Issue 33
Third Quarter
1999

Xcell

PRODUCT INFORMATION

New - CoolRunner CPLDs

DEVELOPMENT SYSTEMS

New - Alliance/Foundation Series 2.1i

New - WebPACK and WebFITTER

APPLICATIONS

ADSL Modem

Unusual Clock Dividers

COVER STORY

Xilinx Online technology allows you to reconfigure your designs remotely

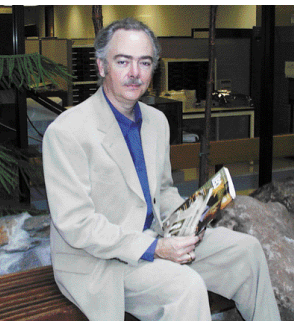
COLUMNS

Industry Analyst - New!

HDL Advisor

Applications Q & A

FROM THE EDITOR



Xcell

ilinx, Inc.
L00 Logic Drive
an Jose, CA 95124-3450
hone: 408-559-7778
AX: 408-879-4780

cell is published quarterly. XILINX,the linx logo, and CoolRunner are registered trademarks of Xilinx,Inc. Virtex, giCORE,IRL,Spartan,SpartanXL, liance Series, Foundation Series, CORE enerator, XPLA, Fast Zero Power, ebFITTER, WebPACK,ChipViewer, elect RAM,Block Ram,Xilinx Online, nd all XC-prefix products are trademarks, and The Programmable Logic ompany is a service mark of Xilinx,Inc. ther brand or product names are trademarks or registered trademarks of their spective owners.

Xilinx Online - The End of Single-use Hardware

We are on the verge of an exciting new paradigm in logic design—one that could forever change not only how you design but also change what you design.

A computer can "become" a word processor, a communication terminal, a calculator, a video game, or an almost unlimited number of other machines, simply by loading in a new machine description or "program." It is this reprogrammable aspect of computers that make them so useful. Now imagine that the hardware of your computer (or any other device) is just as programmable as the software; that the machine itself can "morph" into almost any function, instantly adapting to new requirements, and eventually becoming things not yet dreamed of. Imagine that this upgradable system is connected to the Internet, allowing it to be automatically modified, remotely, as often as needed. The implications are staggering; the possibilities are unlimited; it's called Xilinx Online™.

This futuristic capability, a combination of several technologies, has just arrived, yet it's already beginning to have a dramatic impact on the way new systems are designed. These enabling technologies include:

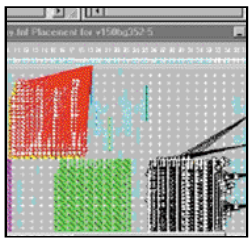
- Device Architecture - The Xilinx Virtex™ FPGA family now has the speed, density, and system-level features you need to create complete systems in programmable logic, systems that are easily modified remotely.
- Process Technology - With our latest deep sub-micron manufacturing technologies, Xilinx programmable logic devices are not only much more capable, they are much less expensive as well, opening many new applications that once required custom, inflexible ASICs.
- Development Tools - With our tools you can create very large, complex designs, and then simulate and debug them quickly and easily. Plus, with our new team-based design capabilities, multiple designers, in separate locations, can easily collaborate.
- Intellectual Property - Many new cores are being developed every day, giving you a low cost headstart on your next design. Xilinx LogiCOREs and third party AllianceCOREs also give you the advantage of fast, predictable performance, no matter where the cores are placed, in any combination, saving you a lot of time and effort.
- Networks - The Internet is everywhere, which means that you have a standard, built-in, infrastructure for remotely reprogramming your designs. Through the Internet (or any communication medium) you can repair, upgrade, or enhance existing equipment, saving you and your customers a lot of time and expense.
- Software Enabling Technologies - The Java™ language enables Xilinx to create universal applications for remotely programming, testing, and verifying your designs. These tools make it easy for you to manage your systems in the field.

Xilinx Online presents the obvious next step in the evolution of logic design, and this issue of Xcell shows you what it's all about.

ARTICLES

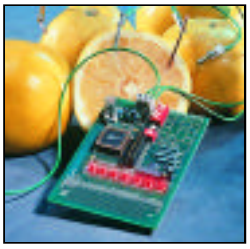
COVER STORY 4

Xilinx Online - A revolution in logic design. With our latest technology, you can create unique new systems that can be remotely reconfigured and maintained at your customers' premises, anywhere in the world.



DEVELOPMENT SYSTEMS 14

New 2.1i software – shortens design cycles and improves productivity. The latest release of the Xilinx Alliance Series and Foundation Series software is now available.



PRODUCT INFORMATION 19

Xilinx acquires the CoolRunner® Line of CPLDs - combining very low power with high speed, high density, and high I/O counts.

APPLICATIONS 24

Low cost Spartan FPGAs used in ADSL modems - managing DMA transfers and implementing the complex system-level glue logic required for the USB interface.

COLUMNS 32

New Industry Analyst column - programmable logic continues to outpace the overall electronic industry growth.

Inside This Issue:

Xcell

Xilinx Online-

A Revolution in Logic Design

With our latest technology, you can create unique new systems that can be remotely reconfigured, upgraded, and maintained at your customers' premises, anywhere in the world.

by Wim Roelandts,
CEO, Xilinx

Xilinx Online represents a whole new concept in logic design. Now for the first time, it's very practical and cost-effective to design hardware that easily adapts to a changing environment; the advantages of this new technology are enormous and far reaching. This capability is expected to soon replace many conventional fixed-logic designs because the benefits are very compelling.

New Possibilities

Using the reconfigurable logic capability of our Virtex FPGAs, and our new software technologies that enable its use, you can create unique new designs such as:

- **Systems that can accept new features and bug fixes over any network, including the Internet.** For example, imagine shipping new systems, even before their final specifications are complete. This is very useful, because in our industry there is an increasing level of competition between many new standards, for everything from interface specifications to communication protocols. This competition between emerging standards creates open-ended delays for new product introductions and it can significantly decrease the life span of your designs. However, using Xilinx Online technologies, you can enter the market far ahead of your competition and easily adapt to the changing standards, long after your systems are installed.
- **Systems that adapt themselves to a changing environment, on-the-fly.** For example, imagine MPEG decoders or pattern recognition devices that can automatically adapt their algorithms to match the quality of the incoming data. Plus, you can download new algorithms, as they are developed, without interrupting system operation.

In the future it will become even more crucial for you to have the capability of adapting to many different evolving standards and changing environments. If you wait to see which competing standards win, or to see which options your customers are buying, you run the risk of always being behind the technology curve, and your product life span (and profitability) will be significantly reduced.

By creating reconfigurable hardware, you will extend the life and profitability of your systems, and you can also generate new, on-going revenue from existing systems. For example, its not difficult to imagine "universal" Internet-based appliances that can become whatever your customer chooses (and is willing to pay for). With this type of device you can continue to create and sell new designs, long after your initial system was purchased, without shipping any new hardware.

(Continued)

It's Happening Now

Researchers have been investigating this concept for years, and a few forward-looking companies have already created FPGA-based products that can be reconfigured from afar. IBM, for example, currently markets an ATM switch whose FPGA-based logic can be changed over the network to bring it into accord with the latest changes in the ATM standard. While such reconfigurable hardware is not yet mainstream, a number of Xilinx customers—including large communications companies—are very interested in the idea and are well along the way with major designs efforts.

The types of systems that could benefit from being field upgradable are wide-ranging. Almost any system that has some type of connectivity to the "outside world" can benefit from the Xilinx Online technologies. Typical products include network appliances, set-top boxes, security systems, network diagnostic equipment, cellular base stations, and satellite communications systems. Other likely applications are HDTV, video and image processing, encryption, military communications, surveillance, radar, and sonar.

The Enabling Technologies

Three enabling technologies have recently come together to make it possible for you to easily upgrade your hardware remotely:

- **Pervasive networks such as the Internet** provide the infrastructure that allows you to easily communicate with any type of system, anywhere in the world. Through networks (or any type of communication medium), new FPGA designs can be downloaded and tested, remotely.
- **The Java language** makes it easy to implement universal software applications. Using Java we can create applets that perform the necessary functions of transporting, programming, and verifying FPGA designs, on any type of host system. A Java "virtual machine" can also be implemented in an FPGA if necessary, so a separate microprocessor is not required.
- **The new system-level, Virtex FPGAs from Xilinx** now have enough speed and capacity to implement high performance applications. Our advanced sub-micron fabrication technologies have enabled us to significantly reduce our prices while increasing density and performance.

Our Virtex family was designed from the beginning to allow complete or partial reconfigurability. This means that you can change part of the FPGA while the other parts are still running. These features, along with abundant on-chip RAM (of various types), advanced clocking capability (using Delay Locked Loops), and support for many different I/O types (up to eight different standards simultaneously), makes the Virtex family the perfect choice for remotely upgradable applications.

Most systems today already come with some form of built-in communications or microprocessor interface, making the addition of remote field update capability a simple matter. And, many new tools are being developed that will make it even easier for you to create field upgradable products.

Summary

The capability to remotely upgrade and debug your systems brings many compelling advantages over conventional fixed-logic designs; you can get to the market much sooner, stay in the market much longer, and sell new features as they are developed. Clearly, manufacturers who create remotely upgradable systems today will be the ones who lead their markets in the not-too-distant future.

By incorporating Xilinx Online capability, your equipment is cheaper to maintain, and it will not become prematurely obsolete—benefits that your customers will appreciate and pay for. Σ

Industry Leaders Discuss the Role of the Internet and Programmable Logic

Xilinx Webcasts from DAC '99 present the views of industry leaders.

by Ann Duft, Public Relations Manager, Xilinx,
ann.duft@xilinx.com

This year at the Design Automation Conference (DAC) '99 in New Orleans, Xilinx presented Webcasts of prominent leaders in our industry—another industry first. There were talks by John Chambers (CEO of Cisco Systems), Scott McNealy (CEO of Sun Microsystems), Aart de Geus (CEO of Synopsys), Geoffrey Moore (author of Crossing the Chasm), and Wim Roelandts (CEO of Xilinx).

Geoffrey Moore spoke on the emerging system-on-a-chip market. "One of the most exciting things that Xilinx has brought to the market is this notion of a reprogrammable chip that can be functionally changed and modified after shipping. This is particularly attractive in my view for people in the hand-held device markets, notably wireless phones, but increasingly in the Internet appliance market currently infiltrating our lives.

"The challenge for the systems provider, who is deploying a service, is that the appliances become an expensive consumable. God forbid there's a bug in one of these things and you have to do a recall on the product. What's exciting to me about the Xilinx offering is that as the technology moves forward and the customer wishes to increase their subscription in a service, someone at a remote systems management monitoring console can actually reprogram the device, live, without the customer having to do anything.

"This is going to be extremely compelling to the systems houses. And these houses with hand-held devices that will need modification, may very well be the "crossing the chasm" target segment that pulls the system-on-the-chip market into the marketplace. In that case, I would see Xilinx providing the platform product."

John Chambers compared the Internet revolution to the industrial revolution. Plus he offered a five-year look ahead. "Voice will almost ride for free in a connection with voice and video over a single network. Imagine the challenges the major phone companies will have when 90 percent of their revenues and profits will become commodities and be free. The Internet will level the playing field between big companies and small companies. There will be a globalization of companies and resources at a pace that we're just beginning to understand. Almost every electronic device in our home, in our work, in our cars, or even on our bodies will be networked."

Chambers noted that the Internet will change the attitudes of business and the business principles; it will be viewed as the competitive advantage; change and rapid adaptation to change will become key requirements for corporate culture. It will be an era where business must lead and government must follow. In short it will change everything. To quote their primetime commercial, he asks, "Are you ready?"

When asked about the importance of programmable logic in his future product plans, Chambers responded, "Very simply, it's extremely important to our future in terms of cost-of-ownership and flexibility for our customers."Σ

Xilinx Receives Three *IRL* Industry Awards

The Xilinx IRL methodology combines computer networks, the Java programming language, and the new Xilinx Virtex FPGAs, to create a new class of electronic systems that can be fixed, modified, or updated after installation at the end users' premises.

by Mike Seither, Director of
Public Relations, Xilinx,
mike.seither@xilinx.com

Internet Reconfigurable Logic (IRL™), the new methodology that enables you to create Xilinx Online upgradable systems, continues to win awards for innovation from the electronics industry.

Electronique

The latest IRL award came from Electronique, the monthly magazine of record for the electronics industry in France. The magazine ranked the IRL methodology as one of the industry's best innovations for 1998. In its June 1999 issue, the magazine called the approach for creating Xilinx Online upgradable systems "a great leap forward for programmable logic."

Xilinx won top honors from Electronique for active components, one of several product categories recognized for excellence. An independent panel of customers, consultants, and design services judged more than 80 products nominated for the magazine's annual awards. Xilinx accepted the award at a recognition dinner in June in Paris.

International Engineering Consortium

Citing market impact and customer benefit, the International Engineering Consortium selected the Xilinx IRL methodology as a winner of the organization's 1999 InfoVision Award. Xilinx was a winner in the Internet category and will be recognized with other winners at an award ceremony in October in Chicago during the National Communications Forum, a networking and communications conference.

The International Engineering Consortium is a cooperative, public service organization dedicated to positive change in the information industry and university communities. For more than 50 years, the IEC has provided educational opportunities for industry professionals and promising students.

Electronique International Hebdo

Last November, Electronique International Hebdo, the Paris-based weekly newspaper for the electronics business in France, chose the Xilinx IRL methodology as the top technical achievement in the semiconductor industry for 1998. The publication said that a majority of electronic products would be using the Xilinx IRL technology in the 21st Century. These Xilinx Online upgradable systems can range from multi-use set-top boxes and wireless telephone cellular base stations to communications satellites and network management systems.

"We're once again honored to receive this recognition for our IRL methodology, which has clearly resonated with the design engineering community," said Xilinx president and CEO Wim Roelandts. "Since we announced our unique approach for expanding the reach of programmable logic, there has been a growing interest in this exciting technology. IRL is the backbone for creating Xilinx Online upgradable systems, and we expect these products to provide tremendous new levels of flexibility for our customers' customers." Σ

Who's Using Virtex and Spartan FPGAs *in Xilinx Online Applications?*

Though it was only recently introduced, Xilinx Online technology is already being used by some leading-edge companies to create unique new field upgradable systems.

by Wallace Westfeldt,
Marketing Manager IRL, Xilinx,
wallace@xilinx.com

Many companies are now taking advantage of the enormous benefits provided by the Xilinx Online program, which was created to enable, identify, and promote field upgradable applications based upon the award winning Internet Reconfigurable Logic (IRL) methodology from Xilinx. These field upgradable applications are defined as being those that are connected to a private or public network (such as the Internet), with the ability to be updated, fixed, or modified after they have been deployed in the field. Such applications are being widely developed today by a growing number of leading-edge companies; many use Virtex devices for their size, speed, and flexibility; others use Spartan FPGAs for their lower cost.

Apex

Apex is one of the fastest growing network companies; they were recently ranked fifth in Business Week's list of "100 Hottest Growth Companies," and the company has been in the top five of this list for three consecutive years. Apex uses SpartanXL FPGAs in their latest remote server management product introduced in August.

The Emerge 2000™ system from Apex is a breakthrough remote server device that can dynamically update its performance characteristics depending on the workload and systems to which it is connected. "The Emerge family addresses several needs regarding remote server management," said Kevin J. Hafer, president and CEO of Apex. "IT managers require speed, security, ease-of-use, and the ability to manage servers without burdening server or network performance. Our team worked very closely with Xilinx to ensure that Emerge 2000 meets all these key requirements."

Emerge 2000 is expected to be a key cost-reduction component of server management strategies for data centers, Internet service providers, and "server farms" of all types, especially for round-the-clock mission-critical applications. It allows operators to manage large numbers of servers connected to local area networks, wide area networks, or the Internet using only a single console, mouse and key-

APEX Emerge 2000

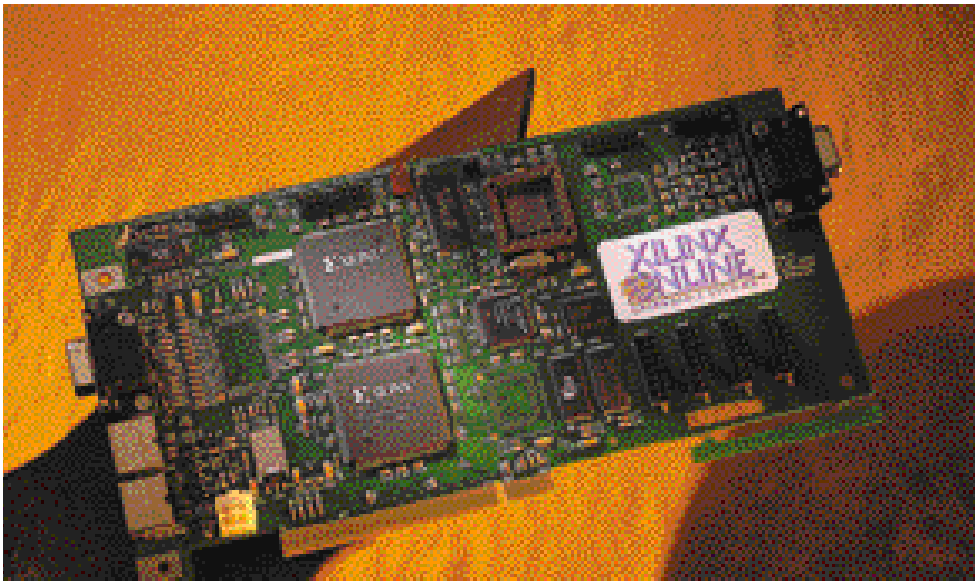


Figure 1

board, without the need to load software or hardware directly on each server.

Emerge 2000 customers can access racks of servers from remote locations through dial-in modems, networks, or Internet protocol addresses. When combined with Apex's OutLook® and ViewPoint® console switches, Emerge 2000 permits system administrators to remotely switch among, view, and control up to hundreds of servers.

Emerge 2000 can be reconfigured to support the operating system, BIOS, video subsystem, keyboard map, or mouse pointing system of a given server, and it provides compatibility with the display resolution, color depth, and refresh rate parameters for the server's console. Additionally, Emerge supports various physical connections between the console and the server farm, ranging from broadband optical and T-1 to dial-up modem.

The key element of Emerge is a compactly designed PCI circuit board (Figure 1). One of its Xilinx SpartanXL FPGAs serves as a 32-bit, 33-MHz PCI interface (based on the Xilinx Real PCI LogiCORE) and a video signal processor. The other SpartanXL FPGA is reconfigured dynamically, with digital signal processing algorithms, to overcome problems associated with remote access. The Emerge 2000 system swaps communications optimization algorithms in the SpartanXL FPGA almost every time it switches to another server. The reconfiguration data is stored on disk and read into the SpartanXL FPGA in milliseconds.

"Apex is to be congratulated on its forward looking design of the Emerge 2000 and its use of the Xilinx Internet Reconfigurable Logic methodology to create advanced server management products," said Xilinx CEO Wim Roelandts. "The Emerge 2000 systems exemplifies the beginning of a new generation of Xilinx Online field upgradable applications whose FPGA-based hardware can be modified after installation at the customer's premises."

TSI TelSys

"Our customers include many of the world's space agencies and leading aerospace firms," said Toby Bennett, Vice President, TSI TelSys. "Their satellite ground station facilities, which utilize our protocol agile communication products, are often located in remote, inhospitable regions such as near the North Pole. TSI TelSys develops Xilinx Online applications so that our systems can be easily maintained and updated over the Internet." Figure 2 shows a TSI TelSys board.

NDS

NDSalready makes great useof field upgrades," said John Simmons, Project Leader, NDS, Ltd., speaking on their downloading procedures for making post-installation upgrades. "For example, all our FPGA design data is stored in flash memory and the flash is upgradable via the Ethernet. Field upgradability is essential to our success. It allows fast, cheap resolutions to the rare problems that we get in the field, and is a delight to our customers." NDS provides broadcast systems for HDTV.

Xilinx Online Reference Designs on the Web

In addition to these customer examples, Xilinx continues to enable field upgradable applications by providing, information, discussion, and reference designs through the Xilinx Online website (http://www.xilinx.com/xilinxonline/). Recently published on this website is a description of an Internet-upgradable application for Virtex, based on the WildCard general purpose reconfiguration card from Annapolis Microsystems.

This application, demonstrated at DAC '99, allows users to select Photoshop filters (configured for a Virtex FPGA) from an Internet appliance website, and securely download them using Java-based technology. These filters, are downloaded to a laptop and then programmed

(Continued)

TSI TelSys Board

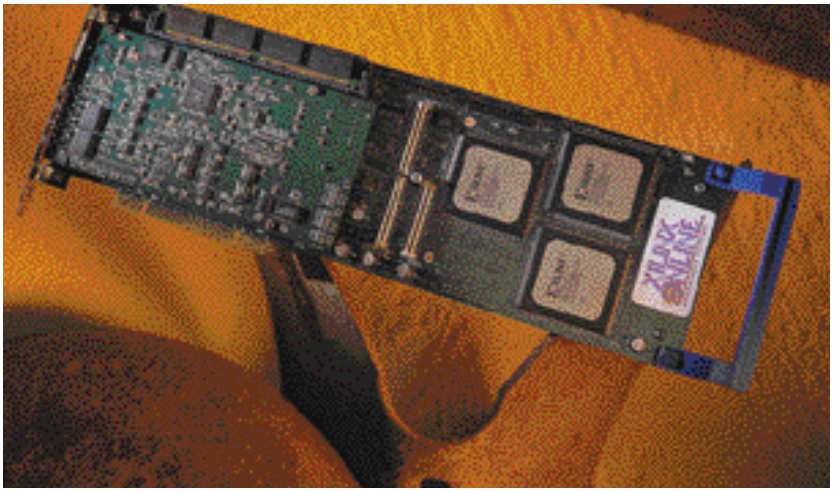


Figure 2

into a Virtex FPGA on a PCMCIA card, for high speed graphics processing.

"In this demonstration application, which provides hardware acceleration for Photoshop filters, we wanted to show a complete system using standard tools available today for doing field upgrades," said Rich Sevcik, senior vice president software, cores, and support at Xilinx. "With our tools, the Virtex FPGAs, our design service partners, and the reference designs, we are providing a complete environment to enable our customers to build the next wave of programmable logic applications."

Conclusion

The number of Xilinx Online applications is growing quickly because of the many benefits offered by this revolutionary technology. Xilinx Online field upgradable applications are currently being used in network equipment such as ATM switches, cellular base stations, and satellite communications systems, and the Xilinx Online program is additionally targeting opportunities in emerging markets such as network appliances, multi-use set top boxes, and mobile network devices. Σ

Virtex FPGAs

The Virtex family was developed to address system-level designs and field upgradable systems, integrating 200 MHz system interfaces and timing management capabilities within a million–gate FPGA that includes a hierarchy of memory resources. Virtex FPGAs support partial reconfiguration, thus allowing new circuitry to be downloaded while standard operation continues within the device.

A fast 400Mb/second reconfiguration rate ensures that a full reconfiguration can be done in milliseconds and a partial reconfiguration can be done in microseconds. Reference designs will be available on the Xilinx Online Web page within the Xilinx website. These Web pages also contain frequently asked questions, a user forum, application notes, white papers, links to third-party reconfiguration companies, and links to design service companies supporting the development of Xilinx Online field upgradable applications.

