1.8V	384 Macrocells (9,600 Gates), 9 I/O Standards, Slew Rate Control, Clock Doubler, Clcok Divider, CoolClock, DataGate, Bus Hold, I/P Hysteresis. Ultra low power.
XC2C512	10 ns/100 MHz	PQ208	1.8V	512 Macrocells (12,800 Gates), 9 I/O Standards, Slew Rate Control, Clock Doubler, Clcok Divider, CoolClock, DataGate, Bus Hold, I/P Hysteresis. Ultra low power.

Note: See page 96 for CPLD IQ devices Package Options and User I/O.

Part Number	Speed Grade	Package	Voltage	Description
Spartan-XL				
XCS05XL	-4	VQ100	3.3V	Low cost FPGA with power down pin, 5V tol I/O, 5,000 Gate, 238 logic cells, 100 CLBs.
XCS10XL	-4	VQ100	3.3V	Low cost FPGA with power down pin, 5V tol I/O, 10,000 Gate, 466 logic cells, 196 CLBs.
XCS20XL	-4	TQ144, PQ208	3.3V	Low cost FPGA with power down pin, 5V tol I/O, 20,000 Gate, 950 logic cells, 400 CLBs.
XCS30XL	-4	TQ144, PQ208	3.3V	Low cost FPGA with power down pin, 5V tol I/O, 30,000 Gate, 1,368 logic cells, 576 CLBs.
XCS40XL	-4	PQ208, BG256	3.3V	Low cost FPGA with power down pin, 5V tol I/O, 40,000 Gate, 1,862 logic cells, 784 CLBs.
Spartan-II				
XC2S15	-5	TQ144	2.5V	High volume FPGA, on-chip RAM, 16 I/O standards, 15,000 Gate, 432 logic cells, 96 CLBs, 4 block RAM blocks, 4 DLLS.
XC2S30	-5	TQ144, PQ208	2.5V	High volume FPGA, on-chip RAM, 16 I/O standards, 30,000 Gate, 972 logic cells, 216 CLBs, 6 block RAM blocks, 4 DLLS.
XC2S50	-5	TQ144, PQ208, FG256	2.5V	High volume FPGA, on-chip RAM, 16 I/O standards, 50,000 Gate, 1,728 logic cells, 384 CLBs, 8 block RAM blocks, 4 DLLs.
XC2S100	-5	TQ144, PQ208, FG256	2.5V	High volume FPGA, on-chip RAM, 16 I/O standards, 100,000 Gate, 2,700 logic cells, 600 CLBs, 10 block RAM blocks, 4 DLLs.
XC2S150	-5	PQ208, FG256	2.5V	High volume FPGA, on-chip RAM, 16 I/O standards, 150,000 Gate, 3,888 logic cells, 864 CLBs, 12 block RAM blocks, 4 DLLs.
XC2S200	-5	PQ208, FG456	2.5V	High volume FPGA, on-chip RAM, 16 I/O standards, 200,000 Gate, 5,292 logic cells, 1,176 CLBs, 14 block RAM blocks, 4 DLLs.
Spartan-IIE				
XC2S50E	-6	TQ144, PQ208, FT256	1.8V	High volume FPGA, on-chip RAM, 19 I/O standards, 50,000 Gate, 1,728 logic cells, 384 CLBs, 8 block RAM blocks, 4 DLLs.
XC2S100E	-6	TQ144, PQ208, FT256	1.8V	High volume FPGA, on-chip RAM, 19 I/O standards, 100,000 Gate, 2,700 logic cells, 600 CLBs, 10 block RAM blocks, 4 DLLs.
XC2S150E	-6	PQ208, FT256	1.8V	High volume FPGA, on-chip RAM, 19 I/O standards, 150,000 Gate, 3,888 logic cells, 864 CLBs, 12 block RAM blocks , 4 DLLs.
XC2S200E	-6	PQ208, FT256	1.8V	High volume FPGA, on-chip RAM, 19 I/O standards, 200,000 Gate, 5,292 logic cells, 1,176 CLBs, 14 block RAM blocks, 4 DLLs.
XC2S300E	-6	PQ208, FG456	1.8V	High volume FPGA, on-chip RAM, 19 I/O standards, 300,000 Gate, 6,912 logic cells, 1,536 CLBs, 16 block RAM blocks, 4 DLLs.
XC2S400E	-6	FT256, FG456, FG676	1.8V	High volume FPGA, on-chip RAM,19 I/O standards, 400,000 Gate,10,800 logic cells, 2,400 CLBs, 40 block RAM blocks, 4DLLs.
XC2S600E	-6	FG456, FG676	1.8V	High volume FPGA, on-chip RAM,19 I/O standards, 600,000 Gate,10,800 logic cells, 3,456 CLBs, 72block RAM blocks, 4DLLs.

