XCell

PRODUCT INFORMATION

QPRO - For the Aerospace and Defense Market

Multilinx – New High-speed Download Cable

APPLICATIONS

Spartan FPGAs Used in Digital Modems

Two Simple Solutions for Tricky Problems
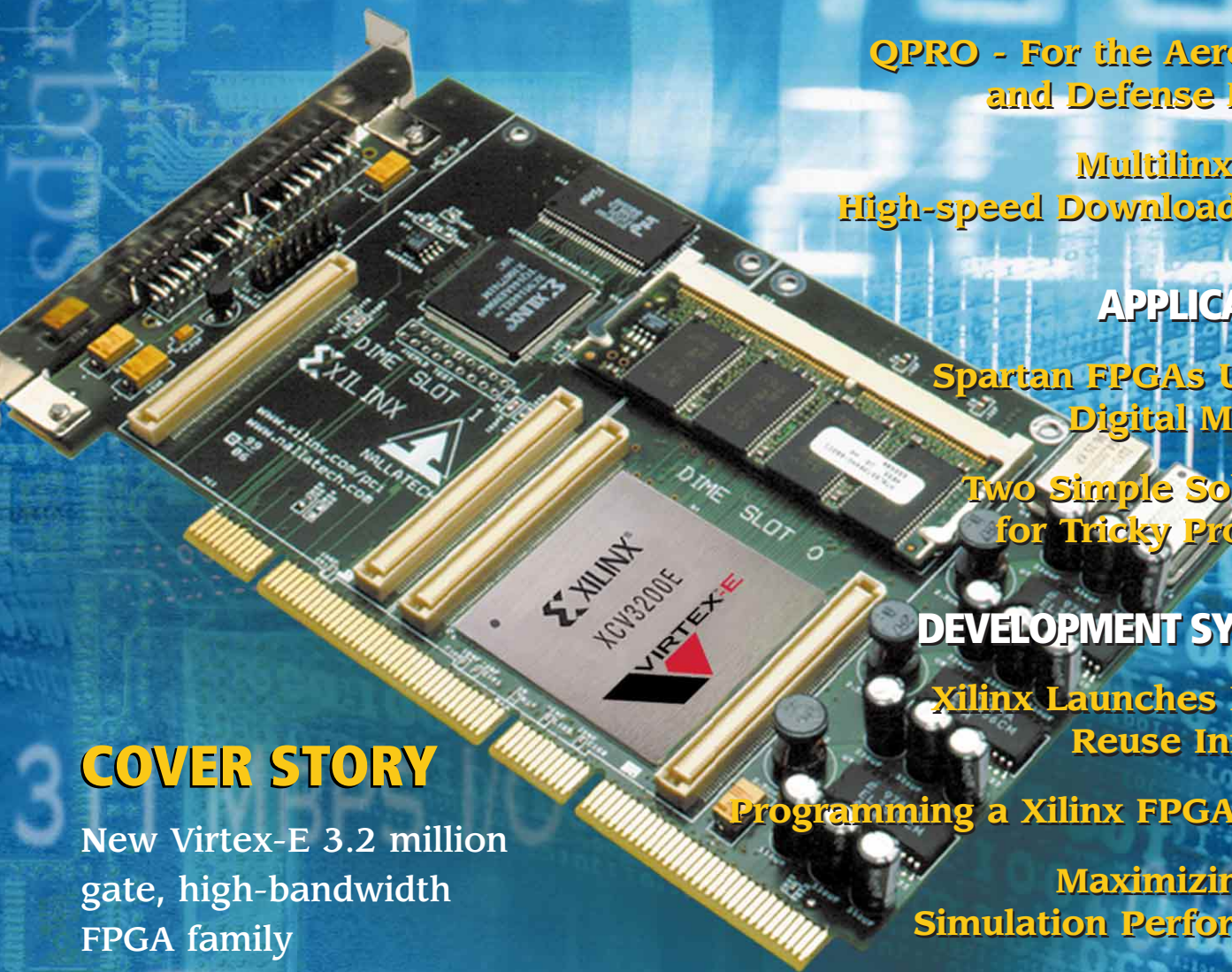
DEVELOPMENT SYSTEMS

Xilinx Launches Design Reuse Initiative

Programming a Xilinx FPGA in "C"

Maximizing HDL Simulation Performance

COVER STORY

New Virtex-E 3.2 million gate, high-bandwidth FPGA family

XILINX®

FROM THE EDITOR

What's Ahead in the Year 2000...

What do you think will be the big technology news in the year 2000?

It has been a great year for the programmable logic industry. During 1999, we have seen dramatic advancements in every area: device densities have skyrocketed to two million gates, with unprecedented performance; development tools have become very fast, efficient, and easy to use; intellectual property has become plentiful and widespread; device prices have fallen; unique new applications have arisen.

So, what can you expect in the year 2000?

- **Ever Denser FPGAs** - We already have 2-million gate Virtex™-E FPGAs available, and soon we will be shipping our 3.2-million gate, very high performance devices. This much power, in a programmable logic device, gives you unprecedented design freedom and helps you get very complex designs to market as quickly as possible.

- **Very Low Cost FPGAs** - You will continue to get more gates for less money. Our Spartan™ family of low cost FPGAs keeps getting better and less expensive. Why develop costly, time consuming, risky ASICs, when you can get the flexibility, ease of use, low risk, and low cost of FPGAs? High volume, consumer applications will take full advantage of this breakthrough.

- **Design Reuse** - To reach the marketplace sooner, with robust designs that work right the first time, companies will create libraries of in-house and third-party intellectual property. Xilinx already provides the tools you need to create and manage intellectual property, and we will continue to lead the industry in this critical area through our Design Reuse Initiative.

- **Internet Reconfigurable Logic** - You will begin to see many more companies designing hardware that can be reconfigured remotely. The era of single use hardware is over, because it is so easy to build "universal" hardware systems that can change and adapt to new requirements. Using the evolving Xilinx Online™ technology, you can create field upgradable systems that last longer and provide more benefits to your customers for less cost.

- **Advanced Development Tools** - It takes a full team to design with multi-million gate devices, and our new software tools will make it easy to collaborate on designs with engineers anywhere in the world. Look to Xilinx to lead the effort in creating the high-level, fully-integrated, development tools you will need for both large and small designs.

In the year 2000 you will not only see bigger, better, faster, cheaper devices, you will also see the decline of single use, fixed logic hardware and the beginning of a new paradigm of field upgradable logic. ✗

ARTICLES

COVER STORY

5

Now you can achieve up to 622 MHz differential I/O performance with the new Virtex-E 3.2 million gate, high-bandwidth FPGA family.

PRODUCT INFORMATION

20
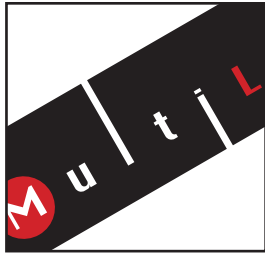
Xilinx has just introduced a new bitstream download cable that reduces download times by 10X over the older XChecker™ solution.

DEVELOPMENT SYSTEMS

44
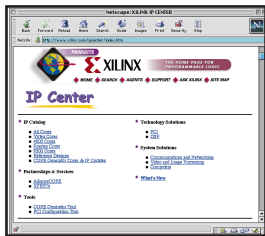
New tools allow you to capture and share your own intellectual property over the Internet.

APPLICATIONS

49
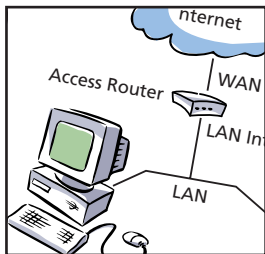
An overview of digital modem technologies and how Xilinx high-volume programmable devices can be used to implement complex system-level glue logic in digital modem designs.

COLUMN

58

The market has shifted dramatically in the last ninety years. Today the buzzword is "mass customization"— unique products at a price comparable to that of mass-produced standard goods.

www.xilinx.com

Xcell

For A FREE Subscription To The Xcell Journal

E-mail your request to: literature@xilinx.com

Please be sure to include:

1. Your Full Name and Mailing Address

2. Your Title

3. The Name of Your Company

4. Your E-mail Address

5. Is This New Subscription or a subscription Renewal?

Programmable Logic
in the Next Millenium

I expect that within the first ten years of the next millennium we will see programmable logic devices inside every piece of electronic equipment, because hardware will become just as programmable as software.

by Wim Roelandts, President and CEO, Xilinx

Wall Street market analysts expect the programmable logic industry to grow another 25-35% in the year 2000, following a 25% increase in 1999. This increase is being driven, in part, by a worldwide economic recovery; we are seeing dramatic increases in Southeast Asia, and Japan is also beginning to improve. However, the recovering world economy is just one of many reasons for the dramatic growth in programmable logic.

The primary reason for this phenomenal growth is the tremendous technological advances that have taken place over the last two years. In 1996, our largest FPGA contained just 50,000 gates, and no BlockRAM™. Today, the Xilinx Virtex-E family offers up to 3.2 million system gates, with 832 kbits of BlockRAM. This means that FPGAs are large enough and fast enough to interest designers who once relied on gate array or standard cell technology. Using FPGAs, designers can now get the performance and density of gate arrays, and something gate arrays never offered—fast time-to-market and affordable pricing.

Xilinx will soon introduce a new Spartan family which will offer 100,000 system gates for less than $10.00. This downward pricing trend is expected to continue, and 500,000 system gates will be available for under $10.00 by the year 2003. These dramatic improvements will further expand the influence of programmable logic in new, fast growing, high volume equipment such as cable and DSL modems, cell phones, hand-held electronic games, and DVD players.

The total cost of ownership for an FPGA has also stayed flat, as compared to substantial increases in the cost of ownership for standard cell and gate-array technology. FPGAs eliminate the need for expensive NRE (Non Recurring Engineering), development, system integration, and field deployment costs. However, NRE charges for standard cell and gate array ASICs have risen sharply as feature sizes have dropped below 0.5 micron. In addition, for gate array and standard cell chip development, the required board debugging, system integration, and field trial costs have also gone up sharply.

Another significant attraction of programmable logic is field upgradability. Because FPGAs can be reprogrammed an infinite number of times, equipment can easily be upgraded over a network after it has been installed at a customer's premises. This allows forward-thinking manufacturers to quickly get their new products to market and then add new features, comply with evolving standards, or fix bugs, remotely, without physically replacing any hardware.

The year 2000 is clearly a turning point for the programmable logic industry, as we quickly expand into new high-volume markets with advanced high-density, low-cost technologies. Σ

THE NEW
Virtex-E
3.2-Million Gate, *High-bandwidth* FPGA Family

With up to 622 MHz differential I/O performance, Virtex-E FPGAs are the high-bandwidth solution for your next generation high performance systems.

by Bruce Jorgens, Virtex Product Marketing Manager, Xilinx, jorgens@xilinx.com

The new Virtex-E FPGA family is built on the highly successful Virtex architecture. Leveraging our latest 0.18-micron, six-layer metal technology, Virtex-E devices now give you up to 3.2 million system gates, 804 I/Os, and up to 622 MHz differential I/O performance. Combined with system-level features for clock management, multiple I/O standards, and embedded True Dual-Port™ memory, the Virtex-E family is designed to support the high bandwidth requirements of next generation high performance DSP and communication systems.

Many designs will require multiple high bandwidth data ports with I/O bandwidth distributed across the required ports as shown in Table 1. Virtex-E devices range from 44 Gbps to 200 Gbps I/O bandwidth. For applications such as OC-192, Virtex-E devices can easily support the multiple 10-Gbps data ports that are required.

Bandwidth-enabling Technology

The Virtex-E devices contain advanced system-level technology that is specifically designed to support high bandwidth applications. Figure 1 shows a block diagram of the Virtex-E bandwidth-enabling technology including digital Delay Lock Loops (DLLs), True Dual-Port embedded memory, and SelectI/O+ technology.
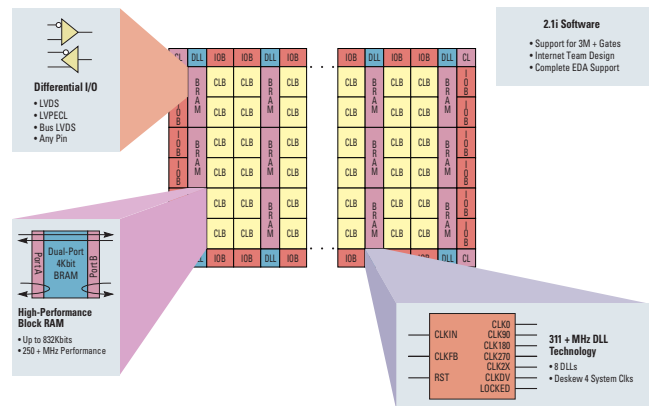
Figure 1 - Virtex-E Architecture Overview

Device	Logic Cells	Dual-Port Block Memory (Kbits)	Maximum User I/O	I/O Bandwidth (Gbits/sec)
XCV50E	1728	56	176	44
XCV100E	2700	80	176	44
XCV200E	5292	112	284	71
XCV300E	6912	128	316	79
XCV400E	10800	160	404	100
XCV600E	15552	288	512	127
XCV1000E	27648	384	660	164
XCV1600E	34992	576	724	180
XCV2000E	43200	640	804	200
XCV2600E	57132	736	804	200
XCV3200E	73008	832	804	200

1 I/O bandwidth = (80% max. user I/O) x (311 Mbps)

Table 1 - Virtex-E Features

Eight High Performance DLLs

The Virtex-E advanced DLL technology provides the system clock management that is necessary for high bandwidth chip-to-chip or backplane applications. The DLL circuitry allows very precise synchronization of external and internal clocks. Xilinx was the first to deliver this technology by offering four 200 MHz DLLs in each Virtex device. The Virtex-E family takes this technology to the next level offering eight DLLs in each Virtex-E device.

As a fully digital implementation, the Virtex and Virtex-E DLLs do not have the typical problems encountered with analog phase locked loops (PLLs); PLLs are extremely sensitive to noise on the power and ground pins. Many systems cannot provide the isolation and decoupling required for the proper operation of a PLL, while the Virtex-E DLLs have no special requirements. Also, because the DLL is not sensitive to process variations, it is offered as a standard feature in every device, and every speed grade.

Virtex-E DLLs provide precise clock edges through phase shifting, frequency multiplication, and frequency division. Table 2 shows the basic characteristics of the Virtex-E DLLs.

Parameter	Value
Maximum Output Frequency	320 MHz*
Maximum Output Jitter	100 ps
Output Frequency Duty Cycle	50%+/- 100ps

* Based on Virtex-E –7speed grade product

Table 2 - Bandwidth-critical Specifications of the Virtex-E DLL

Maximizing Memory Bandwidth with Virtex-E DLLs

A key technique for increasing the bandwidth of a particular data port is to have signals change on both edges of the clock, commonly referred to as the "Double Data Rate" technique. At high frequencies, signal integrity limits the clock performance, which limits the bandwidth of the data. Bandwidth for the port is immediately doubled if the architecture can change data at each edge of a system clock. Memory suppliers have already started to support this type of high performance technique to increase the memory bandwidth of their devices.

For this technique to work, it is critical that the clock duty cycle be as close to 50 percent as possible. Because Virtex-E DLLs can generate clocks with a duty cycle guaranteed to be within 100 picoseconds of 50 percent, you can achieve the maximum memory bandwidth when interfacing to the fastest DDR memories. Figure 2 demonstrates how Virtex-E DLLs, coupled with the SelectI/O+™ technology, help achieve maximum bandwidth in a 266MHz DDR application.

No Bottleneck Memory Bandwidth

Figure 2 - Virtex-E Interfacing to a 266 MHz DDR SDRAM Memory

SelectI/O+ Technology

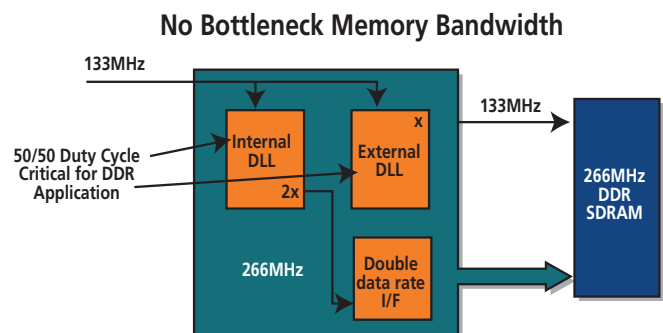To meet high bandwidth requirements, electrical signals must switch at over 100 MHz; standard TTL and CMOS signal technology cannot keep pace. With the original Virtex family, Xilinx pioneered the SelectI/O™ technology which supports 200 MHz I/O and allows a single FPGA to interface with any other device without external converters. Virtex-E SelectI/O+ technology expands the performance and flexibility by supporting high performance I/O standards such as HSTL and SSTL at over 300 Megabits per second (Mbps) per pin. In addition, Virtex-E devices are the first programmable logic devices to directly interface with differential I/O standards including LVDS, Bus LVDS (BLVDS), and LVPECL.

Standard	Typical Application
LVTTL	3.3 V General Purpose
LVCMOS2	2.5 V General Purpose
LVCMOS18	1.8 V General Purpose
PCI33_3	33 MHz 3.3 V PCI Backplane
PCI66_3	66 MHz 3.3 V PCI Backplane
SSTL2 (I,II), SSTL3(I,II), CTT	SDRAM, DDR SRAM
HSTL(I,III,IV)	SRAM, DDR SDRAM, Backplanes
GTL, GTL+, AGP	Backplanes, Microprocessor Interfacing
LVDS	Point to Point and Multi-drop Backplanes High Noise Immunity
BLVDS	Bus LVDS Backplanes, High Noise Immunity, Bus Architecture Backplanes
LVPECL	High Performance Clocking, Backplanes, Differential 100MHz+ Clocking, Optical Transceiver, High Speed Networking, and Mixed-Signal Interfacing
5 V TTL* (4mA Iol)	Legacy 5V TTL Interfacing

*Requires 100 Ohms external resistor

Table 3 - I/O standards Supported by Virtex-E Family

High-performance Differential Signaling: LVPECL, LVDS, and Bus LVDS

Increasingly, leading systems designers are turning to differential signaling as the mechanism of choice for backplane applications. Differential signaling enables high bandwidth while reducing power, increasing noise immunity, and decreasing EMI emissions. Virtex-E devices meet this emerging challenge with unprecedented capabilities and support for high-performance differential signaling.

The Virtex-E family supports a hierarchy of differential solutions including up to 36 pairs of LVDS and/or LVPECL operating at 622 MHz, and up to 344 differential pairs operating up to 311 MHz. This gives you a maximum differential I/O bandwidth of over 100 Gbps, which can be distributed over the three differential signal standards as needed. For the first time in a programmable device, you can leverage the high bandwidth and noise immunity characteristics of these standards.

LVPECL I/O is widely used in 100+ MHz inter-chip signaling in high-speed data communications and instrumentation systems. Fiber-optic network interfaces and gigahertz analog-to-digital converters, for example, rely on LVPECL I/O to achieve gigabit per second bandwidth. All Virtex-E differential I/Os support LVPECL input, output, and I/O signaling.