Spartan FPGAs

Spartan FPGAs are targeted as gate array replacements for low-cost, high-volume designs under 40,000 system gates that require on-chip RAM and can benefit from pre-defined software cores. Current Spartan devices operate at 3.3 volts and 5 volts.

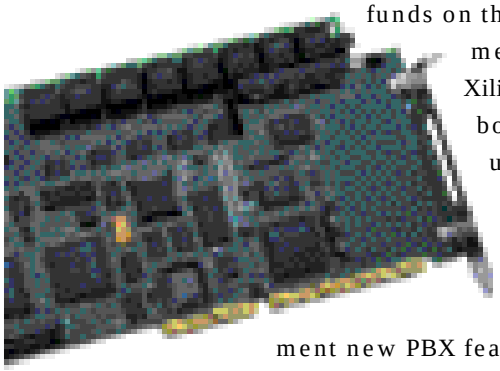
Voice Technologies Group Creates Adaptable, Netlist Controls

Xilinx has played a major role in VTG's product development since the inception of the company over ten years ago.

by Gene Kielty, Director of Engineering, Voice Technologies Group, kieltyee@vtg.com

Voice Technologies Group is the leader in PBX integration technology and Xilinx has been the implementation tool of choice in our product designs. Our product architecture is based on adaptable technology that allows us to conform to the ever-changing horizon of the computer telephony industry. Xilinx provides the silicon that enables us to move quickly into the market and make changes to meet the needs of our customers.

Xilinx has enabled VTG to develop adaptable, cost effective solutions. VTG's Solar project is a good example because it was our first high-volume desktop product. The OEM we were dealing with wanted an ASIC solution for cost reasons. Xilinx worked closely with VTG and we were able to hit the OEM's cost target with the Xilinx XC5210 FPGA. This allowed VTG to provide a cost-effective solution without having to spend large amounts of ASIC NRE funds on the development. Having Xilinx "on board" saved us in the later stages when VTG was able to implement new PBX features with a simple software update rather than hardware changes to the board. This type of flexibility and price competitiveness has made VTG a staunch Xilinx user.



In another product development effort, VoiceBridge 2000, VTG was able to again use Xilinx to develop a better, cost-effective solution. The predecessor to this product, VoiceBridge-PC, had ten XC3042s, an XC3064, and an ISA interface consisting of numerous discrete logic parts and CPLDs. Use of the Xilinx XC5210s and XC9500 CPLDs allowed us to increase the board's capabilities while reducing overall cost and board space.

Xilinx has enabled VTG to build superior products with minimal development time to capitalize on market opportunities. The industry awards and recognition achieved by VTG products are in no small way attributed to the superior and cost-effective technology of Xilinx.

Our intellectual property runs on Xilinx and will continue to do so into the next millennium, based on the new Spartan series and beyond. VTG must stay ahead of its competitors on the road to voice and data network convergence. We plan to continue to use Xilinx as a key technology to enable VTG to play a central role in this convergence.

Since its inception in 1989, Voice Technologies Group has maintained and continues to generate record growth in both the OEM and end-user CTI markets. VTG is recognized as a leader in the telecommunications industry, using its expertise in computer and telephony integration to design, manufacture, and market high-performance, leading-edge products, including intelligent digital PBX integration server products and desktop packages. For more information on VTG, see: www.vtg.com.Σ

How to Add Features and Fix Bugs - *Remotely*

Here's what you need to consider when designing a Xilinx Online application.

by Tom Branca, Applications Engineer, Xilinx,
tomb@xilinx.com

Having the ability to remotely update hardware with new features or the latest bug fix can accelerate your time-to-market, extend the useful life of existing systems, and significantly cut production, maintenance, and support costs. If you plan for remote updates during your initial specification and design process, your systems can easily reap all the benefits of the Xilinx Online capability.

The Remote Field Upgrade Process

FPGAs are SRAM-based, so you can reconfigure them an unlimited number of times. To use this capability for field upgrades, you must include a mechanism for updating the configuration bitstream. To support remote field updates a system must have some sort of communication channel across which a replacement bitstream can be transferred; this could be a cable or modem connection; or a satellite, infrared or radio interface.

To initiate a remote field update, a command is sent across the communication interface to the communication processor, signaling to the system that the FPGA needs to be updated. The communication processor can be as complex as a microprocessor, or as simple as a CPLD.

Once the communication processor knows that an update is required, it can reconfigure the FPGA directly using the slave, Boundary-Scan, or peripheral FPGA configuration modes. Alternatively the processor can update a non-volatile memory bank (typically an EEPROM or Flash memory) and then simply initiate a standard FPGA reconfiguration cycle.

Planning for Remote Upgrade

Once you have made the decision to take advantage of remote field updates, a number of issues must be addressed.

Data Transmission

The type of communication channel will affect the speed, security, and integrity of the data that is used to update the FPGA. A communication interface already being used for sending and receiving data in the system (for normal operation or for firmware type updates) can usually be reused to perform the FPGA remote update.

Data Integrity and Verification

It's important that you verify the integrity and reliability of the update data before the FPGA configuration process even begins. Xilinx FPGAs have a cyclic redundancy check (CRC) built into each frame of the configuration data so that an error in the bitstream will cause the FPGA configuration to fail. You should design the system to be able to detect transmission errors, and request a re-send of the data, if necessary.

Security

If FPGA update information is sent over an unsecured network, design security may be an issue. However, it is practically impossible to decipher a configuration bitstream, to extract

(Continued)

information on the functionality of a design or make intelligent modifications to it. Xilinx keeps the specifications of the bitstream a closely guarded secret. If you feel the need for an additional level of security to keep your update data confidential, you can also use encryption.

Compression

As FPGA densities increase, the amount of data required to configure a device increases, and for larger designs compression can be beneficial. This would require some kind of additional software or hardware support to manage the data compression and extraction.

Planning for Adding New Features and Bug Fixes

Designing your system so that it can implement the current functionality and still be flexible enough to meet the requirements for future design revisions requires some advanced planning.

Choosing the Optimal FPGA Density

When you choose an FPGA for a remotely updateable application you should consider future expandability and compatibility. To determine the optimal FPGA density, you must consider the device resource requirements of the current design and that of any potential future design enhancements. An advantage of Xilinx FPGAs is that for any device in a given architecture, in the same package, all device sizes are footprint compatible. This gives you the ability to select and work with a specific device and still have the flexibility to easily change to a larger or smaller device before going to production.

Configuration Memory

You can choose either volatile or non-volatile memory for storing FPGA configuration data. The advantage of non-volatile (EEPROM or Flash) memory is that when reconfiguring the FPGA after a power cycle, the system will still have the most recent configuration data in memory and will not require extra clock cycles to update its configuration data across the communication interface.

Most FPGA configuration modes require the entire bitstream to be loaded into the FPGA during the configuration cycle. Some Xilinx FPGAs (including the Virtex family) allow partial configuration. The Xilinx data book has more information on the different configuration modes available for specific device architectures.

Design Expandability

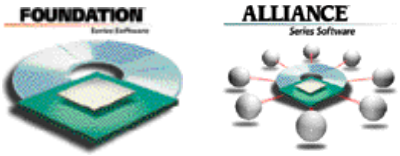
If additional connections between the FPGA and other devices are required in future revisions, then defining the interface between devices will need to be done during the initial product revision. You can create test programs that toggle these future I/O connections, so you can test specific interfaces, without actually completing the design. Or, you can perform an EXTEST using the FPGA's built-in Boundary-Scan functionality.

Archiving the Design for Future Updates

It is critical to not only archive all source, implementation, and constraint files that were used to create the existing revision of the design, but to also document the entire flow used to create the final bitstream. Also, depending on the expected life of the application, it may be a good idea to archive the tools (save the CDROM) used in the creation of the bitstream. This will make it much easier to update the devices in the field, even years from now.

Conclusion

With a little planning, your next design can easily have the ability to be remotely fixed or upgraded, a feature that your customers will find very appealing. Σ



New 2.1i Software – Shortens Design Cycles, Improves Productivity

The latest releases of the Xilinx Alliance Series and Foundation Series software are now available.

by Arne Barras,Product Marketing Manager, Xilinx, arne.barras@xilinx.com

There are many new productivity-enhancing features in the new 2.1i software release from Xilinx. Faster runtimes, quicker constraint entry, better timing reports, and the most recent update of FPGA Express all combine to shorten the amount of time it takes to complete a Xilinx design. Almost every tool, from mainstream applications like the Design Manager/Flow Engine to advanced ones like the FPGA Editor, has been enhanced with new features that make it easier than ever to get great results quickly when using Xilinx programmable logic.

2.1i Overview

Xilinx Development systems are packaged in two product families, each configured to provide the tools necessary to support any customer's design flow needs. Both families of design solutions include the world-class Xilinx implementation tools, the industry's leading solution in advanced programmable logic design methodologies.

The Xilinx Alliance Series™ solutions emphasize seamless integration into Alliance EDA partner design flows and methodologies. This release of the Alliance Series software features support for LMG Smartmodels, enabling board-level simulation, and chip-level I/O timing models written in the Stamp format, enabling board-level static timing analysis. Alliance Series solutions also drove the requirements for the addition of other ASIC-like design capabilities, such as minimum delay reporting, and temperature and voltage prorating.

The Xilinx Foundation Series™ solutions are ready-to-use programmable logic design packages. Incorporating all facets of design into a complete, front-to-back design environment. The Foundation Series emphasizes ease of use in simplifying the programmable logic design flow.

While simplifying the design process, Xilinx has also managed to incorporate some of the industry's best schematic and HDL design and synthesis technologies in a package with a price tag that is hard to beat. The v2.1i Foundation Series solutions feature FPGA Express™ v3.2, synthesis technology that delivers the robust language compilation capabilities designers have become accustomed to receiving from Synopsys, with new Xilinx-specific optimization capabilities. A more detailed review of the new features in the 2.1i products is provided below.

New Features in the v2.1i Implementation Tools

This section discusses the new features that are common to both the Alliance Series and Foundation Series products.

Runtime Improvement

Development efforts toward reducing runtimes were focused on the Virtex family, and this has paid off with a 50% reduction in runtimes for the larger Virtex devices. This runtime improvement is concentrated mostly in the placement step, which can be anywhere from four to ten times faster than the time required for placement in version 1.5i. This improvement in runtime comes with no performance penalty; clock

(Continued)

rates should be as fast or faster than those obtained from the 1.5i software.

2.1i Hierarchical Timing Report Browser

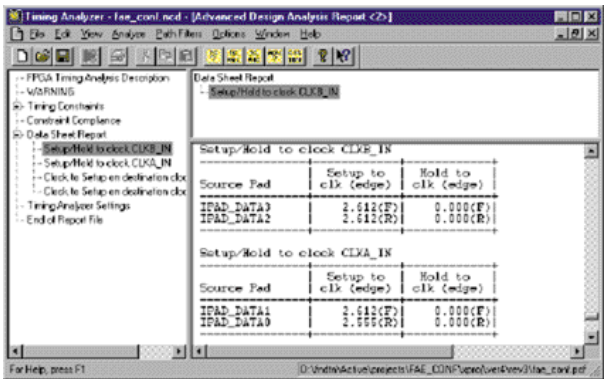


Figure 1

Timing Reporting and Analysis

One of the most visible enhancements in the 2.1i software release is in the area of timing reporting. A new "datasheet style" timing report saves you time by simply and clearly displaying all I/O timing of the chip as well as clock frequency. The I/O timing includes input setup and hold, as well as output clock-to-pad analysis, and the clock frequency is reported on a per clock basis.

For more complex designs such as those that include multiple clocks, separate tables are provided that report the phase relationships between each clock domain. By default, this report will list the delays for any I/O or clocks with timing constraints on them. For designs without timing constraints the report can be generated by selecting the "Advanced Analysis" option in the Timing Analyzer.

Another great new timesaving feature is the hierarchical timing report browser (Figure 1) available from the within the Timing Analyzer. By generating and viewing timing reports from this tool, even the most detailed and complex timing reports are easily navigated.

Virtex Floorplanner

The Xilinx Floorplanner (Figure 2), first introduced in our 1.5i release, has been enhanced to support the Virtex device family. It offers a simple graphical view of the physical design, simplifying the process of floorplanning a design, and thus leveraging user knowledge to simplify the algorithmic intensive process of laying out a design. Floorplanning has proven to be most useful for larger, highly structured designs. The Floorplanner is accessible both from the Design Manager and from the command line.

2.1i Floorplanner

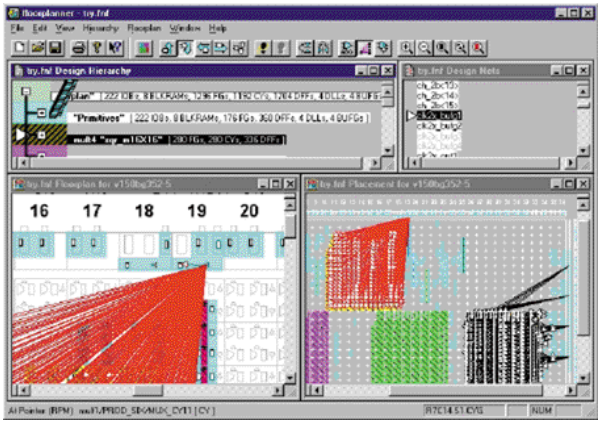


Figure 2

Support for Board-level Static Timing Analysis

The value of static timing analysis, as a complement to simulation to reduce timing verification, has long been recognized within the FPGA design community. This methodology is now being extended to support board-level design as well. Xilinx can now generate a Stamp timing model of the device I/O, for use with BLAST™ from Viewlogic, or Tau™ from Mentor Graphics - two of the more popular and powerful board-level static timing tools available today.

The Stamp timing model is generated using the **-stamp** command line option of TRCE, the static timing analysis command, or by adding the **-stamp** option to the Design Manager through its Customize capability.

(Continued)

Design Manager

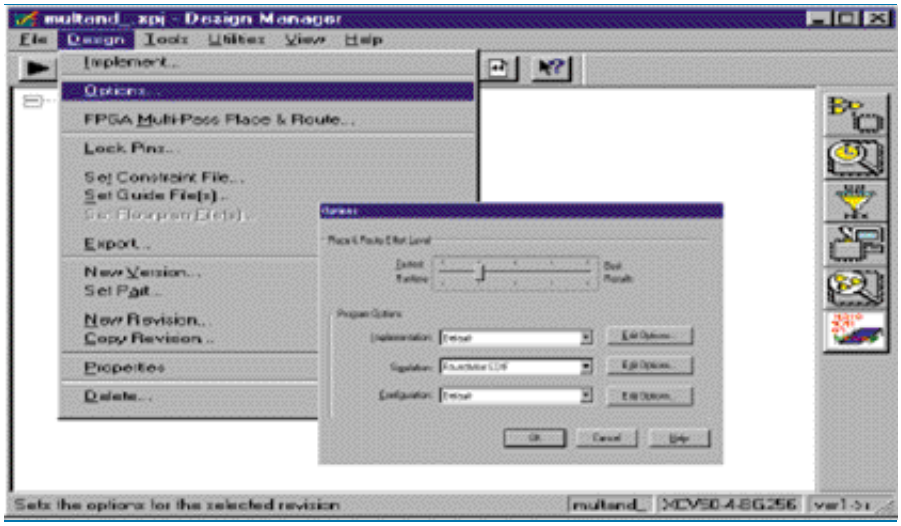


Figure 3

Design Flow Automation Improvements

Compiling a design through the Xilinx Design Manager has become more streamlined through several new enhancements:

- Run-time options have been made more visible and easy to find by moving them up a level from the Implement dialog, to an "Options" command directly accessible from the "Design" menu.
- Fewer revisions are created in the 2.1i release because the Design Manager will continue to implement a design within the current revi sion unless the input netlist has changed.
- The new, "Smart" Flow Engine automatically detects which options have been changed or if constraint tools like the Constraints Editor or Floorplanner have been used, and implements the modifications by starting at the appropriate step in the flow. This simplifies the design flow by not requiring you to manually choose which steps are to be run, and shortens runtimes by removing the requirement to start the implementation process from the beginning as a result of any design constraint modification.

Another significant enhancement in the Design Manager (Figure 3) is the capturing of options and control files such as the constraints file, guide file, and floorplan into the revision itself. Combined with the "seed file" capability that is already in the 1.5i release, any revision can now be considered a full snapshot of the files and options to produce that revision. At any point in time, you can review not only the results of a revision, but the options and files used to produce it.

Real-time Debugging

A new real-time design debug capability has been added in the 2.1i release. This capability allows you to identify any internal signal that you wish to see externally, and it will be automatically routed to an unused I/O pin. Called "PROBE", it is accessed through an easy two-step process of selecting the desired internal signal and choosing one or more available I/O pins to display it on.

There is no limit to the number of internal signals that can be probed, and they can be selected and brought out without having to re-implement the design. PROBE is available in the new FPGA_Editor. Replacing the older EPIC tool, the new FPGA_Editor is a fully MFC-compliant rewrite which brings many advantages such as being able to open multiple windows and view more than one area of the FPGA at one time.

Constraints Editor

High speed designs often require a complete set of timing constraints, and the 2.1i Constraints

(Continued)

Constraints Editor

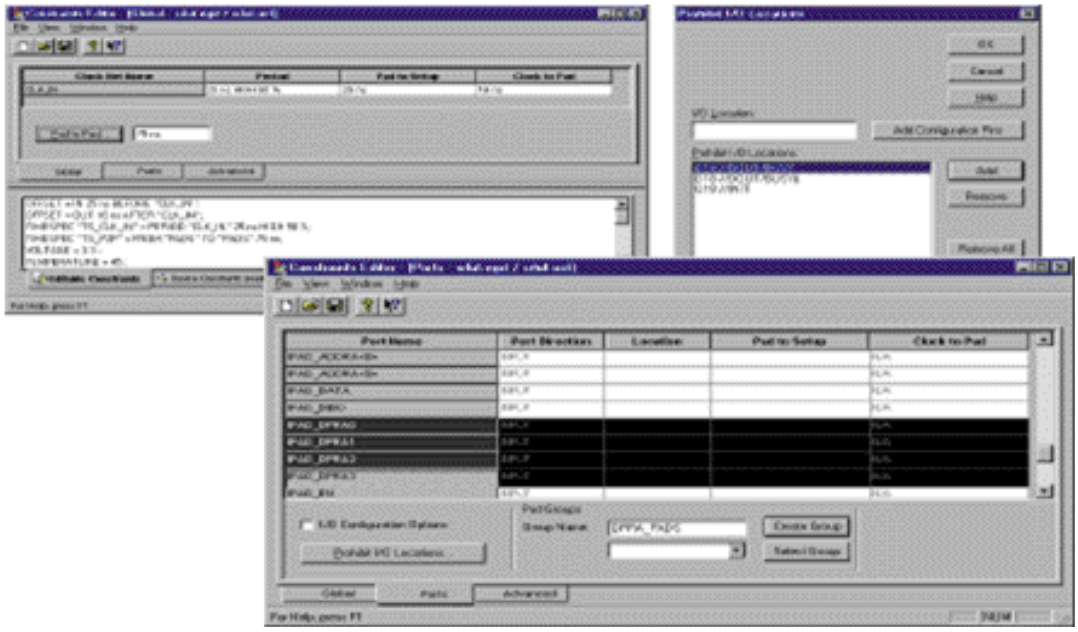


Figure 4

Editor (Figure 4) has been enhanced to shorten the amount of time it takes to apply them. These enhancements include direct cell entry similar to that found in spreadsheet programs, the ability to select multiple cells at once, and the ability to easily group pads together and apply a single constraint. Together these all help reduce the effort needed to enter timing constraints.

Easing the overall design constraining process is the addition of a command to prohibit the dual-purpose configuration pins during layout, and the capability to specify the Virtex I/O attributes such as Pullup, Pulldown, Keep, and the various voltage standards and drive capabilities. DLL support for Virtex has also been improved, now automatically adjusting and applying the constraint to the DLL outputs, and allowing just the input clock to be specified.

Minimum Delays and Prorating

Minimum and prorated delays, which were initially introduced for the XC4000XL family in the1.5i release, have been extended to support the XC4000XLA™ and SpartanXL families in the 2.1i release. The 2.1i release has also been designed to support minimum and prorated delays for Virtex FPGA as soon as the values are provided in an upcoming speedfile, which is on support.xilinx.com.

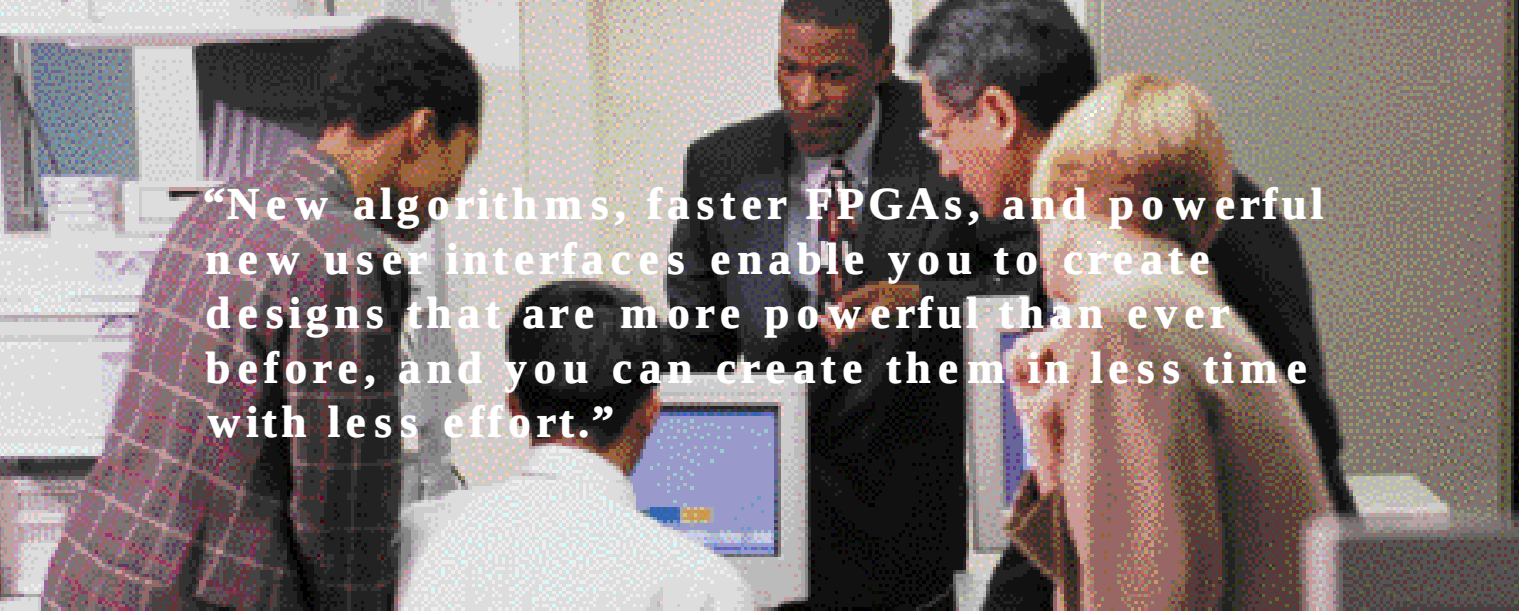
New XFLOW Scripting Tool

For those who prefer to run the tools using a command line, through batch files, or want to encapsulate the Xilinx implementation tools into their own shell programs, a new tool named XFLOW has been added. It allows you to pick from a set of predefined flows and option files and run the entire implementation process as a single command. In addition, you can modify the flows and options to create your own preferred set.