Note: See page 93 for Spartan IQ devices Package Options and User I/O.

Xilinx Configuration Storage Solutions

SystemACE	Memory Density	Number of Components	Min board space	Compression	FPGA Config. Mode	Multiple Designs	Software Storage	Removable	IRL Hooks	Max Config. Speed	Non-Volatile Media
SystemACE CF	up to 8 Gbit	2	25 cm^2	No	JTAG	Unlimited	Yes	Yes	Yes	30 Mbit/sec	CompactFlash
SystemACE MPM	16 Mbit 32 Mbit 64 Mbit	1	12.25 cm^2	Yes	SelectMAP (up to 4 FPGA) Slave-Serial (up to 8 FPGA chains)	Up to 8	No	No	Yes	152 Mbit/sec	AMD Flash Memory
SystemACE SC	16 Mbit 32 Mbit 64 Mbit	3	Custom	Yes	SelectMAP (up to 4 FPGA) Slave-Serial (up to 8 FPGA chains)	Up to 8	No	No	Yes	152 Mbit/sec	AMD Flash memory

PROM	Density	PD8	VO8	SO20	PC20	PC44	VQ44	Core Voltage	I/O Voltage 2.5V	I/O Voltage 3.3V
In-System Programming (ISP) Configuration PROMs										
XC18V256	256Kb			Y	Y		Y	3.3V	Y	Y
XC18V512	512Kb			Y	Y		Y	3.3V	Y	Y
XC18V01	1Mb			Y	Y		Y	3.3V	Y	Y
XC18V02	2Mb					Y	Y	3.3V	Y	Y
XC18V04	4Mb					Y	Y	3.3V	Y	Y
One-Time Programmable (OTP) Configuration PROMs										
XC17V01	1.6Mb		Y	Y	Y			3.3V	Y	Y
XC17V02	2Mb				Y	Y	Y	3.3V	Y	Y
XC17V04	4Mb				Y	Y	Y	3.3V	Y	Y
XC17V08	8Mb					Y	Y	3.3V	Y	Y
XC17V16	16Mb					Y	Y	3.3V	Y	Y

FPGA	PROM Solution	PD8	VO8	SO20	PC20	PC44	VQ44	Core Voltage	I/O Voltage 2.5V	I/O Voltage 3.3V
OTP Configuration PROMs for Spartan-II/IIE										
XC2S50E	XC17S50A	Y	Y	Y				3.3V	Y	Y
XC2S100E	XC17S100A	Y	Y	Y				3.3V	Y	Y
XC2S150E	XC17S200A	Y	Y				Y	3.3V	Y	Y
XC2S200E	XC17S200A	Y	Y				Y	3.3V	Y	Y
XC2S300E	XC17V04				Y	Y	Y	3.3V	Y	Y
XC2S400E	XC17V04				Y	Y	Y	3.3V	Y	Y
XC2S600E	XC17V04				Y	Y	Y	3.3V	Y	Y
XC2S15	XC17S15A	Y	Y	Y				3.3V	Y	Y
XC2S30	XC17S30A	Y	Y	Y				3.3V	Y	Y
XC2S50	XC17S50A	Y	Y	Y				3.3V	Y	Y
XC2S100	XC17S100A	Y	Y	Y				3.3V	Y	Y
XC2S150	XC17S150A	Y	Y	Y				3.3V	Y	Y
XC2S200	XC17S200A	Y	Y				Y	3.3V	Y	Y

FPGA	PROM Solution	PD8	VO8	SO20	PC20	PC44	VQ44	Core Voltage	I/O Voltage 3.3V	I/O Voltage 5.0V
OTP Configuration PROMs for Spartan-XL										
XCS05XL	XC17S05XL	Y	Y					3.3V	Y	Y
XCS10XL	XC17S10XL	Y	Y					3.3V	Y	Y
XCS20XL	XC17S20XL	Y	Y					3.3V	Y	Y
XCS30XL	XC17S30XL	Y	Y					3.3V	Y	Y
XCS40XL	XC17S40XL	Y		Y				3.3V	Y	Y