In addition to high-speed interfacing, LVPECL is the industry standard for transmission of precise, on-board clocks at frequencies in excess of 100 MHz. While traditional LVTTL clock sources are typically limited to 100 MHz and below (due to the fundamental signal integrity limits), LVPECL clock sources provide operation up to 400 MHz. As FPGA system clock frequencies exceed 100 MHz, LVPECL clocking becomes an essential requirement. Virtex-E devices support high-performance LVPECL clock inputs for global and local clocking, with frequencies in excess of 300 MHz. In addition, through the use of its multiple DLLs coupled with SelectI/O+ technology, the Virtex-E devices
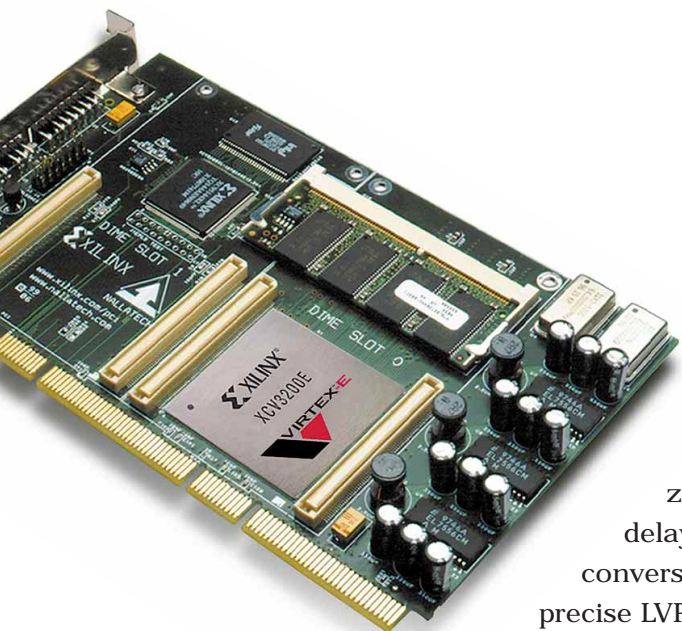
enable zero-delay conversion of precise LVPECL clocks into any required I/O standard. Thus, Virtex-E FPGAs are an integral part of high-performance board-level clock distribution strategies.

In addition to LVPECL, the Virtex-E family has the industry's first programmable devices to support Low-Voltage Differential Signaling (LVDS). LVDS exists in two commonly available variants: LVDS and Bus LVDS. LVDS is optimized for high-speed point-to-point links, while Bus LVDS is optimized for backplane applications employing multi-drop (one transmitter, multiple receiver), and multi-point (multiple transmitters and receivers) configurations.

Virtex-E devices provide unparalleled support for both LVDS and Bus LVDS, with support on all devices and speed grades. You can use up to 688 pins (344 pairs) of LVDS and Bus LVDS capabilities on the largest device, providing differential I/O bandwidth in excess of 100 Gbps.

True Dual-Port Embedded Block Memory

Whether used as FIFOs, caches, or ATM packet buffers, the system requirements for more memory grows much faster than logic requirements. Xilinx pioneered using embedded distributed memory (with its SelectRAM technology) in its XC4000 FPGAs, allowing the configurable logic block to support logic or memory. With the Virtex series, this technology was enhanced to include up to 128 Kbits of True Dual-Port block RAM. The Virtex-E family again provides a quantum leap in internal memory bandwidth by supporting up to 832 Kbits of True Dual-Port RAM (208 blocks of 4Kbits memory) capable of 250 MHz performance.

To emulate most of the functionality of a dual-port memory, two-port memory architectures require twice the number of memory bits and multiplexing of address and data as shown in Figure 3. This results in two-port memory at roughly half the bandwidth and half the efficiency of the Virtex-E True Dual-Port memory in any given configuration.
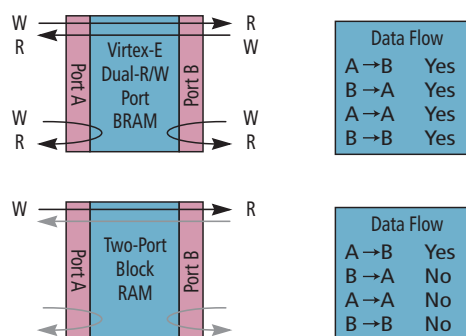
Figure 3 - Typical Memory Configurations

Managing Bandwidth Using True Dual-Port Memory

Each Virtex-E True Dual-Port memory block supports 4 Kbits of memory, and each port can be configured separately to support a variety of depth/width combinations. Embedded memory can buffer high bandwidth data as well as reduce the internal processing speed by transparently converting from one data width to another.
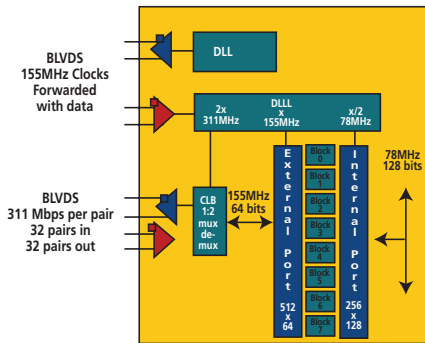
Figure 4 - An OC-192 Application Example

Figure 4 demonstrates an OC-192 application example. A data port with OC-192 bandwidth comes in on 32 BLVDS pairs running at 311 Mbps per pair. Eight blocks of embedded RAM are used to buffer the data internally. The port taking data from the I/O register to the memory is configured as 512 rows deep by 64 bits wide. The port leading to the internal processing of the data is configured as 256 by 128. Internal processing of the 128-bit data need only run at 78 MHz to keep up with the OC-192 bandwidth. An outgoing port would be configured similarly.

Packaging

To support a 10-Gbps bandwidth, the device package must be capable of packing many high performance I/Os in limited board space. The package must also be able to dissipate several watts of power. Virtex-E devices continue the tradition of offering the industry's most reliable and flexible packaging, including

- PQFP - Plastic quad flat pack.

- BGA - 1.27 mm ball grid array.

- CSP - Leading-edge 0.8 mm chip scale package.

- FG - 1.0 mm fine pitch BGA.

These packages are supported across the family.

For the fine pitch 1.0 mm BGA offering, the Virtex-E family introduces three new FG packages:

- FG900 - 31 mm X 31 mm.

- FG1156 - 35 mm X 35 mm.

- FG860 - 42.5 mm X 42.5 mm (thermally enhanced).

The Virtex-E family can now support up to 804 I/Os using board area as small as 35 mm by 35 mm. These packages set new standards in I/Os per square inch as well as maximum bandwidth per square inch.

Summary

The new Virtex-E FPGA family helps you meet the bandwidth requirements of the next generation high performance systems by giving you significant performance and flexibility enhancements in the areas of clock management, SelectI/O+ technology, True Dual-Port block memory, and high performance differential signaling. ☒

For more detailed information on the Xilinx Virtex-E series, including data sheets and applications notes, visit the product section of the Xilinx website at www.xilinx.com.

Virtex-E Package
Compatibility Guide

This package compatibility guide describes the Virtex-E pin-outs and establishes guidelines for package compatibility between Virtex and Virtex-E devices.

by Robert Le, Sr. Applications Engineer, Xilinx, robertle@xilinx.com

The 1.8V Virtex-E FPGA family combines 0.18 mm technology with a synthesis-friendly silicon architecture to provide a new level of FPGA performance and density. Virtex-E FPGA availability in packages compatible with Virtex FPGAs allows systems to migrate from using Virtex family devices to Virtex-E family devices. Package pinout and pin functionality differences between Virtex-E and Virtex FPGAs are covered in this article.

Power Supplies

As with the Virtex family, the Virtex-E positive supply is divided into two separate power supplies: VCCO and VCCINT. VCCO powers output pins and LVTTL, LVCMOS, and PCI output and input pins. VCCINT powers internal logic and all input pins except LVTTL, LVCMOS, and PCI inputs.

The Virtex-E VCCINT is 1.8V (while the Virtex VCCINT is 2.5V). This is a result of more advanced processing and 0.18 mm design rules, which also offer reduced die size, reduced power consumption, and increased speed. VCCO is adjustable, up to 3.3V, depending on the I/O standard used.

Voltage regulator modules with programmable output voltages can be used to power the VCCO and VCCINT inputs and accommodate the lower 1.8V VCCINT in the Virtex-E family.

I/O Standards

Virtex-E devices can be used with 20 high-performance interface standards, including the LVDS and LVPECL differential signalling standards. A new LVCMOS I/O standard based on 1.8V VCCO is also supported. All I/O pins are 3V tolerant, and can be 5V tolerant with an appropriate external resistor. PCI 5V is not supported. Table 1 shows a complete listing of the supported I/O standards.

I/O Banking

There are eight I/O banks in the Virtex-E family, as in the Virtex family, and each bank has multiple VCCO pins. All of the VCCO pins in one bank must be connected to the same voltage level, as determined by the I/O standard in use.

In Virtex-E devices the banking rules are different because the input buffers with LVTTL, LVCMOS, and PCI standards are powered by VCCO instead of VCCINT. For these standards, only input and output buffers that have the same VCCO can be mixed together in the same bank.

Low Voltage Differential Signals

The Virtex-E family incorporates differential signalling (LVDS and LVPECL). Two pins are utilized for these signals to be connected to a Virtex-E device. These are known as differential pin pairs. Each differential pin pair has a Positive

I/O Standard	Output VCCO	Input VCCO	Input VREF	Board Termination Voltage (VTT)
LVTTL	3.3	3.3	N/A	N/A
LVCMOS2	2.5	2.5	N/A	N/A
LVCMOS18	1.8	1.8	N/A	N/A
SSTL3 I & II	3.3	N/A	1.50	1.50
SSTL2 I & II	2.5	N/A	1.25	1.25
GTL	N/A	N/A	0.80	1.20
GTL+	N/A	N/A	1.0	1.50
HSTL I	1.5	N/A	0.75	0.75
HSTL III & IV	1.5	N/A	0.90	1.50
CTT	3.3	N/A	1.50	1.50
AGP-2X	3.3	N/A	1.32	N/A
PCI33_3	3.3	3.3	N/A	N/A
PCI66_3	3.3	3.3	N/A	N/A
BLVDS/LVDS	2.5	N/A	N/A	N/A
LVPECL	3.3	N/A	N/A	N/A

Table 1 - Supported I/O Standards

(P) and a Negative (N) pin. These pairs are labeled in the following manner:

I/O_L#[P/N]

where   L= LVDS or LVPECL pin

= Pin Pair Number

P = Positive

N = Negative

I/O pins for differential signals can either be synchronous or asynchronous, input or output. The pin pairs can be used for synchronous input and output signals as well as asynchronous input signals. However, only some of the differential pairs can be used for asynchronous output signals.

Pin Name	Description
IO_L#[P/N] Example: IO_L22N	Represents a general I/O or a synchronous input/output differential signal. When used as a differential signal, N means Negative I/O and P means Positive I/O.
IO_L#[P/N]_Y Example: IO_L22N_Y	Represents a general I/O or a synchronous input/output differential signal, or a part-dependent asynchronous output differential signal.
IO_L#[P/N]_YY Example: O_L22N_YY	Represents a general I/O or a synchronous input/output differential signal, or an asynchronous output differen-tial signal (for all devices within the same package.)
IO_LVDS_DLL_L#[P/N] Example: IO_LVDS_DLL_L16N	Represents a general I/O or a synchronous input/output differential signal, or a differential clock input signal or a DLL input. When used as a differential clock input, this pin is paired with the adjacent GCK pin. The GCK pin is always the positive input in the differential clock input configuration.

Table 2 - Differential Pin Pairs

Differential signals require the pins of a pair to switch almost simultaneously. If the signals driving the pins are from IOB flip-flops, they are synchronous. If the signals driving the pins are from internal logic, they are asynchronous. Table 2 defines the names and function of the different types of differential pin pairs in the Virtex-E family.

Part	Package	Pins	Virtex	Virtex-E
XCV200	FG456	E11, U11	I/O	No Connect
		B11, AA11	No Connect	IO_LVDS_DLL
XCV300	BG432	B26, C7, F1, F30, AE29, AF1, AH8, and AH24	No Connect	V_{CCINT}
XCV400	FG676	D13, Y13	I/O	No Connect
		B13, AF13	No Connect	IO_LVDS_DLL
XCV400/600	PQ240/HQ240	P215, P87	IO_V_{REF}	IO_LVDS_DLL
		P216, P86	I/O	IO_V_{REF}
All	PQ240/HQ240	P232, P207, P176, P146, P116, P85, P55, and P25	I/O	V_{CCO}
		P231	I/O	IO_V_{REF}

Table 3 - Virtex Family Compared to Virtex-E Pin-out Differences

Differential Clock Pins

In addition to the four GCLKs in the Virtex family, the Virtex-E family has four IO__LVDS_DLL pins that can be paired with GCLKs to support up to four differential clocks. A differential clock input pair always includes one GCLK and the adjacent IO__LVDS_DLL pin. The GCLK pin is always the positive input in differential clock input configurations.

When differential clocks are not in use, these IO__LVDS_DLL pins can be used as single-ended I/Os or as DLL input pins.

DLL Input Pins

Four additional DLL input pins (IO_LVDS_DLL) can be used as inputs to the DLLs, for a total of eight usable inputs for DLLs in the VIrtex-E family. This is very useful in clock mirroring applications.

Pinout Differences

The Virtex-E and Virtex families are pin-compatible with some minor exceptions as shown in Table 3. Some of the exceptions are described below:

- XCV200E Device, FG456 Package - The Virtex-E XCV200E has two I/O pins swapped with the Virtex XCV200 to accommodate differential clock pairing.

- XCV300E Device, BG432 Package - The Virtex-E XCV300E has eight pins (B26, C7, F1, F30, AE29, AF1, AH8, and AH24) connected to VCCINT that are no-connect in the Virtex XCV300.

- XCV400E Device, FG676 Package - The Virtex-E XCV400E has two I/O pins swapped with the Virtex XCV400 to accommodate differential clock pairing.

- All Devices, PQ240 and HQ240 Packages - The Virtex devices in PQ240 and HQ240 packages do not have VCCO banking, but Virtex-E devices do. To achieve this, eight Virtex I/O pins (P232, P207, P176, P146, P116, P85, P55, and P25) are now VCCO pins in the Virtex-E family. This change also requires one Virtex IO_V_{REF} pin to be swapped with a standard I/O pin.

Additionally, accommodating differential clock input pairs in Virtex-E caused some IO_V_{REF} differences in the XCV400E and XCV600E devices only. Virtex IO_V_{REF} pins P215 and P87 are Virtex-E IO_V_{REF} pins P216 and P86, respectively. Virtex-E pins P215 and P87 are now IO__LVDS_DLL.

Conclusion

Though the new Virtex-E family has many enhancements, it can easily be interchanged with the Virtex family, with only minor considerations. ⚡

This information is subject to change. For the latest information, see the Xilinx web site at http://www.xilinx.com.

Xilinx Receives
Cisco's Highest
Honor for a Supplier

Xilinx has been named one of the top suppliers for 1999 by the worldwide leader in networking for the Internet.

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

During the recent eighth annual supplier appreciation ceremonies, Cisco CEO John Chambers presented the Xilinx team with the President's Customer Satisfaction Award, Cisco's highest honor for a supplier. Chambers said that programmable logic technology developed by Xilinx has helped Cisco reduce the time necessary to bring innovative new networking products to market.

Cisco CEO, John Chambers, presented this award to Xilinx.

"We are tremendously pleased to receive this award from Cisco and their recognition of the value of Xilinx programmable logic solutions," said senior vice president Dennis Segers, who accepted the award on behalf of Xilinx. "What pleases me most is it recognizes the results of the outstanding teamwork on the part of many, many Xilinx employees, as well as our suppliers, sales and distribution partners who are contributing to Cisco's success."

This was the first time Xilinx was nominated for the President's award.

"This award is given to the supplier who makes significant contributions that directly impact the customer satisfaction level of Cisco customers," said Michael J. Campi, vice president of global supply management for Cisco Systems. "Xilinx has helped Cisco exceed our customers' expectations."

In addition to being nominated for the President's Customer Satisfaction Award, Xilinx was also nominated as best semiconductor supplier. Cisco presented awards to winners in 11 categories in all. Suppliers in each of the categories were selected by a panel of representatives from Cisco's Corporate Supply Management and Global Supply Management organizations. Σ

Stackable Virtex
FPGA Board
For General Use

A new basic board to help you quickly test and implement your Virtex-based design.

by Dr. Stefan Schafroth, Hardware and Software Development Engineer, ErSt Electronic GmbH, stefan.schafroth@erst.ch

The EVALXCV-HQ240 board has been designed to provide all the necessary basic components needed in most Virtex-based designs. All I/Os are routed to header connectors where you connect your special purpose interfaces. By stacking several boards you can easily cope with the complexity of a design which exceeds the scope of a single FPGA.

To overcome the annoying task of supplying the board with several supply and reference voltages, we have also created a power module which does this for you. It can be attached directly to the FPGA board such that the two boards form a single unit.

Key Features

The heart of the board is a Virtex FPGA in an HQ-240 package (XCV800/600/400) . Vital components for a basic system are placed around the FPGA. These include two crystal oscillators, three push buttons, DIP switches, and nine status LEDs. All configuration modes of the FPGA are supported. You can provide configuration data either by serial configuration PROMs (SCPs) sitting in onboard sockets or by connecting a Xilinx MultiLINX, XChecker™, or JTAG cable. A functional diagram detailing the
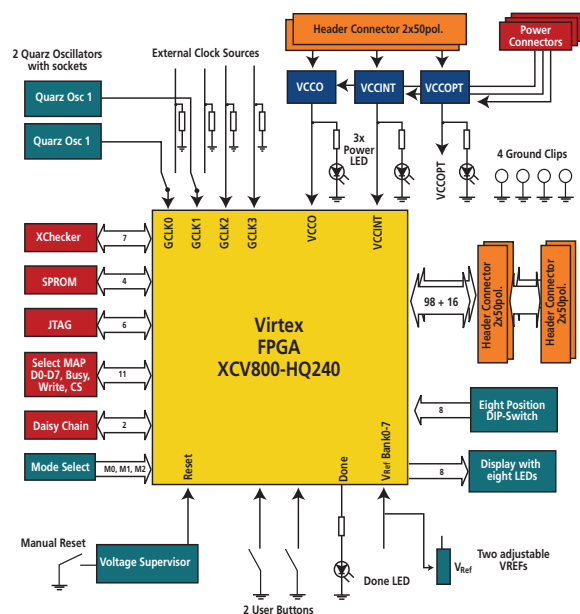
Figure 1- Functional Diagram of the FPGA Board Module

building blocks of the Virtex FPGA board is shown in figure 1.

The board is very well suited to:

- Evaluate the larger members of the Virtex FPGA family.

- Implement custom designs utilizing the full power of Virtex FPGAs.

- Quickly and easily expand the complexity of the system by stacking several boards.

Figure 2 - Top View of the FPGA Board

Figure 4 - View of the Power Module with Attached FPGA Board

Figure 2 shows a top view of the board. On the right side you see the connectors used to supply configuration data, sockets for SCPs, crystal oscillators, and LEDs. The push buttons and DIP switches are placed on the left side. Header connectors for I/O signals run along the top and bottom side of the board.