Foundation Series Software Improvements

The Foundation Series software builds on the strength of the features described above by incorporating the latest version of the Xilinx implementation tools. This update has integrated user access to each of the advanced analysis tools offered by Xilinx in the Foundation tools, and expands the design implementation flow options available to Foundation users. This feature expansion also adds the options of guided

(Continued)



implementation flows as well as utilization of floorplan data. In addition, the Foundation Series software further improves productivity and shortens the design cycle time by adding several other productivity enhancing features.

Foundation Design Entry Tools

One of the productivity enhancements that schematic users will notice is the addition of several Design Rule Check (DRC) options in the Foundation Series schematic editor. This release also so you can select the DRCs that are performed, allowing you to avoid unnecessary warnings and messages.

The second item that you will notice is the integration of the Xilinx CORE Generator™ tool. You now have access to Xilinx cores without leaving the Foundation Series environment. Simulation models, HDL instantiation templates, and schematic symbols are all automatically added to your environment.

Foundation Series Synthesis Engine

The Foundation Series synthesis flow has become more efficient with the integration of the latest FPGA Express synthesis technology. The 3.2 version of the FPGA Express synthesis engine dramatically improves multiplier implementations, delivering smaller and faster multipliers. The FPGA Express synthesis engine also offers options for buffering internal nets, so you can automatically take advantage of unused global routing resources for internally generated high fanout nets.

The implementation timing constraints, forward annotated by the synthesis engine, have also been improved to reduce the overall design cycle. The FPGA Express engine writes timing constraints which have been optimized for the Xilinx implementation tools, requiring fewer system resources and improved runtimes.

Summary

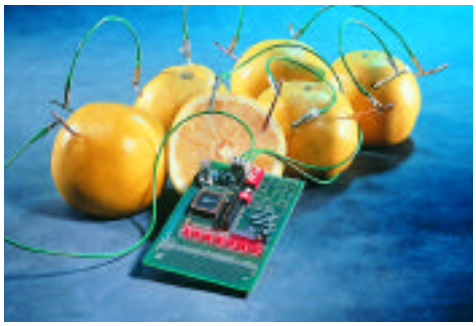
Version 2.1i of the Xilinx Alliance Series and Foundation Series software adds many new productivity enhancements. While the previous release of these solutions, (v1.5i) was highly acclaimed as the fastest and easiest to use in the industry, the 2.1i release is even faster and easier to use. New algorithms, faster FPGAs, and powerful new user interfaces enable you to create designs that are more powerful than ever before, and you can create them in less time with less effort.Σ

For more information on Xilinx development systems, visit our website at:
http://www.xilinx.com/products/software/software.htm

Xilinx Acquires CoolRunner Line of CPLDs

The CoolRunner line is the first family of CPLD products to combine very low power with high speed, high density, and high I/O counts in a single device.

by Mike Seither, Director of Public Relations, Xilinx, mike.seither@xilinx.com



Xilinx recently signed a letter of intent with Philips Semiconductors to acquire their line of low-power complex programmable logic devices (CPLDs). Under the terms of the agreement, Xilinx will purchase the Philips CoolRunner® CPLD technology as well as the Philips XPLA™ Professional suite of design tools. Xilinx intends to retain approximately 40 Philips employees associated with CoolRunner product development, including an IC development team in Albuquerque, New Mexico, and a software team in Sunnyvale, California.

The CoolRunner line offers the largest CPLD available today, the CoolRunner PZ3960 device with 960 macrocells. The CoolRunner devices are available in 3.3-volt and 5-volt versions with densities ranging from 32 to 960 macrocells. The line also includes 3.3-volt and 5-volt 22V10 CoolRunner devices.

The CoolRunner line features Fast Zero Power™ technology, drawing virtually no power in standby mode, making them ideal in the fast growing market for battery operated portable electronic equipment such as laptop PCs, telephone handsets, personal digital assistants, and electronic games. CoolRunner CPLDs also use far less dynamic power during actual operation compared to conventional CPLDs, an important feature for high-performance, heat-sensitive equipment such as telecom switches, video conferencing systems, simulators, high end testers, and emulators.

"Our current family of XC9500 CPLDs has given Xilinx a competitive advantage in speed and price," said Evert Wolsheimer, vice president and general manager of the Xilinx CPLD Business Unit. "With the CoolRunner devices in our portfolio, we also will have a commanding position in the areas of low power and high density CPLDs. As a programmable logic supplier, Xilinx is now able to offer everything from 22V10s and CPLDs to million-gate FPGAs. This acquisition will also provide Xilinx with an additional talented engineering team to help us continue to develop advanced products for our customers."

Under the agreement, Philips Semiconductors retains the right to incorporate the Fast Zero Power technology, used in the CoolRunner CPLDs, in embedded applications such as system-on-chip designs. The CoolRunner products will continue to be made at Philips Semiconductors' wafer fabs in Europe and at Taiwan Semiconductor Manufacturing Corporation in Taiwan. Xilinx intends to market the products under the current CoolRunner XPLA product name.

"This agreement allows Philips Semiconductors to better focus its efforts in the company's core areas such as consumer, telecommunications and automotive markets, as well as in the discrete, logic, and microcontroller product line," said Arthur van der Poel, Chairman and CEO of Philips Semiconductors. "Additionally, we look forward to working with Xilinx on future joint developments such as system-on-chip designs."Σ

WebFITTER–
Now *Better* Than *Ever*

Here are the new features and capabilities for WebFITTER, our Internet-enabled XC9500 fitting and device evaluation tool.

by Dave Grace, CPLD Software
Product Manager, Xilinx,
dave.grace@xilinx.com

Even if you have tried the industry's first on-line CPLD design and evaluation system, it's time to take another look. With the first version of the WebFITTER™ we gave you an easy to use graphical interface (Figure 1) that made it quick and simple to use. We provided you with all the necessary reports to see how Xilinx CPLDs could perform in your design, and all of the essential files (HDL simulation models and JEDEC programming files) to fully verify our solution.

In our latest release, Xilinx has once again raised the bar by providing you with the very best in on-line EDA solutions. By combining ABEL v7.1 (the newly updated version of the popular PLD design language), with our new Xilinx Synthesis Technology (VHDL and Verilog synthesis that augments Synopsys FPGA Express V3.2), Xilinx now offers the best suite of HDL design tools available. This combination of Synopsys synthesis, Xilinx synthesis, and ABEL, allows you to realize the full performance and cost effectiveness of the XC9500 family of ISP CPLDs.

WebFITTER Design Input Interface

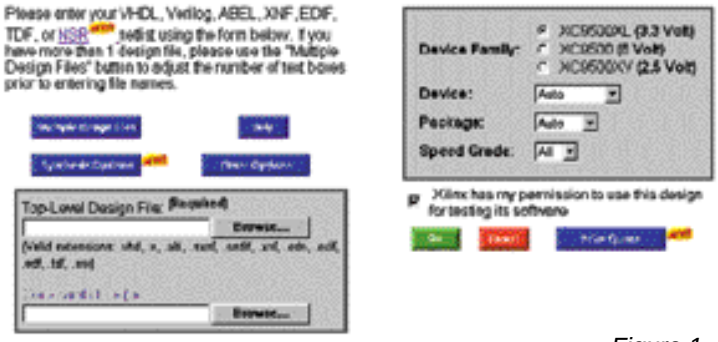


Figure 1

Design Statistics and Price Quote Option

Figure 2

In addition to making it easier to implement your design, we have also made your overall device selection much faster through on-line device price quotes. Whether you need to know the price for a specific device/package/speed grade combination or the price for all devices in a specific package, it is just one click away (Figure 2).Σ

Check out the latest version of the WebFITTER at: www.xilinx.com/sxpresso/webfitter.htm.

New WebPACK– Packs a CPLD Punch

WebPACK is a collection of downloadable modules that give you the best possible CPLD development solution while offering complete EDA tool flexibility

by Dave Grace, CPLD Software
Product Manager, Xilinx,
dave.grace@xilinx.com

Xilinx has taken the industry's best CPLD implementation tool suite, coupled it with the latest enhancements in ABEL and HDL synthesis technology, and incorporated JTAG programming capability, in a free, downloadable solution called WebPACK™.

WebPACK Modules

WebPACK includes three, self contained, downloadable modules. Each module can be downloaded, installed, and run individually, and works in conjunction with other EDA tools, such as synthesis, simulation and schematic capture programs. This gives you the flexibility to use your existing tools and simply interface them to our CPLD solution.

When two or more of the modules are downloaded and installed, they become an integrated design environment. This allows you to go from entry, synthesis, and optimization to device fitting and programming, all from within a single environment.

WebPACK Project Navigator, Simulation Waveform Viewer, and Color-coded HDL Editor

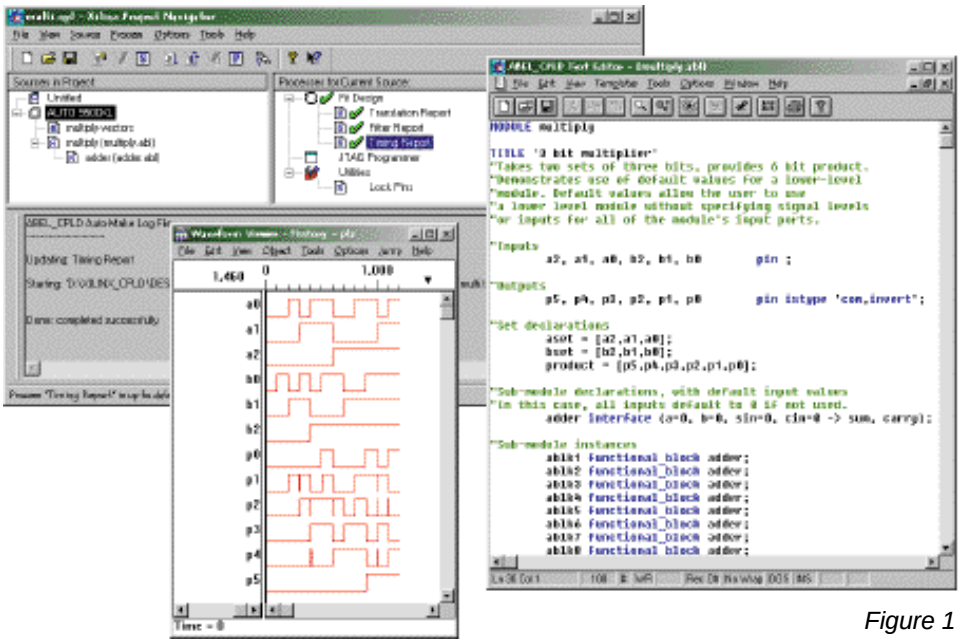


Figure 1

XC9500 HDL-ABEL Synthesis Module

This module (Figure 1) provides the latest version of ABEL v7.1, Xilinx Synthesis Technology, and a project navigation system specifically designed for the XC9500 series. These two synthesis engines provide VHDL, Verilog, and ABEL language design entry within our color-coded HDL editor.

XC9500 Fitting Tools Module

This module (Figure 2) includes the traditional CPLD implementation tools. Included are the Xilinx Design Manager and Flow Engine, Constraints Editor, Iterative Timing Analyzer, and the newly introduced CPLD ChipVIEWER™. These tools can be used in conjunction with the HDL-ABEL Synthesis module or in a more traditional environment as part of an overall EDA solution with other 3rd-party tools.

Device Programming Module

This module (Figure 3) consists of our JTAG Programmer software for both CPLDs and FPGAs. This tool can be run stand-alone to support any Xilinx CPLD or FPGA application and is automatically run from within the WebPACK project navigator, if installed.

Conclusion

Each of these free downloadable modules, as well as the WebFITTER on-line CPLD fitting and evaluation tool, are available from the Xilinx website at: www.xilinx.com/products/software/webpowered.htm. Both tools offer the easiest and most cost effective way to evaluate, price, design, and implement a Xilinx XC9500 CPLD solution.

Xilinx Design Manager, Constraints Editor and CPLD ChipViewer.

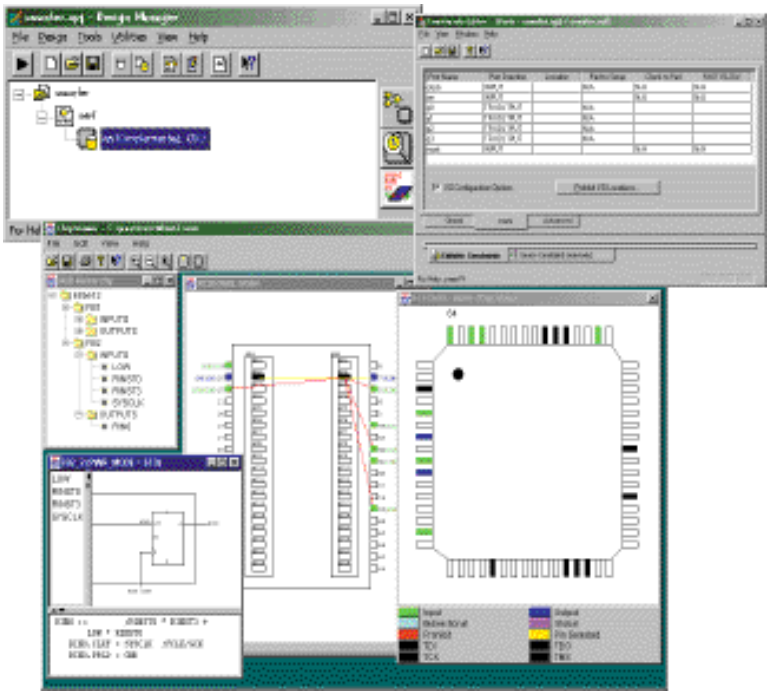


Figure 2

Xilinx JTAG Programmer Graphical User Interface

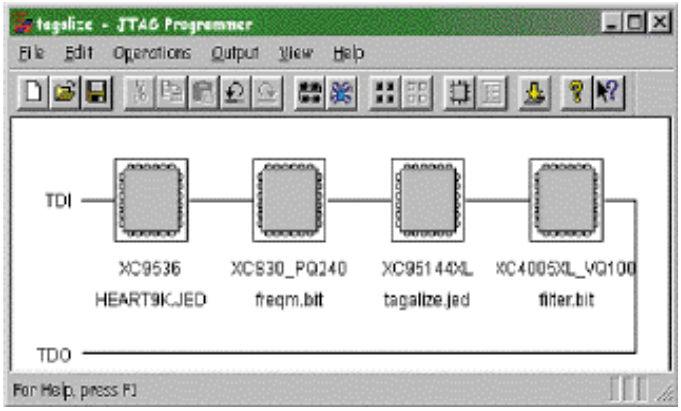


Figure 3

Get a Head Start with a New Virtex Development Board

To demonstrate the versatility of the new Virtex FPGAs, Avnet Design Services has created a new development tool—the Virtex Development System.

by Filip Verhaeghe, Director of Technical Services, Avnet Design Services, filipv@hh.avnet.com

Built by the engineers in Avnet Design Services' Technical Service Center, the Virtex Development System is a circuit board that gives you a significant advantage—it helps you reduce your development cycle and allows you to immediately develop and test your Virtex-based designs. The board is part of Avnet Design Services' ongoing series of reference designs and development systems that help you use the latest components without a prohibitive learning curve. In addition to the development board, the Virtex Development System comes bundled with either the Foundation Series or Alliance Series development software from Xilinx and includes a Xilinx fitter cable and documentation.

The Virtex Development System is available exclusively from Avnet Electronics Marketing and features on-board peripherals, which:

- Demonstrate the high performance of Virtex FPGAs.
- Use various FPGA configuration modes.
- Interface to the Xilinx download cable.
- Demonstrate multiple system clocks and delay locked loops (DLLs)
- Demonstrate various memory interfaces.
- Demonstrate core support and functionality.
- Provide a development/reference platform for developers.

The Virtex Development System is available in the following configurations:

Alliance Series: DS-ALI-STD-PC-SK/B
RESALE $1195.00 USD

Foundation Series: DS-FND-EXP-PC-SK/B
RESALE $2795.00 USD

To order the Virtex Development System, visit the Avnet Electronics Marketing Semiconductor website at: www.em.avnet.com/semis/ads/virtex.html.

The Virtex Development System board (Figure 1) provides maximum flexibility by using the XCV300BG432, which is surrounded by popular emerging technologies. The board includes physical layer interface chips, memory interfaces, bus interfaces, a Xilinx JTAG download cable interface, and a RAMDAC interface for CRT display support. With the supporting electronics in place and electrical interfaces defined, the function of the Virtex Development System depends on the FPGA logic and cores that you implement. You can use board in a 3.3V PCI slot, or operate it standalone.Σ

Virtex Development System Board

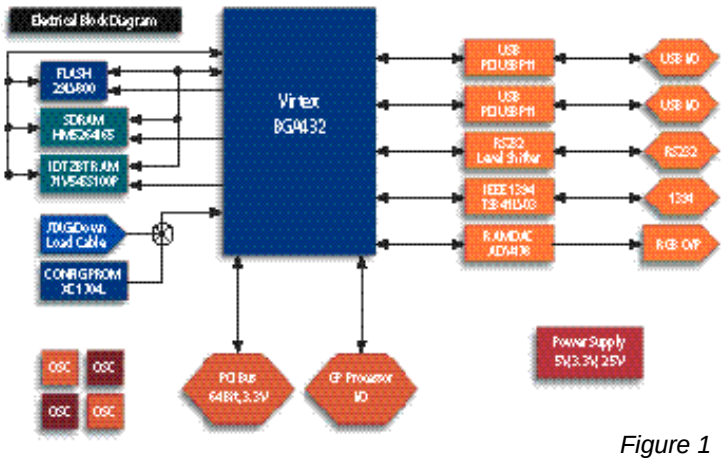


Figure 1

Low-cost Spartan FPGAs Used in ADSL Modems

Using Spartan FPGAs to manage DMA transfers and to implement the complex system-level glue logic required for the USB interface—Spartan devices are very cost effective in these applications.

by Marc Baker, Applictions Engineer, Xilinx,
marc.baker@xilinx.com

ADSL modem technology can expand the useable bandwidth of existing copper telephone lines, delivering high speed data communications at rates of up to 8 Mbps. The recent G.Lite standard allows for a lower-speed, lower-cost implementation.

In this ADSL Modem design example (Figure 1), the Spartan device sits between the CPU, USB interface controller, and the ADSL chipset, and manages DMA transfers of ATM cells. While the design is targeted at solving a specific problem, interfacing an ADSL chipset to USB, it illustrates solutions to a number of general technical issues, including the implementation of Utopia interfaces for ATM devices and remote configuration of Spartan devices.

Overview

The design objectives for this application were:

- To achieve the lowest possible cost. In this case, the target cost for the USB interface was significantly less than $10 in high volumes.
- To deliver the best possible performance. Current solutions are able to deliver 2 to 3 Mbps of bandwidth across USB at a much higher price. The minimum target for this design was to support the full 1.5 Mbps data rate of G.Lite and at the same time get as close as possible to the full G.992.2 (6.1 Mbps) data rate.
- To configure the Spartan device from the host via the USB interface. This has the dual benefit of eliminating the requirement for FPGA configuration memory in the modem and the ability to update the configuration in the field.

ADSL Modem System Block Diagram

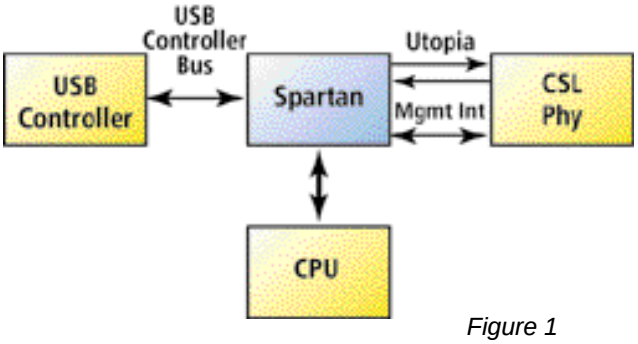


Figure 1

(Continued)

Interfacing ASSPs

Much of the complex logic required for implementing an ADSL modem is provided in highly integrated chipsets known as Application-Specific Standard Products (ASSPs). However, these ASSPs are often not designed to talk to each other. Xilinx low-cost FPGAs, such as the Spartan Series, provide the complex system-level glue required to complete these applications.

The Alcatel DynaMiTe ADSL three-chip set consists of the MT-20134 Analog Front End, the MT-20135 ADSL Modem and ATM Framer, and the MT-20136 ADSL Transceiver Controller. The Spartan device interfaces to the MT-20135, and MT-20136.

The MT-20135 sends and receives data in the form of ATM over a standard Utopia level 2 interface. The function of the Spartan device is to handle the handshaking required to convert the full duplex ATM cell stream from the Utopia interface into half-duplex transfers to and from the USB controller. Because both the MT-20135 and the USB controller have internal FIFOs capable of storing a complete ATM cell, the transfers consist of moving data one cell at a time between these FIFOs.

The MT-20136 is essentially a single-chip processor dedicated to managing the ADSL modem. The Spartan device interfaces to this device via a specialized management interface called CTRLE, used to control the modem and to query status.

The USB interface in the design is based on a National Semiconductor USBN9602 controller. This device contains all of the logic necessary to transfer data frames to and from the host with minimal processor intervention.

The Spartan device supports two primary functions:

- DMA logic that manages the transfer of ATM cells between the USBN9602 and the Utopia interface.
- Arbitrating access to the USBN9602 by an 80C51 microcontroller. The 80C51 provides a low-cost microcontroller with functionality that includes initializing the Spartan device at startup and responding to status queries from the host received via USB messages.

Spartan Device Selection

The following criteria were used to select the device used in this application:

- I/O Pins – The design requires a total of 60 pins.
- Voltage – The design operates at 3.3V.
- Density – The estimated size of the design is 8K gates, broken down as follows: 4K for the USB controller interface, 1K each for the Utopia Tx and Rx state machines, 1K for the microcontroller interface, and 1K for the remaining logic.
- Performance – The highest clock speed used in the device is 48 MHz, used to clock the USB interface bus state machine. The remaining logic runs at 3 MHz (the Utopia data rate) or 16 MHz for the microcontroller.
- Packaging – The size constraints imposed on most modem designs dictate a high density surface mount package.

Based on these criteria the device selected for this design is the XCS10XL-4VQ100C. This device offers 10K gate density, 3.3V operation with 5V compatibility, 77 user I/O, and is packaged in a space saving VQ100 package. The -4 speed grade is sufficient for this design's performance requirements.

Spartan Device Configuration

The microcontroller loads configuration data into the Spartan device using slave serial mode. The data comes from the host via the USB interface. On the host side, the configuration data image is stored in the device driver. The advantages of this approach are that the modem does not need dedicated configuration storage, and updating the configuration is easily accomplished by distributing an updated device driver.

At startup the USB interface pins on the Spartan device are placed into a 3-state condition

(Continued)

and the microcontroller can read and write registers in the USBN9602 by manipulating the I/O port pins. When the Spartan device has been configured and is ready to take control of the USB interface, the microcontroller reconfigures these port pins as inputs to avoid contention.