Xilinx Home Page
http://www.xilinx.com

Xilinx Online Support
http://www.xilinx.com/support/support.htm

Xilinx IP Center
http://www.xilinx.com/ipcenter/index.htm

Xilinx Education Center
http://www.xilinx.com/support/education-home.htm

Xilinx Tutorial Center
http://www.xilinx.com/support/techsup/tutorials/index.htm

Xilinx WebPACK
http://www.xilinx.com/sxpresso/webpack.htm

Xilinx CPLD Product Selection Matrix

PRODUCT SELECTION MATRIX

						I/O Features		Speed			Clocking	
	System Gates	Macrocells	Product terms per Macrocell	Input Voltage Compatible	Output Voltage Compatible	Max. I/O	I/O Banking	Min. Pin-to-Pin Logic Delay (ns)	Commercial Speed Grades (fastest to slowest)	Industrial Speed Grades (fastest to slowest)	Global Clocks	Product Term Clocks per Function Block
CoolRunner-II Family — 1.8 Volt												
XC2C32	750	32	40	1.5/1.8/2.5/3.3	1.5/1.8/2.5/3.3	33	1	3.5	-3 -4 -6	-6	3	17
XC2C64	1500	64	40	1.5/1.8/2.5/3.3	1.5/1.8/2.5/3.3	64	1	4	-4 -5 -7	-7	3	17
XC2C128	3000	128	40	1.5/1.8/2.5/3.3	1.5/1.8/2.5/3.3	100	2	4.5	-4 -6 -7	-7	3	17
XC2C256	6000	256	40	1.5/1.8/2.5/3.3	1.5/1.8/2.5/3.3	184	2	5	-5 -6 -7	-7	3	17
XC2C384	9000	384	40	1.5/1.8/2.5/3.3	1.5/1.8/2.5/3.3	240	4	6	-6 -7 -10	-10	3	17
XC2C512	12000	512	40	1.5/1.8/2.5/3.3	1.5/1.8/2.5/3.3	270	4	6	-6 -7 -10	-10	3	17
CoolRunner XPLA3 Family — 3.3 Volt												
XCR3032XL	750	32	48	3.3/5	3.3	36		5	-5 -7 -10	-7 -10	4	16
XCR3064XL	1500	64	48	3.3/5	3.3	68		6	-6 -7 -10	-7 -10	4	16
XCR3128XL	3000	128	48	3.3/5	3.3	108		6	-6 -7 -10	-7 -10	4	16
XCR3256XL	6000	256	48	3.3/5	3.3	164		7.5	-7 -10 -12	-10 -12	4	16
XCR3384XL	9000	384	48	3.3/5	3.3	220		7.5	-7 -10 -12	-10 -12	4	16
XCR3512XL	12000	512	48	3.3/5	3.3	260		7.5	-7 -10 -12	-10 -12	4	16
XC9500XV Family — 2.5 Volt												
XC9536XV	800	36	90	2.5/3.3	1.8/2.5/3.3	36	1	5	-5 -7	-7	3	18
XC9572XV	1600	72	90	2.5/3.3	1.8/2.5/3.3	72	1	5	-5 -7	-7	3	18
XC95144XV	3200	144	90	2.5/3.3	1.8/2.5/3.3	117	2	5	-5 -7	-7	3	18
XC95288XV	6400	288	90	2.5/3.3	1.8/2.5/3.3	192	4	6	-6 -7 -10	-7 -10	3	18
XC9500XL Family — 3.3 Volt												
XC9536XL	800	36	90	2.5/3.3/5	2.5/3.3	36		5	-5 -7 -10	-7 -10	3	18
XC9572XL	1600	72	90	2.5/3.3/5	2.5/3.3	72		5	-5 -7 -10	-7 -10	3	18
XC95144XL	3200	144	90	2.5/3.3/5	2.5/3.3	117		5	-5 -7 -10	-7 -10	3	18
XC95288XL	6400	288	90	2.5/3.3/5	2.5/3.3	192		6	-6 -7 -10	-7 -10	3	18