Power Module

Although you can use the Virtex board standalone, we recommend that you use our PWR3 power module which has been designed to eliminate the burden of manually applying several low level supply voltages. It generates

three regulated output voltages and eight reference voltages from a single unregulated power supply. Its output power is sufficient to satisfy the needs of several FPGA boards stacked together. A functional diagram is shown in figure 3. In figure 4 you see the power module attached to the bottom side of the Virtex board.

Conclusion

The EVALXCV-HQ240 gives you an ideal platform for evaluating, implementing, testing and extending Virtex FPGA based custom designs. You can also easily integrate the board into a larger system. The power module eliminates the need for several supply voltages and forms a compact unit with the FPGA board. Σ

For additional information on EVALXCV-HQ240 and PWR3 see: www.erst.ch, or contact us at info@erst.ch.

Figure 3 - Functional Diagram of the Power Module

15

Q-Pro - for the
Aerospace
& Defense market

With densities up to one million system gates, Virtex QPRO™ devices provide an off-the-shelf system-level ASIC solution, without the non-recurring engineering costs, long prototype cycles, and minimum volumes required of custom ASICs.

by Howard Bogrow, Marketing Manager, Aerospace and Defense Products, Xilinx, Inc., howard@xilinx.com

The Aerospace and Defense market has been undergoing significant changes in recent years, and continues to challenge both system designers and component suppliers. Designers no longer have to specify mil-spec parts for every application, and as a result numerous mil-spec suppliers have exited the market. Thus, it has become increasingly difficult for aerospace and defense customers to obtain many needed devices. Additionally the use of custom or semi-custom ASICs may no longer be a viable option.

Even with all the changes that are taking place, the aerospace and defense market still has some unique product and supply chain requirements:

Design Evolution Using FPGAs

- **Operating temperature range** - This is typically a harsh operating environment, and while many new designs are able to use industrial temperature components, there are still numerous applications that require guaranteed performance over a wider temperature range. QPRO Virtex FPGAs provide this guarantee.

- **Packaging** - Many new applications are able to take advantage of the more cost-effective plastic packages. The QPRO Virtex family offers a wide range of solutions for these applications in both flat pack and ball grid technology. However, ceramic packaging is still available for those applications that require hermeticity, such as shipboard and space. The latest addition to our hermetic package offering is the CG560, a ceramic surface mount column grid array, which is available for the XQV1000 device.

- **COTS and DMS** - There is a very strong trend towards the use of COTS (commercial off-the-shelf) components, and away from custom or specially processed parts. Also, a big concern is DMS (diminished material supply) or product obsolescence. The QPRO Virtex family are COTS products in every sense of the definition. They are commercially available standard products that require no special processing or specifications. And,

16

because the QPRO Virtex family is now available with up to 1,000,000 system gates, the functions of many obsolete components such as discrete logic, semi-custom ASICs, processors, memories, and interface chips can now be incorporated into a single QPRO Virtex FPGA.

- **IP Cores** - Many aerospace and defense systems use a PowerPC architecture, and military versions of the peripherals for this processor have become very difficult to obtain. To address this problem Eureka Technologies of Los Altos, California, has developed an IP core solution that can be incorporated into QPRO Virtex devices. This core includes a PowerPC bus master and slave, PCI host bridge, bus arbiter, SDRAM controller, DMA controller, and UART.

Device	System Gates	Packages
XQV100	100,000	PQ240, BG256, CB228
XQV300	300,000	PQ240, BG352, BG432, CB228
XQV600	600,000	HQ240, BG432, CB228
XQV1000	1,000,000	BG560, CG560

Table 1 - QPRO Virtex Product Offering

Summary

QPRO Virtex devices provide unsurpassed flexibility as a replacement for ASICs for the aerospace and defense market. These products offer system-level integration and performance, and because they are reconfigurable, field upgrades are possible. QPRO Virtex FPGAs bring state-of-the-art technology to this market, while solving the critical supply management issues. 𝚺

SEU Mitigation Techniques for Virtex FPGAs in Space Applications

SRAM-based logic devices such as FPGAs have some susceptibility to Single Event Upsets (SEU) and functional interruption. This paper describes several reliable mitigation techniques for the Virtex series FPGA architecture, which will retain functional integrity while static upsets are detected and corrected.

Additionally, this paper demonstrates how an SEU in an FPGA can be corrected in 3us without disrupting operation of the device, how to build hardened voting circuits, and that a single event has only 1 chance out of 3.25 million of causing a functional interrupt.

Re-configurable computing and adaptive hardware is an emerging technology for space applications. The basis for this technology is the capability for device- and system-level functional changes to be implemented in-system and transmitted remotely. FPGAs provide an array of logic resources, which may be interconnected, and configured for specific functions. All logic definitions and block connections are controlled by static RAM cells. Thus, this technology is sometimes referred to as "SRAM Logic," which allows for on-the-fly reconfiguration of the circuits' functional definition.

The Xilinx XQVR product line is a radiation-tolerant version of the of the commercially popular Virtex series FPGA. Virtex has become a common ASIC replacement in commercial markets due to its density, performance, and wide range of capabilities. The XQVR utilizes an epitaxial process that renders it latch-up immune to an LET of 125MeV-cm2/mg.

This is an excerpt of a paper presented by Carl Carmichael, Xilinx Applications Engineer, at the 1999 MAPLD Conference, held at Johns Hopkins University. For the complete paper, go to http://www.xilinx.com/products/hirel_qml.htm

Gamma-ray
Large Area Space Telescope (GLAST) Tower CPU

Sapphire Computers Inc. recently designed the prototype Tower CPU for the GLAST space telescope program, using XC4000XL FPGAs.

by Dan Rudolf, President, Sapphire Computers, Inc. drudolf@fpgaConfigurator.com, Phone: (937) 767-1062

GLAST will provide astronomers and physicists a direct view of galactic gamma-ray sources in the 10MeV to 300GeV range (such as black holes, neutron stars, blazars, and pulsars). The 16 Tower CPU (TCPU) cards planned for the telescope will provide high-speed parallel processing of the vast amount of data captured by each of 16 sensor arrays.

The GLAST TCPU provides a very high level of computing power for a space processor board design, to handle the bandwidth of the data stream from the sensor arrays. It incorporates a 100MHZ PowerPC 603e with 64-bit data paths to the level 2 (L2) cache and main DRAM memory. A split-level bus architecture ensures fast accesses to the L2 cache while allowing simultaneous DMA access to the main memory array. This level of processing power does not come without an increase in power consumption, so the TCPU design includes power saving sleep modes and allows programmable reduction of the bus clock rates. To provide solid operation in a space environment, the L2 cache, the flash program store, and the DRAM memory are all error correction coded.

Specifications

The prototype TCPU is a VME card running the VxWorks operating system, with one megabyte of L2 cache, 256 megabytes of 64-bit-wide Reed-Solomon corrected DRAM, Flash memory, serial I/O, and Ethernet capability. The prototype board's goal is to give the GLAST software developers a development platform that is a functional equivalent to the final space-qualified board without the expense of using radiation hardened components.
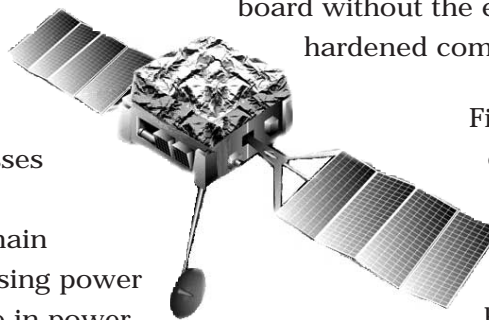
Finding space-qualified device equivalents turned out to be one of the more challenging aspects of this design. There are few rad-hard component vendors and very few high integration devices available in radiation hardened versions. As a result, a number of system functions that are normally available in ASICs had to be custom designed in FPGAs. One of these is the

PowerPC's companion L2 and cache memory controller, the MPC106. While the 603e processor is relatively rad-hard, the MPC106 is not.

All the memory and cache control functions had to be implemented in programmable logic. Additionally the interrupt controller, the system timers, the DRAM error corrector, and the DMA arbiter all had to be custom designed in FPGAs. Xilinx XC4000XL FPGAs were chosen because of their flexibility and ease of debugging.

Corrector) function for the 96-bit-wide DRAM array. The EDAC is capable of correcting up to two four-bit errors in each 96-bit word.

Conclusion

The typical development flow for a space design is to prototype the system using commercial grade parts and RAM-based FPGAs to allow low development cost, easy debugging, and easy upgrading. The next step traditionally was to migrate to space grade parts and away from RAM-based FPGAs. The recent
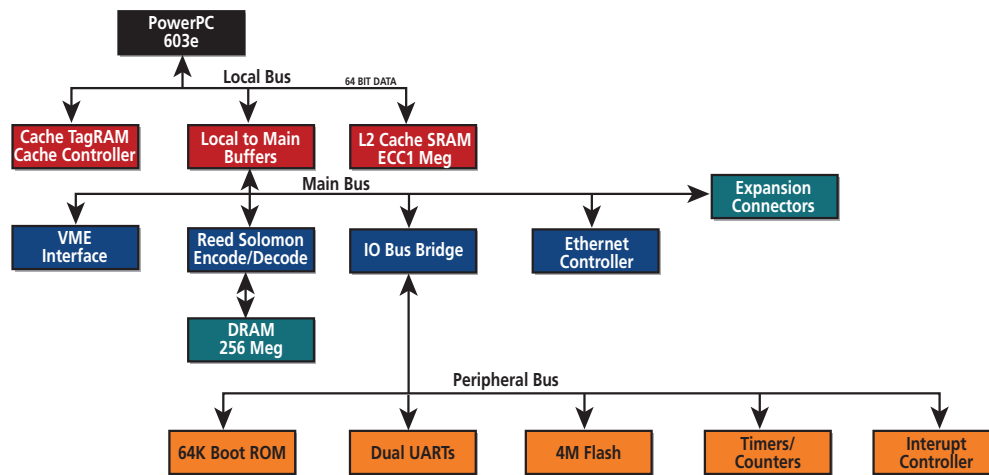
TCPV Block Diagram

The functions were divided into three FPGAs on the board. An XC4036XL is used to implement the L2 cache and memory controller which also does sleep and clock speed control. Another XC4036XL is used to implement the peripheral bus controller incorporating system timers, interrupt controller, and DMA arbiter. The third FPGA is an XC4062XL which does the Reed-Solomon EDAC (Error Detector And

availability of Xilinx QPRO™ devices means that GLAST may be able to stay with RAM-based FPGAs for flight as well. This would allow unprecedented flexibility, allowing during-mission hardware updates and system failure workarounds. Σ

For more information on GLAST please see the Stanford GLAST website: http://glast.stanford.edu.

19

Xilinx introduces-
New High-Speed
Download Cable

MultiLinx reduces download times by 10X over the older XChecker solution.

by Frank Toth, Marketing Manager, Configuration Solutions, Xilinx, Frank.Toth@Xilinx.com

Our new download cable, called MultiLINX™, features a high-speed USB input (or RS232 if you prefer) and a full set of interfaces for the SelectMAP (Virtex), Slave Serial, and IEEE JTAG Download Modes Using a USB cable (with an 8-Mbit per second throughput) the cable can download an XCV1000 in less than a second compared to more than 14 minutes using the older XChecker Cable.

MultiLINX supports all Xilinx FPGAs and CPLDs (Virtex, XC4000XL/XLA/XV, XC5000, Spartan/XL, XC9500). It automatically switches to a wide range of interface voltages including TTL (5 volts), LVTTL (3.3 volts), LVCMOS (3 volts), JESD8-7 (2.5 volts normal and 2.5 volt wide), without the need for a separate external adapter. It's also compatible with 1.8-volt devices (with VCCIO set to 2.5 volts), and it

MultiLinx Download Cable

accepts any voltage from 3.0 to 5.0 volts as a power supply input (from either a target application or external power supply).

Download speeds are increased by 50% even when using the standard 25-pin RS232 cable with the MultiLINX on-board bitstream compression.

MultiLINX offers both flexibility and high-performance to make downloading and programming Xilinx programmable logic easier and faster than ever. ∑

For more information see:
http://www.xilinx.com/support
/programr/cables.htm#Multilinx

A New Family of In-System
Programmable *FLASH*
Serial/Parallel PROMs

Programming, storing, updating, and delivering bit streams for programmable logic has just become a lot easier.

Eric Thacker, Marketing Manager,
Xilinx, eric.thacker@xilinx.com

Xilinx recently introduced a whole new family of JTAG in-system programmable, serial/parallel PROMs. The new XC1800 family uses the IEEE 1149.1 Boundary-Scan interface (commonly know as JTAG) and enables you to easily and cost-effectively configure an FPGA. The XC1800 family can easily interface to any Xilinx FPGA, including the Virtex and Spartan families, using a simple interface requiring only one user data pin. For the ultimate in download speed, these PROMs accommodate the 8-bit wide Select Map and Express modes that enable bit stream rates as fast as 500 Mbits per second.

Device	Configuration Bits
1804	4,194,304
1802	2,097,152
1801	1,048,576
18512	524,288
18256	262,144
18128	131,072

Table 1 - Product listing

Flexible Bit Stream Reconfiguration

The XC1800 series makes it very easy to do power-on downloading and field updating of bit stream files. The power-on loading of the FPGA is enabled automatically, and if the configuration of the FPGA needs to be changed during operation (to remotely download hardware enhancements, debug configurations, or fix bugs, for example), the XC1800 series can be reprogrammed remotely using the JTAG port and then the bit stream is downloaded to the FPGA, without first powering down the FPGA. The FPGA can also be reconfigured directly from the JTAG port, without affecting the bit stream stored in the PROM.

Features:

- Densities ranging from 128 Kbits to 4 Mbits.

- Endurance of 10,000 program/erase cycles.

- Fast programming and configuration speeds.

- Dual configuration modes.

  - Serial Slow/Fast configuration (6 to 15 mHz).

  - Parallel

- PROM controlled initiation of configuration without powering down FPGA.

- IEEE Std 1149.1 Boundary-Scan support.

- JTAG command initiation of standard FPGA configuration.

- Cascadable for storing longer or multiple bitstreams.

- I/O pins accept 5 V, 3.3 V, and 2.5 V.

- 3.3 V or 2.5 V output capability.

- Available in PC20, SO20, PC44, and VQ44 packages. Σ

For more information see:
http://www.xilinx.com/products/
configsolu.htm

Simultaneous 4-channel, 12-bit
A/D Conversion

Using Virtex FPGAs you can now process 600 Mbytes/s of data, a previously impossible task even with today's state-of-the-art DSP Processors.

by Neil Harold, Design Engineer, Nallatech Ltd,
n.harold@nallatech.com

Nallatech Ltd. has created a very high performance analog-to-digital module that provides unrivalled data conversion abilities. The "Ballyriff" module, built using the company's DIME standard (DSP and Image processing Modules for Enhanced FPGAs), can incorporate a wide range of Virtex FPGAs from the XCV400 to XCV3200E parts, giving it maximum flexibility that is further enhanced by a wide variety of input options, as illustrated in Figure 1.
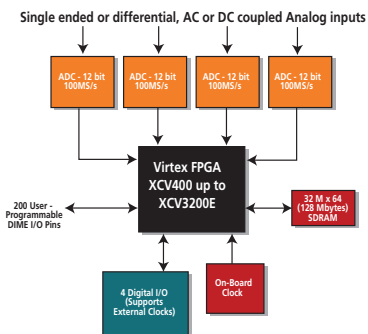
Figure 1 - Ballyriff Block Diagram

This A/D module has been meticulously planned to optimize the space available on the board, and so, Nallatech Ltd. has produced another DIME module with groundbreaking performance characteristics packed onto a few square inches.

The high-speed data conversion ability of the Ballyriff module comes from four of the very latest 12-bit, 100 Msps analog to digital converters. However, the benefits are not limited to data conversion, and include:

- Four digital input lines–for direct digital access to the Virtex FPGA, for external control and trigger functions.

- 128 Mbytes of SDRAM memory–more than any other DIME module.

- A variety of analog input options.

- A variety of clock sources–including external, on-board, and fed from DIME Motherboard

By using Nallatech's "Ballynuey" PCI Carrier Card, this module can be set up in minutes, and FPGA designs can be downloaded dynamically via the PCI bus directly, eliminating the need for PROMs or download cables.
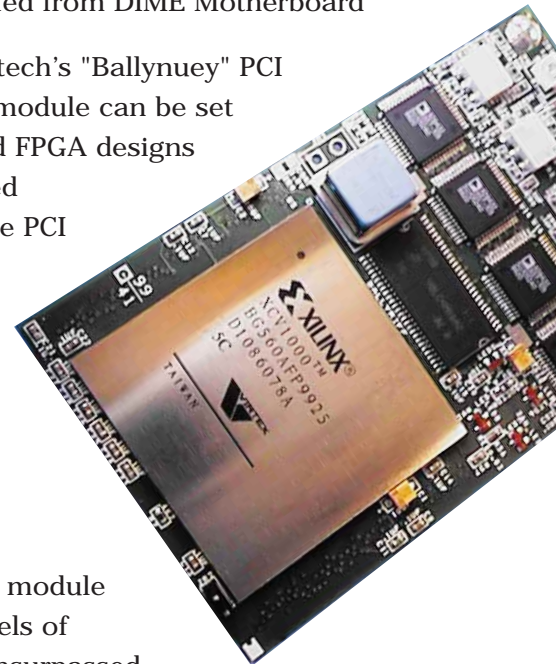
Conclusion

The Ballyriff DIME module gives you high levels of innovation with unsurpassed performance. Other DIME modules provide video capture and display, high-speed communications, and data capture functions. For more information on Ballyriff, or other DIME modules, contact Nallatech at: www.nallatech.com.

Spartan FPGAs Help Create
Movie-quality
Video on the PC

The Dazzle Digital Video Creator is designed around a low cost Spartan FPGA, and provides everything the PC user needs to record, edit, manage, and publish video.

by Kurt Wong, Product Manager, Xilinx,
kwong@xilinx.com

Dazzle Multimedia designs, manufactures, and markets innovative and affordable MPEG digital video products for business and personal computer users. They turned to the Xilinx Spartan FPGA family for their latest groundbreaking product. "We looked very carefully at several different options, and Xilinx simply made the most sense in terms of gates, memory, reliability and price," said Sajid Sohail, Founder, and CEO for Dazzle Multimedia. "The Spartan FPGA was the optimum device for our design and made the process very easy for us. The design required several memory blocks that we implemented as a FIFO. The embedded distributed RAM blocks within the Xilinx device were much more efficient than any other product on the market."

Derived from the industry standard Xilinx XC4000 architecture, the 5-volt Spartan FPGA family combines unparalleled performance with low prices to provide the highest price performance ratio available in the industry. Fabricated on a leading edge silicon process, the Spartan family provides a full feature set including on-chip dual-port RAM, aggressive power management, densities ranging from 5,000 to 40,000 system gates, and small form factor packaging. The Spartan family is the first FPGA family that meets all the key requirements of ASIC designs for high volume production, and delivers unmatched benefits over competing PLDs.

"This product has been so well accepted by consumers, we can't seem to build them fast enough," said Sohail. "It is the first product of its kind at an affordable price." The Digital Video Creator is currently shipping at a high volume rate and retails for $249. Dazzle chose to continue to utilize the Spartan FPGA throughout production. "There is certainly the advantage of quick time to market combined with the flexibility to modify our design. We found a bug in our software within the first six months of shipment - instead of a crisis, this was an easy fix. We simply fixed the problem and sent out our new software to everyone. This was a great win for us and for the FPGA."