Interface Architecture

Figure 2 and Figure 3 illustrate how all of this fits together. Sitting in the middle of the interface, the Spartan device is the glue that pulls the whole thing together. The design lets the USB controller and the ADSL modem directly interchange ATM cells with no intervention from the microcontroller. At the same time, the microcontroller has access to control and status registers in all of the devices. This allows the software in the host driver to directly manage the modem.

(Continued)

USB Interface Connections

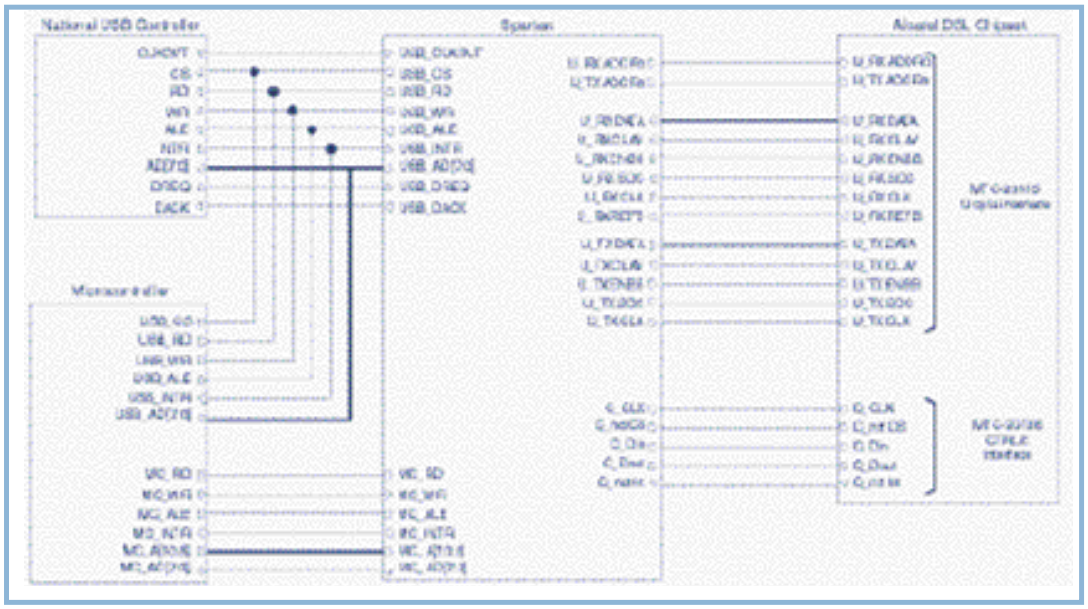


Figure 2

USB Interface Connections

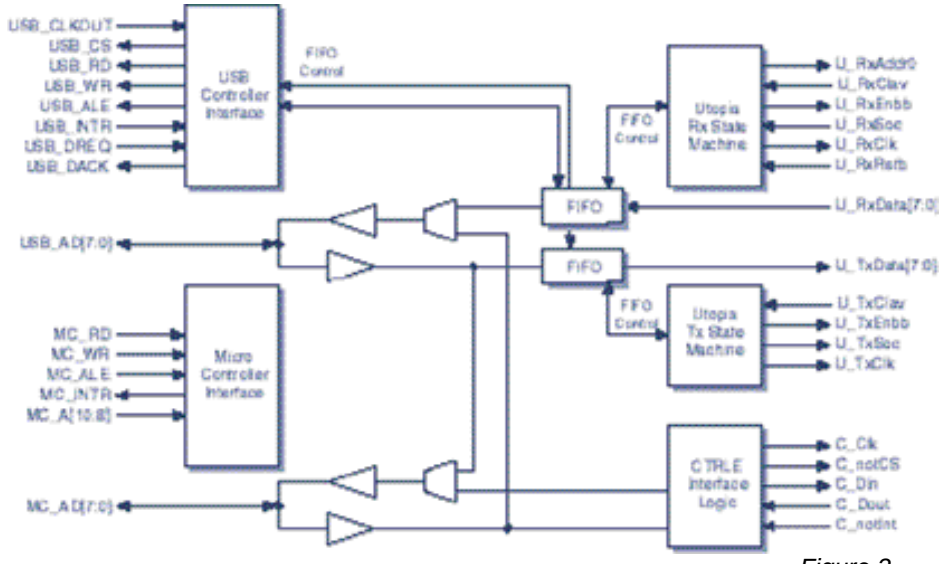


Figure 3

Spartan Device Implementation

The Utopia Tx and Rx state machines interact with the USB interface via 2-word, 9-bit wide interface FIFOs. The ninth bit is set to mark the start of a cell during transfers and reflects the state of the Tx and Rx Start Of Cell (SOC) indicators on the Utopia interface. The use of FIFOs provides a simple means of abstracting the interface between the Utopia state machines and the USB controller interface state machine.

The SpartanXL SelectRAM™ memory is used for the FIFOs. The Utopia Rx state machine continually polls the MT-20135 device to determine if there are any cells in its Rx FIFO. If data is present, it signals this to the USB controller interface and waits for it to indicate that it is ready to transfer a cell to the USB controller. Upon receiving this indication it transfers the cell to the Rx FIFO one byte at a time.

The Utopia Tx State machine continually polls the MT-20135 to determine if there is space for a new cell in its Tx FIFO. When this occurs, it signals the USB controller interface that it is ready to start a transfer. When the USB controller interface is prepared to transfer a cell, it starts loading the Tx FIFO. When the Utopia Tx state machine detects the assertion of the Tx FIFO full flag, it starts transferring the cell data across the Utopia interface. The USB interface arbitrates between the three functions that need to transfer data across the USBN9602 I/O bus. These are ATM cell read operations, ATM cell write operations, and read or write operations initiated by the microcontroller.

The microcontroller interface includes logic for latching the address, and decoding address ranges for the USBN9602 and CTRLE interfaces. In addition it contains a command register for initiating read and write cycles to the USBN9602.

Conclusion

The cost of this solution is well within the design target: $7.68 in 100k quantities (Table 1). The cost of the complete semiconductor solution is $72.68 of which the Alcatel chipset represents almost 90%. Note that the street price of the Alcatel chipset is well below the budgetary number quoted here.

In terms of performance, the design is capable of transferring cells at twice the line rate. While this is just one factor in the overall system performance, we believe that the current limitation in the USB host controllers will be the limitation for system-level performance.

The third objective of in-system configuration was met by supporting the transfer of configuration data across the USB interface. This capability should prove valuable in this and other applications where standard practice, if not the standard itself, is in a state of flux.Σ

For more information on Xilinx applications in digital modems and other high-volume applications, see the Xilinx website at: www.xilinx.com/products/xaw.

ADSL Modem Semiconductor Bill of Materials

Supplier	Part Number	Description	Per System	Unit Cost
Alcatel	MTK-20131	DynaMiTe DSL Modem Chipset	1	$65.00
National Semi.	USBN9602	USB Full Speed Function Controller	1	$1.63
Xilinx	XC510XL-4VQ100C	Spartan FPGA	1	$3.55
Philips	Sc80C51BCCB44	8-bit microcontroller	1	$2.50

Prices are based on budgetary quotes at 100K volume

Table 1

ADVANCED

Chip Scale & BGA Packaging

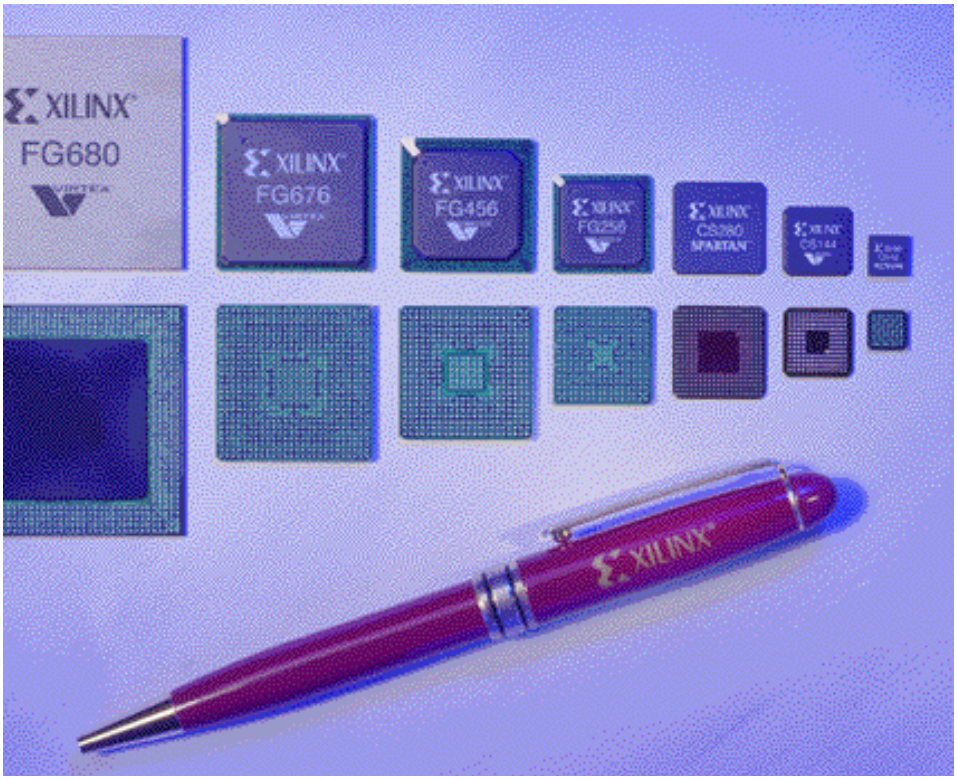
New packaging technology for FPGAs and CPLDs reduces board space and increases I/Os.

by Jay Aggarwal, Spartan Product Marketing Manager, Xilinx, jay.aggarwal@xilinx.com

Xilinx recently announced new 144-ball and 280-ball, 0.8-millimeter pitch chip scale packages(CSP) for the SpartanXL FPGA and XC9500 CPLD families. In addition, new 1.0-millimeter FinePitch packages with ball grid arrays ranging from 256 to 680 balls are available for Virtex FPGAs, along with a 144-ball chip scale package for the two smallest Virtex devices.

"Today Xilinx provides more I/Os in less space than any competitor in the industry," said Sandeep Vij, vice president of Marketing and general manager of the High Volume FPGA Business Unit at Xilinx. "We are also the first programmable logic company to be able to offer 100,000 system gates in a compact chip scale package, a remarkable achievement in itself."

New Xilinx Packaging



Chip scale packages are ideal for applications requiring low power and small form factors. The packages are targeted at high-volume, cost-sensitive designs such as digital modems, DVDs, and camcorders. CSP packaging for Xilinx CPLDs and FPGAs offers higher I/O density in less board space than the 1.0 millimeter pitch offerings available from competing programmable logic suppliers.

(Continued)

Chip Scale Packaging Leadership

Last year, Xilinx was the first programmable logic supplier to offer a 0.8-mm pitch, 48-ball package for the XC9536™ CPLD, providing the smallest form factor package in the industry. This proven chip scale packaging technology is now available for all members of the XC9500 CPLD and SpartanXL FPGA families. The CSP 144–ball package is also available for the Virtex XCV50 and XCV100 devices, which offer 50,000 and 100,000 system gates, respectively. The chip scale packages are designated as the CS48, CS144, and CS280.

Xilinx is the first programmable logic supplier to offer a CSP package that meets the JEDEC Level 3 moisture sensitivity requirements. This level of reliability enables you to reduce standard manufacturing cycle times and further minimize overall system cost.

New Package Lineup

	0.8 mm ChipScale			1.0mm FinePitch			
Leads/balls	CS48	CS144	CS280	FG256	FG456	FG676	FG680
Virtex		x		x	x	x	x
SpartanXL		x	x				
XC9500XV	x	x	x				
XC9500XL	x	x	x				
XC9500	x						

Table 1

FinePitch BGA Packaging for Virtex FPGAs

The new FinePitch ball grid arrays for the Virtex FPGAs are gaining wide market acceptance and feature a 1-millimeter pitch versus the 1.5- and 1.27-millimeter pitch of conventional BGAs. The Virtex series is the first Xilinx FPGA family to fully support these advanced FinePitch BGA packages.

The FinePitch BGAs are available in 256-, 456-, 676-, and 680-ball arrays, require less than half the board space of the previous generation of BGAs, and offer up to 512 user I/Os. Plus, the Virtex series provides footprint compatibility within the FinePitch BGA packages of different density devices. These packages are available for all Virtex devices offering from 100,000 to one million system gates, and are designated as the FG256, FG456, FG676, and FG680.

In addition, Xilinx offers a new dimension of flexibility by supplying vertical pin-out compatibility between the FG456 and FG676 packages. This gives you the ability to layout one printed circuit board and accommodate different solder ball count FinePitch BGA packages, significantly reducing design costs and cycle time.Σ

"Today Xilinx provides more I/Os in less space than any competitor in the industry, we are also the first programmable logic company to be able to offer 100,000 system gates in a compact chip scale package, a remarkable achievement in itself."

Unusual Clock Dividers

Sometimes you need to divide a clock by odd or non-integer numbers – here are four circuits that are efficient and simple, plus they are cheaper and faster than any external PLL alternative.

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

This article describes how to divide clocks by 1.5, 2.5, and by 3, and 5 with a 50% duty-cycle output. Dividing an incoming clock frequency by any integer number is trivial, and division by any even number always generates a 50% duty cycle output. However, sometimes it is necessary to generate a 50% duty cycle frequency that is not an even integer sub-multiple of the source clock.

These circuits are useful in XC4000-family and Spartan-family devices, where they are simple and efficient, and both cheaper and faster than any external phase-locked-loop alternative. Virtex devices do not need to use these tricks because they can implement these and many other functions in either of their four dedicated delay-locked-loop circuits, if the incoming clock rate is higher than 25 MHz.

How They Work

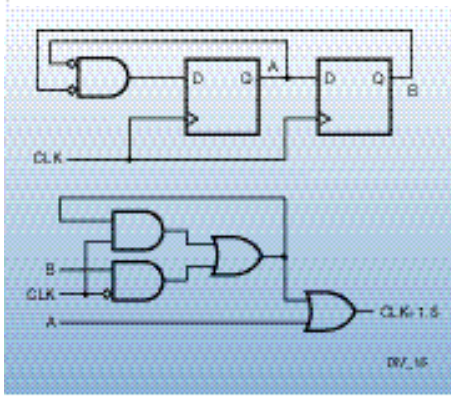
Each circuit assumes a 50/50 duty cycle of the incoming clock, otherwise the fractional divider output will jitter, and the integer divider will have unequal duty cycle. All four circuits use combinatorial feedback around a look-up table, which works perfectly and is glitch-free, but may cause your circuit simulator to fail.

These circuits have a look-up table input driven from the clock signal, with minimal skew between the A and B inputs. The chosen vertical clock line must, therefore, also have access to a LUT input. This is best achieved by coding the design as a Hard Macro.

Divide by 1.5 in One CLB.

This circuit divides the clock by 1.5, generating 60 MHz from a 90 MHz input for example (Figure1). The two flip-flops form a ÷3 circuit, and the G and H look-up tables together generate two output periods at the H output. The first output pulse is driven by the A flip-flop, the second output pulse is derived from the B flip-flop, but is delayed half an incoming clock cycle. The output stays Low while the clock is High, and stays High, after B has gone Low, until the clock goes Low again. It is this latch circuit that may cause simulator problems.

Divide by 1.5

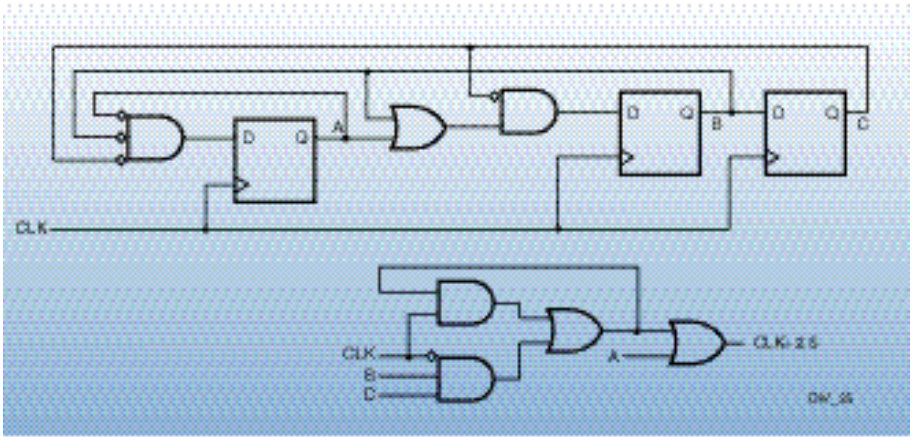


CLK	A	B	OUT
H	0	0	1
L	0	0	0
H	1	0	1
L	1	0	1
H	0	1	0
L	0	1	1

Figure 1

(Continued)

Divide by 2.5



CLK	A	B	C	OUT
H	0	0	0	0
L	0	0	0	0
H	1	0	0	1
L	1	0	0	1
H	0	1	0	0
L	0	1	0	0
H	0	1	1	0
L	0	1	1	1
H	0	0	1	1
L	0	0	1	0

Figure 2

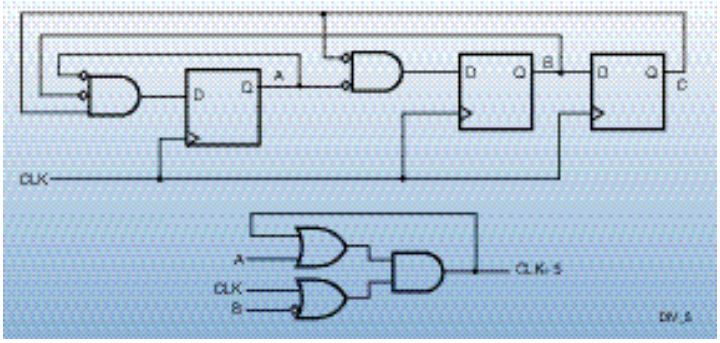
Divide by 2.5 in Two CLBs

This circuit divides the clock by 2.5, generating 40 MHz from a 100 MHz input for example (Figure 2). Three flip-flops form a ÷5 circuit, and the G and H look-up tables together generate two output periods at the H output. The first output pulse is driven by the A flip-flop, the second output pulse is derived from the B AND C signal, but is delayed half an incoming clock cycle. The output stays Low while the clock is High, and stays High after B has gone Low, until the clock goes Low again. It is this latch circuit that may cause simulator problems.

Divide by 5 with 50% Output Duty Cycle

This two-CLB circuit divides the clock by five and maintains a 50/50 output duty cycle (Figure 3). Three flip-flops form a ÷5 circuit, and the G look-up tables generate the divided output. The first output pulse is started by the A flip-flop and terminated by the B flip-flop, when the clock is Low. It is this latch circuit that may cause simulator problems.

Divide by 5



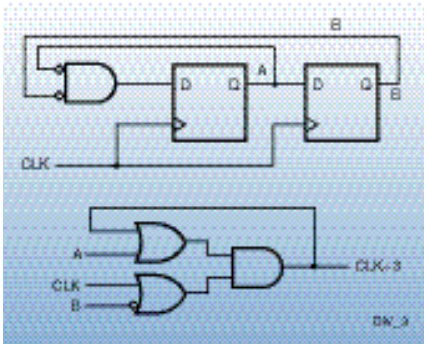
CLK	A	B	C	OUT
H	0	0	0	1
L	0	0	0	1
H	0	1	0	1
L	0	1	0	0
H	0	1	1	0
L	0	1	1	0
H	0	0	1	0
L	0	0	1	0
H	1	0	0	1
L	1	0	0	1

Figure 3

Divide by 3 with 50% Output Duty Cycle

This one-CLB circuit divides the clock by three, and maintains a 50/50 output duty cycle (Figure 4). The two flip-flops form a ÷3 circuit, and the G look-up tables generate the divided output. The first output pulse is started by the A flip-flop and terminated by B flip-flop, when the clock is Low. It is this latch circuit that may cause simulator problems.

Divide by 3



CLK	A	B	OUT
H	0	0	0
L	0	0	0
H	1	0	1
L	1	0	1
H	0	1	1
L	0	1	0

Figure 4

The Long-term Industry Outlook

Programmable logic continues to outpace the overall electronic industry growth.

by Rebecca E. Burr, Manager of Market Analysis, Xilinx, burr@xilinx.com

Understanding the trends and business cycles that affect our industry will help you make better decisions when you choose the critical technologies that affect your designs.

Industry Growth Cycles – 1980-1998

From 1980 to 1998, the global semiconductor industry has grown at a 13.0 percent compound annual growth rate (CAGR),as shown in Figure 1. During this period, the industry endured four dramatic business cycles, with the deepest correction occurring in 1985, when semiconductor shipments dropped 16.5% (A net of 29.5% in a one-year period).

1996-1999

The past three years have been below average for the overall industry, and 1996 was the first time since the 1985 downturn that semiconductor shipments were down on a sequential basis. However, the 13.0 percent longer-term growth rate for the semiconductor industry is still almost 5 percent higher than other segments of the electronics industry; U.S. shipments of computer and office equipment (SIC 371) and communication equipment (SIC 366) have only grown at a CAGR of 8.4 and 8.1 percent respectively between 1980 and 1998.

Excess wafer fab capacity, the Asian economic crisis, and tighter inventory management by computer manufacturers has contributed to these below-average results, and has hampered the industry recovery. And although the business outlook is steadily improving, the economics of shortage and oversupply radically affect the semiconductor industry, causing extreme variations.

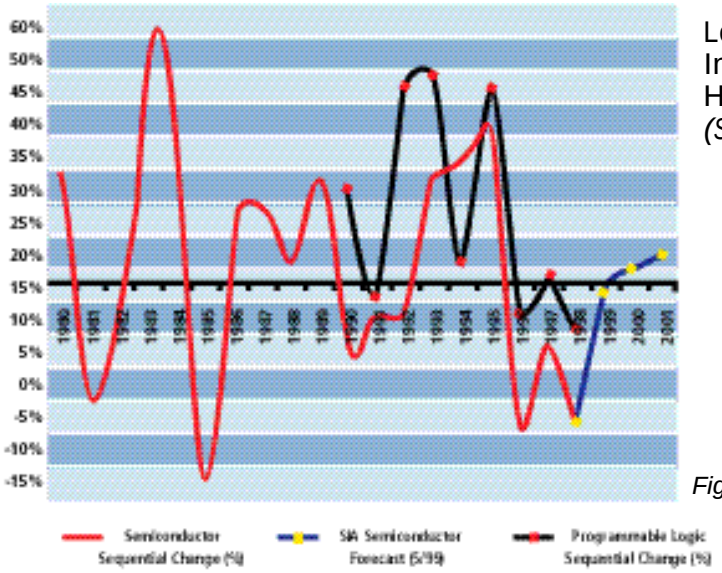
Semiconductor Industry Recovery

On a global basis, Japan's nascent recovery appears underway as the economy grew 1.9 percent in Q199. South Korea is already well on it's way to recovery from a dismal 5.8 percent GDP decline in 1998 with GDP growth of 4.6 percent in Q199.

A good way to gauge the health of the semiconductor industry is by measuring the utilization of the available wafer fabrication capacity, as shown in Figure 2. This graph shows three solid quarters of increasing utilization and the trend is likely to continue upwards, due to strong demand.