PACKAGE OPTIONS AND USER I/O

		CoolRunner-II						CoolRunner XPLA3						XC9500XV				XC9500XL			
	Body Size	XC2C32	XC2C64	XC2C128	XC2C256	XC2C384	XC2C512	XCR3032XL	XCR3064XL	XCR3128XL	XCR3256XL	XCR3384XL	XCR3512XL	XC9536XV	XC9572XV	XC95144XV	XC95288XV	XC9536XL	XC9572XL	XC95144XL	XC95288XL
PLCC Packages (PC)																					
44	16.5 x 16.5 mm	33	33					36	36					34	34			34	34		
PQFP Packages (PQ)																					
208	28 x 28 mm				173	173	173				164	172	180				168				168
VQFP Packages (VQ)																					
44	12 x 12 mm	33	33					36	36					34	34			34	34		
64	12 x 12 mm																	36	52		
100	16 x 16 mm		64	80	80				68	84											
TQFP Packages (TQ)																					
100	14 x 14 mm														72	81			72	81	
144	20 x 20 mm			100	118	118				108	120	118*				117	117			117	117
Chip Scale Packages (CP) — wire-bond chip-scale BGA (0.5 mm ball spacing)																					
56	6 X 6 mm	33	45						48												
132	8 X 8 mm			100	106																
Chip Scale Packages (CS) — wire-bond chip-scale BGA (0.8 mm ball spacing)																					
48	7 x 7mm							36	40					36	38			36	38		
144	12 x 12 mm									108						117				117	
280	16 x 16 mm										164						192				192
BGA Packages (BG) — wire-bond standard BGA (1.27 mm ball spacing)																					
256	27 x 27 mm																				192
FGA Packages (FT) — wire-bond fine-pitch thin BGA (1.0 mm ball spacing)																					
256	17 x 17 mm				184	212	212				164	212	212								
FBGA Packages (FG) — wire-bond Fineline BGA (1.0 mm ball spacing)																					
256	17 x 17 mm																192				192
324	23 x 23 mm					240	270					220	260								

*JTAG pins and port enable are not pin compatible in this package for this member of the family.

Important: Verify all Data with Device Data Sheet and Product Availability with your local Xilinx Rep