Dazzle used Verilog, Exemplar, and the Xilinx Foundation Series software to create their design. "The Foundation Series software provided a ready-to-use solution combined with a powerful interface," said Sohail. This fully integrated design environment allows the user to access a complete set of design entry tools, and automates the creation of schematic and HDL symbols for inclusion within a design, making it easy to design complex circuits in minutes.

Dazzle Multimedia expects to continue its strong presence in the retail channel by introducing an array of new SpartanXL-based products later this year. Σ

23

Infrared
Digital Voice Module

Nextel, France, has created a better way for motorcyclists to communicate, by using digital infrared transceivers controlled by Xilinx Spartan FPGAs.

by Christian Petiot, Technical Director
Nextel France, nextel.france@wanadoo.fr

Talking to someone located just a few inches away from you may seem easy, but on a motorcycle it can be quite difficult and hazardous. Infrared IrDa devices are widely used in computers and mobile telephones, which makes them inexpensive. Digitizing voice is also a simple task, because there are many codecs available on the market. The problem is to design the interface between the half-duplex IrDa transceiver and the codec to make a full-duplex infrared voice transceiver. The logic design is quite complex.

Once the prototype breadboard had been set up, we had to find a way to reduce the size of the board so that it could be integrated into a helmet, we had to greatly reduce the power consumption using low voltage (3.3V) devices, and we obviously had to keep the price as low as possible to address the consumer market. We found all our answers at Xilinx, thanks to the new Spartan FPGAs.

We selected the XCS05XL 5000-gate FPGA. With a sample from our local distributor, Compress, we made a new breadboard, but this time with very few components. For a few French francs we got very powerful yet easy to use software from Xilinx (Foundation Series) to develop the FPGA, and after just two weeks the design was complete. Development time was

The Infrared Module

Once the prototype breadboard had been set up, we had to find a way to reduce the size of the board so that it could be integrated into a helmet, we had to greatly reduce the power consumption using low voltage (3.3V) devices, and we obviously had to keep the price as low as possible to address the consumer market. We found all our answers at Xilinx, thanks to the new Spartan FPGAs.

greatly reduced thanks to everyday support from the Compress application engineers.

The FPGA was used to generate the clocks for the codec, transform the NRZ signal from the codec to RZ signal for the infrared transceiver (and vice versa), and synchronize the incoming data. There were also enough gates left over to add features such as low battery audio alert and communication quality control.

The Finished Product

The module is mounted in a hermetic box with an infrared transparent front. It is connected to the microphone and the speaker inside the helmet by a 4-wire cable. The box is tied to the helmet with an elastic strap. The box may be removed from the helmet in just a few seconds by removing the strap and disconnecting a small connector hidden in the helmet as shown in the pictures.

Conclusion

Xilinx Spartan FPGAs are used in many unique applications. This Digital Infrared Voice Module is the smartest way to communicate on a motorbike; there is no interference at all, it is totally confidential, and it is inexpensive. Σ

For more information e-mail Nextel at:
nextel.france@wanadoo.fr

Programming a
Xilinx FPGA in "C"

Hardware designers are realizing they will need to use higher levels of abstraction to increase their productivity.

by Doug Johnson, Business Development Manager, Frontier Design, doug_johnson@frontierd.com; Marc Defossez, Field Applications Engineer, Xilinx, Inc. - BeNeLux, Belgium, marc.defossez@xilinx.com

Today, many complex communications and digital signal processing (DSP) systems are described using ANSI C or C++ with floating-point mathematics. ANSI C or C++ is the language most commonly used by system engineers because it is powerful and popular, and a variety of environments are available for code development, compilation, and debugging. In addition, the simulation speed using C or C++ can be substantially faster than an equivalent design environment in Verilog or VHDL.

Typically, DSP functions modeled in C are algorithms that perform filtering, modulation, demodulation, compression, coding, and other operations on digital signals. However, most hardware designers are using a design methodology based on VHDL or Verilog HDL. System designers typically deliver a C language specification which has been simulated extensively with internal system-level simulators or commercial products like SPW (Cadence), COSSAP (Synopsys), or HP-ADS (HP EESOF). The hardware designer must then rewrite the specification at least once by hand in VHDL or Verilog and re-simulate the behavior. This manual re-write is difficult, time-consuming, and error prone. In addition, the floating-point C specification must be transformed into a fixed-point implementation in hardware—a daunting task requiring extensive collaboration between the system and hardware engineers.

Frontier Design's C to HDL Solution

The EDA software tools from Frontier Design bridge the gap between the system engineers, working on a design at a high level of abstraction, and the hardware engineer, tasked with the implementation of the design in hardware such as an FPGA or an ASIC. Frontier Design has embodied its 15 years of experience (in transforming DSP algorithms into working silicon) into a methodology that is called "Algorithm to Register Transfer," or A|RT.

Two Frontier products incorporate the A|RT methodology: A|RT Library and A|RT Builder. A|RT Library extends ANSI C and C++ with fixed-point data-types that allow bit-accurate modeling of arithmetic operations as well as modeling the overflow and quantization effects associated with finite precision operations. A|RT Builder supports automatic conversion from ANSI C to VHDL or Verilog. The tools thus enable architectural design in C. As a practical test of the tools, Xilinx - BeNeLux employed the A|RT methodology on three different test cases, one of which is covered here.

Design Methodology Using A|RT

Xilinx - BeNeLux tested a "Coordinate Rotation on a Digital Computer," or CORDIC(1,2) algorithm supplied by Frontier Design using both A|RT Library and A|RT Builder. The algorithm is an iterative computing technique that is capable of evaluating mathematical functions like

multiplication, square roots, and logarithms. The CORDIC algorithm is used extensively because its implementation in hardware utilizes shifts and adds only. A very important aspect of the test was to evaluate the tools to determine their applicability in implementing an algorithm based on C in a Xilinx FPGA, using HDL generated by the A|RT Builder product.

The design flow used for the evaluation is shown in Figure 1. A|RT Builder is the only C-to-HDL

Figure 1 - AIRT Design Flow

The A|RT software is available for both Unix and PC environments; Windows NT for the PC, HP-UX for Hewlett-Packard, and Solaris 2.6 for Sun platforms. The user interface is simple to use, and to ease the code development process, a cross-highlighting capability has been added to A|RT Builder. This feature allows you to highlight the generated HDL code by selecting the associated C code fragment.

Figure 2 shows the cross-highlighting built into the A|RT Builder user interface.
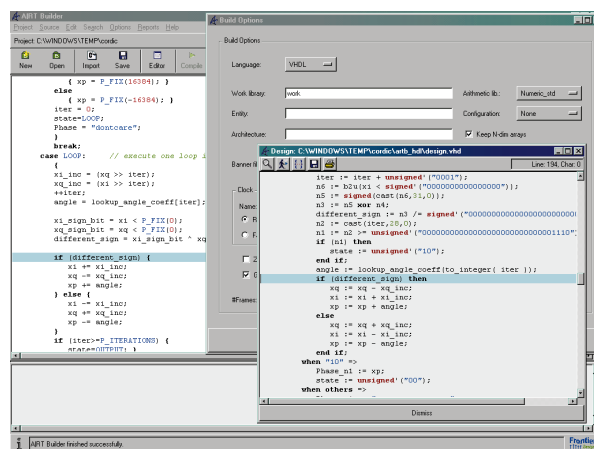
Figure 2 - AIRT Builder User Interface with Cross-highlighting

conversion tool that fully supports fixed-point algorithms. The tool employs a "What You Write Is What You Get" (WYWIWYG) paradigm. What WYWIWYG really means is that if you code in behavioral C, the output VHDL or Verilog will be behavioral. If you code in RTL C, you get RTL VHDL or Verilog. It is a supported subset of C that is utilized by A|RT Builder to generate the HDL code. Because some C constructs have no meaningful realization in hardware, there are ANSI C constructs that A|RT Builder does not support and for some constructs there are restrictions on the way they are supported. A|RT Builder automatically generates both C and HDL test benches so that the HDL and C simulations can be compared.

The Evaluation

First, the CORDIC algorithm was designed and simulated in a C-based environment such as Visual C++ or GNU. Floating-point variables in the CORDIC algorithm were easily specified in fixed-point form by using the A|RT Library data type "Fix<w,p>", where "w" is the fixed word width and "p" is the fixed number of precision bits. For example, a variable with an 8-bit word length and 5 bits of precision is specified as follows: "Fix<8,5>".

Fixed-point values can also be specified as unsigned fixed-point, Ufix<w,p>, or as signed or unsigned integers, Int<w> and Uint<w>. In addition, several overflow and quantization

characteristics can be applied to C variables to model the potentially adverse effects of finite precision arithmetic.

Once the desired response was achieved in C simulation, A|RT Builder was used to automatically convert the fixed-point CORDIC algorithm to Verilog or VHDL. For the evaluation, a number of different C coding "styles" were employed to optimize the data path length, to introduce pipelining, and to share resources and hierarchy in the generated VHDL and Verilog files.

```
Int<32> MultA;
Int<32> MultB;
MultA = InpA * InpB;
MultB = InpC * InpD;
OutP = MultA + MultB;
```

Figure 3 - No Resource Sharing In C Code

```
COMPUTE_PROC: process(
.....
 begin
 InpA_n1 := signed(InpA);
 InpB_n1 := signed(InpB);
 InpC_n1 := signed(InpC);
 InpD_n1 := signed(InpD);
 OutP_n2 := signed(OutP);
 MultA := InpA_n1 * InpB_n1;
 MultB := InpC_n1 * InpD_n1;
 OutP_n2 := MultA + MultB;
 OutP_n1 <= std_logic_vector(OutP_n2);
 end process;
.....
```

Figure 4 - No Resource Sharing in VHDL Code

The examples in figures 3 and 4 illustrate two C coding styles: Figure 3 shows a non-resource shared operation coded in C and Figure 4 shows the corresponding VHDL code fragment

generated by A|RT Builder to get a two multiplier, one clock-cycle, non resource shared implementation in VHDL.

```
static Uint<1> cycle = 0u;
 static Int<32> MultA;
 Int<32> MultB;

switch (cycle) {
        case 0:
         MultA = InpA * InpB;
         break;
        case 1:
         Int<32> MultB = InpC * InpD;
         OutP = MultA + MultB;
         break;
}
++ cycle;
```

Figure 5 - Resource Sharing In C Code

```
begin
 OutP_n2 := "———————————————————";
 — copy state to local variables
 MultA := signed(MultA_r);
 — compute new state and outputs
 InpA_n1 := signed(InpA);
 InpB_n1 := signed(InpB);
 InpC_n1 := signed(InpC);
 InpD_n1 := signed(InpD);
 OutP_n2 := signed(OutP);
 cycle := unsigned'("0");
 case cycle is
 when "0" =>
 MultA := InpA_n1 * InpB_n1;
 when "1" =>
 MultB := InpC_n1 * InpD_n1;
 OutP_n2 := MultA + MultB;
 when others =>
 assert false
 report "Invalid (possibly unknown or dontcare)
value for cycle"
 severity warning;
 end case;
 OutP_n1 <= std_logic_vector(OutP_n2);
 — copy local variables to next value for state
 MultA_nxt <= std_logic_vector(MultA);
 end process;
.....
```

Figure 6 - Resource Sharing in VHDL Code

To get a resource-shared implementation in VHDL, a different C coding style is employed. Figure 5 illustrates the C code required to get a single multiplier, two clock-cycle implementation in VHDL. Figure 6 shows the resulting VHDL code fragment with only a single multiplier in the VHDL process description. Because a switch statement has been used in the C code, A|RT Builder generated a VHDL case statement (or switch in Verilog), indicating that the two branches are exclusive, so the operators used in them can be fully shared.

The Synplify FGPA synthesis tool was employed to synthesize the HDL code and the resulting netlist was input to the Xilinx Alliance series tools. Thus, the CORDIC algorithm was implemented in an XC4000 series FPGA. The design can also be implemented in Virtex FPGAs.

The evaluation proved the validity of the design flow from an algorithm written in C to an implementation in a Xilinx FPGA. The quality of VHDL or Verilog HDL generated by A|RT Builder is comparable to that of hand-crafted code. However, the HDL code generated by A|RT Builder is only as good as the C code source. You have full control over the code that is generated and by adhering to specific coding styles, generation of very efficient behavioral and RTL code by A|RT Builder can greatly decrease the time required to map a C algorithm into HDL code.

Importing Components

One of the most powerful capabilities of the A|RT methodology uncovered during the evaluation was the ability to import pre-defined, optimized components into the C code, such as those created by the Xilinx Core Generator, those offered by the Xilinx LogiCORE™ and AllianceCORE programs, or any custom-made

modules. Then you can use A|RT Builder to automatically map the instantiated components in VHDL.

For example, to import a Xilinx CORE Generator module, the C code would contain a function call referencing the name of the specific module. The Xilinx CORE Generator tool can be used to create custom parameterized, pre-optimized modules which are then imported as black boxes into A|RT Builder as explained. All of the parameters and pin names are named the same in the C function as they are on the CORE Generator instance. To simulate in C, you write a behavioral model in C for the CORE Generator module. The behavioral code, intended to be used only for C simulation and HDL code generation for the module, is inhibited by encapsulating the behavioral code between "#ifndef __SYNTHESIS__" and "endif" C statements. A|RT Builder is then employed to translate the C code into HDL. The C function then becomes a component declaration in VHDL or a module declaration in Verilog HDL.

For logic synthesis, these black-boxes are defined as "don't touch" to the synthesis tool. If you want to perform behavioral simulation of the CORE Generator modules in HDL, you have two options:

- Use behavioral HDL that A|RT Builder generates from your behavioral C code without inhibiting the HDL translation. Replace the behavioral HDL with a black-box component or module declaration before synthesis.

- Use the behavioral HDL code generated by the CORE Generator for the module, and replace it with a black-box declaration before synthesis. The Xilinx Alliance Series tools can automatically merge the black-boxes after synthesis

The benefits to the DSP algorithm designer and hardware engineer are enormous because the LogiCORE and CORE Generator blocks are now usable by both the system engineer and the hardware engineer. The system engineer can now incorporate pre-defined hardware functionality directly in C code. The hardware engineer gets a VHDL or Verilog HDL model from A|RT Builder that directly instantiates area- and performance-optimized LogiCORE and CORE Generator DSP building blocks. Also, if a DSP algorithm requires extensive memory elements such as register files or block RAM, the same code technique can be employed to target the block RAM capabilities of the Virtex devices.

Conclusion

Hardware designers realize that they will have to move to higher abstraction levels to increase their productivity; there is just no question about this. A|RT Builder is an ideal first step; it allows you to use C, while still performing a WYWIWYG

translation for maximum control over the generated HDL code. The main benefits of the A|RT methodology are:

- **Automatic conversion from C/C++ to HDL**-Eliminates error-prone manual re-coding from C to VHDL/Verilog.

- **Architectural design from C/C++**-C/C++ simulation can be substantially faster than HDL; richer and more attractive development environments; more-compact code.

- **Open System**-Easily include existing, pre-optimized modules like LogiCORE, CORE Generator, and AllianceCORE components. ∑

For further information about Frontier Design and the A|RT tools, please contact Doug Johnson at: doug_johnson@frontierd.com.

Consult the following references for more information about the CORDIC algorithm:

(1) Jack E. Volder, "The CORDIC Trigonometric Computing Technique" IRE Transactions on Electronic Computers, Vol EC-8, pp. 330-334.

(2) Vladimir Baykov, CORDIC Bibliography, http://devil.ece.utexas.edu/cordic.html

New Java Tool Increases Pipelining *Speed*

Genesis One Technologies has developed a Java-based tool that automatically creates pipelining in Xilinx designs.

by Ben Israel, President, Genesis One Technologies, Inc, ben@genesis-one.com

AutoPipe is a new tool that reads Xilinx-generated EDIF simulation files, rebuilds the structure, and writes out a pipelined Verilog file with the same logic. This verilog file is then re-synthesized with FPGA Express. AutoPipe is written in Java, and is controlled from a command line. The number of stages, the EDIF input file, and the Verilog output file are its only parameters; you apply it to your time-critical modules and no source code changes are necessary. The result is greater design speed and less design time.

Pipelining is basically an assembly line; results are produced every clock. In a pipelined design the first stages can begin processing new inputs while the last stages are finishing outputs. Without it, input stages must wait unproductively, for the output stages to finish

To use pipelining without an automated process, you must first select the number of stages, then place or schedule all the logic in the appropriate stage. Each time you change the number of stages, all the logic must be rescheduled. This greatly complicates the design process, is error prone, and is the most likely reason why most designs today are not pipelined.

Pipelined designs get maximum speed in the least area. Figure 1 shows the design tradeoffs for a network switch. This example has 12 logic levels, was synthesized and implemented in a Virtex V300FG456-6 FPGA with automatic placement and routing.
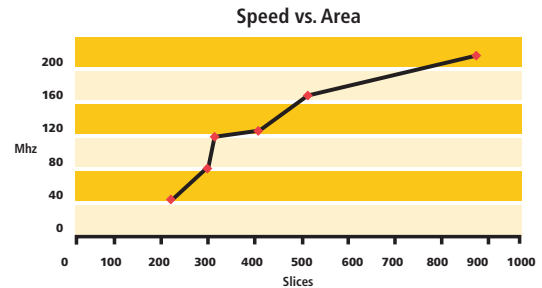
Figure 1 - Speed vs. Area Tradeoffs

Running AutoPipe with one stage is the equivalent of no pipelining. This creates a version of 218 slices that runs at 45 Mhz. A three stage version has 310 slices and runs at 102Mhz; over twice the frequency with a less than 50% area increase. A maximally pipelined version with 12 stages runs at 177Mhz using 889 slices; four times the frequency with four times the area.

Conclusion

With AutoPipe, you don't have to think about pipelining until your design is complete. Then, you can test different versions and choose the most efficient one. It can take a single design and easily create pipelined versions with up to four times the clock frequency. Because AutoPipe is applied after your design is done, you do not have to change the way you work to see the benefits. Finally, AutoPipe is not subject to human error and can even be applied to old designs. ∑

For more information on AutoPipe contact Ben Israel President at: ben@genesis-one.com; 561-852-2146

Maximizing HDL Simulation Performance

How do you know what is happening during simulation? Here's one way, using the new ModelSim SE Performance Analyzer from MTI.

by Darron May, Technical Marketing Engineer, Model Technology Inc, darronm@model.com

Simulation can be time consuming, and often there is no visibility into what is actually happening. That's why Profiling or Performance Analysis can very valuable. Performance Analysis shows where the simulator is spending time, and can be extremely useful at all levels of the design abstraction from behavioral to gate level. The new ModelSim SE simulator (Special Edition) has a built-in Performance Analyzer, available in the 5.3 version of VSIM, with the potential to save many hours of regression test time.

Performance Analyzer

The Performance Analyzer provides a graphical representation of where ModelSim is spending run time. The graphical representation is useful in quickly and easily determining what is impacting your design environment's simulation performance—it is rare that a simulation environment is fully tuned for performance.

At any moment the simulator will be executing a specific action, which can be categorized into a number of groups. This action could be attributed to a line of code or an internal housekeeping event. Profile samples are accumulated for each group and line of code, and then presented as the amount of execution time with respect to the total run time.