The Semiconductor Industry Association (SIA) forecasts that total semiconductor sales are expected to reach about $140.8 billion in 1999, a jump of 12.1 percent over 1998. Industry growth will be fueled by sales of PCs and Internet-related equipment. This compares with an 8.4 percent decline to $125.61billion in 1998 (Figure 3).

(Continued)



Long Term Semiconductor Industry Outlook (%) History And Forecast *(Source: WSTS Inc./SIA)*

Figure 1

Semiconductor Industry Outlook (Billion $) *(Source: WSTS Inc./SIA)*

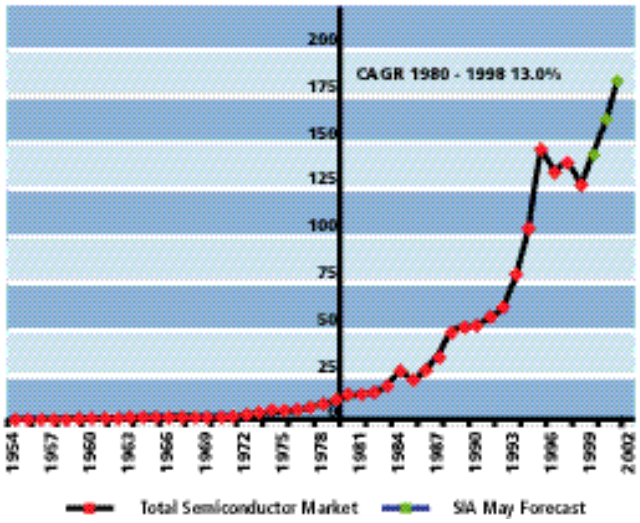


Figure 2

Programmable Logic is Leading the Industry

While the semiconductor industry has exhibited low double-digit growth, the programmable logic sector has grown more rapidly. From 1989, the first year when these statistics were available, to 1998 the PLD industry grew at a CAGR of 21.4 percent. Programmable logic is forecasted to continue to outpace the overall semiconductor industry at 18.6 percent growth in 1999. According to industry prognosticators, the longer-term outlook for PLDs will remain positive with growth rates of around 20% for the next few years.

This growth is fueled in part by the increased capabilities of our latest programmable logic and software technologies. Plus, many futuristic applications such as those made possible by the Xilinx Online technologies, can only be accomplished through field-reconfigurable programmable logic. These features, combined with very efficient development tools and significantly lower costs, are compelling many companies to view programmable logic technology as their primary design solution. Σ

32-Channel (Duplex) ADPCM Transcoder for Virtex FPGAs

Digital signal processing without the complicated DSP chip—that's the power of Xilinx DSP solutions in Virtex FPGAs.

by David Mann, Marketing Communications, Integrated Silicon Systems Ltd, dmann@iss-dsp.com

Thanks to the advent of high-density programmable logic device technology, you can now do high-performance ADPCM in off-the-shelf silicon without all the problems associated with a programmable digital signal processor. Xilinx AllianceCORE partner, Integrated Silicon Systems (ISS), offers reusable cores for ITU-compliant ADPCM Transcoders.

Up to 32 Duplex ADPCM Channels

Xilinx recently released an ISS-designed 8-channel (duplex) ADPCM AllianceCORE for the Virtex family. This solution supports ITU G.721, G. 723, G.726, G.726a, G.727a, or G.727, and has been tested and verified to be fully compliant using the ITU specified test vectors. This same IP core was originally targeted at the XC4000 family, an FPGA architecture with distributed RAM features. However, thanks to the enhanced Block RAM features of the new Virtex FPGA architecture, ISS can realize a complete 32-channel (duplex) ADPCM Transcoder on a single V150BG352-4 Virtex device.

ISS achieved this unusually compact FPGA implementation by mapping all the multiplication calls in the transcoder algorithm into a single multiplier. What's more, ISS can also deliver fully customized variants. For example, the PCM input channel multiplexing and serial-to-parallel conversion circuitry may be added as required to suit your Virtex-based target system.

Online Configuration

The 8-channel (duplex) version ADPCM AllianceCORE for Virtex is "on-line configurable" in terms of compression rate and PCM companding algorithm: A-law (for Europe), µ-law (for USA). This means that each channel can be individually programmed.

ISS ADPCM AllianceCORE deliverables include netlists, test-bench files, and command scripts, together with technical documentation, on-line, and hot-line technical support for one year. You can get the Xilinx AllianceCORE datasheet (.pdf) for the ISS ADPCM core at: www.iss-dsp.com/adpcm.

Conclusion

The Virtex architecture is capable of hosting a 32-channel (duplex) ADPCM Transcoder that is on-line configurable for compression rate, µ/A-law, and number of channels. This demonstrates the attractiveness of the Virtex family as a viable platform for high-performance DSP applications using high-quality IP cores. Σ

Adaptive Differential Pulse Code Modulation

ADPCM (Adaptive Differential Pulse Code Modulation) is a dynamically adaptive form of Differential PCM. DPCM is a more bandwidth-efficient than PCM because voice data is represented more succinctly as the difference between the present signal value and the previous. ADPCM cuts the bandwidth of straight PCM by around 50 percent through a process called companding.

The Virtex Family– a Powerful ASIC Alternative

With the success of the Virtex FPGA family, programmable logic technology has reached density, feature, and performance levels that make it a viable ASIC alternative in system-level applications.

by Shelly Davis, Virtex Marketing Manager, Xilinx,
sdavis@xilinx.com

The ASIC industry is moving toward deep sub-micron processes and standard cell methods instead of gate array design implementations. However, standard cell technology, though it's great for enabling system-level designs, is complex in both architecture and implementation. Standard cell design cycles are becoming more complex, with rigorous verification and testing required in the pre-silicon phase.

Non Recurring Engineering charges (NRE) and minimum volume requirement hurdles are becoming more and more prohibitive, and leading standard cell vendors must choose both customers and applications carefully, because of the high up-front investment required by customer and vendor. However if you simply want to get to market quickly, with designs that work and fit within your budget, Virtex FPGAs may offer the right solution for you.

Design Cycles Stretch Out

Throughout the early 1990's, ASIC software developers focussed on creating algorithms and point tools that reduced die size. Process technology suddenly jumped from 1.0-µ geometries to the deep-submicron 0.13-µ processes in development today. ASIC tools lost their foothold and are just now strengthening their ability to comprehend the issues of interconnect, power, and simultaneously switching outputs (SSO) that are so critical in deep-submicron design. ASIC Designers spend less and less time in the "window of innovation" (authoring, creating, and determining optimal feature sets) and too much time in the simulation and verification stages of the design cycle. In fact, studies show that only 20% of the typical ASIC design process is spent in the innovation stage.

Figure 1 shows the relative time spent in each of the design tasks required in today's ASIC environment. More point tools are required, and the time spent in verification becomes the overriding task. Test development must also be factored in. Studies have shown that up to 40% of the entire design process can be spent in test development.

Typical ASIC Design Process
(Source: Electronic Systems, Jan. 99)

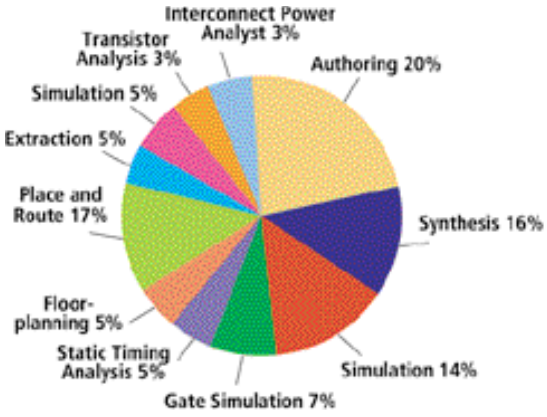


Figure 1

(Continued)

In the ASIC environment, exhaustive simulation and verification is crucial, because designers can't afford mistakes. NRE charges that cover the costs of prototyping were actually decreasing throughout the 1980's and early 1990's, as third party EDA companies picked up more of the design tasks. But NRE charges for deep-submicron devices radically reverse that trend. Mask sets for one 0.18µ device can run well over $200,000. With average standard cell design cycles running 9 to 12 months, a single mistake found during test, after the prototype cycle is complete, can put a project six months behind schedule and require a second NRE.

Virtex FPGAs Solve Deep-Submicron ASIC Design Dilemmas

Because our devices are programmable, we have "pre-engineered" many of the issues that plague system-level ASIC designers today. Each Virtex device is already specified, verified, and tuned for proper I/O performance. SSO, power, and interconnect issues have been resolved prior to the production release of the FPGA family.

In the development phase, you are working with actual silicon, so modeling, simulation, and verification results reflect actual silicon performance. Thus, you can spend more time in the "window of innovation," working on authoring, feature development, and system-level integration, without sweating through a silicon prototyping phase before any real in-system testing can begin.

Figure 2 shows the relative design-cycle times for ASICs vs. FPGAs. With decreases in process technology, design cycles have actually increased. FPGAs still have the shortest design cycles, and Integrated Circuit Engineering estimates that development costs are 92% less for and FPGA than for a comparable ASIC solution.

Design Tools Further the FPGA Advantage

Time to market is one of the most crucial issues you face. Shortening the design cycle is critical. As an ASIC alternative, FPGA design tools play a major role in increasing the advantage through dramatic reductions in compile times and a streamlined design flow.

Integrated Circuit Engineering estimates that development costs are 92% less for an FPGA than for a comparable ASIC solution.

In the past, FPGA designers were accustomed to compile times on the order of 10,000 gates/hour. Through new algorithms, Xilinx place and routing tools compile at the rate of 100,000 gates/minute. This provides the ability to achieve multiple design turns per day, which allows more time to spend in the window of innovation, creating an optimal design. The advantage of quickly realizing a design change in actual, qualified silicon can not be accomplished in an ASIC design flow.

The ASIC design flow is powerful but complex, and requires lots of tools which are usually very expensive. Ultimately you are not in full control of the tools. The FPGA design flow requires fewer steps, which gives you the flexibility to control the entire design methodology.

A significant time saving example is the built-in FPGA scan insertion. Scan is used to test "stuck-at" faults within the silicon, and FPGAs are 100% tested. However, for ASICs,

(Continued)

Relative Design-Cycle Times

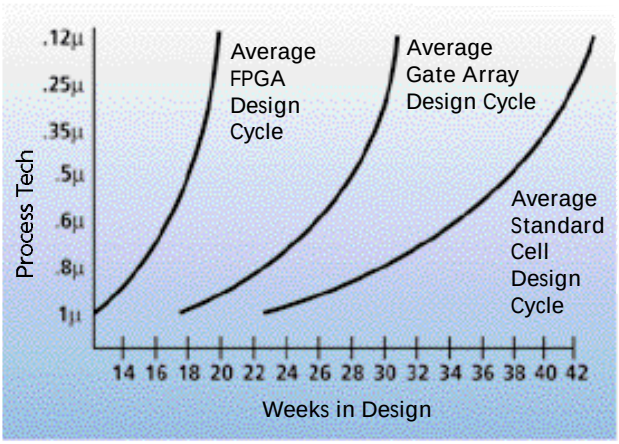


Figure 2

there is a lengthy verification process to ensure minimum coverage. The FPGA design tools are designed to place, route, and download the bitstream, allowing in-system verification. In the event of an Engineering Change Order (ECO), you can implement the change, perform functional verification, and place and route the design on real silicon.

The place and route tool outputs a standard delay format (SDF) file for use with static timing analyzers to aid in reaching timing convergence. Thus, an ECO can be implemented within the original design, instead of having to fix a board, or add components.

The ASIC Bar is Raised

Not only have system-level ASIC design cycles stretched out and NREs increased, but minimum order quantities have been raised as well. Eight- to twelve-inch wafers can yield hundreds of 0.18µ die. Wafer fabrication economics dictate that steady, high volume brings down costs, stabilizes manufacturing lines, and best utilizes expensive resources.

Many pure-play foundries require minimum purchases of 20,000 to 50,000 ASIC devices as a starting point for engagement. Because the investment is high for the ASIC vendor and for the customer, many first tier ASIC vendors have very exclusive customer criteria and accept only the highest volume, and most-stable projects from well-established customers. If your projects do not fit 1st tier ASIC vendor criteria, having the Virtex FPGA path for system-level design is a crucial alternative.

Summary

Our feature-rich Virtex devices, with up to one million system gates, including Block RAM, DLL's, and pre-verified core solutions, could not have been introduced at a more appropriate moment in the system-level design revolution.

As complex ASIC design becomes more expensive and tougher to complete in today's time-critical markets, Virtex FPGAs create real, programmable alternatives without the prohibitive development costs and drawbacks of an ASIC solution. Σ

ASIC Designers spend less and less time in the "window of innovation" (authoring, creating, and determining optimal feature sets) and too much time in the simulation and verification stages of the design cycle. In fact, studies show that only 20% of the typical ASIC design process is spent in the innovation stage.

Verilog GSR/GTS Simulation Methodology– Changes in the Alliance Series 2.1i Software

by Roberta Fulton, Technical
Marketing Engineer, Xilinx,
roberta.fulton@xilinx.com

With the release of Alliance Series 2.1i software, Xilinx has added support for compiled Verilog simulators such as Synopsys VCS, Model Technology VLOG, and Cadence NC-Verilog. These simulators are full Verilog simulators but behave differently in a few significant ways from interpretive Verilog simulators like Verilog-XL. Please see the individual tool manuals for details.

The previous GSR/GTS approach allowed you to define the GSR/GTS signal as a text macro with the **‘define** construct, at any time prior to the actual simulation of the GSR signal by an interpretive simulator like Verilog-XL. For example, **‘define GSR my_gsr_signal** could be entered on the command line at any time prior to simulation. But in compiled Verilog it is a requirement that all constructs be fully elaborated at the time they are compiled. This prohibits a text macro from being used by a **‘ifdef** construct contained in a library module if the text macro is defined in a higher level module or on the command line after the library has been compiled.

The new methodology works for both compiled and interpretive simulators. It defines a module called **glbl** that contains the GSR signal. The Unified Library component modules drive their internal GSR behavior by the glbl.GSR wire. The glbl.GSR wire can in turn be driven by any user-defined wire attached to it in the test fixture.

The New Simulation Methodology

While GSR alone is described here for simplicity, this methodology equally applies for the CPLD PRLD signal and the GR signal in older FPGA technologies, and works similarly for the global GTS signal on the output buffers. For a more complete explanation of the GSR/GTS methodology, please refer the 2.1i version of the Xilinx Synthesis and Simulation Guide on the Alliance Series 2.1i Documentation CD, or www.support.xilinx.com.

Describing the GSR Behavior (Verilog)

For Verilog simulation, all behaviorally described (inferred) and instantiated registers should have a common signal (GSR) that asynchronously sets or resets the register. Toggling GSR emulates the automatic Reset-on-Configuration mode of the FPGA. This is similar to the Power-on-Reset of an ASIC. If you do not do this, the flip-flops and latches in your simulation may initialize to an unknown state in the simulation.

The GSR net is present and may be pulsed in your implemented design (as it is in silicon) even if you do not instantiate the STARTUP block in your design. The function of STARTUP is to give you the option to control the global reset net from an external pin or from user-defined internal circuitry.

Often, mismatches between the Register Transfer Level (RTL) and gate-level simulations are caused by not fully defining the GSR behavior in the RTL. The new methodology allows the GSR behavior to be fully defined in the RTL, while properly synthesizing and implementing to a post-route netlist that can use the same test fixture. The Verilog UniSim library is used in RTL or gate-level simulations prior to implementation. Simulation at other points in the flow uses the Verilog SimPrims Libraries.

(Continued)

Describing the GSR Behavior (UniSim)

For UniSim functional simulation, you must set the value of the **glbl.GSR** net to the same name value as the GSR net, qualified by the appropriate scope identifiers. The scope identifiers are a combination of the test module scope and the design instance scope. The scope qualifiers are required because the scope information is needed when the **glbl.GSR** wires are interpreted by the Verilog UniSim simulation models to emulate a GSR signal.

For post-route timing simulation the test fixture template (.tv file), produced by running NGD2VER with the **-tf** option, contains most of the code required for defining and toggling GSR.

- **Without a STARTUP module -** Add the following to the test fixture file:

```
reg GSR;
assign glbl.GSR = GSR;
assign testfixture_name.instance_name.GSR=GSR;
// Only for RTL modeling of GSR
```

  For post-route timing simulation, you must omit the "assign" statement for GSR. This is because the net connections exist in the post-route design; retaining the assign definition causes a possible conflict with these connections.

- **With a STARTUP module -** If you do have a STARTUP block in your design, the signal you toggle is the external input port that controls the global reset pin of the STARTUP block. You should add the following to the test fixture module for the RTL modeling of the global reset pin:

```
reg port_connected_to_GSR_pin;
assign glbl.GSR = port_connected_to_GSR_pin;
```

For post-route timing simulation, you must omit the assign statement for the global reset signal. This is because the GSR net connections explicitly exist in the post-route design, and retaining the assign definition causes a possible conflict with these connections.

A Verilog global signal called **glbl.GSR** is defined within the STARTUP and STARTUP_VIRTEX modules to make the connection between the user logic and the global GSR net embedded in the Unified Library models. For post-route timing simulation, **glbl.GSR** is defined in the Verilog netlist that is created by NGD2VER.

You can compile the Verilog source files in any order because Verilog is compiled independently of the source module hierarchy except in cases of constructs like **‘define** or **‘defparam** as previously noted. However, Xilinx recommends that you specify the test fixture file before the Verilog netlist of your design, as in the following examples.

Invoking the Simulators

- **For RTL simulation,** enter the following:

```
verilog -y $XILINX/verilog/src/unisims
design.stim design.v $XILINX/verilog/src/glbl.v
```

  The path specified with the **-y** switch points the simulator to the UniSim models and is only necessary if Xilinx primitives are instantiated in your code. When targeting a device family other than the XC4000E/L/X, Spartan/XL, or Virtex families, change the UniSims reference in the path to the targeted device family.

- **For post-route simulation,** enter the following:

```
verilog design.stim time_sim.v
$XILINX/verilog/src/glbl.v
```

  In this example, the test fixture file is declared first followed by the simulation netlist created by the Xilinx tools. The name of the Xilinx simulation netlist may change depending on how the file was created. For Verilog-XL, it is also assumed that the -ul switch was specified during NGD2VER to specify the location of the SimPrims libraries using the **uselib** directive.

(Continued)

MTI ModelSim

- **For RTL simulation,** enter the following:

```
vlog design.stim design.v
$XILINX/veilog/src/glbl.v
vsim -L unisims testfixture_name glbl
```

This example targets the XC4000E/L/X, Spartan/XL, or Virtex families and assumes the UniSim libraries are properly compiled and named "unisims". For more information on the compilation of the ModelSim libraries, refer to http://www.xilinx.com/techdocs/1923.htm.

- **For post-route simulation,** enter the following:

```
vlog design.stim time_sim.v
$XILINX/verilog/src/glbl.v
vsim -L simprims testfixture_name glbl
```

This example is based on targeting the SimPrim libraries, which have been properly compiled, named, and mapped to simprims. Also, the name of the simulation netlist may change depending on how the file is created.

Note: Xilinx recommends giving the name "test" to the main module in the test fixture file. This name is consistent with the name of the test fixture module that is written later in the design flow by NGD2VER during post-route simulation. If this naming consistency is maintained, you can use the same test fixture file for simulation at all stages of the design flow with minimal modification.

Examples

```
module my_counter (CLK, D, Q, COUT);
input CLK, D;
output Q;
output [3:0] COUT;

wire GSR;
reg [3:0] COUT;

always @(posedge GSR or posedge CLK)
 begin
 if (GSR == 1'b1)
 COUT = 4'h0;
 else
 COUT = COUT + 1'b1;
 end

// FDCE instantiation
// GSR is modeled as a wire within a global module. So,
// CLR does not need to be connected to GSR and the flop
// will still be reset with GSR.

FDCE U0 (.Q (Q), .D (D), .C (CLK), .CE (1'b1), .CLR (1'b0));

endmodule
```

Because GSR is declared as a floating wire and is not in the port list, the synthesis tool optimizes the GSR signal out of the design. GSR is replaced later by the implementation software for all post-implementation simulation netlists. In the test fixture file, set GSR to "**test.uut.GSR**" (the name of the global set/reset signal, qualified by the name of the design instantiation instance name and the test fixture instance name). Because there is no STARTUP block, a connection to GSR is made in the test fixture via an assign statement.

In this example, the active high GSR signal in the XC4000 family device is activated by driving it High. 100 ns later, it is deactivated by driving it Low. (100 ns is an arbitrarily chosen value.)

You can use the same test fixture for simulating at other stages in the design flow if this method is used.

```
`timescale 1 ns / 1 ps
module test;
reg CLK, D;
wire Q;
wire [3:0] COUT;

reg GSR;
assign glbl.GSR = GSR;
assign test.uut.GSR = GSR;

my_counter uut (.CLK (CLK), .D (D), .Q (Q), .COUT (COUT));

initial begin
$timeformat(-9,1,"ns",12);
$display("\t T C G D Q C");
$display("\t i L S O");
$display("\t m K R U");
$display("\t e T");
$monitor("%t %b %b %b %b %h", $time, CLK, GSR, D, Q, COUT);
end

initial begin
CLK = 0;
forever #25 CLK = ~CLK;
end

initial begin
#0 {GSR, D} = 2'b11;
#100 {GSR, D} = 2'b10;
#100 {GSR, D} = 2'b00;
#100 {GSR, D} = 2'b01;
#100 $finish;
end

endmodule
```

The Evans & Sutherland Ensemble Image Generator

Virtex and Spartan FPGAs help Evans & Sutherland accelerate their development schedule, and roll in new features during and after the main design cycle.

by John F. Snow, PC Simulation Engineering, Evans & Sutherland, jsnow@es.com

Since the early 1970s, E&S has produced industry-leading computer image generators for commercial and military pilot training. These special-purpose graphics computers generate realistic and detailed out-the-window and sensor simulations in real-time for training people to operate a wide range of vehicles including aircraft, ships, trains, tanks, and automobiles.

Ensemble System Requirements

In April 1998, a small engineering team began working on the product definition of the company's first PC-based image generator for the simulation market, using graphics accelerator boards based on the company's latest generation of REALimage technology. Other image generators made by the company use custom ASICs designed specifically for the demanding task of real-time image generation.

While based on a standard graphics chip, these new PCI form-factor graphics boards have been specifically tailored for the simulation market and contain simulation-specific features not generally available on standard PC-based graphics accelerator boards. This new image generator system, Ensemble™, is software compatible with the company's current high-end image generator, Harmony™.

The engineering team defined three graphics accelerator boards, Quartet™, Duet™, and Solo™, with different levels of performance to meet different price points. All of the cards have the following features designed specifically for the simulation market:

- Order-independent antialiasing with four subsamples per pixel generated with no impact to pixel fill rate.
- Video genlock for synchronizing the video rates of multiple cards.
- Edge blending to roll off the video intensity along the edges to allow seamless overlaps of channels when using projectors.
- Own-ship lighting to simulate the effect of vehicle lights illuminating the scene implemented in hardware so as not to impact the rendering performance.
- Postprocessing effects to simulate infrared or night vision goggle sensor channels.
- A double-buffered overlay plane used for displaying the symbols typically seen on a heads-up display.