Automotive products are highlighted: -40C to +125C ambient temperature for CPLDs

Xilinx IQ Solutions for Automotive Intelligence

Xilinx Software

	Feature	ISE *WebPACK*	ISE *BaseX*	ISE *Foundation*	ISE *Alliance*
Devices	Virtex™Series	**Virtex-E:** V50E – V300E **Virtex-II:** 2V40 – 2V250 **Virtex-II Pro:** 2VP2	**Virtex:** V50 – V300 **Virtex-E:** V50E – V300E **Virtex-II:** 2V40 – 2V250 **Virtex-II Pro:** 2VP2	ALL	ALL
	Spartan™ II/IIE Families	ALL	ALL	ALL	ALL
	CoolRunner™ XPLA3 / CoolRunner-II	ALL	ALL	ALL	ALL
	XC9500™ Series	ALL	ALL	ALL	ALL
Services	Educational Services	Yes	Yes	Yes	Yes
	Design Services	Sold as an Option	Sold as an Option	Sold as an Option	Sold as an Option
	Support Services	Web Only	Yes	Yes	Yes
Design Entry	Schematic Editor	Yes	Yes	PC Only	No
	HDL Editor	Yes	Yes	Yes	Yes
	State Diagram Editor	Yes	Yes	PC Only	No
	CORE Generator System	No	Yes	Yes	Yes
	PACE (Pinout and Area Constraint Editor)	Yes	Yes	Yes	Yes
	Architecture Wizards DCM – Digital Clock Management MGT – Multi-Gigabit Transcievers	Yes	Yes	Yes	Yes
	3rd Party RTL Checker Support	Yes	Yes	Yes	Yes
	Xilinx System Generator for DSP	No	Sold as an Option	Sold as an Option	Sold as an Option
Embedded System Design	GNU Embedded Tools GCC – GNU Compiler GDB – GNU Software Debugger	No	No	Yes	Yes
	WindRiver Xilinx Editon Development Tools Diab C/C++ Compiler SingleStep Debugger visionPROBE II target connection	No	No	Sold as an Option	Sold as an Option
Synthesis	Xilinx Synthesis Technology (XST)	Yes	Yes	Yes	No
	Synplicity Synplify/Pro	Integrated Interface	Integrated Interface	Integrated Interface (PC Only)	Integrated Interface (PC Only)
	Synplicity Amplify Physical Synthesis Support	Yes	Yes	Yes	Yes
	Leonardo Spectrum	Integrated Interface	Integrated Interface	Integrated Interface	Integrated Interface
	Synopsys FPGA Compiler II	EDIF Interface	EDIF Interface	EDIF Interface	EDIF Interface
	ABEL	CPLD	CPLD	CPLD (PC Only)	No
Implementation	iMPACT	Yes	Yes	Yes	Yes
	FloorPlanner	Yes	Yes	Yes	Yes
	Xilinx Constraints Editor	Yes	Yes	Yes	Yes
	Timing Driven Place & Route	Yes	Yes	Yes	Yes
	System ACE Configuration Manager	Yes	Yes	Yes	Yes
	Modular Design	No	Sold as an Option	Sold as an Option	Sold as an Option
	Timing Improvement Wizard	Yes	Yes	Yes	Yes
Board Level Integration	IBIS Models	Yes	Yes	Yes	Yes
	STAMP Models	Yes	Yes	Yes	Yes
	LMG SmartModels	Yes (Available from Synopsis)	Yes (Available from Synopsis)	Yes (Available from Synopsis)	Yes (Available from Synopsis)
	HSPICE Models*	Yes	Yes	Yes	Yes
Verification	HDL Bencher™	Yes	Yes	PC Only	No
	ModelSim® Xilinx Edition (MXE II)	ModelSim XE II Starter**	ModelSim XE II Starter**	ModelSim XE II Starter**	ModelSim XE II Starter**
	Static Timing Analyzer	Yes	Yes	Yes	Yes
	ChipScope PRO	No	Sold as an Option	Sold as an Option	Sold as an Option
	FPGA Editor with Probe	No	Yes	Yes	Yes
	ChipViewer	Yes	Yes	Yes	Yes
	XPower (Power Analysis)	Yes	Yes	Yes	Yes
	3rd Party Equivalency Checking Support	Yes	Yes	Yes	Yes
	SMARTModels for PPC and Rocket I/O	No	Yes	Yes	Yes
	3rd Party Simulator Support	Yes	Yes	Yes	Yes
Platforms		PC Only	PC, Sun Solaris, Linux	PC, Sun Solaris, Linux	PC, Sun Solaris, Linux
IP/CORE	For more information on the complete list of Xilinx IP products, visit the Xilinx IP Center at http://www.xilinx.com/ipcenter				

* HSPICE Models available at the Xilinx Design Tools Center at www.xilinx.com/ise.

** MXE II supports the simulation of designs up to 1 million system gates and is sold as an option. For more information, visit the Xilinx Design Tools Center at **www.xilinx.com/ise**

Xilinx Global Services

Xilinx Global Services

Xilinx Design Services (XDS)

- XDS provides extensive FPGA hardware and embedded software design experience backed by industry recognized experts and resources to solve even the most complex design challenge.
- **System Architecture Consulting** — Provide engineering services to define system architecture and partitioning for design specification.
- **Custom Design Solutions** — Project designed, verified, and delivered to mutually agreed upon design specifications.
- **IP Core Development, Optimization, Integration, Modification, and Verification** — Modify, integrate, and optimize customer intellectual property or third party cores to work with Xilinx technology. Develop customer-required special features to Xilinx IP cores or third party cores. Perform integration, optimization, and verification of IP cores in Xilinx technology.
- **Embedded Software** — Develop complex embedded software with real-time constraints, using hardware/software co-design techniques.
- **Conversions** — Convert ASIC designs and other FPGAs to Xilinx technology and devices.