The benefit of using this type of statistical analysis is that you typically don't have to run an entire simulation to get enough information to analyze the environment.

Improving RTL Performance

The following example shows how the Performance Analyzer was used to improve simulation run time at the RTL level; it takes a customer design and shows how the simulation run time was improved. This VHDL design was from a data communications application, used as a buffer between two different data stream rates; the concepts hold true for an equivalent model in Verilog. The Performance analyzer was used to take around 3000 samples. After simulation it could be seen that 98% of the simulation time was spent executing two lines of code as shown in Figure 1, each using a similar style but in separate parts of the design. These lines of code were VHDL loop statements. The run time for this example was 20 minutes 34 seconds (Code Fragment 1).

Figure 1 - Profiler Report Window

```
Code Fragment 1 - ORIGINAL
for i in 0 to (buffer_size - 1) loop
IF (i = ramadrs((counter_size - 1) downto 0)
THEN
rd0a <= buffers(i);
END IF;
end loop ;
```

A loop was used to make an assignment to each of the buffer locations in the design if the input counter address matched the currently indexed location. It was possible to add an "exit" command into both of these loops which would break the loop when the match occurred. After the code modifications, the simulation was re-run and the run time was reduced to 10 minutes and 3 seconds, a 2X run time improvement (Code Fragment 2).

```
Code Fragment 2 - EXIT
for i in 0 to (buffer_size - 1) loop
 IF (i = ramadrs((counter_size - 1) downto 0)
 THEN
 rd0a <= buffers(i);
 exit;
 END IF;
end loop ;
```

However, the loop was still consuming the majority of the simulation time as detailed by the profiler output. As the loop is being used to compare an address pointer to a location it was possible to re-write this model to use an indexed array to achieve the same function. When the simulation was re-run with the new array version of the model the run time reduced to 1 minute 2 seconds, which was 20X faster than the original model. Code fragments for each change are shown.

The profile then showed a better distribution, however there was now another part of the design consuming the majority of the time. This was the control block and was centered around a large counter that was modeled using a std_logic_vector. In VHDL integers take less memory than std_logic_vectors and can be manipulated as part of the language. Therefore if this model could be re-written to use integer math, then time could be saved in conversion, addition, and comparison functions (Code Fragment 3).

```
Code Fragment 3 - ARRAY
address := conv_integer(ramadrs((counter_size-1) downto 0));
rd0a <= buffers(address);
```

The model was again re-written to reduce the number of events by using less processes, integer math, and variables for connecting

functions instead of signals. Signals cause events when they change whereas variables do not. The simulation was re-run with the new model and the simulation time was further reduced to 27 seconds.

Due to the profiler showing where time was spent in the simulation, the use of better modeling techniques allowed an overall speed improvement of over 40X from the original model.

Gate Level Performance

The Performance Analyzer can also be used at the gate level. ModelSim automatically recognizes that a VITAL function is being referenced from the IEEE library and generates code to call hand-optimized built-in routines. All of the functions within the VITAL_primitives and VITAL_timing packages are accelerated in this way for both ModelSim PE and ModelSim EE. There is an extra level of optimization that occurs in the EE product. To qualify for this global acceleration, the VITAL cell has to be VITAL Level 1 compliant. The model is then transformed into a state machine that does the equivalent behaviour more efficiently. This can have a significant effect on the simulation performance. The Performance Analyzer can be used to see the effects of an non-accelerated model as the execution of accelerated code is shown in a separate category.

Conclusion

The Performance Analyzer has already been used by a number of customers at all stages of the design cycle to save many hours of regression test time, or to allow more verification in the same amount of time. With the growing time-to-market pressures, allowing more verification in the same time period will increase product quality. ∑

For more information see Model Technology at: www.model.com.

What's New
in V2.1i for XC9500 CPLDS?

Our latest Alliance Series and Foundation Series software, v2.1i, offers an uncompromising level of performance while improving ease of use.

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

The Xilinx Alliance Series™ and Foundation Series™ software provides a complete development environment for Xilinx CPLDs (and FPGAs). The v2.1i improvements for CPLDs include:

Ease of use enhancements

- ChipViewer - the new, interactive, graphical device constraints editor
- Implementation template enhancements

Performance enhancements

- New "Balanced" optimization for even logic distribution and pin placement
- More aggressive "Speed" optimization for the fastest clock speeds possible
- Complete XC9500/XL/XV fitting and programming file support
- XC9500XL-based tutorials along with CPLD quick install
- STAMP models and XFLOW support
- XPORT capability in Foundation

Ease of Use Enhancements

There are many new ease-of-use improvements in the v2.1i release, including the following.

ChipViewer

ChipViewer is our new graphical constraints editor; an interactive way to view and control the logic design routing within a Xilinx CPLD. It also allows you to either pre-assign the pinout or see exactly how the design is routed in your selected device. This provides a more intuitive approach to assigning constraints within a CPLD. Additionally, it allows you to verify and trace the actual design's routing and resource usage. See the companion article on page 48.

Implementation Templates

Xilinx makes it easy for you to select how your design should be routed, by using a new GUI conveniently located under the Options menu in the Alliance Series and Foundation Series Design Managers, as shown in Figure 1. This GUI offers you a choice of four different optimizations, and each forces the fitter to route the design with different end objectives:

- Area - Provides the best density optimization.
- Speed - Provides the best possible clock frequency response.
- Balanced - Provides even logic distribution and best pin-locking; new in v2.1i.
- User Defined - Provides full control over design implementation options.

Pre-defined optimization templates
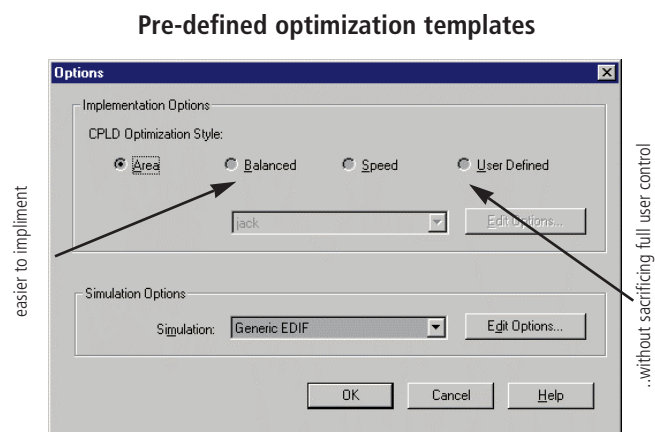
Figure 1 - CPLD Implementation Template Settings Under "Options"

Performance Enhancements

There are many new performance improvements in the v2.1i release, including the following.

Complete XC9500/XL/XV Fitting and Programming File Support

The v2.1i software now includes full fitting, programming, and file support, not only for the XC9500/XL families but also for the newer 2.5V XC9500XV product.

XC9500XL-based Tutorials Along with CPLD Quick Install

The XC9500XL is now being used as the basis for Xilinx CPLD tutorials. Additionally, The Alliance Series and Foundation Series software provide the option for a "CPLD-only" quick install (without the FPGA tools) to help speed you on your way.

STAMP Models

STAMP models are useful in specifying pin-to-pin timing delay and constraint information for components. Xilinx is the first PLD supplier to output the Stamp file format from its implementation software.

XFLOW Support

XFlow is a non-graphical tool that encapsulates the latest implementation and simulation flows. It is device independent and has a simple interface to the Xilinx tools that is flexible, extensible, and user customizable.

XPORT Capability in the Foundation Series

XPort 4.1 converts ABEL designs to Verilog or VHDL format. ABEL test vectors are converted as well and put in a separate test bench file. XPort 4.1 also converts AHDL designs to Verilog or VHDL format. Hierarchical AHDL designs are also supported. Each AHDL module file will have a resulting Verilog or VHDL output file.

New v2.1i Benchmark Tests

Xilinx recently completed a benchmark study comparing our previous implementation tools (v1.5i) with our latest v2.1i release. These benchmarks clearly show that the new v2.1i software has once again raised the industry standard in both ease-of-use and performance improvements.

Fifty-seven VHDL and Verilog designs were compiled and benchmarked. No additional timing constraints were used during the EDIF netlist creation. The various design EDIFs were used as the input files

to both the v1.5i (service pack 2) and v2.1i fitter tools. Two different implementations were run using v1.5i. The first implementation attempted to fit the EDIF file using an area optimization algorithm. The second implementation used a speed optimization algorithm. In v2.1i the same two implementations were run along with a third, the balanced implementation.

The output results were compared for area, speed, and default settings. The implementation comparisons were performed for both the 5V CPLD device families (XC9500) and again for the 3.3V /2.5V device families (XC9500XL/XV).

The results were impressive; the 57 designs were compared on a design-by-design basis with the difference between them computed as a percentage. The range of improvement in clock frequency was 3% to 96%. Figure 2 shows the average change for both the XC9500 and XC9500XL/XV. As you can see, the new v2.1i software generated fewer macrocells and product terms while increasing the design speed.
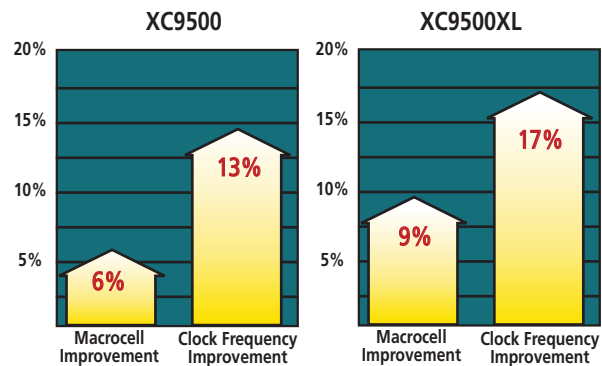
New aggressive speed optimization dramatically improves clock frequency response.

Notes:
• Average % improvement for designs affected using the speed template
• Range of improvement in clock frequency: 3% to 96%
• 57 test designs, 13 affected by V2.1i improvement

Figure 2 - v2.1i vs. v1.5i Software Benchmark Results

Conclusion

The new v2.1i release of Xilinx Alliance Series and Foundation Series software sets new standards of excellence in quality of results, ease-of-use, and features. Partnered with an XC9500 architecture rich in features, and the industry's lowest CPLD prices, Xilinx clearly provides the industry's best CPLD solution. ⚡

Timing Correction for Flight Time Compensation

With the HyperLynx signal integrity simulation software, you can easily verify the overall timing of your high performance designs.

by Lynne Green, Signal Integrity Engineer
HyperLynx, lgreen@hyperlynx.com

As signal and clock speeds increase, and driver edge rates decrease, the routing segments on a printed circuit board become transmission lines. Today, the routing on the board may contribute more to the delay than does the capacitance of the receiver IC. Therefore, modeling the load as a fixed capacitance does not provide good delay prediction on a printed circuit board or multi-chip module. In addition, I/O buffers and logic delays are often modeled and simulated separately, making it difficult for you to create a consistent timing model.

You must correct the fixed timing delays (obtained from the data book) to get the actual values after layout is complete, because the flight time of the signal down the routed nets is usually comparable to the delay through the integrated circuits. Once you have compensated for this "flight time," you can verify timing constraints such as setup and hold times. The compensation calculation starts with the data book delay under fixed loading conditions, and then arrives at a corrected delay for the actual board load (including the wire routing and the receiver) by using signal integrity simulation software such as HyperLynx.

Timing Correction on PC Boards

Integrated circuit (IC) delays presented in data books are specified at a fixed test load, which is selected by the manufacturer as representative of loading for nets with moderate to high fanout and no wires. The data book delay includes both the logic delays and the delay through the I/O driver. On the other hand, I/O buffers are usually modeled using IBIS (I/O Buffer Information Specification) for simulation. In an IBIS model, only the buffer is represented, and there is no information about any logic delays inside the integrated circuit. Fortunately, the IBIS model provides an easy method to compensate board-level delays for flight time effects.

Figure 1 shows the delay between a driver and a receiver for three different loads:
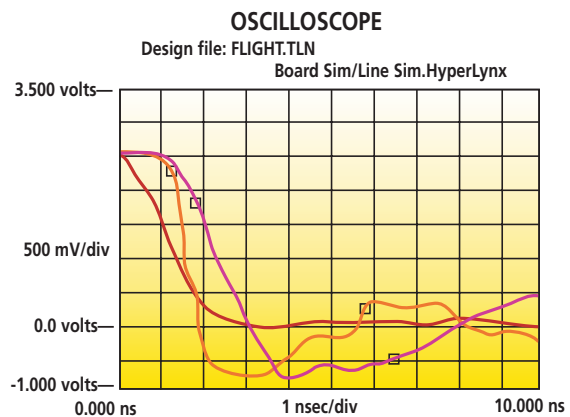
Figure 1: Comparison of delays with loads of (a) 35 pF CREF, (b) 6.5 pF receiver, 6" (11 pF, 68 nH) transmission line and 17.5 pF load capacitor to total 35 pF, (c) 6.5 pF receiver and 6" transmission line.
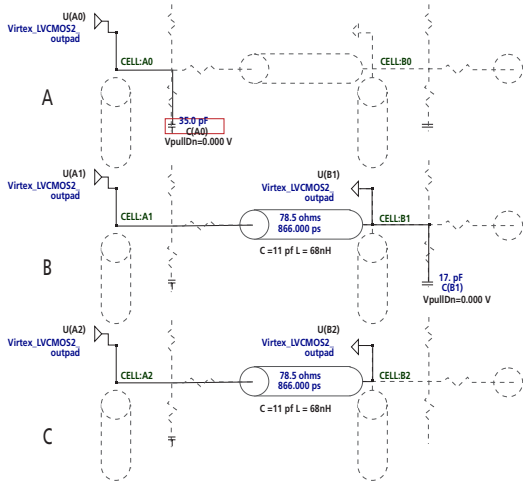
Figure 1.1 - Loading Diagram

(A) The CREF 35pF capacitive load (left curve).

(B) A 6.5 pF receiver, with a 6" transmission line (11 pF), and a 17.5 pF capacitor for a total load of 35 pF (right curve).

(C) Just the 6.5 pF receiver and 6" transmission line (center curve).

The IBIS models used here are for a Xilinx Virtex LVCMOS I/O buffer. This is a dramatic illustration of the effects of transmission line delays being comparable to capacitive loading delays.
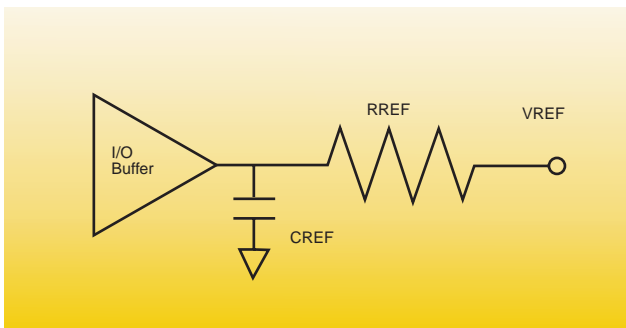
Figure 2 - The IBIS timing model.

The Data Book Delay

The delay presented in a typical data book is from an input (such as CLK or A0) to the output (such as Q or Y2) of an FPGA or other IC. This delay includes two effects: the delay through the IC between the specified input and the input of

the I/O buffer, and the time for the I/O buffer to pull the load up (or down) to the manufacturing test voltage VMEAS. This test load may be any combination of CREF and RREF, as shown in Figure 2. The manufacturer assumes that the board-level user will then start any timing measurements from VMEAS.

The data book delays are measured with lumped loads to make manufacturing testing easier. However, on a real printed circuit board or multi-chip module, each routing segment is actually a transmission line—a distributed LC component, as shown in Figure 3. On the typical PC board, the inductance contributes
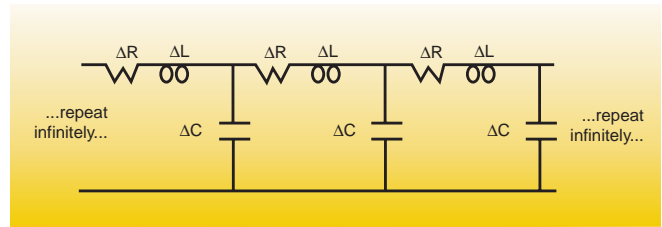
Figure 3 - Electrical model of a transmission line.

significantly to the delay. On a PC board or multi-chip module, for example, the driver and receiver pins may be an inch or more apart. While an inch may not seem like much at first glance, it represents a delay of roughly 200 ps—comparable to the transition time of a fast driver. Therefore, transmission line delays must also be included in timing analysis, even though they are not in the data book delay specifications.

Timing Correction Factors

Figure 4 shows the sequence of events. At T=0, the input changes on the driver IC. At the data book load CREF, TCREF is when the signal has gone through the IC's logic path and the output driver voltage crosses VMEAS. Finally, Tr_max is when the receiver is guaranteed to have changed state. Note that the total delay (Tr_max) may be either positive or negative; this is caused by the manufacturer's choice of CREF and the slew rate of the input signal.
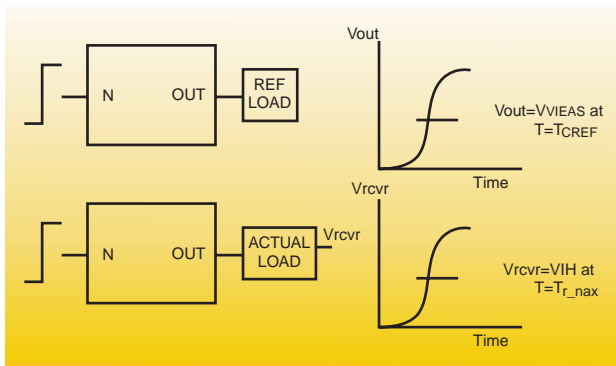
Figure 4 - The timing diagram for TCREF and delay Tr_max.

The compensation time, T_{comp}, and pin-to-pin delay, $T_{pin-to-pin}$, are required to correct the predicted delay from the data book value to the actual value. Once T_{comp} and $T_{pin-to-pin}$ have been determined, all data book delays to the output pin can be corrected. A separate correction is required if the load is changed, because both T_{comp} and $T_{pin-to-pin}$ are load dependent.

Normally, separate timing calculations are required for the rising edge and for the falling edge. This is because the output buffer does not have the same drive current capability when rising as it does when falling. For some buffers, these two times may be comparable, but for others these times may differ by a factor of 2 or more. In particular, open drain (and open collector) drivers have a strong pull down on a bus but very weak pull up, depending on other drivers on their bus to pull the bus up quickly.

Extracting Compensation Time

The compensation time is the time difference between the output voltage crossing VMEAS at the manufacturer's test load and at the actual load. Once calculated, the same T_{comp} applies to all delays from any input pin to this output pin with this load. For other output pins or other loads, the time must be re-measured, because the driver characteristics and load characteristics determine T_{comp}.

The simulated delay T_{CREF} with the manufacturer's test load on the I/O buffer is extracted using a signal integrity simulator (such as HyperLynx) and the buffer's IBIS model. Then the delay T_{vm} under the actual loading (transmission line, receivers, and any terminating components on the signal net) are extracted the same way. The difference $T_{comp}=T_{vm}-T_{CREF}$ is a calibration of the output voltage delay change caused by loading. The value of T_{comp} depends on the simulator; T_{comp} may be positive or negative.