(Continued)

Quartet Board Block Diagram

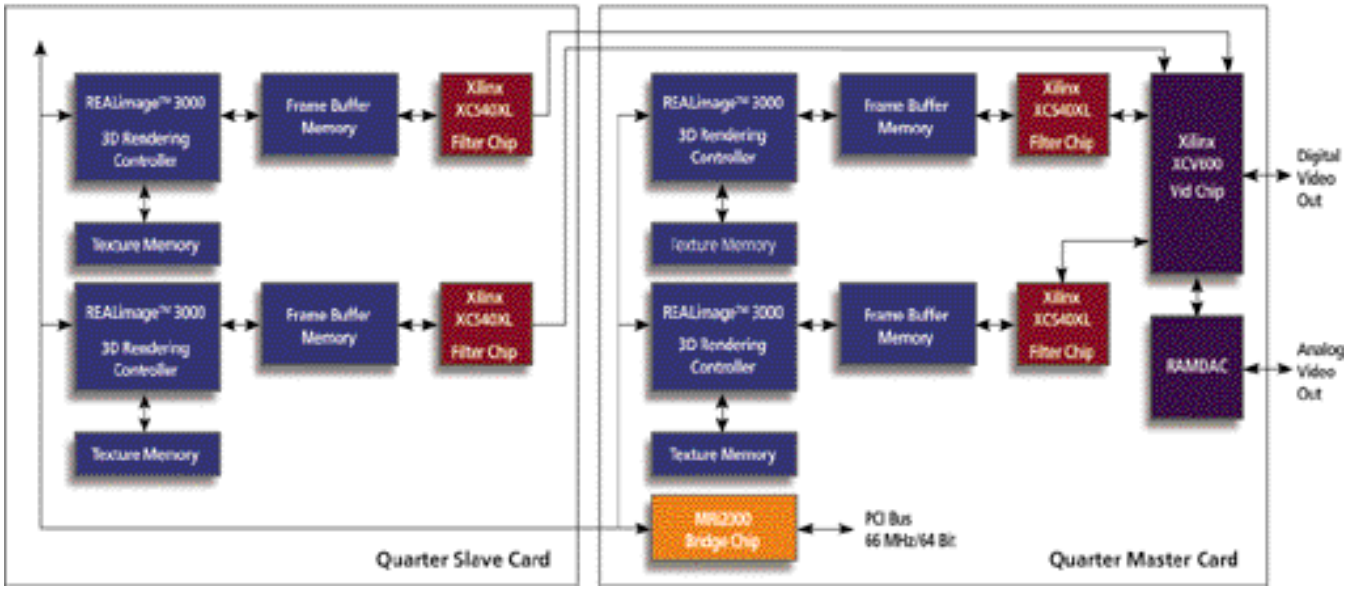


Figure 1

Selecting a Technology

Most of these simulation-specific features are implemented in a postprocessing device, called the Vid chip, which fits between the frame buffer and the RAMDAC, and is the key element in differentiating the Ensemble graphics accelerators from boards developed for the more traditional PC workstation, gaming, or home markets.

Due to its central role in the operation of the boards, the engineering team focused on selection of the technology for the Vid chip early in the design process. The engineering team wanted to use an FPGA for the Vid chip, but were not sure that FPGAs would be dense enough to fit all of the feature set, or fast enough to meet the requirements of postprocessing video "on-the-fly."

Standard REALimage-based graphics accelerator boards have a gate array called the Pixel Converter 2000 that fits in the same place that the Vid chip occupies on the Ensemble boards. The Vid chip implements most of what is in the Pixel Converter 2000 and adds to that all the simulation-specific post-processing features.

Evaluating the Alternatives

The engineering team took the Verilog code for the Pixel Converter 2000 ASIC and, without modification, compiled the code using the Xilinx and Altera synthesis tool chains. The initial Xilinx target was a member of the XC4000 family.

The results showed that the Pixel Converter code, which is implemented in a 50K gate device, almost filled up the largest Xilinx and Altera parts available at the time, and ran only at half the required frequency. Note that this was without modifying the code in any way to make it map more efficiently to the FPGA architecture, but the results left the team uneasy about the prospects of implementing the Vid chip in a FPGA.

At this time, Xilinx provided E&S with preliminary information about their new Virtex family. As soon as Xilinx delivered an early release of the tool chain with Virtex support, we recompiled the Pixel Converter code and targeted for Virtex. This time the results showed that the Pixel Converter code occupied about 50% of an XCV300 and ran at about 75% of the required speed. These results led us to

(Continued)

believe that, with some work, the Vid chip could be implemented in a Virtex part and meet the system timing requirements. The team targeted the Vid chip to run at 80 MHz, which would support a 1280x1024-resolution screen refreshed at 85 Hz.

Implementing the Design

The Quartet board (Fig. 1), which is actually a two-board set with four REALimage 3000 chips, has the highest pixel fill performance of any PC-based graphics accelerator card currently produced by E&S. It can process 400 million textured, shaded, and antialiased pixels per second.

The Quartet boards are designed to use the Virtex XCV600 for the Vid and four Spartan XCS40XL "Filter" chips. The Filter chips implement the antialiasing filter and reduce the pin count to the Vid chip.

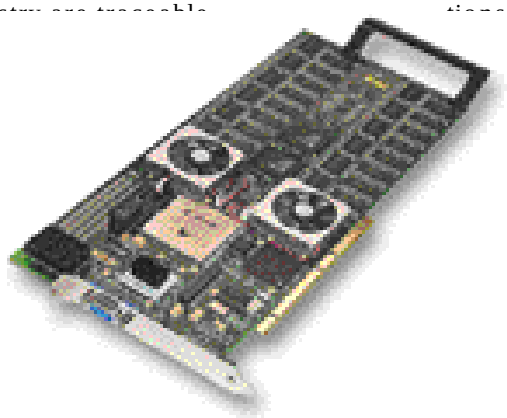
Detailed design of the Quartet board and the Vid chip began in August of 1998. The first prototype Quartet boards were powered on for the first time in early November 1998. A prototype Ensemble system was demonstrated three weeks later at a trade show for the simulation industry.

Using Virtex FPGAs to implement the Vid chip allowed the Ensemble engineering team to pursue this aggressive design schedule. The team was able to commit to pin-outs of the Vid and Filter chips early, to begin the circuit board layout while continuing to roll features into the design of the FPGAs. The availability of large amounts of block SelectRAM in the Virtex device also allowed us to implement many of our sophisticated post-processing algorithms.

Conclusion

Without the density and speed of the Virtex parts, the Vid chip probably would have been implemented in a gate array, causing a much longer development cycle for the Ensemble boards. The engineering team would have had less opportunity to roll in new and enhanced features during and after the main design cycle. Σ

Many of the major milestones in the evolution of the computer graphics indu
to Evans & Sutherland Compu
Founded in 1968 by Dr. Ivan S
Dr. David Evans, E&S is headq
in Salt Lake City, Utah. E&S bu
units deliver high-quality visua
systems for simulation and training in defense and commercial applica-
as well as high-performance
systems for workstation
graphics, digital video, and
other applications throughout
world.
ore information on Evans &
nd see www.es.com



Questions & Answers

From the Xilinx Applications Engineering Staff

by Kamal Koraiem, Product Applications Manager, Xilinx, kamalk@xilinx.com

Virtex

Q: What's the recommended way to asynchronously set or reset flip-flops in a Virtex design, and why? Is it still necessary to use the STARTUP_VIRTEX block?

A: Write this high fan out reset/set signal explicitly in the HDL codes, and do not use the STARTUP_VIRTEX block. There are two advantages:

- This method gives you a faster design. The reset/set signal will be routed onto the secondary long lines in the device, which are global lines with minimal skews and high speed. Therefore, the reset/set signal on the secondary lines is much faster than the the GSR net of the STARTUP_VIRTEX block. Because Virtex is rich in routing, placing and routing this signal on the global lines is easily done by our software.
- Our trace program will analyze the delays of this explicitly written reset/set signal. You can read the .twr file (report file of the trace program) and find out exactly how fast it is. Trace does not analyze the delays on the GSR net of the STARTUP_VIRTEX block. It is not necessary to use the STARTUP_VIRTEX block if the reset/set signal is explicitly coded. However, you still have the option of using the STARUP_VIRTEX block if you want, if GSR speed is not a concern.

Core Generator

Q: I just tried generating a CORE Generator module using the new 2.1i software release. I specified that I wanted a VHDL (or Verilog) behavioral model, but did not get one in my project directory. All I see is a .VHO (or .VEO) file.

A: The HDL behavioral simulation flow is new in the 2.1i version of the CORE Generator. The 2.1i CORE Generator does not generate a .VHD or .V file in the project directory. Instead, it creates a .VHO (for VHDL) or .VEO (for Verilog) template file containing the code snippets required to integrate the core into a higher level design block's behavioral simulation netlist.

In addition, before any behavioral simulation of a core can be done, you must:

- Run the **get_models** utility to extract the models into a separate source library.
- Analyze the library, if required by your simulator.
- Set your simulator to point to the extracted (and analyzed) library.

Refer to the Design Flows chapter of the CORE Generator User Guide (available from within the CORE Generator under Help-> Online Documentation) for more details.

(Continued)

Implementation

Q: I hear that Xilinx has a new command line program that will allow me to run the full suite of Xilinx implementation tools. What is it and where can I find it?

A: This new command line tool is called XFLOW. It allows you to run the full suite of Xilinx implementation, simulation, and configuration flows. It is device independent and has a customizable interface to the Xilinx tools. XFLOW accepts design, flow, and option files as input. This tool is making its debut in the 2.1i release. To run it, simply type in "xflow" from the command line. For detailed information and examples, please refer to the XFLOW chapter of the 2.1i Development System Reference Guide.

Q: Do the implementation tools have a feature that allows me to automatically specify the copying of files into version and revision directories for the purpose of maintaining source netlist information (.EDN, .EDF, .VHD, .VER, .SCH, etc...) along with implementation output?

A: Yes. The Xilinx Design Manager has a "backdoor" feature that allows you to automatically specify the copying of files into each and every newly created version or revision directory. To access this "File List" feature, you must select your project name in the Design Manger window, then right-click and select "Properties". For more detailed instructions, please reference:
http://support.xilinx.com/techdocs/6704.htm

Timing Analysis

Q: What is the Hierarchical Report in the Timing Analyzer?

A: The Timing Analyzer displays FPGA and CPLD analysis reports in a hierarchical format. The report window (used to display all .twr files) now has three panes (Figure1). The lower-right pane is the text view. It just displays the text of the .twr file. The left pane is the index or outline view. Click on the labels to scroll the corresponding line in the report to the top of the text view. Click on the "+" or "-" buttons to expand or collapse topics. The other standard mouse and keyboard manipulations for tree views also work.

The upper-right pane is the context or path view. It shows the path through the topic hierarchy to the item currently selected in the index view. The contents of the context view change only when the selection in the index view changes. Click on items in the context view to scroll the text view to the appropriate place.

Report Window

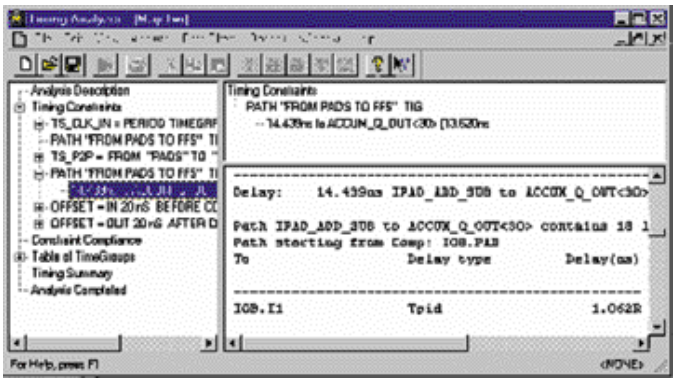


Figure 1

This can be handy if you're deep in a list of paths for a timing constraint and need to jump back to the timing constraint definition.

Q: How do I add a probe in FPGA Editor?

A: The following steps walk you through the procedure to create a probe in FPGA Editor. Also refer to Solution 6616, at: http://www.xilinx.com/techdocs/6616.htm:

1. Launch FPGA Editor either via command line **(fpga_editor)** or the GUI.
2. Load the design's ncd file, either from map (map.ncd) or par (design.ncd), and select the Edit Mode to be "Read Write".
3. Select the "Probes" button on the leftside.

(Continued)

4. Select the "Add" button, to add a probe.
5. Hilight the net that you want to probe, in the "Select Net" box.
6. Choose the "Method," either Automatic or Manual, in the "Select Pin Numbers" box.
   - Automatic - FPGA Editor picks the closest pin to the net to be probed.
   - Manual - You choose which pin is used for the probe.
7. Select the "OK" button, to add the probe.
8. The probe is listed in the "Probes" window with the delay and the pin number.

Q: How do I apply a PERIOD timing specifications in 2.1i when using a Virtex CLKDLL?

A: The rules regarding property tracing through the DLL have changed in 2.1i (Figure 2). When a **TNM_NET** property is traced into the CLKIN pin of a Virtex CLKDLL component, the TNM group and its usage will be examined. The TNM will be pushed through the CLKDLL only if the following conditions are met:

- The TNM group name is used in exactly one PERIOD specification.
- The TNM group name is not used in any FROM-TO or OFFSET specifications.
- The TNM group name is not referenced in any user group definition.

If any of the above conditions are not met, the TNM will not be pushed through the CLKDLL, and the following error message will be issued: ERROR:NgdHelpers:702 - The TNM "PAD_CLK" drives the CLKIN pin of CLKDLL "$I1".

This TNM cannot be traced through the CLKDLL because it is not used in exactly one PERIOD specification. This TNM is used in the following user groups and/or specifications:

```
TS_PAD_CLK=PERIOD PAD_CLK 20000.000000 pS HIGH 50.000000%
TS_01=FROM PAD_CLK TO PADS 20000.000000 pS
```

If the above conditions are met each clock output pin on the CLKDLL will be examined to see if it is connected to a net with at least one other connection (it is not a dangling net). If the output pin has a net, a new TNM group will be created on that net, and a new PERIOD specification will be created for that group. The new specification will be copied from the original PERIOD specification, and then modified as follows:

Output Pin	Modifications to PERIOD Specification
CLK0 CLK90 CLK180 CLK270	If the DUTY_CYCLE_CORRECTION=TRUE property is found on or above the CLKDLL, the duty cycle will be adjusted to 50%. If DUTY_CYCLE_CORRECTION=FALSE, the duty cycle will be unchanged from the original PERIOD specification. If no DUTY_CYCLE_CORRECTION property is found, the default value of TRUE will be assumed.
CLK2X	The PERIOD value will be doubled (if originally expressed as a frequency) or divided in half (if originally expressed as a delay). The duty cycle will also be adjusted to 50%.
CLKDV	The PERIOD value will be divided (if a frequency) or multiplied (if a delay) by the value in the CLKDV_DIVIDE property. If no such property is found on or above the CLKDLL, the default value of 2.0 will be used. The duty cycle will also be adjusted to 50%

Figure 2

(Continued)

- If the original TNM_NET property is pushed only into the CLKDLL CLKIN pin (it does not trace to any appropriate elements without going through the CLKDLL), the original TNM group and the original PERIOD specification will be eliminated from the design.
- If a newly-created TNM group is pushed forward from a CLKDLL output and encounters the CLKIN input of a second CLKDLL (such as in the 4X configuration), the above process will be repeated to fur ther adjust the PERIOD specifications per the behavior of the second CLKDLL.
- If the group created for the first CLKDLL traces only into the second CLKDLL, that group and its PERIOD specification become unnecessary and will be eliminated. For further information regarding property tracing through the CLKDLL refer to the Developmental System Reference Guide Chapter 6.

Board-Level Timing Analysis

Q: Can I do a board-level timing analysis that includes my FPGA as well as other chips?

A: Yes. You can do a board-level static timing analysis using either the Mentor Tau tool or the Viewlogic Blast tool. Both can take in a STAMP file created by the Xilinx tool trace. Please see the following document for trace options and the method to create the stamp files: http://support.xilinx.com/pub/documentation/interfaces/tau_blast.pdf

Virtex Configuration

Q: Is Readback supported in Virtex?

A: Readback in Virtex is supported through the SelectMAP and JTAG ports. Readback may be used for in-system configuration verification, or internal state capturing and logic debugging. Xilinx Application Note XAPP 138 Virtex Configuration and Readback describes the process and procedures for performing readback operations in detail. XAPP 138 also describes the readback files produced by the 2.1i software that are used in readback operations. While the options to produce these readback files are also included in the 1.5i software, some of the resulting files needed for verification are generated incorrectly. Therefore, the 2.1i software is required to support this functionality.

Q: Which BitGen options should be used when configuring Virtex devices in a daisy-chain?

A: When configuring multiple Virtex devices in a serial daisy-chain the DONE pins must be connected together and the DONE_cycle must be first in the startup sequence. This is most easily accomplished by using default settings. If not specified by the user, the settings for DONE_cycle, GTS_cycle, GSR_cycle, and GWE_cycle will default to the startup sequence states 4, 5, 6, 6, respectively.

When the first device has received all of its configuration data it will also have received the command to run through the startup sequence. With all the DONE pins tied together, unconfigured devices will externally hold the DONE pins of configured devices LOW. When the startup sequence releases the DONE pin (DONE_cycle) the sequencer will not progress passed this state until the DONE pin has externally gone HIGH.

If it were allowed to do so, once the GTS_cycle state was achieved, the DOUT pin would no longer be capable of passing configuration data to the daisy-chained FPGAs. This would also be the case if the GTS_cycle was set to an earlier state than the DONE_cycle.

FlibGen, FlibTime, and ChipView – Power Tools for FPGA Design

No matter how awe-inspiring your latest FPGA design is, the next one will have to be even better. Three tools from Fliptronics can help you produce faster, denser designs in less time with less effort.

by Philip Freidin,
Fliptronics,
philip@fliptronics.com

Fliptronics has introduced three software power tools that can produce faster, more compact Xilinx FPGA designs. Using these tools, you can generate highly-optimized logic modules with predictable timing, and you can quickly view chip layout and critical timing paths.

Customized Modules (FlibGen)

The first tool, FlibGen, works in concert with Viewlogic ViewDraw to produce over 35,000 modules (Table 1) that have been optimized for speed and area. Using an intuitive module design menu invoked within ViewDraw (Figure 1), you can choose the function type, data width (from 2 to 32 bits) and other characteristics. FlibGen then creates a complete detailed schematic of the desired module, as well as a module symbol that can be placed in the design schematic.

FlibGen modules are particularly suited to high-performance designs, for the following reasons:

- Modules are consistently floorplanned, with two bits of datapath implemented in each CLB row. This consistency allows you to easily and efficiently tile the modules by attaching RLOC properties to the module symbols. The module schematic created by FlibGen contains a floorplanning diagram describing CLB usage, module shape, and bit assignments.
- Modules contain only the features and functions you specify.
- Because the modules are completely floor planned, they offer predictable timing.
- No special processing is required for functional simulation. Signals can be followed into and through modules.
- You can see exactly what's been built and, if desired, modify it.
- Modules are optimal. They are designed the same way a power user would design the function.

Predictable Timing (FlibTime)

The second tool, FlibTime, is an adjunct to FlibGen that reports detailed timing information for each module generated. Using FlibTime, you can:

- Create a detailed timing data sheet for any module (Figure 2). Included in this data is timing from any source pin to any destination pin, as well as timing starting and ending in internal registers.
- Quickly determine the differences in module timing among speed grades and Xilinx FPGA families.
- Make tradeoffs early in the design by determining the highest possible operating frequency for various key modules.

Suppose, for example, that a design depends on a 25-bit counter with parallel load running at 66MHz. The question, "What XC4000XL speed grade will support this requirement?" can be answered in a few seconds by specifying the module with FlibGen and analyzing its timing

Function Type	Types Available
Accumulators	23,808
Adders	2,232
Adder/subtractors	2,232
Comparators	310
Decrementors	1,116
Down counters	186
Incrementors	1,116
Incrementor/decrementors	1,116
Min/max Selectors	248
Multiplexers	868
Registers	62
Subtractors	2,232
Up counters	186
Up/down counters	186

Table 1

FlibGen can create over 35,000 building blocks optimized for high speed and low area.

(Continued)

with FlibTime. (The answer, by the way, is –09.) Similarly, you can determine the best performance obtainable from a given device type and speed grade, without generating any modules, or running place and route.

Viewing the Results (ChipView)

Once you have placed and routed your design, ChipView provides a quick way of viewing placement results and critical paths. Using PAR output files as its input, ChipView:

- Generates a 2-D plot of chip placement (Figure 3).
- Through shape and color coding, identifies function generators, flip-flops, single- and dual-port RAMs, ROMs, and carry logic.
- Shows the positions and names of all I/O pins, and indicates which are locked down.
- Identifies floorplanned and unfloorplanned portions of the design.

In addition to showing the layout, ChipView can also identify worst-case timing paths. It post-processes the output of the TRCE program, lists timing delays in a tree format, and graphically depicts timing paths on the chip layout diagram. By examining worst-case paths visually, you can quickly determine the portions of a design that need further floorplanning. One such path is shown in Figure 3; It starts at the green circle (R11C5), extends up column 5 and ends at the red circle (R3C1). ChipView works with both schematic- and HDL-based design flows.

Summary

FlibGen, FlibTime, and ChipView provide you with a means of generating highly-optimized building blocks with predictable timing, and of quickly appraising FPGA layout and timing-critical paths. All three programs include online context-sensitive help, run under Windows NT 4.0, and support the Xilinx XC4000E, XC4000XL, XC4000XLA, Spartan, andSpartanXL families. These programs can be purchased separately or together; the price of each is $500. Σ

For more information, call Fliptronics at (408) 737-0295, or visit the Fliptronics website at http://www.fliptronics.com.