Part Number	Product Description	Duration	Availability
Education Services		**hours**	
FPGA13000-5-ILT	Fundamentals of FPGA Design	8	Now
FPGA23000-5-ILT	Designing for Performance	16	Now
FPGA33000-5-ILT	Advanced FPGA Implementation	16	Now
LANG11000-5-ILT	Introduction to VHDL	24	Now
LANG21000-5-ILT	Advanced VHDL	16	Now
LANG12000-5-ILT	Introduction to Verilog	24	Now
PCI18000-5-ILT	PCI CORE Basics	8	Now
PC128000-5-ILT	Designing a PCI System	16	Now
ASIC25000-5-ILT	Designing for Performance for the ASIC User	24	Now
DSP2000-3-ILT	DSP Implementation Techniques for Xilinx FPGAs	24	Now
DSP10000-4-ILT	DSP Design Flow	24	Now
RIO22000-5-ILT	Design with Rocket I/O Multi-Gigabit Transceiver	16	Now
PROMO-5004-5-ILT	FPGA Essentials	15	Now
PROMO-5003-5-LEL	Designing for Performance, Live Online	9	Now
Platinum Technical Service			
SC-PLAT-SVC-10	1 Seat Platinum Technical Service w/10 education credits	N/A	Now
SC-PLAT-SITE-50	Platinum Technical Service site license up to 50 customers	N/A	Now
SC-PLAT-SITE-100	Platinum Technical Service site license for 51-100 customers	N/A	Now
SC-PLAT-SITE-150	Platinum Technical Service site license for 101-150 customers	N/A	Now
Titanium Technical Service			
PS-TEC-SERV	Titanium Technical Service (minimum 40 hours)	N/A	Now
Design Services			
DC-DES-SERV	Design Services Contract	N/A	Now
Xilinx Productivity Advantage			
DS-XPA	Custom XPA Packaged Solution	N/A	Now
DS-ISE-ALI-XPA	XPA Seat, ISE Alliance	N/A	Now
DS-ISE-FND-XPA	XPA Seat, ISE Foundation	N/A	Now

Titanium Technical Service

- Dedicated application engineer at your site or ours
- Raise skill level of your design engineers
- Increase your Xilinx design knowledge
- Minimize risk

Platinum Technical Service

- Access to Senior Applications Engineers
- Dedicated Toll Free Number*
- Priority Case Resolution
- Ten Education Credits
- Electronic Newsletter
- Formal Escalation Process
- Service Packs and Software Updates
- Application Engineer to Customer Ratio, 2x Gold Level

**Toll free number available in US only, dedicated local numbers available across Europe*

Education Services Contacts

North America: 877-XLX-CLAS (877-959-2527)
http://support.xilinx.com/support/training/training.htm

Europe: +44-870-7350-548
eurotraining@xilinx.com

Japan: +81-03-5321-7750
http://support.xilinx.co.jp/support/education-home.htm

Asia Pacific: +03-5321-7711
http://support.xilinx.com/support/education-home.htm

Design Services Contacts

North America & Asia:
Richard Fodor: 408-626-4256
Mike Barone: 512-238-1473

Europe:
Alex Hillier: +44-870-7350-516
Martina Finnerty: +353-1-4032469

designservices@xilinx.com

XPA Contacts

North America: 800-888-FPGA (3742)
fpga.xilinx.com

Europe:
Stuart Elston: +44-870-7350-532

Titanium Technical Service Contacts

North America:
Telesales: 1-800-888-3742

Europe:
Stuart Elston: +44-870-7350-532

The most complete environment for logic, embedded, and system design

The Xilinx ISE 5.1i release takes performance to new heights. With a single design environment supporting all leading Xilinx silicon devices, ISE delivers clock rates exceeding 300MHz, and compile times 6X faster than our nearest competitor. Whether it's embedded processors, logic or system level design, only Xilinx gives you all the speed you need.

Finish faster with the industry's highest performance solutions

Our intuitive Architecture Wizard simplifies high speed communication design. Now you can easily take advantage of the programmable RocketIO™ interfaces and the advanced Digital Clock Management system in our Virtex-II Pro™ Platform FPGA. It's unique features like these that enable fast and flexible support of PCI Express, 10G Ethernet XAUI and all emerging standards. Finish fast. Finish first. Every time.

Incremental Design means change without risk

ISE 5.1i has the industry's only true incremental design capability. Now, the time from HDL editing to debugging is only minutes, not hours - even for our largest Virtex-II Pro devices. That means you can make those last minute changes and still make your deadline.

Visit *www.xilinx.com/faster* today and find out why ISE is the number one choice for programmable systems design.

XILINX®
The Programmable Logic Company℠

www.xilinx.com/faster

FORTUNE® 2002
100 BEST COMPANIES TO WORK FOR

PN 0010683