In Figures 5 and 6 the simulation results for a Xilinx Virtex LVCMOS I/O buffer are shown. The LVCMOS output has VMEAS=1.25V and CREF=35pF. For Cload=CREF=35pF, the output delay was 1.2 ns in the HyperLynx simulator. For the actual load of 6" of transmission line and

Figure 5 - Simulated delay at C_{load}=CREF. Scope settings are 0.5V/div and 200 ps/div. Vout=VMEAS at T_{CREF}=1.20 ns.

one receiver, the delay was 0.7 ns. We can now calculate the compensation time for the simulator:

$$T_{comp} = T_{vm} - T_{CREF} = 0.7 - 1.2 = -0.5 \text{ ns}$$

Extracting the Pin-to-Pin Delay

The pin-to-pin delay, caused by the transmission line and receiver, can be measured directly from a plot of simulation results in a signal integrity simulator. The maximum delay to the receiver is

38

Figure 6 - Simulated delay with 6" transmission line and one receiver. Scope settings are 0.5V/div and 200 ps/div. Vout=VMEAS at T_{vm}=0.7 ns, V_{in}=VIL=0.8V at T_{in}=1.7 ns, and $T_{pin-to-pin}$= T_{in}-T_{vm}=1.7-0.7 = 1.0 ns.

when the receiver crosses VIH (rising) or VIL (falling). The delay depends on the signal's interaction with the transmission line, and on the termination strategy. Figure 6 shows the simulated pin-to-pin delay, for the example 6" of transmission line and a 6.5 pF receiver, to be 1.0 ns.

The delay caused by the transmission line is simply the length of the line divided by the speed of the signal in the line: T(line) = Length/vel. The transmission line delay is always positive. Transmission line delays can be anywhere from a few picoseconds to several nanoseconds, depending on the length of the line. For example, the 1" line mentioned previously had a delay of about 200 ps.

The delay of an unterminated receiver is due to the RC impedance at the load, where R is the characteristic impedance of the transmission line and C is the capacitance of the receiver. The time for the receiver to respond is the time required for this RC response to transition from VMEAS to VIH (rising) or to VIL (falling). The

receiver delay can be positive or negative, depending on the values of VMEAS and VIH or VIL.

The receiver delay is further influenced by the termination strategy. A series resistor at the driver will reduce the voltage drive available on the transmission line and therefore the drive available to change the voltage on the receiver's internal capacitance, while a parallel (R or RC) termination at the receiver reduces the signal current available to change the voltage on the receiver's internal capacitance. Termination choices can cause significant variation in delay; -50% to +400% variation can be observed in some designs.

The pin-to-pin delay due to the transmission line and receiver (and any associated termination components) is difficult to calculate, particularly when the transmission line branches and connects multiple receivers. Simulation software is normally used for this purpose. Fortunately, the time between the driver's voltage crossing VMEAS and the receiver's state change (maximum delay) can be measured quickly and easily, even for the most complex routing topologies, using the HyperLynx simulator.

Calculating the Total Delay

Once a signal integrity simulator has been used to measure the time delay between the driver pin and the input pin of the receiver, the goal of finding the actual delay can be realized.

The simulator delay includes the buffer delay, the delay down the transmission line, the effects of the receiver capacitance and all termination components, and the voltage change from measuring VMEAS at the driver to VIH or VIL at the receiver. On the other hand, the data book delay includes the buffer delay and the delay

39

through the IC's logic stages as well as the output buffer driver delay, but only at the manufacturer's test load. By referring back to the diagram in Figure 4, we can now calculate the total delay to the receiver.

The total delay can be calculated, including flight time and simulator calibration effects, as:

$$T_{correction} = T_{pin\text{-}to\text{-}pin} + T_{comp}$$
$$T_{r\_max} = T_{databook} + T_{pin\text{-}to\text{-}pin} + T_{comp} = T_{databook} + T_{correction}$$

For our example, with the 6" of transmission line, the Virtex LVCMOS I/O, and using the values from Figures 5 and 6, the correction factor becomes:

$$T_{correction} = 1.0 + (\text{-}0.5) + = +0.5 \text{ ns.}$$

So, if the data book delay from pin A1 to pin Y9 is Tdatabook=3.5 ns (including logic), then the maximum delay to the receiver is:

$$T_{r\_max} \text{ (A1 to Y9)} = T_{databook} + T_{correction} = 3.5 + 0.5 \text{ nsec} = 4.0 \text{ ns}$$

Similarly, if there is a data book delay of 2.7 ns from another input, say A7, to the same output, Y9, then the maximum delay for that path would be:

$$T_{r\_max} \text{ (A7 to Y9)} = T_{databook} + T_{correction} = 2.7 + 0.5 \text{ ns} = 3.2 \text{ ns}$$

The maximum delay can be shorter or longer than the data book delay. This corrected delay information can be used to check whether design timing constraints, such as setup and hold times, are satisfied.

Note that it took only two simulations to arrive at the total delay. The first simulation was used to obtain the delay for the manufacturer's test load. The second simulation used the actual net, with its receiver IC and driver IC IBIS

models, and the actual net routing and termination components. Finally, the compensation time and the pin-to-pin delay from those simulations are used to correct the total time delay from the input of the first IC to the input of the second IC.

Conclusion

With today's very high speed designs you must consider the transmission line effects of the interconnections between components. By using the HyperLynx signal integrity simulation software, you can precisely determine the overall timing of your design and be assured that your design will work flawlessly under all specified conditions. ∑

For more information on the HyperLynx software, see: www.hyperlynx.com

Dr. Lynne Green has over twenty years of design experience. She recently joined HyperLynx, a division of PADS Software, Inc, as the Product Marketing Engineer for High Speed Products. Previously, at Duet Technologies, Dr. Green designed I/O library cells, and was president of Green Streak Programs, where she did consulting in modeling and simulation. She earned MSEE and PhD degrees in Electrical Engineering at the University of Washington.

The *POWER* of the Xilinx Foundation Series Software

Why is it so important to have fully integrated development tools?

by Karen Fidelak, Integrated Solutions Technical Marketing, Xilinx, karen.fidelak@xilinx.com and Loren Lacy, Integrated Solutions, Product Marketing, Xilinx, loren.lacy@xilinx.com

How would you define the ultimate set of design automation tools? You would probably include best-in-class point tools, the ability to run on a variety of operating systems and computing platforms, the ability to share data between operating systems and computing platforms, and the ability to seamlessly pass design data from one tool to the next.

Until recently, most companies focused more on the individual point tools than the flow of data between the tools. Large digital design companies would then connect each of the point tools in a customized flow. This need for connecting the various point tools led to the development and support of standard information exchange interfaces such as EDIF (Electronic Design Interchange Format). While EDIF is a very flexible standard its flexibility takes away from its simplicity when interfacing to various tools. So, while EDIF standardizes tool integration, it could be argued that it does not simplify the overall job.

Looking at the EDA landscape today, we see the beginning of a new focus. As more and more companies look at solving the "system on a chip" problem, it has become obvious that there is actually more value in the integration of tools than in the individual point tools themselves. This is apparent in the level of cooperation between traditional rivals such as Cadence and

Synopsys, along with the demise of individual point tool providers like Zycad.

Design Flow Management

In addition to the easy passing of design information between point tools, you also value the ability to specify common information, just once, for multiple tools. This includes the location of simulation libraries, macro libraries, and timing information, for example. While the specification of this information can often be automated, updating one of the point tools within the flow often calls for a complete rewrite of the setup information. An integrated tool suite automatically communicates this information to each tool.

Using various tools within a design flow often requires the creation of additional design data files. Unlike a homegrown flow automation process, an integrated design suite is aware of the downstream tool requirements. For example, if you want to perform timing simulation after place and route, the place and route tools must be instructed to produce the timing simulation netlist, so it can be read by the simulator.

Project Management

Given the large number of source files, control files, and implementation files associated with a given design project, it becomes desirable and

necessary to have the software tool take control of managing all of these various files.

A design project may consist of schematics, HDL files, IP cores, netlists, user constraints, or any combination of these. With the complexity of designs today, it can become difficult to manage all of these design modules as pieces of the design are modified. An integrated tool suite will know about all modules in the design and will determine when sources have been changed and therefore when design netlists must be updated and processes re-run. The tool will also clearly display all the design sources and implementation results and provide easy access to the appropriate editing tool for the given source file.

Schematic Entry tools as well as many HDL compilers require that the device family library be specified up front o provide the appropriate library symbols and components for the given architecture. Additionally, if a design is retargeted to a new device architecture in the middle of the design process, it is necessary to change the project libraries to match the new architecture. In an integrated tool suite this is done automatically by the tools, leaving you with nothing to do but select the device family, once. This selection will set the appropriate device libraries for design entry, as well as pass the device information forward to the place and route tools.

Designs are likely to be implemented many times in the course of a design cycle. For example, modifications to timing constraints, target device, and place and route options may be made in the pursuit of the best overall design implementation. As these design iterations are made, it is convenient to have access to previous results for comparison and archiving. An integrated tool suite provides revision control by archiving each implementation for future reference or use.

Optimization of the Design

You usually have some overall design strategy that you are looking to accomplish in your design flow. This strategy may place the highest priority on fitting the design in the smallest possible device, or getting the fastest performance, for example. The synthesis tool may be able to optimize the design based on the given timing requirements, but then the place and route tools also need to receive the same direction to appropriately place and route the design. This can mean setting these requirements twice. In an Integrated design environment, these settings need only be entered once.

Software Development

When placed in an integrated environment, software tools are given the difficult task of working well together, a task often left to the user with a collection of point tools. The tools must communicate with each other to pass design data. The data which is being communicated must be understood by all tools involved, and consideration must be taken for enabling the features and benefits of all tools involved. When an integrated tool suite is created, the software developers must collaborate. Interaction in the development phase is critical to ensure a working environment in the final product.

Software quality assurance is a collaborative effort in an integrated tool suite. If various partners' point tools are involved, the majority of the individual point tool testing is done by the respective vendor. Integration testing is done by

all partners involved to ensure the proper flow of data between tools. An integrated tool suite undergoes thorough testing from design entry through design implementation to ensure that all tools communicate properly with each other and that data is successfully processed.

Software Support

Running into a problem with the software while doing a programmable logic design can be very frustrating. What can be even more frustrating is hunting around trying to find the vendor who can actually answer the question. With a true integrated tool environment, each vendor involved should be familiar with the entire design flow and with the other tools involved. All of the partners work very closely to develop and test the product. Support for the entire product should be available from a single source, where the support engineers are well trained in the individual point tools as well as the interfaces between them.

Xilinx Foundation Series – a Completely Integrated Tool Suite

The Xilinx Foundation Series provides software for every phase of the design flow including design entry (schematic, HDL, state machine), synthesis, simulation, implementation, and verification—all under a single project management environment. Synopsys FPGA Express HDL Synthesis and Aldec schematic entry and simulation are combined with Xilinx implementation tools to provide a first-class integrated design environment.

Libraries necessary for the selected device architecture are automatically included in the project, and synthesis optimizations are tailored to the selected device as well as overall design strategy. Timing constraint information, which is entered in the synthesis phase of the flow is passed directly, to the place and route tools. Files necessary for timing simulation or floorplanning are automatically created by the implementation tools and later loaded into the respective point tool upon invocation.

Source files and implementation revisions are marked in or out of date as appropriate, and automatic updating of the design is available when necessary.

These are just a few examples of features within the Foundation Series software which demonstrate the true integration of the product.

Conclusion

Software engineers from Synopsys, Aldec, and Xilinx work closely through the product development to ensure tight integration, ease of use, and high overall quality of results. The entire Foundation series design flow is fully tested by Xilinx SQA engineers, and support for all aspects of the flow is available from Xilinx technical support. The effort put forth by all partners involved has led to a tightly integrated, full-featured, well-supported product which ultimately increases the productivity of the end designer. Σ

Xilinx **Launches Design Reuse** Initiative

New tools allow you to capture and share your own IP over the Internet.

by Jim Burnham, CORE Generator Product Manager, Xilinx, jim.burnham@xilinx.com

Xilinx recently announced a design reuse initiative aimed at helping you implement designs for reuse and share your intellectual property (IP) internally. The first offerings include:

- **The IP Internet Capture™ tool** - allows you to easily capture and package intellectual property (IP) you have already created for Xilinx FPGAs and share it inside your organization with other design teams, over the Internet.

- **The IP Remote Interface™ tool** - provides you (and third-party developers) with a method to integrate your own parameter-driven IP with the Xilinx CORE Generator system. These cores can be distributed or sold over the Web in a secure environment.

As part of the initiative, Xilinx also created a manual named "Xilinx Design Reuse Methodology for ASIC and FPGA Designers," a supplement to the "Reuse Methodology Manual" from Synopsys and Mentor Graphics. The Xilinx Reuse manual provides guidelines for designers who want a common strategy for reusing intellectual property, regardless of whether it was developed for ASICs or for FPGAs.

"Xilinx customers have been using million-gate FPGAs for more than a year, and they have benefited from leadership products such as the Xilinx 64-bit, 66-MHz PCI core for those devices" said Xilinx CEO Wim Roelandts. "While cores are important to large designs, we also want to help customers leverage the tremendous value of the intellectual property they have developed internally for FPGAs. This initiative goes hand-in-hand with our recent announcement of the new Virtex-E multi-million-gate FPGAs, and it will help customers address challenges of reuse, modular design techniques, and team-based design."

IP Internet Capture Tool

The new IP Internet Capture tool, the first of a planned series of programmable logic design reuse software products from Xilinx, provides you with an automated method to identify, capture, and document a module of synthesizable VHDL or Verilog code, or a fixed-function netlist. The IP Internet Capture tool prompts you for key reuse information such as module naming and cataloging information,
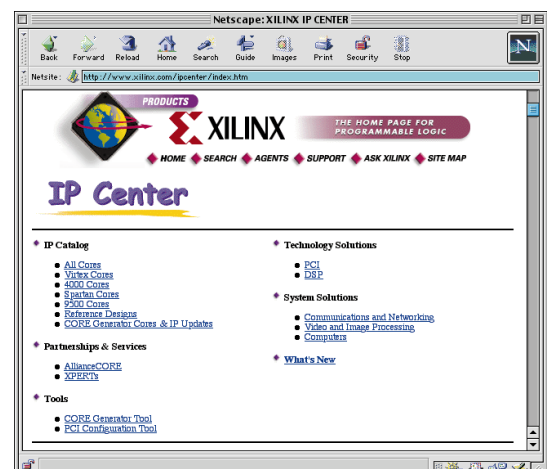
Figure 1 - Design Reuse Tools Available from the Xilinx IP Center

module description, and details about technical support and module source files. It supports both PDF files and HTML-based Web pages to document cores, and it allows you to create a directory for copying and creating files and creating Web files.

By creating a Web page with links to download the module, the IP Internet Capture tool allows distribution of your intellectual property over your network, as well as over internal or external web sites. Once the new module has been captured and posted, other engineers can use standard Internet browsers to download the IP and install it in their copy of the Xilinx CORE Generator system.

The IP Internet Capture tool supports all Xilinx FPGA families.

IP Remote Interface Tool

The new IP Remote Interface tool is a new feature of the Xilinx CORE Generator system, providing the ability to access IP over the Web. Third-party Xilinx AllianceCORE partners or customer teams can use the IP Remote Interface tool to create parameter-driven cores. It allows IP designers to create their own graphical user interface and software executables. These applications with customer-specified parameters can then select from a set of fixed net lists, modify generic values in VHDL or Verilog source code, or drive algorithmic software code implemented in programming languages such as Perl, C++, or Java.

This tool provides a high level of security, and it controls access to IP source code, which can be encrypted or compiled. Xilinx AllianceCORE partners can develop parameter-driven cores that access encrypted source code and then write an encrypted or clear-text netlist. Partners can also control access to their cores by

requiring authentication or a password. IP developers can also catalog their cores in the Xilinx CORE Generator, and then, through a Web address, provide secure access to their application. This feature provides for both authenticated access and source code security.

"Xilinx has been successfully offering customers access to IP, such as our PCI core, BaseBlox, DSP cores, and reference designs through the IP Center™ on the Xilinx website," said Rich Sevcik, senior vice president of software, cores and support at Xilinx. "Providing our AllianceCORE partners and customers with this capability extends our commitment, made a year ago in the Silicon Xpresso initiative, to step up use of the Internet to increase the productivity of designers."

Xilinx Design Reuse Methodology Manual

The new "Xilinx Design Reuse Methodology for ASIC and FPGA Designers" manual is intended for designers who need to target both ASIC and FPGA architectures with the same RTL code. This Xilinx supplement to the Synopsys and Mentor Graphics manual provides an overview of FPGA system-level features and contains general RTL synthesis coding guidelines that have the most impact on improving system performance.

Conclusion

The IP Internet Capture and IP Remote Interface tools, as well as "Xilinx Design Reuse Methodology for ASIC and FPGA Designers" manual are available from the IP Center on the Xilinx website (www.xilinx.com). The Xilinx IP Center has long been a resource offering access to complete IP solutions. Design reuse is the latest addition to the Xilinx list of IP solutions. ∑

Try HDL Simulation for Free

Xilinx and Model Technology have partnered to give you a risk-free introduction to HDL simulation.

by Dave Kresta, Product Line Manager, Model Technology, davek@model.com
Craig Willert, Software Marketing Manager, Xilinx, cnw@xilinx.com

You probably already know why using VHDL or Verilog is such a good idea, and you may already be using a high-level description language (HDL) to represent your design. You may be coupling HDL with synthesis, and simulating your design with a gate-level simulator to confirm its behavior after synthesis. However, when it comes to verifying your design, are you still relying on the time consuming method of testing programmed devices in a prototype? If so, here is a better way.

As your designs get bigger and your design cycles get shorter, physical verification becomes too time consuming and costly. However, with the ModelSim Xilinx Edition Starter (XE Starter) and ModelSim Xilinx Edition (XE) software, you can try VHDL or Verilog simulation and begin to leverage the full power of an HDL-based design flow.

Why Use HDL Simulation?

Even if you are not trying to squeeze three whole circuit boards onto a single Virtex device, your next design will probably be bigger than your current design. The benefit of a full HDL design flow that includes HDL simulation is simple: you will get your designs fully debugged and working in the system, faster.

HDL simulation allows you to debug your design at the source code level, pinpointing design problems directly to the line of code responsible for the failure. For example, you can stop the simulation whenever a variable changes, or step through a piece of code line by line, or trace a signal's flow through a design.

You can write tests in VHDL or Verilog which not only apply stimulus to your design, but which can check for the correct response. The same "self-checking testbenches" can be used before synthesis, after synthesis, and after place-and-route. Not only is it easier to let the testbench inspect the design for failures, but it will improve quality as well.

HDL simulation also allows you to explore alternatives and determine design tradeoffs quickly. You can create powerful designs with thousands of logic gates in just a few small modules of high level VHDL or Verilog code. And you can make major changes to the design quickly, and simulate those changes without running any synthesis or implementation tools.