FlibGen Menu for Building Adders

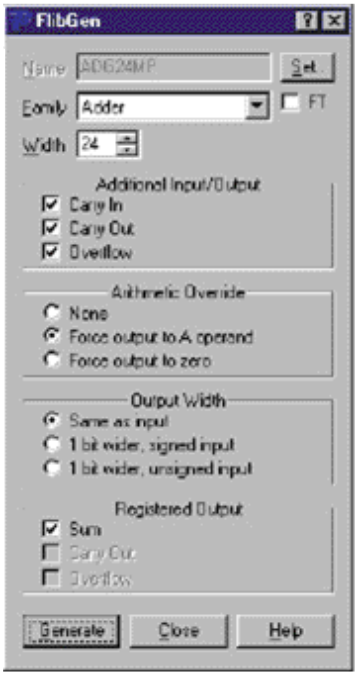
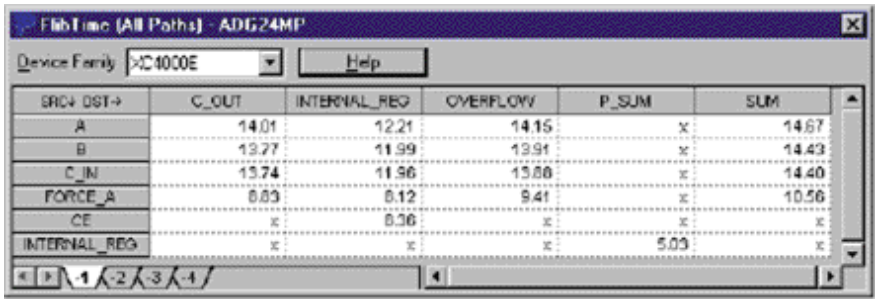


Figure 1

FlibTime Timing Report Window for 24-bit Adder, in XC4000E, -1 Speed Grade



SRC\ DST→	C_OUT	INTERNAL_REG	OVERFLOW	P_SUM	SUM
A	14.01	12.21	14.15	x	14.67
B	13.77	11.99	13.91	x	14.43
C_IN	13.74	11.96	13.88	x	14.40
FORCE_A	8.83	8.12	9.41	x	10.56
CE	x	8.36	x	x	x
INTERNAL_REG	x	x	x	5.03	x

Figure 2

ChipView Displaying a 16 Bit RISC CPU

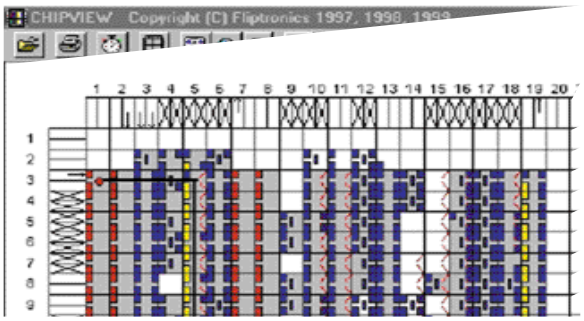


Figure 3

New *FPGA* Compiler II– For Million-Gate Designs

Achieve dramatic increases in productivity and perfor mance for your million-gate Xilinx Virtex FPGA designs.

by Mark Bollar, FPGA Marketing Program Manager, Synopsys, mbollar@synopsys.com

You now have a revolutionary new synthesis solution crafted to meet your advanced design needs. The Synopsys FPGA Compiler II v3.2 implements ASIC-compatible design flows and methodologies for million-gate FPGAs, while also effortlessly delivering "push-button" synthesis with unprecedented quality of results.

FPGA Compiler II Defined

Since it was introduced, FPGA Express has been extremely well received in the market place for its easy-to-use GUI and its highly efficient synthesis technology. Synopsys now introduces a second distinct FPGA synthesis product: FPGA Compiler II. FPGA Compiler II builds on the success of FPGA Express by encompassing all of the features and functionality of the FPGA Express tool. It also adds many new breakthrough features that allow you to implement a million-gate design within your tight schedule and high performance design requirements.

You can use FPGA Compiler II as a standalone tool in a traditional FPGA design flow or, if you are an existing ASIC designer or plan on implementing an ASIC style flow for additional performance reasons, FPGA Compiler II is designed to plug into your existing or enhanced flow.

Key features supported by FPGA Compiler II v3.2:

- Reads and writes Design Compiler shell scripts.
- Outputs Synopsys .db database files.
- Support for DesignWare Foundation IP library.
- VISTA gate-level and RTL-level schematic viewer.
- Available on Solaris and HPUX as well as Windows 95/98/NT.

Design Flow

The FPGA Compiler II flow (Figures 1 and 2) offers you many key advantages when migrating an ASIC design to an FPGA or when performing ASIC prototyping within a standard ASIC flow.

First is the capability to utilize your industry standard Design Compiler ASIC synthesis shell

(Continued)

FPGA Compiler II Input Flow

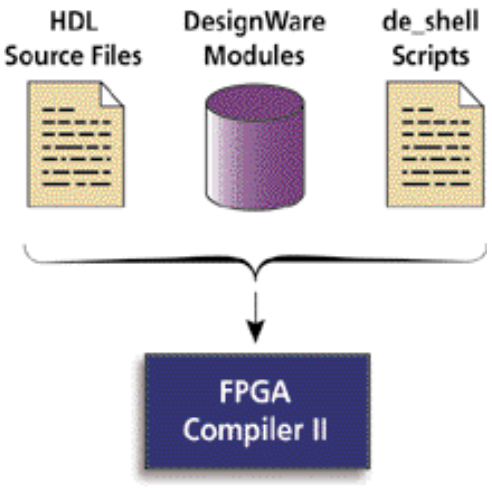


Figure 1

scripts with your FPGA synthesis tool. FPGA Compiler II is unique in that along with its intuitively simple GUI (Figure 3) you can also run a translated Design Compiler shell script in an interactive command shell (Figure 4) or as a transparent batch job.

Second is the capability to output the Synopsys .db database format that is supported by Design Compiler, PrimeTime (Static Timing Analysis), and many other Synopsys tools. ASIC designers have found that using static timing analysis allows for much faster design verification for million-gate designs.

Third, you can now make use of the DesignWare Foundation library. DesignWare Foundation is an off-the-shelf IP library provided by Synopsys that has been available to Design Compiler customers for many years, and now is available to you with FPGA Compiler II to efficiently and predictably implement your Xilinx solutions.

Summary

With the introduction of million-gate FPGAs, the need for you to have ASIC-compatible shell scripting, DesignWare IP libraries, and a proven synthesis product that leverages your existing investment and knowledge, is quickly becoming a reality. In today's market, if you are going to complete your design on schedule and on specification, then now is the time to implement these flows and methodologies. The Virtex Family and FPGA Compiler II v3.2 will make your success a reality.

FPGA Complier II Output Flow

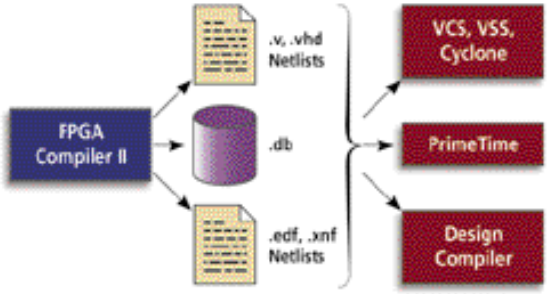


Figure 2

For more information on Synopsys FPGA Compiler II go to: www.synopsys.com/fpga

FPGA Compiler II GUI

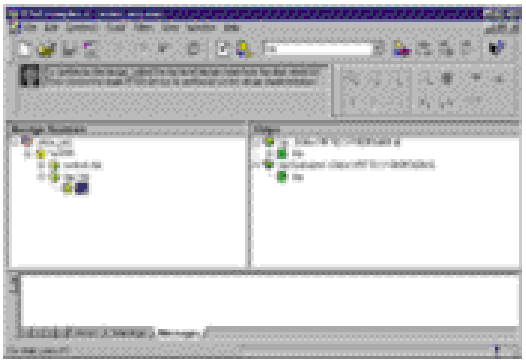


Figure 3

FPGA Compiler II Interractive Command Shell

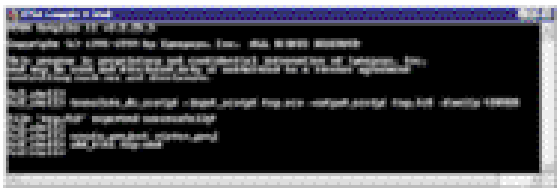


Figure 4

New EDIF Netlist Controls

Synplicity provides you with the ability to control the formatting of EDIF netlists for use with Xilinx FPGAs.

by Margaret E. Albrecht, Technical Marketing Manager, Synplicity
maggie@synplicity.com

There are several commonly used conventions for delimiting busses in netlists such as EDIF. Mixing conventions can cause problems with your design flow, so Synplicity has added additional user control to allow easier conformance to your chosen convention.

As of Synplify 5.1 and later releases, you can specify the bus format and the case of nets in the Xilinx EDIF netlist. The two new attributes**, syn_edif_bit_format** and **syn_edif_scalar_format**, allow you to specify the format either directly in HDL code, or by applying the attribute preferences graphically in Synplify's constraints editor (SCOPE).

How to Apply the syn_edif_bit_format Attribute

Applied to the top level, and all black box modules and components in a design, **syn_edif_bit_format** can be specified for vectors and takes the following values:

- "%C<%i>" – Use <> to delimit bits of a bus.
- "%C[%i]" – Use [ ] to delimit bits of a bus.
- "%C(%i)" – Use ( ) to delimit bits of a bus.
- "%C_%i" – Use _ to delimit bits of a bus [implies **syn_noarray_ports**=1].
- "%C%i" – Append bit to the name of a bus [implies **syn_noarray_ports** = 1].

The value of C can be:

- u – To upshift the case of the base name of a bus.
- d – To downshift the case of the base name of a bus.

  n – To preserve the case of the base name of a bus.

How to Apply the **syn_edif_bit_format** Attribute in a Constraint File (.sdc) or Graphically in SCOPE:

```
# Attributes
define_global_attribute syn_edif_bit_format {%n[%i]}
```

Note that %u and %d can be substituted for %n for upshifting and downshifting the base name.

Figure 1, shows a screen shot of Synplify (upper left), HDL Analyst – RTL View (upper right), SCOPE graphical constraints editor (left center), Verilog source file (lower left), and the Synplify EDIF netlist (lower right). This screen shot shows how you can set an EDIF netlist attribute in SCOPE (left center). The side by side netlists (bottom left and right) illustrate the change in the bit format from your HDL source (bottom Left: [3:0] Y) to the EDIF netlist output (bottom right: Y<3:0>).

(Continued)

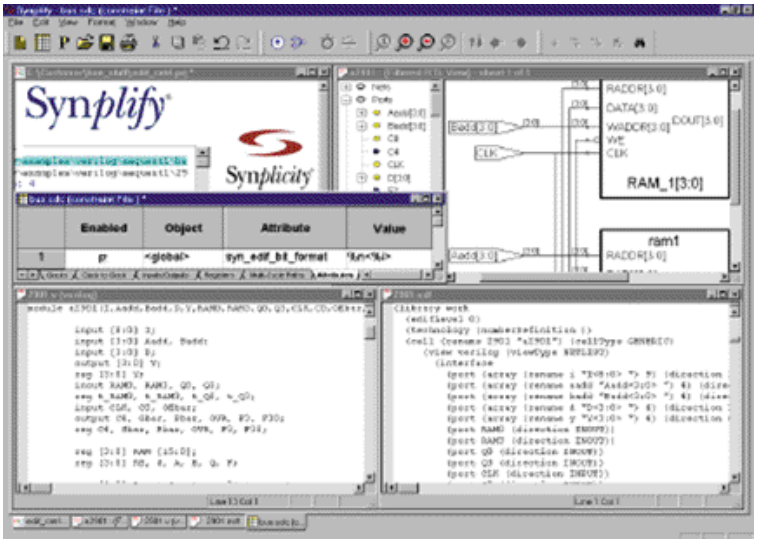


Figure 1

Applying the syn_edif_scalar Attribute

Applied to top level and black box modules and components in a design, the **syn_edif_scalar_format** can be applied on scalars and takes the following values:

- "%u" – Upshifts the case of the base name of a scalar.
- "%d" – Downshifts the case of the base name of a scalar.
- "%n" – Preserves the case of the base name of a scalar.

Verilog

Apply on a module basis:

```
module test(...) /* synthesis syn_edif_bit_format =
"%d[%i]" syn_edif_scalar_format = "%u"
*/
```

VHDL

Apply on an architecture/component basis:

```
Component
                component test
                 port( ...
        );
                end component;
                attribute syn_edif_bit_format string;
                attribute syn_edif_scalar_format :
                string;
                attribute syn_edif_bit_format of test
                component is "%u(%i)";
                attribute syn_edif_scalar_format of
                 test component is "%u";
Architecture
                architecture rtl of test is
                attribute syn_edif_bit_format string;
                attribute syn_edif_scalar_format :
                string;
                attribute syn_edif_bit_format of rtl
                architecture is "%u(%i)";
                attribute syn_edif_scalar_format of
test component is "%u";
```

Applying the **syn_edif_scalar_form**at Attribute in a Constraint File (.sdc) or Grapically in SCOPE:

```
# Attributes
define_global_attribute syn_edif_scalar_forma t {%n}
```

Note that %u and %d can be substituted for %n for upshifting and downshifting of the base name.

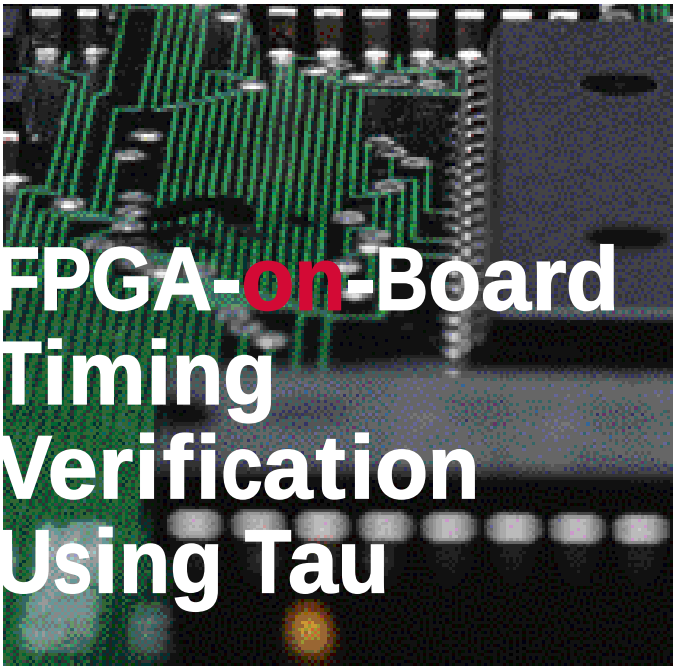
Attribute Examples given a bus definition, My_Bus[1:0]:

- %n<%i> should rename My_Bus to: My_Bus<1:0>.
- %u(%i) should rename My_Bus to: MY_BUS(1:0).
- %d[%i] should rename My_Bus to: my_bus[1:0].
- %n_%i should rename My_Bus to: My_Bus_1 My_Bus_0.
- %u%i should rename My_Bus to: MY_BUS1 MY_BUS0.

Summary

This article shows how to control the formatting of Synplicity-Xilinx EDIF netlist in your designs. By providing an interface to customize the Synplify netlist, you can integrate high-performance IP or legacy code with a variety of net nomenclatures into your existing Synplify design flow.

For more information about Synplify, SCOPE, and HDL Analyst, please see the Synplicity website at http://www.synplicity.com



FPGA-on-Board Timing Verification Using Tau

by Ajay J. Daga, Tau Product Line Manager, Mentor Graphics Corporation, ajay_daga@mentor.com

Timing verification is increasingly becoming an important step in board design flows. With clock speeds frequently exceeding 50 MHz, timing margins have become progressively tighter. When you factor in the impact of clock skew, interconnect delay, and the delay through custom logic (FPGAs, for example), it is no longer possible to rely on Excel spreadsheets, back-of-the-envelope calculations, or stress tests of prototypes, to ensure that board designs will work reliably, at intended speeds, under worst-case conditions. Indeed, timing verification is now the fastest and most thorough approach to finding problems early in your design cycle.

Roger Yang, senior verification engineer at Cisco Systems says, "Timing verification is a very important step in our board design flow and we use Tau™ for this purpose. We find that an automated solution to board timing analysis is essential to fully, and in a timely manner, consider the impact on circuit timing of issues such as clock skew, interconnect delay, and delay through FPGAs and ASICs."

Tau, from Mentor Graphics, is a board timing verification solution. Tau takes as input a netlist, models for components in the netlist, and interconnect delay information (Figure 1). Tau uses this information to exhaustively, and in one pass, verify all timing constraints (set-up/hold times, or pulse-width times) on a design. Tau reports timing margins for each constraint and the skew between clocks, taking only a few seconds to completely analyze most board designs.

To illustrate the board timing verification process, consider the example circuit in Figure 2. It contains an IDT79RV5000 processor communicating with memory through a custom memory controller implemented using a Xilinx FPGA. The processor/memory controller interface is synchronous and is driven by a 100-MHz clock that is generated by the processor. The memory interface is asynchronous to the processor interface and is driven by a 40-MHz clock on the memory controller. The netlist for the design is provided to Tau either from a schematic capture tool such as Design Architect (from Mentor Graphics) or through EDIF.

Board Timing Verification with Tau

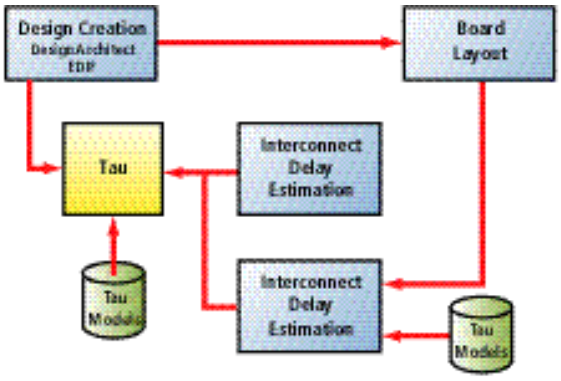


Figure 1

Models

Tau models provide black-box timing information for a component (pin-to-pin delays and constraints). Additionally, you can attach timing diagrams to a Tau model that describe read/write cycles. Tau uses the information in a timing diagram to automatically eliminate the reporting of false timing violations (for example, checking set-up/hold times every clock

(Continued)

Example Board Design

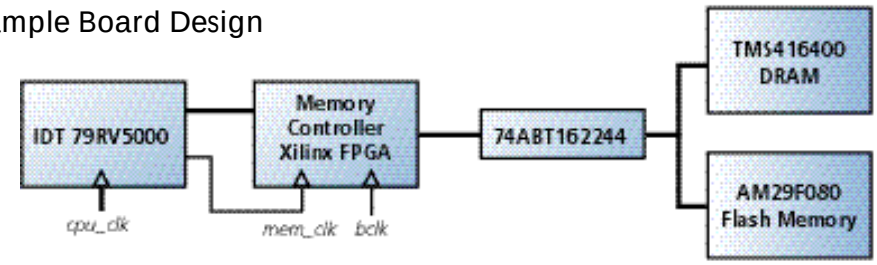


Figure 2

cycle when data is really latched at every other clock edge).

Models for the off-the-shelf components (IDT79RV5000, 74ABT162244, AM29F080, and TMS416400) on a design are obtained either from the Tau Model Library provided by Mentor Graphics (at no charge), or through Mentor's Tau project modeling service. This service provides models, with a two-week turnaround time for all off-the-shelf components on a design, for a fee of $3,000.

To model custom components, such as the Xilinx FPGA in Figure 2, Tau interfaces with model formats. Xilinx provides the ability to generate black-box timing information for FPGAs in the Stamp format. Stamp files are directly imported into Tau and design engineers do not have to manually re-enter this information. This is particularly important given the many revisions an FPGA goes through and the need to have immediate access to up-to-date FPGA timing information for accurate FPGA-on-board timing verification.

Timing Diagrams are a Source of Component models

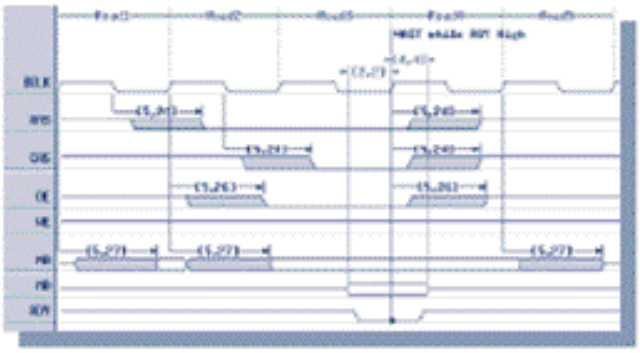


Figure 3

Finally, to prevent Tau from reporting false timing violations, you can import timing diagrams drawn using Chronology's TimingDesigner (such as that shown in Figure 3) that describes the read access on the memory controller. Timing diagrams capture the sequence of clock cycles that constitute a memory access. They specify when outputs change, when inputs are latched, and the values on control signals. Timing diagrams are useful for accurate analysis of memory subsystems because they capture phase and multi-cycle relationships between signals on an asynchronous interface.

Interconnect Delay

Interconnect delay is imported into Tau using one of two approaches. The first approach is to estimate interconnect delay based on placement information and a user-provided value for interconnect delay as a function of length. This approach is useful early in the design process when a board has been placed but not routed. It provides a "ball-park" estimate for interconnect delay and helps focus your attention on the potential problem areas of your design.

For a more thorough and accurate characterization of interconnect delay, you can use transmission-line simulation tools such as IS_Analyzer™ (from Mentor Graphics) or XTK™ (from Viewlogic). These tools use IBIS models to accurately characterize interconnect delay under different "corner" conditions while taking into account the impact of crosstalk, net topology, reflected-wave switching, and so on. Transmission line analysis is traditionally performed when a design is routed and all the physical information is in place for accurate delay characterization.

(Continued)

Report of Timing Margins

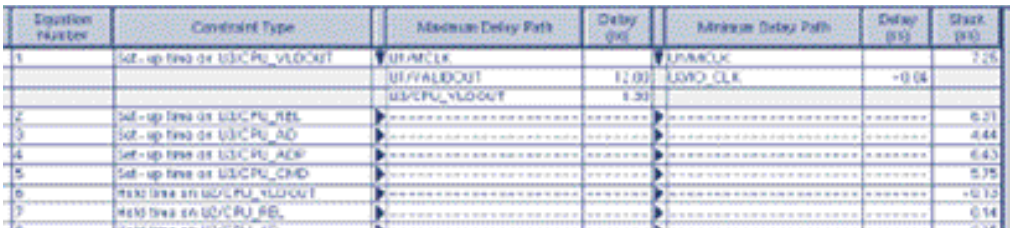


Figure 4

Timing Verification

In addition to a netlist, component models, and interconnect delay information, Tau requires a description of the clocks on a design. You specify the frequency, duty cycle, and peak-to-peak jitter at the source of a clock tree and this information is propagated to the clock nets driven by the source. So, for the design in Figure 2, a clock frequency of 100 MHz, with a 50% duty cycle and 250ps of peak-to-peak jitter, is defined at CPU_CLK. Tau automatically propagates this information to MEM_CLK. Asynchronous clocks are easily specified by allowing multiple clock sources on a design. So, in Figure 2, BCLK (used to drive the memory interface) is also a source clock.

Worst-case timing verification of the entire design is performed in one analysis run. Tau computes the skew between clocks and does so taking into account inversion in the clock tree and output skew (skew between output pins) on a component. Correlation in delay between clock and data paths is also taken into account. For example, in Figure 2, the IDT79RV5000 generates both clock and data signals for the memory controller. If the delays within the IDT79RV5000 track, as they probably will, then this is captured in the Tau model for the component in either percentage terms or picoseconds of skew.

Tau reports timing margins, or slack times, for each constraint verified, as shown in Figure 4. Negative slack indicates that a constraint is violated. Each constraint can also be viewed in block diagram form, as shown in Figure 5, to see the propagation paths for the constraint and component/interconnect delays for each segment in these paths. The block diagram view is particularly useful in helping isolate and fix timing problems.

Through the Stamp interface to Xilinx, the timing information for each revision of the memory controller FPGA is imported into Tau and the impact on timing margins is quickly and easily established. This is important because the memory controller timing is key to the overall board timing;

Block diagram view of the paths for a timing constraint

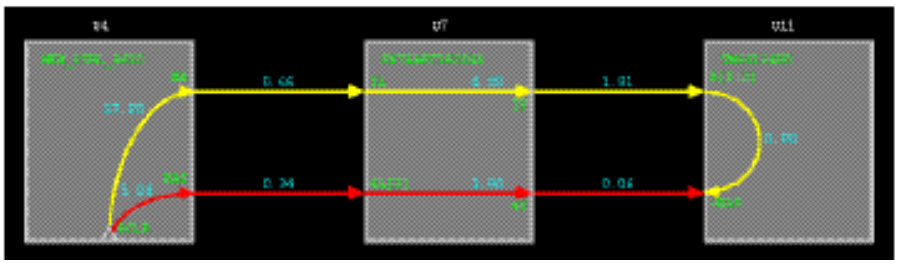


Figure 5

the controller communicates with the IDT processor at 100MHz and also drives an asynchronous interface to DRAM and Flash Memory.