The Right HDL Simulator for YOU

ModelSim is the most popular HDL simulator on the market, with over 40,000 licenses sold to date. This success is grounded in ModelSim's performance, value, ease of use, and broad industry support. Now, with the introduction of ModelSim XE Starter and ModelSim XE, Model Technology and Xilinx are working together to ensure that you have the tools necessary to meet all your HDL design and verification needs. With Model Technology and Xilinx as your partners in

	ModelSim XE Starter	ModelSim XE	ModelSim PE	ModelSim SE/EE
	Available from Xilinx	Available from Xilinx	Available from Model Technology	Available from Model Technology
100% compilant VHDL or Verilog simulation	X	X	X	X
Complete HDL debugging environment	X	X	X	X
Windows 95,98,NT Support	X	X	X	X
Unix Support				X
Appropriate for Device Densities in the range of:	Less than 500 lines of HDL code	Xilinx 9500, Spartan, low-density 4KX and Virtex FPGAs up to approximately 60k gates*	Most CPLD/FPGA designs	All designs, including extremely large FPGA and ASIC designs
Typical simulation times for medium sized design and test set	10 minutes	10 minutes*	2 minutes	30 seconds
Support for mixed VHDL/Verilog designs			X	X
Language-neutral licensing				X

Table 1 - ModelSim Versions

*Simulation performance for ModelSim XE diminishes by a factor of 2x for designs with more than 4000 lines of RTL HDL code, and by an additional 10x for designs with more than 30,000 lines of RTL HDL code. See www.model.com for more ModelSim PE/EE/SE differences.

design, you are certain to get to market quicker, helping make you and your project successful.

Features:

- **Proven technology**–the most widely used HDL simulator in the world.

- **Easy-to-use**–fast, comprehensive debugging with a full-featured graphical interface.

- **Powerful**–has the performance and features for even the most demanding designs.

- **Flexible**–handles VHDL, Verilog, or even mixed-language designs.

- **Scalable**–configured to meet a variety of performance, capacity, and budget demands.

ModelSim Xilinx Edition Starter (XE Starter)

The ModelSim XE Starter version is a risk free solution for learning the benefits of VHDL or Verilog simulation. This product enables new HDL designers to experiment with the simulation of small HDL designs. ModelSim XE Starter is provided free of charge to all registered Xilinx customers who have current maintenance contracts.

ModelSim Xilinx Edition (XE)

ModelSim XE provides a powerful step into the world of HDL simulation with capacity and performance designed for verifying Xilinx

XC9500 CPLD series and Spartan FPGA series of programmable logic devices, as well as for the lower-density versions of the XC4000 and Virtex FPGAs. ModelSim XE supports behavioral, RTL, and gate-level simulation of Xilinx cell libraries and is available in both VHDL and Verilog versions.

Conclusion

The ModelSim family of products also includes the ModelSim Special Edition (SE) and Elite Edition (EE) for cutting-edge ASIC and high-end FPGA development on both PCs and Unix workstations. Also available is ModelSim Personal Edition (PE), the price/performance leader for PC-based FPGA designs. ModelSim SE, EE, and PE are available for purchase through MTI and its distributors (see www.model.com for more details).

The industry's most friendly upgrade program guarantees that you can begin using ModelSim XE today, and leverage your investment tomorrow for an upgrade to the more flexible SE, EE, or PE products as your verification needs change. ⚡

For more information on ModelSim XE Starter and ModelSim XE see: www.xilinx.com/products/software/mxe.htm

CPLD ChipViewer - Graphical Design Control Made Simple

An interactive way to view and control your logic design routing within a Xilinx CPLD.

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

ChipViewer™ is a powerful tool that allows you to see how your logic is implemented in a Xilinx CPLD. Its usefulness is realized at both the start and end of your design implementation.

In the Pre-fit mode, ChipViewer acts a graphical constraints entry tool, and allows you to pre-assign the pinout for your device. Your design I/O is displayed in a hierarchical form showing inputs, outputs, and global resources. With a simple mouse click you can assign I/O pinouts or prohibit pin usage. Package "thumbnails," which give you a top and bottom view, are also available to assist in intelligent logic and I/O placement. Macrocell register initial states, the power mode, and output slew rate values are also easily specified. Saving the modifications generates a user constraints file (*.ucf) which is used by the Xilinx fitting tool during implementation.

In the Post-fit mode, ChipViewer reads in the implemented design file and displays a representation of the logic placement within the device. Figure 1 illustrates the graphical features showing the color-coded interconnections that exist within the part. Clicking on an active macrocell shows either the inputs to the macrocell, the outputs from the

macrocell, or both simultaneously. Double clicking on this same macrocell brings up an additional window that provides detailed information, such as the implemented equations along with a schematic representation of the logic that is contained in that macrocell.

Conclusion

ChipViewer allows you to intelligently interact with the logic in your Xilinx XC9500 CPLD designs, saving you time and making your job easier. ∑
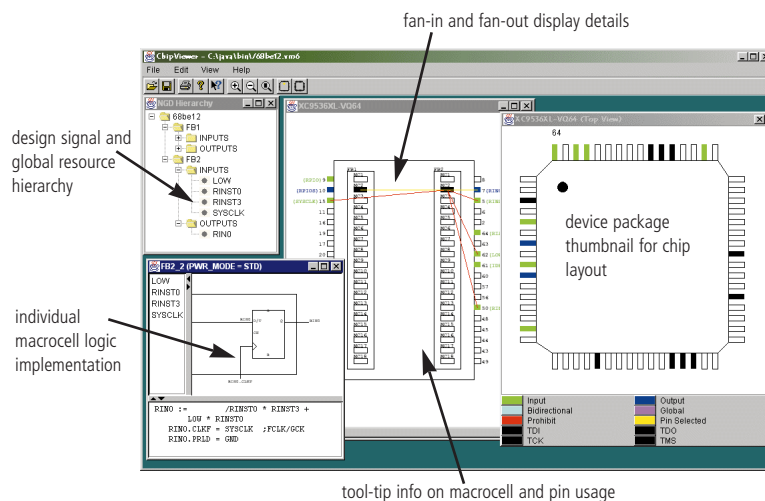
Figure 1 - CPLD Chip Viewer Capabilities

For more information see www.xilinx.com.

48

Xilinx at Work in
Digital Modems

How Xilinx high-volume programmable devices can be used to implement complex system-level glue logic.

by Robert Bielby, Strategic Applications, Xilinx, robert.bielby@xilinx.com

Internet users are continuing to demand higher bandwidth access to the net. In the face of this demand analog modem technology has hit the end of the road with the 56K generation of devices. Increasing Internet access bandwidth requires a migration to digital modem technologies.

While there are numerous approaches being proposed, including wireless and satellite technologies, it is generally accepted that the bulk of users will get their high bandwidth Internet access using either cable modem or Digital Subscriber Line (DSL) modem technologies. Cable modems offer data rates up to 10 Mbps. DSL technologies offer a variety of access rates up to 6 Mbps.

Cable Modems

Cable modems use the same coaxial cable that is used to deliver cable TV service to provide a connection to the Internet. Digital data is RF modulated and transmitted on 6-MHz channels reserved for data services.

In spite of the higher bandwidth that cable modems appear to offer over DSL technologies, the multidrop configuration of cable requires that the bandwidth be shared among users in a neighborhood. This means that all of the homes in a neighborhood are effectively wired together and all see the same data. Unless data is encrypted before it is transmitted, it is vulnerable to interception by other users.

DSL

DSL technologies use the same twisted pair copper wiring that is used for voice telephone service to deliver high speed digital data. In fact the same line that is used for DSL service can be used for voice service using a standard analog phone. Due to standardization issues, DSL services have gotten off to a slow start. However, it is expected that DSL services will be rapidly rolled out over the next few years.

DSL Technologies

Driving high-speed data down copper pairs from a central office to a subscriber's home requires sophisticated digital signal processing technology. The problem is compounded by the variances in line length, cross talk, wire gauge and other factors. In response to this the various DSL technologies that have appeared make different tradeoffs between data rate, the distance that data can be driven, and complexity of the line-coding scheme used (Table 1).

Fortunately the key players have agreed on a common ADSL technology described in ANSI Standard T1.413, and based on the Discrete Multi Tone (DMT) coding scheme. G.992.1 is the original full rate, 6.1 Mbps, specification. The problem with this standard is that it requires the installation of a "splitter" at the customer premises to support both voice and data over the same line. The G.992.2 (formerly known as G.Lite) was developed to eliminate the need for a

Table 1 DSL Technology Comparison

Technology	Pairs Required	Upstream Data Rate	Downstream Data Rate	Reach
HDSL	2	2 Mbps	2 Mbps	4 km
HDSL2	1	2 Mbps	2 Mbps	4 km
SDSL	1	768 Kbps	768 kbps	4 km
ADSL	1	up to 768 Kbps	up to 6.1 mbps	4 km
VDSL	1	6 or 13 Mbps	13, 26, 52 Mbps	1.5 km
IDSL	1	144 Kbps	144 Kbps	11 km

splitter at the customer end of the line. The tradeoff for the simplified lower cost installation is a reduced data rate, 1.5 Mbps.

Digital Modems and SOHO Routers

A modem has two interfaces: a Wide Area Network (WAN) interface, connected to the phone line, and the host interface, connected to a computer. The function of the modem is to modulate and demodulate (hence the term modem) data for transmission across the WAN interface.

The explosive growth of Local Area Networking (LAN) has created a class of devices targeted at Small Office/Home Office (SOHO) applications that combine a modem with an Access Router. An access router is used to determine whether traffic on a local LAN segment needs to be forwarded to the WAN. SOHO routers typically look like external modems with an Ethernet connection rather than a host connection.

The advantage of a SOHO router over a standard modem is that it lets multiple users
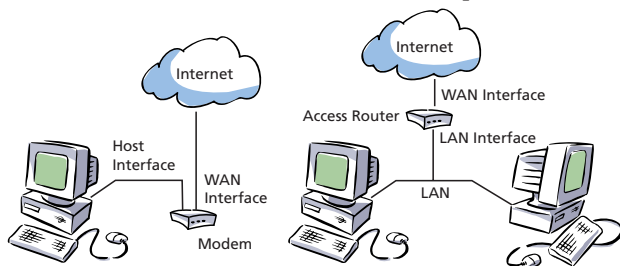
Figure 1 - Digital Modems and SOHO Routers

share a single Internet connection. The increasing interest in home LAN technology is expected to increase the demand for this class of product.

Digital Modem Architecture

Whether the product is a cable modem, DSL modem, or SOHO router, all of these digital modem products share the common functional blocks shown in Figure 2.
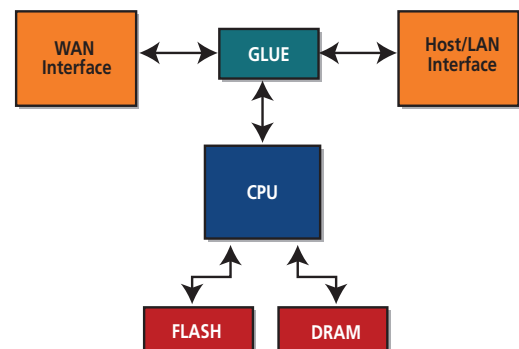
Figure 2 - Digital Modem Architecture

The functional blocks that make up the system are:

- A WAN interface containing the modem functions.

- A CPU complex consisting of the CPU plus RAM and ROM, responsible for configuring and managing the system.

- A Host Interface used to connect a modem to the host computer.

- Or a LAN Interface used to connect a SOHO router to a LAN.

Each of these blocks is typically implemented by a small number of Application Specific Standard Products (ASSPs). In most cases there are mismatches between the ASSPs used to implement each of these blocks. Xilinx FPGAs and CPLDs provide the solution for the system-level glue logic needed to interface these blocks.

Digital Modem WAN Interfaces

WAN interfaces are implemented with specialized signal processing ASSPs, each of which has been targeted at a specific application. Functions that are included in these interfaces include:

- Data encoding and decoding.

- Clock recovery - for encoding and extraction of clock information with the data.

- Adaptive equalization - to deal with line reflections.

- Error Detection and Correction - to deal with data errors.

- Transmission Convergence - to provide framing and ATM cell delineation.

The current price structure for digital modems makes it impractical for these functions to be implemented in current FPGA technology.

LAN Interfaces

LAN interfaces are used on the local side of a SOHO router. This interface gives all users on the LAN access to the WAN connection.

The most popular interface for this application is Ethernet; usually the 10 Mbps

twisted pair version (10-BASET). You can also expect to see products being introduced that support the new Phone Networking Alliance (PNA) version of Ethernet. The PNA technology supports 1 or 10 Mbps Ethernet networking over existing phone wiring. Better yet, the technology supports the simultaneous use of the same wiring for phone service.

Another network interface that has been used for first generation DSL modems is 25 Mbps ATM. Ethernet or USB will likely replace this interface in next generation products.

LAN interface ASSPs are targeted at the PC adapter card market. As a result virtually all include a PCI host interface. In addition software support comes in the form of drivers for PC operating systems.

Host Interfaces

Host interfaces are used on the local side of a modem, and connect the modem to a PC, server, or other networking equipment. For an internal modem this interface is the I/O bus of the computer, typically ISA or PCI.

In the past, the most popular interface for external modems has been RS-232. Unfortunately this interface is not fast enough to support the data rates provided by digital modems. As a result manufacturers of DSL and cable modems have had to move to other interfaces.

The most popular choice for new designs has been Universal Serial Bus (USB). A key advantage of this interface is that USB has been incorporated into PC core logic for over a year and as a result is included as a standard feature in all new PCs. The downside to USB is that

while the raw data rate is 12 Mbps, most vendors have not been able to get more than 2 to 3 Mbps from existing implementations. While not adequate for full-rate DSL and cable modem applications, this is not a limitation for the 1.5 Mbps supported by G.Lite DSL which is expected to make up the bulk of DSL modem shipments. In addition, USB 2 will support data rates in excess of 120 Mbps.

DSL Modem Add-In Card

An internal modem is still the most cost-effective solution. When using an off-the-shelf ADSL ASSP, such as the Dynamite chipset from Alcatel, there is a need to connect the Utopia interface of the chipset to the PCI bus within the PC. Although the ADSL chipset contains a significant amount of functionality, there is still the need to pack and unpack data into data blocks called cells. This process is commonly referred to as segmentation and reassembly.

In the past this has been accomplished through the use of an ATM device called a segmentation and re-assembly controller (SAR). These devices are relatively expensive. One of the more cost-effective versions available costs $20 in volume and requires a 32-bit wide pool of SRAM to do its job.
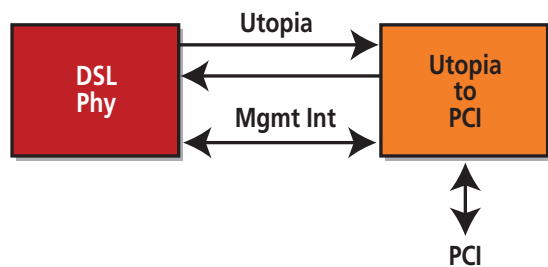
Figure 3 - DSL Modem PCI Card

A significant amount of the complexity and resulting cost of this interface can be eliminated by transferring the SAR functions to the host's CPU and implementing only bus interface and DMA functions in the Utopia to PCI interface. Offloading the SAR function to the CPU allows the interface to be implemented in a Spartan device for less than $10.

DSL Modem with USB

A DSL modem that provides a USB interface is attractive since it not only eliminates the need for users to open their systems but also provides a means of supporting non-PC systems such as the popular iMac.
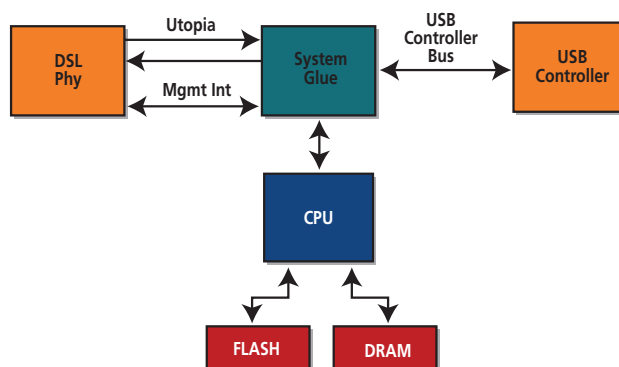
Figure 4 - DSL Modem with USB Host Interface

The problem for the designer of such a product is that it means gluing together several ASSPs that were not designed to interconnect directly. The system glue must interface the Utopia bus that transfers ATM cells to and from the DSL chipset, the proprietary micro controller interface that is provided by the USB controller and the micro controller itself. In addition to just connecting the pins the glue logic, it also needs to implement DMA functionality so that the

micro controller is not overwhelmed with transferring the data via software.

There is also this issue: the protocol between the modem and the host system across USB has not been standardized. This means that each modem will require driver support from the manufacturer. This is not the case with most analog modems, which can use a driver that comes with Windows. Therefore the ability to update the design in the field to be compatible with emerging standards is a valuable feature to vendors trying to get products to market quickly.

Both the glue logic complexity and the lack of standardization make this type product an ideal candidate for FPGA-based glue logic. The cost sensitive nature of this high volume market makes Spartan FPGAs the ideal solution.

DSL SOHO Router

Designers of DSL SOHO routers also face the
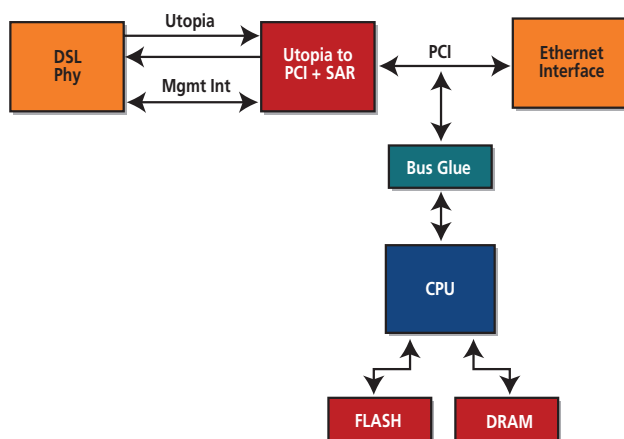
Figure 5 - DSL SOHO Router

task of gluing together a system from ASSPs with differing interfaces. One of the features that differentiates a router from a modem, the interaction of the CPU with each packet that passes through it, also means that a higher performance CPU is needed and the interface between the CPU and the network interfaces must be more efficient.

Because the LAN interface usually includes an Ethernet interface, a natural approach is to glue the other two blocks to PCI. In the case of the DSL chipset this usually means interfacing Utopia to PCI, and as in our DSL add-in card example, a Spartan device provides a very low cost means of accomplishing this.

Many embedded RISC controllers now come with PCI interfaces built in. If you have chosen a CPU that doesn't, then the same Spartan device can implement a CPU host bridge at a lower cost than off-the-shelf devices designed for that purpose. The Spartan FPGA can also be used to implement routing-specific functions such as header parsing, IP checksum calculation, and buffer management to leverage CPU MIPs more effectively.

Conclusion

Designers of digital modem products are faced with the task of interfacing a variety of devices with incompatible interfaces. Xilinx high volume FPGA and CPLD technologies provide you with cost effective solutions that retain the traditional PLD time to market advantage. ✗

53

Two Simple Solutions
for *Tricky Problems*

Here are two solutions that can help you create trouble-free designs.

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

How to Eliminate False Triggering

As FPGA flip-flops become faster, they can respond to very short clock pulses and undesired glitches from clock reflections and ground bounce. Here's a simple solution.