Summary

Increasing clock speeds, the growing impact of interconnect delay, and the frequent use of FPGAs in the timing-critical portion of a design, necessitate board timing verification. There really is no other alternative to ensure the mass production of reliable boards that work at intended speeds under worst-case conditions. The Stamp interface from Xilinx, in conjunction with Tau, provides you with a viable and easy-to-use solution.

Xilinx Design Series Text Books

Used by over 150,000 students per year in colleges and universities worldwide, Xilinx programmable logic devices are more than twice as popular as those of our nearest competitor.

by Patrick Kane, University Program Manager, Xilinx, xup@xilinx.com

Xilinx, along with Prentice Hall, the premier publisher of engineering books and software for the academic market, recently announced the publication of a new engineering text that includes a copy of the Xilinx Student Edition software. The text, Modeling Synthesis and Rapid Prototyping with the Verilog HDL, by Michael D. Ciletti, is the first textbook to be offered as a package with the popular Xilinx programmable logic tools. Xilinx and Prentice Hall have been partners for two years, providing the Xilinx Student Edition software for college-level instruction.

Ciletti's text on Verilog synthesis and rapid prototyping is designed for the undergraduate computer science, computer engineering, and electrical engineering courses in digital design and hardware description languages. It provides in-depth instruction combined with the sophisticated tools students will need to put those theories into practice in the laboratory environment. This offering builds on the success of the Xilinx Student Edition software and furthers the Xilinx commitment to our university program.

"Prentice Hall is pleased to offer a logic design simulator that complements the content of Michael Ciletti's textbook," commented Tom Robbins, publisher for Prentice Hall. "The Xilinx Student Edition allows students to put into immediate practice the tools and techniques they learn from the text right on their own computer."

Other engineering texts in the Xilinx Design Series published by Prentice Hall will also include the Xilinx Student Edition software:

- The Xilinx Practical Designer, by Dave Van Den Bout.
- VHDL: Starter's Guide From Simulation to Synthesis, by Sudhakar Yalamanchili.
- Digital Design: Principles and Practices (Third Edition), by John F. Wakerly.
- Other titles, to be announced.

Instructors who wish to make the Xilinx Student Edition software part of their curriculum have the option of packaging the Student Edition lab book with any Prentice Hall digital design book for additional savings and course completeness. For more information, see: www.xilinx.com/programs/xds1.htm.

The Modeling Synthesis and Rapid Prototyping with the Verilog HDL text packaged with the Xilinx Student Edition software is now available for a list price of under $100. A special website dedicated to the lab exercises will always have the latest up-to-date files.

About Prentice Hall

Prentice Hall Engineering is a part of the Higher Education publishing group within Pearson Education, the global education publishing business of Pearson plc, the international media group. Pearson Education is the world's leading education business, enlightening and enriching the lives of students of all ages, everywhere, with books, multiple media, and learning programs in all major academic disciplines. Its leading imprints include Prentice Hall, Addison Wesley, Scott Foresman, Computer Curriculum Corporation, Allyn & Bacon, Longman, and QUE. For more information on Prentice Hall, access their website at www.prenhall.com. Σ

Web-integrated Software Manuals

Quickly and easily find the technical information you want from the Xilinx software manuals.

by the Technical Publications Staff, Xilinx, swmonline@xilinx.com

Whether you want to find a book associated with a design step, search for a term in a book or collection, or just read part of a manual, finding the information you want in the Xilinx Software Manuals is now quicker and easier with the new document viewer and your Web browser. The following sections show you how to use the new document viewer, which works inside your Java-enabled Web browser.

Reading the Manuals

To read the manuals, use any of the following methods:

- Go to http://support.xilinx.com. Click the Software Manuals button in the left column.
- Open the manuals directly from the 2.1i Software Documentation CD.
- Download the manuals from the 2.1i Software Documentation CD to a local hard drive and open the manuals from your hard drive.

The manuals appear in your default Web browser. The browsers that work best are Netscape Navigator™ version 4.x and higher, and Microsoft Internet Explorer™ 4.x and higher. If you do not have a Web browser, you can install Netscape Navigator from either the 2.1i Implementation Tools or the 2.1i Documentation CD-ROMs.

Document Viewer Table of contents

Figure 1

Navigating the Table of Contents

The document viewer's left frame provides a table of contents for the book collection, either Alliance Series, Foundation Series, or both. The table of contents also includes direct links to the Xilinx Support website and the Xilinx Home website.

If you click the "+" to the left of a collection, each book title in the collection displays. Directly click title text to do the following.

- Click Alliance Series or Foundation Series for more information on each product line.
- Click a book title to display the book's preface.
- Click a chapter or heading title to display that section of the book.

Finding a Book Associated with a Design Step

The right frame of the document viewer contains a graphic of the design flow.

(Continued)

Design Flow

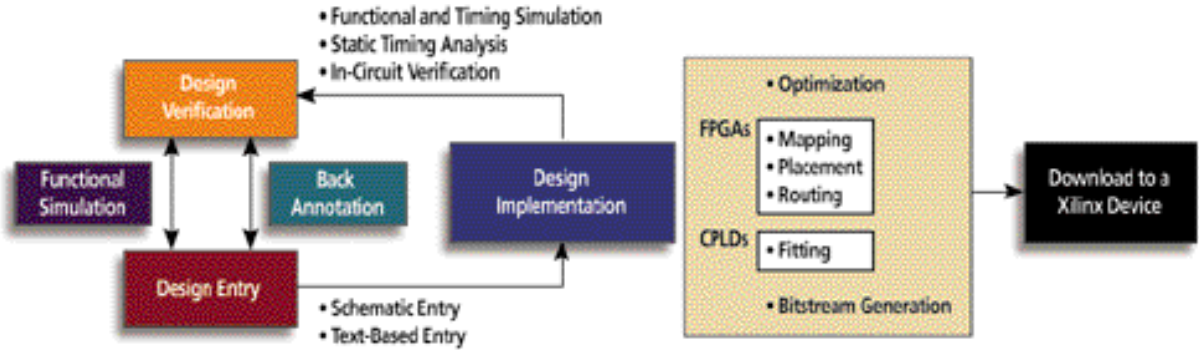


Figure 3

Each design step in the graphic is linked to a table that lists and describes books related to that step. Click the book title in the table to display the first page of the book.

Searching for Terms in Books or Collections

Click the Search button in the left frame of the document viewer to search on a term. Use the Options tab to change the way your search results display.

Options Tab

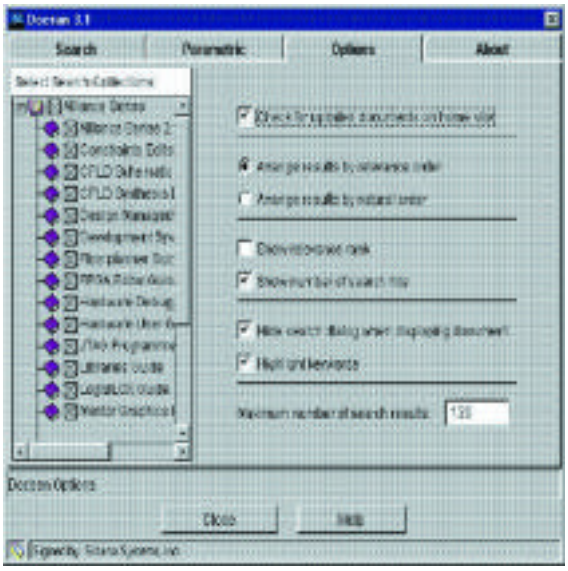


Figure 2

To constrain your searches, do the following.

- Specify the collection or book to search using the Options tab.
- Search set categories within the Libraries Guide using the Parametric tab.

Printing PDF Copies:

You can print PDF copies of the manuals from the Web or from the 2.1i Software Documentation CD. The viewer's online Help includes instructions for accessing and printing the PDF files.

Summary

Its now easier than ever to find the technical information you need, and use Xilinx software more effectively.

	Usable Gates (K)	Logic Gates	System Gates (see note 1)	Logic Cells (see note 2)	Total Cells	Total Flip-Flops	Max. I/O Package	Max. RAM Bits (K Bits)	Config. Memory (K Bits)	XC17V00 Serial PROM (PDIP, SOIC, VOIC, PC20, VO20)
XC4000EX Family - 5 Volt .5Aaaaaaaaaum Triple Layer Metal Process										
XC4028EX	18-50	28K	50K	2,432	1,024	2,560	256	32.8K	668	XC1701
XC4036EX	22-65	36K	65K	3,078	1,296	3,168	288	41.5K	833	XC1701
XC4000XLA Family - 3.3 Volt Advanced .35µm Five Layer Metal Process										
XC4013XLA	10-30	13K	30K	1,368	576	1,536	192	18.4K	393	XC17512L
XC4020XLA	13-40	20K	40K	1,862	784	2,016	224	25.1K	522	XC17512L
XC4028XLA	18-50	28K	50K	2,432	1,024	2,560	256	32.8K	668	XC1701L
XC4036XLA	22-65	36K	65K	3,078	1,296	3,168	288	41.5K	833	XC1701L
XC4044XLA	27-80	44K	80K	3,800	1,600	3,840	320	51.2K	1,015	XC1701L
XC4052XLA	33-100	52K	100K	4,598	1,936	4,576	352	62.0K	1,215	XC1702L
XC4062XLA	40-130	62K	130K	5,472	2,304	5,376	384	73.8K	1,434	XC1702L
XC4085XLA	55-180	58K	180K	7,448	3,136	7,168	448	100.4K	1,925	XC1702L
XC4000XV Family - 2.5 Volt .25µm Five Layer Metal Process										
XC40110XV	75-200	110K	220K	9,728	4,096	8,704	432	131.1K	2,488	XC1704L
XC40150XV	100-300	150K	300K	12,312	5,184	11,520	448	165.9K	3,373	XC1704L
XC40200XV	130-400	200K	400K	16,758	7,056	15,456	432	225.8K	4,551	XC1704L & XC17512L
XC40250XV	160-500	250K	500K	20,102	8,464	18,400	448	270.9K	5,434	XC1704L & XC1702L
Spartan Family - 5 Volt and 3.3 Volt (XL) Advanced .5µm & .35µm (XL) Process										
XCS05XL	2-5	3K	5K	238	100	360	77	3.2K	54	XC17S05 (XL)
XCS10XL	3-10	5K	10K	466	196	616	112	6.3K	95	XC17S10 (XL)
XCS20XL	7-20	10K	20K	950	400	1,120	160	12.8K	179	XC17S20 (XL)
XCS30XL	10-30	13K	30K	1,368	576	1,536	192	18.4K	249	XC17S30 (XL)
XCS40XL	13-40	20K	40K	1,862	784	2,016	224	25.1K	330	XC17S40 (XL)
Virtex - 2.5 Volt										
XCV50	20-50	20K	50K	1,728	384	1,536	180	57K	559	XC1701L
XCV100	30-100	30K	100K	2,700	600	2,400	196	79K	781	XC1701L
XCV150	45-150	45K	150K	3,888	864	3,456	260	104K	1,040	XC1701L
XCV200	60-200	60K	200K	5,292	1,176	4,704	284	133K	1,335	XC1702L
XCV300	80-300	80K	300K	6,912	1,536	6,144	324	164K	1,751	XC1702L
XCV400	130-400	130K	400K	10,800	2,400	9,600	404	236K	2,546	XC1704L
XCV600	185-600	185K	600K	15,552	3,456	13,824	500	319K	3,608	XC1704L
XCV800	250-800	250K	800K	21,168	4,704	18,816	510	416K	4,715	XC1704L & XC1701L
XCV1000	330-1M	330K	1M	27,648	6,144	24,576	510	524K	6,127	XC1704L & XC1702L

QPRO™ High-Reliability QML Products	Usable Gates (K)	Logic Gates	System Gates (see note 1)	Logic Cells (see note 2)	Total Cells	Total Flip-Flops	Max. I/O Package	Max. RAM Bits (K Bits)	Config. Memory (K Bits)	XC17V00 Serial PROM (PDIP, SOIC, VOIC, PC20, VO20)
XC4000EX Family - 5 Volt .5Aaaaaaaaaum Triple Layer Metal Process										
XC4005E	3-9	5	9	466	196	616	112	6.3	9.5	XC17256D
XQ4010E	7-20	10	20	950	400	1,120	160	12.8	178	XC17256D
XQ4013E	10-30	13	30	1,368	576	1,536	192	18.4	248	XC17256D
XQ4025E	15-45	25	45	2,432	1,024	2,560	256	32.7	422	XC17526Dx2
XQ4028EX	18-50	28	50	2,432	1,024	2,560	256	32.7	422	XC17256Dx2
XQ4000XL Family - 3.3 Volt Advanced .35Aaaaaaaaaum Five Layer Metal Process										
XQ4013XL	10-30	13	30	1,368	576	1,536	192	18.4	393	XQ1701L
XQ4036XL	22-65	36	65	3,078	1,296	3,168	288	41.5	833	XQ1701L
XQ4062XL	40-130	62	130	5,472	2,304	5,376	384	73.8	1,434	XQ1701Lx2
XQ4085XL	55-180	85	180	7,448	3,136	7,168	448	100.4	1,925	XQ1701Lx2
XQ4000XL Family - 3.3 Volt .25Aaaaaaaaaum Five Layer Metal Process										
XQV100	30-100	30	100	2,700	600	2,400	180	79	781	XQ1701L
XQV300	80-300	80	300	6,912	1,536	6,144	316	164	1,751	XQ1701Lx2 or XQ1704L
XQV600	185-600	185	600	15,552	3,456	13,824	500	319	3,608	
XQV1000	330-1M	330	1M	27,648	6,144	24,576	512	524	6,128	

	Macrocells	Max. I/O	Pin-to-Pin Delay (ns)
XC9500 Family - 5 Volt CPLDs			
XC9536	36	34	5
XC9572	72	72	7.5
XC95108	108	108	7.5
XC95144	144	133	7.5
XC95216	216	166	10
XC95288	288	192	10
XC9500XL Family - 3.3 Volt CPLDs			
XC9536XL	36	36	4
XC9572XL	72	72	5
XC95144XL	144	117	5
XC95288XL	288	192	6
XC9500XV Family - 2.5 Volt CPLDs			
XC9536XL	36	36	3.5
XC9572XL	72	72	4
XC95144XL	144	117	4
XC95288XL	288	192	5

Speed Grade Options	Slowest	Fastest
XC4000EX	-4	-2
XC4000XLA	-09	-07
XC4000XV	-09	-07
Virtex	-4	-6
SpartanXL	-4	-5
XQ4005X	-4	-4
XQ4000XL	-3	-3

Serial PROM Package Options	Density	PD8	SO8	VO8	PC20	SO20	VO44	3 Volt	5 Volt
XC1736E	36K	Y	Y	Y	Y				Y
XC1765E (EL)	65K	Y	Y	Y	Y			Y (EL)	Y
XC17128E (EL)	128K	Y		Y	Y			Y (EL)	Y
XC17256E (EL)	256K	Y		Y	Y			Y (EL)	Y
XC17512L	512K	Y			Y	Y		Y	
XC1701	1M	Y			Y	Y			Y
XC1701L	1M	Y			Y	Y		Y	
XC1702L	2M						Y	Y	
XC1704L	4M						Y	Y	
XC1736	36K	Y		Y					Y
XC1765	65K	Y		Y				Y	
XC1765L	65K	Y		Y					Y
XC1765	65K	Y		Y				Y	
XC1765L	65K	Y		Y				Y	
XC1736	176K	Y		Y					Y
XC1736L	176K	Y		Y				Y	
XC1736	248K	Y		Y					Y
XC1736L	248K	Y		Y				Y	
XC1736	328K	Y				Y			Y
XC1736L	328K	Y				Y		Y	

Speed Grade Options	Density	DD8	CC44	SO20
XC17256D	256K	Y	-	-
XQ1701L	1M	-	Y	Y
XQ1704L	4M	-	Y	Y

Notes:

1. System Gates includes 20-30% of CLBs used as RAM

2. A Logic Cell is defined as a 4-input LUT and a Register

Ordering Information (example):

XC4005E - 6 PQ 160 C

Device Type
Speed Grade
Temp. Range
Number of Pins
Package Type

FPGA / CPLD Package Options and User I/O

	XC4028EX	XC4036EX	XC4013XLA	XC4020XLA	XC4028XLA	XC4036XLA	XC4044XLA	XC4052XLA	XC4062XLA	XC4085XLA
IOBs	256	288	192	224	256	288	320	352	384	448
PLCC Packages										
44										
84										
CSP Packages										
48										
144										
280										
PQFP / HQFP Packages										
100										
160			129	129	129	129	129			
208	160		160	160	160	160	160			
240	193	193	192	193	193	193	193	193	193	
304	256	256			256	256	256	256	256	
VQFP Packages										
44										
64										
100										
TQFP / HTFP Packages										
100										
144			113	113						
175			145	145						
CBFP Packages										
228	192									
BGA Packages										
256			192	205	205					
352	256	288			256	288	288			
432		288				288	320	352	352	352
560								352	384	448
PGA Packages										
299	256			256						
411		288			288	320	352			
475									384	
559										448
FinePitch BGA										
256										
456										
676								384		
680										

	XC40110XV	XC40150XV	XC40200XV	XC40250XV	XCS05/XL	XCS10/XL	XCS20/XL	XCS30/XL	XCS40/XL
IOBs	448	448	448	448	80	112	160	192	224
PLCC Packages									
44									
84					61	61			
CSP Packages									
48									
144						112	113		
280								192	224
PQFP / HQFP Packages									
100									
160									
208							160	169	169
240	193	193						192	192
304									
VQFP Packages									
44									
64									
100					77	77	77	77	
TQFP / HTFP Packages									
100									
144						112	113	113	
175									
CBFP Packages									
228									
BGA Packages									
256								192	205
352	289	289							
432	352	352	352	352					
560	432	432	432	432					
PGA Packages									
299									
411									
475									
559		448		448					
FinePitch BGA									
256									
456									
676									
680									

	XC9536	XC9572	XC95108	XC95144	XC95216	XC95288	XC9536XL/XV	XC9572XL/XV	XC95144XL/XV	XC95288XL/XV
IOBs	34	72	108	133	166	192	36	72	117	192
PLCC Packages										
44	34	34					34	34		
84		69	69							
CSP Packages										
48	34						36	38		
144									117	
280										
PQFP / HQFP Packages										
100		72	81	81						
160			108	133	133					
208					166	168				168
240										
304										
VQFP Packages										
44	34									
64							36	52		
100										
TQFP / HTFP Packages										
100										
144		72	81	81				72	81	
175									117	117
CBFP Packages										
228										
BGA Packages										
256										
352					166	192				192
432										
560										
PGA Packages										
299										
411										
475										
559										
FinePitch BGA										
256										
456										
676										
680										

	XCV50	XCV100	XCV150	XCV200	XCV300	XCV400	XCV600	XCV800	XCV1000
IOBs	192	240	288	336	384	480	576	672	768
PLCC Packages									
44									
84									
CSP Packages									
48									
144	98	98							
280									
PQFP / HQFP Packages									
100									
160									
208									
240	166	166	166	166	166	166	166	166	
304									
VQFP Packages									
44									
64									
100									
TQFP / HTFP Packages									
100									
144	98	98							
175									
CBFP Packages									
228									
BGA Packages									
256	180	180	180	180					
352			260	260	260				
432					316	316	316	316	
560						404	404	404	404
PGA Packages									
299									
411									
475									
559									
FinePitch BGA									
256	176	176	176	176					
456			260	284	312				
676							404	444	444
680							512	512	512

Available Now

Xilinx Alliance Series and Foundation Series Features

Features Included	Alliance Series		Foundation Series			
			Design Environment			
	Schematic & Synthesis		Schematic & ABLE		Schematic & Synthesis	
	ALI-BAS	ALI-STD	FND-BAS	FND-STD	FND-BSX	FND-EXP
EDA Libraries and Interfaces for Cadence, Mentor, Synopsys, and ViewLogic	✓	✓				
Turns Engine (Workstation Only)	✓	✓				
Synthesis Constraint Editor and Timing Analyzer						✓
Esperan MasterClass Lite VHDL Tutorial					✓	✓
HDL Synthesis Tools (ABEL, VHDL, and Verilog)					✓	✓
HDL Design Tools: HDL Wizard, Context Sensitive Editor, Graphical State Editor, and Language Assistant			✓	✓	✓	✓
Schematic Editor			✓	✓	✓	✓
Simulator (Functional and Timing)			✓	✓	✓	✓
HDL Synthesis Libraries (UniSim and Simprim)	✓	✓	✓	✓	✓	✓
Implementation Tools: Design Manager, Flow Engine, Timing Analyzer, Hardware Debugger, LogiBLOX, JTAGProgrammer, PROM File Formatter Graphical Constraints Editor, Graphical Floorplanner	✓	✓	✓	✓	✓	✓
EDIF, VHDL (VITAL), and Verilog Back Annotation	✓	✓	✓	✓	✓	✓
LogiBLOX™ Module Generator	✓	✓	✓	✓	✓	✓
Xilinx CORE Generator	✓	✓	✓	✓	✓	✓
CPLD Devices (XC9500 and XC9500XL)	✓	✓	✓	✓	✓	✓
FPGA (Low Density/High Volume Devices): XC4000EXL (Up to XC4010EXL) Spartan and SpartanXL (All) XC3000A, XC3000L, XC3100A, XC3100L XCS200 (Up to XCS210)	✓		✓		✓	
FPGA (Unlimited Device Support): Virtex XC4000EX (All) Spartan and SpartanXL (All) XC3x00A/L (All) XCS200 (All)		✓		✓		✓
Xchecker Cable (Workstation Only)	✓	✓				
JTAG Cable (PC Only)	✓	✓	✓	✓	✓	✓

In today's rapidly changing technology climate, choosing a strategic partner means more than simply buying a part, because it's a safe bet that what solves your problem today probably won't solve your problem tomorrow.

Our solution? It's simple. Offer each and every device necessary to address the needs of our customers, for today's design challenges *and* those of the next millennium.

Starting with our XC9500 CPLD and Spartan™ device families for truly cost effective ASIC replacement, to Virtex™, the industry's first one million gate FPGA, Xilinx offers a programmable logic solution ideally suited to *your* specific design.

And when it comes to development system software, you'll have even more to smile about! Both Xilinx Foundation™ And Alliance™ Series Software solutions are built on one common platform, so they not only look the same and work the same, they both deliver the industry's fastest compile times!

If this all sounds pretty easy, that's because it is. We made it easy by spending over 12% of our annual revenue on R&D. That's equal to the combined spending of our two nearest competitors. All to ensure that our customers have access to the broadest selection of programmable logic devices available anywhere. That's how we became the leader.

Take a look at our website and see the whole drama unfold. You'll learn why the leading company's don't just choose a part they choose a partner. They choose Xilinx.

www.xilinx.com

Q399 - 34162

XILINX®

2100 Logic Drive
San Jose, CA 95124-3450

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