Fast clock transitions often lead to reflections on long printed circuit clock lines, and such reflections can result in overshoot, undershoot, and ringing. This can lead to multiple crossings of the flip-flop input thresholds and thus false triggering. There are well-known methods to keep reflections under control, but they require an understanding of high-frequency, transmission-line effects; they may also be difficult and expensive to implement, and impossible to retrofit.

Systems with multiple un-correlated clocks are especially vulnerable, even when the clock edges are noise free. Ground bounce caused by one clock can occur during the transition of another clock, especially when this transition is slow, and can cause the apparently monotonic transition to be interpreted as multiple clock transitions inside the chip. In the past, FPGAs were slow enough to be forgiving, so that many high-frequency problems could be ignored. Now, double pulses separated by one or a few nanoseconds, as shown in Figure 1, can lead to double triggering and result in system failure.
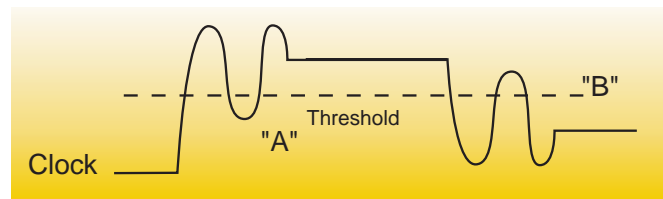
Figure 1- Reflections Causing Clock Noise

Here are two effective remedies against such problems. The description assumes rising-edge triggering, but can easily be modified for falling-edge triggering.

(A) Double clocking on the rising edge – this means that a reflection caused the internal clock signal to go Low-High-Low-High (L-H-L-H) instead of a simple Low-High (L-H) transition. The second L-H transition might clock the flip-flops again, if they are fast enough to respond to two clock edges that are only a few nanoseconds apart.

The simple solution is to give all affected flip-flops sufficient delay in front of the D-inputs, so they cannot change in the short time between the two rising clock edges, as shown in Figure 2. Use redundant LUTs or additional routing to slow the few flip-flops that are sensitive; typically they are the least-significant bits of a counter.
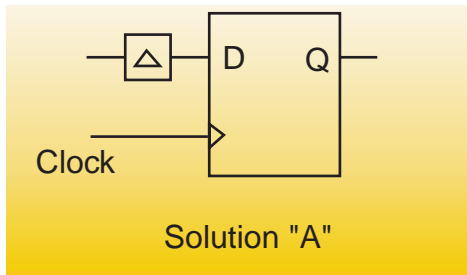
Figure 2 - Delaying the D-input

(B) Clocking on the wrong (falling) edge – this is always caused by a H-L-H-L sequence of clock transitions within a few nanoseconds. (No flip-flop can possibly be affected by the wrong edge. On the other hand, no flip-flop will ever ignore the edge it is supposed to trigger on.)

This false triggering which seems to occur on the wrong edge in the middle of the clock period cannot be cured by slowing down the flip-flop inputs. Instead, a slightly delayed and inverted version of the incoming clock signal must be used as Clock Enable to the flip-flops, as shown in Figure 3. CE will be active High before and during the legitimate rising clock edge, but will be inactive (Low) before and during the unintended clock glitch, caused by a reflection of the falling clock edge.
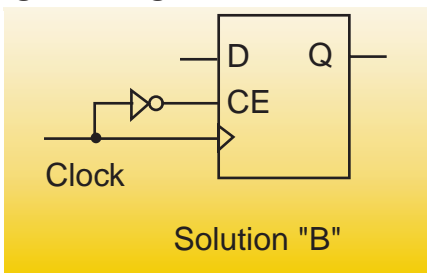
Figure 3 - Inverting the Clock

These two problems (on the rising and the falling edge), although caused by the same phenomenon, require two very different solutions. Luckily, both solutions are simple, and can be used together.

Clock reflections are best avoided by proper printed circuit board design, but accidents do happen, and it is nice to have a simple solution, especially one that is so easily applied in an FPGA without any harmful side-effects. These techniques can be used to retrofit older designs that fail when faster devices (which are more susceptible to noise) are used.

Proper Use of Mode-Pin Pull-Up Resistors

XC4000-series mode pins have an internal pull-up resistor that guarantees a logic High level on an unconnected mode pin during power-up. After configuration, the default bitstream turns these resistors off. Subsequent re-configurations can, therefore, fail.

For all modes except Master Serial (where all three mode pins are being pulled Low), we recommend either external pull-up resistors to guarantee a High level upon reconfiguration, or an explicit change of the configuration bitstream.

Here's how to change the configuration bitstream:

- From command-line: use the following options upon invoking bitgen:
  bitgen -g M0Pin:Pullup -g M1Pin:Pullup -g M2Pin:Pullup <design_name>.ncd

- From the Design Manager GUI: under the Configuration tab of the Edit Configuration Options Window, select the PullUp radio buttons for the M0, M1, and M2 Configuration pins. Σ

55

Trade Shows

Xilinx **Trade Show** Programs

See our International Trade Show schedule and plan to attend.

by Darby Mason-Merchant, Trade Show Manager, darby@xilinx.com

For Xilinx, 1999 has been an incredible year with a brand new custom booth and nearly twice the number of shows in North America as our 1998 season. With this new presence, and so many new opportunities to reach our customers, 1999 was a year of experimentation for Xilinx. From the 7th FPGA/PLD Conference in Japan, to the EDA&T Expo in Bejing, China, we brought our new technology to all corners of the world.

Our most successful experiment was "Club Xilinx" at the 36th Design Automation Conference (DAC) in June. Xilinx brought webcasting technology to the show floor, with live broadcasts of speeches by CEOs such as Aart de Geus of Synopsys, John Chambers of Cisco, Scott McNeally of Sun, and our own Wim Roelandts.

Club Xilinx featured our latest technology including Xilinx Online, our Internet Team Design capabilities, and our latest 2.1i Foundation Series and Alliance Series Software. With our jazz pianist and singing presenters, it was clear that Xilinx has your "hot linx to cool technology".

Xilinx also participated in communications and internet-related shows such as the Embedded Internet Workshop and the 1999 National Communications Forum—InfoVision— where we received the 1999 InfoVision Award for our IRL methodology.

Xilinx In the New Millenium

In the year 2000, Xilinx will be everywhere. You'll see Xilinx taking a broader stance in the portable and wireless arenas with enhanced Spartan and CPLD products on display. You'll see the power of our Virtex and Virtex-E families and our IRL methodology for reconfigurability over the Internet at many more events. Look for the new IP products from Xilinx and our AllianceCORE partners at the DSP Spring Conference 2000, the PLD Days in the UK/Stockholm, and the AC Developers Conference in the Spring of 2000.

Through our worldwide trade show programs, you can get hands-on experience with industry-leading PLD technologies such as our two million-gate Virtex-E family, Xilinx Online and IRL methodology, 2.1i Foundation Series and Alliance Series Software, WebFitter™, WebPACK™, DSP cores, Real PCI 64/66 cores, and support.xilinx.com. Trade show information is listed on our WebLINX page at: www.xilinx.com/company/tradeshows.htm

Year 2000 Xilinx Trade Show Schedules

Here are our preliminary Year 2000 Xilinx Trade Show Schedules. Be sure to check our website for further updates on these and other new events.

Year 2000 North American Trade Show Schedule

January 25-26	Portable Design 2000	San Diego, CA
February 2000	FPGA Conference 2000	Monterey, CA
March 2000	SNUG 2000	San Jose, CA
April 10-12	DSP Spring Conference 2000	San Jose, CA
April 2000	FCCM Conference 2000	Napa, CA
May-Dec	Embedded Computing Shows	US & Canada
May 23-24	AC Developers Expo 2000	Santa Clara, CA
June 5-7	37th Design Automation Conference	Los Angeles, CA
June 21-23	WITI Technology Summit 2000	Santa Clara, CA
July 2000	NSREC 2000	Washington, DC
September 6-7	Embedded Internet Conference 2000	San Jose, CA
September 2000	MAPLD 2000	Laurel, MD

Year 2000 European Trade Show Schedule

April 2000	Intertronic 2000	Paris, France
April 2000	Electronic Design Solutions	NEC Birmingham, UK
May 2000	Advanced FPGA and PLD Day	Sandown, UK
May 2000	Advanced FPGA and PLD Day	Stockholm, Sweden
May 2000	Embedded Systems Show	Olympia, UK
June 2000	Embtech World 2000	Paris, France
September 2000	IBC 2000	Amsterdam, Netherlands
November 2000	IP Europe 2000	Edinburgh, Scotland
November 2000	Embedded Systems Conference	Maastricht, Netherlands

Year 2000 South East Asian Trade Show Schedule

March 23-24	IIC 2000	Guanzhou, China
March 27-28	IIC 2000	Shanghai, China
March 20-21	IIC 2000	Beijing, China
October 2000	EDA&T	Hsinchu,Taiwan
October 2000	EDA&T	Beijing, China

Year 2000 Japanese Trade Show Schedule

January 27-28	EDA Techno Fair 2000	Tokyo, Japan
November 2000	Micon System Tool Fair 2000	Tokyo, Japan

Σ

How **Programmable Logic** Provides More Options for Consumers

The market has shifted dramatically in the last ninety years. Today the buzzword is "mass customization"–unique products at a price comparable to

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

It's been over ninety years since Henry Ford first introduced the Model-T. By 1918, half of all the cars in the United States were Model-Ts. Ford's genius was in the area of precision manufacturing, standardization, the use of interchangeable parts, and division of labor on the assembly line. These factors enabled the mass production of automobiles at an affordable price. Of course, the vehicles that Ford produced were all black and exact carbon copies of one another.

As consumers continue to demand more and more variety, and more options for less, designers are facing the formidable challenge of keeping customers satisfied. To illustrate just how broad a phenomenon this product proliferation is, the model and option combinations for Ford's Escort brand alone result in more than 45,000 possible car configurations.

1999 Ford Escort Options*	
Models	3
Packages	3
Powertrain	2
Exterior	10
Seats	2
Interior Color	3
Audio	3
Tires	2
Options	7

Closer to home, Dell has leveraged mass customization and has continued to deliver top earnings to investors in the highly competitive PC market. Dell has compressed their order-to-manufacture cycle to as little as four hours**.

Programmable Logic is the Key

So what are the rules for success in the electronics mass customization environment? Through concurrent modular engineering, product design cycles may be compressed. At the same time, greater product varieties may be offered as modular designs

*Source: Ford Motor Company, 1999 Ford Escort
**Austin American-Statesman (TX), Sept 28, 1998

are leveraged across a broad range or products. And through design reuse, parts and component count reduction can be achieved through intelligent design and standardization.

Programmable logic devices are uniquely suited to benefit design engineers in meeting demands for a compressed design cycle and for flexible product customization. The intrinsic adaptability of PLDs easily accommodates design customization in many applications:

- Customization for geographic markets and standards.
- Differentiation on a cost or performance basis (such as video game upgrades versus new game add in cards).
- Adaptation to specific customer or key contract requirements.
- Incorporation of new intellectual property, or cores, to meet specific requirements.

As electronic devices become more and more pervasive among consumers, our industry faces the challenge of being everything to everyone. Programmable logic technology offers a compelling design alternative that will help you get your products to market much sooner and keep your products in the market longer.

Conclusion

Although it may appear to be a daunting task, the advantages of product differentiation are enormous in terms of higher customer satisfaction, customer retention and loyalty, and higher profitability through shortening the supply chain. Mass customization, using programmable logic technology, provides an innovative, competitive strategy that will make your product more useful and more profitable. Σ

Software Selection Guide

Features Included	Alliance Series		Foundation Series			
	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, Synopys, 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 Ed tor, 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): XC4000E/XL (Up to XC4010E/XL) Spartan and SpartanXL (All) XC3000A, XC3000L, XC3100A, XC3100L XCS200 (Up to XCS210)	✓		✓		✓	
FPGA (Unlimited Device Support): Virtex XC4000E/X (Ail) Spartan and SpartanXL (All) XC3x00A/L (All) XCS200 (All)		✓		✓		✓
Xchecker Cable (Workstation Only)	✓	✓				
JTAG Cable (PC Only)	✓	✓	✓	✓	✓	✓

SUPPORT.XILINX.COM

Power tools for the designing engineer.

Device Selection Guide

XC4000XV Family - 2.5 Volt   .25µm Five Layer Metal Process

	Usable Gates (K)	Logic Gates	System Gates (see note 1)	Logic Cells (see note 2)	Total CLBs	Total Flip-Flops	Max. I/O Package	Max. RAM Bits	Config. Memory (K Bits)	XC1700 Serial PROM Req.
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
XCS05/XL	2-5	3K	5K	238	100	360	77	3.2K	54	XC17S05 (XL)
XCS10/XL	3-10	5K	10K	466	196	616	112	6.3K	95	XC17S10 (XL)
XCS20/XL	7-20	10K	20K	950	400	1,120	160	12.8K	179	XC17S20 (XL)
XCS30/XL	10-30	13K	30K	1,368	576	1,536	192	18.4K	249	XC17S30 (XL)
XCS40/XL	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 CLBs	Total Flip-Flops	Max. I/O Package	Max. RAM Bits (k)	Config. Memory (K Bits)	XC/XQ PROM Req.
XC4000EX Family - 5 Volt										.5µm Triple Layer Metal Process
XQ4005E	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
XC4000XL Family - 3.3 Volt										Advanced .35µm 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
XC4000XL Family - 3.3 Volt										.25µm 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	

Serial PROM Package Options

	Density	PD8	SO8	VO8	PC20	SO20	VQ44	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	
XC17S05	54K	Y		Y					Y
XC17S05XL	55K	Y		Y				Y	
XC17S10	95K	Y		Y					Y
XC17S10XL	96K	Y		Y				Y	
XC17S20	178K	Y		Y					Y
XC17S20XL	179K	Y		Y				Y	
XC17S30	248K	Y		Y					Y
XC17S30XL	249K	Y		Y				Y	
XC17S40	329K	Y				Y			Y
XC17S40XL	331K	Y				Y		Y	

Speed Grade Options

	Density	DD8	CC44	S020
XQ17256D	256K	Y	-	-
XQ1701L	1M	-	Y	Y
XQ1704L	4M	-	Y	Y

	Macrocells	Max. I/O	Pin-to-Pin Delay (ns)
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			
XC9536XV	36	36	3.5
XC9572XV	72	72	4
XC95144XV	144	117	4
XC95288XV	288	192	5

Speed Grade Options

	Slowest	Fastest
XC4000XV	-09	-07
Virtex	-4	-6
SpartanXL	-4	-5
XQ4005X	-4	-4
XQ4000XL	-3	-3

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

Ordering Information for FPGAs / CPLDs

Example:

XC4036XLA - 08 PQ 160 C

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

Device Selection Guide

FPGA / CPLD Package Options and User I/O

Package	XC40110XV	XC40150XV	XC40200XV	XC40250XV	XCS05/XL	XCS10/XL	XCS20/XL	XCS30/XL	XCS40/XL	XC9536XL/XV	XC9572XL/XV	XC95144XL/XV	XC95288XL/XV	XCV50	XCV100	XCV150	XCV200	XCV300	XCV400	XCV600	XCV800	XCV1000
(gates)	448	448	448	448	80	112	160	192	224	36	72	117	192	192	240	288	336	384	480	576	672	768
PLCC Packages																						
										34	34											
					61	61																
CSP Packages																						
										36	38											
						112	113					117		98	98							
								192	224													
PQFP / HQFP Packages																						
							160	169	169				168									
	193	193						192	192						166	166	166	166	166	166	166	166
VQFP Packages																						
										36	52											
					77	77	77	77														
TQFP / HTFP Packages																						
						112	113	113			72	81		98	98							
												117	117									
CBFP Packages																						
BGA Packages																						
							192	205						180	180	180	180					
	289	289											192		260	260	260					
	352	352	352	352													316	316	316	316		
	432	432	432	432														404	404	404	404	
PGA Packages																						
		448		448																		
FinePitch BGA																						
														176	176	176	176					
																	260	284	312			
																				404	444	444
																				512	512	512

Available Now   SpartanXL available Q4

Available Q199

Go to the Xilinx web site to get the latest product information:
http://www.xilinx.com/products/products.htm

Virtex Series FPGA
Competitive Cross Reference

Virtex™-E – 1.8V Virtex Series Key Features: Density/Performance, SelectRAM+™, SelectI/O™, DLLs, SelectLINK™

Devices	Package	Package I/O	Logic Gates	Typical Gate Range	Flip-Flops*	Output Drive (mA)	Maximum RAM Bits	Maximum Device I/O	Logic Cells
XCV50E	CS144 PQ240 FG256	94 158 176	21K	20K-56K	2064	2/24	80K	176	1728
EP20K60E							32K	204	2560
XCV100E	CS144 PQ240 FG256	94 158 176	32K	30K-95K	2988	2/24	80K	176	2700
EP20K100E							53K	252	4160
XCV200E	CS144 PQ240 FG256 FG456	94 158 176 284	64K	60K-200K	5556	2/24	128K	284	5292
EP20K160E							81K	316	6400
XCV300E	PQ240 FG256 BG432 FG456	158 176 316 312	83K	80K-400K	7116	2/24	224K	316	6912
EP20K200E							106K	382	8320
XCV400E	PQ240 BG432 FG676	158 316 404	130K	130K-560K	10812	2/24	310K	404	10800
EP20K300E							147K	408	11520
XCV600E	HQ240 BG432 FG676 FG680 FG900	158 316 444 512 512	187K	185K-980K	15360	2/24	504K	512	15552
EP20K400E							212K	502	16640
EP20K600E							311K	624	24320
XCV1000E	HQ240 BG560 BG680 FG860 FG900 FG1156 BG560	158 404 512 660 660 660 404	332K	330K-1.5M	26556	2/24	768K	660	27648
XCV1600E	BG680 FG860 FG900 FG1156	512 660 700 724	420K	420K-2M	33276	2/24	1062K	724	34992
EP20K1000E							327K	716	38400
XCV2000E	BG560 FG680 FG860 FG1156	404 512 660 804	518K	520K-2.5M	40812	2/24	1240K	804	43200
EP20K1500E							456K	858	54720
XCV2600E	BG560 FG680 FG860 FG1156 BG560	404 512 660 804 404	685K	600K-3.2M	53196	2/24	1525K	804	57132
XCV3200E	FG680 FG860 FG1156	512 660 804	876K	800K-4M	67308	2/24	1846K	804	73008

Virtex™ – 2.5V Virtex Series Key Features: Density/Performance, SelectRAM+™, SelectI/O™, DLLs

Devices	Package	Package I/O	Logic Gates	Typical Gate Range	Flip-Flops*	Output Drive (mA)	Maximum RAM Bits	Maximum Device I/O	Logic Cells
XCV50	CS144 PQ240 TQ144 BG256 FG256	94 166 98 180 176	21K	34K-58K	1536	2/24	56K	180	1728
EPF10K30E							24K	220	1728
XCV100	CS144 PQ240 TQ144 BG256 FG256	94 166 98 180 176	32K	72K-109K	2400	2/24	78K	196	2700
EPF10K50E							40K	254	2880
XCV150	PQ240 BG256 BG352 FG256 FG456	166 180 260 176 260	47K	93K-165K	3456	2/24	102K	260	3888
EP20K100							52K	250	4160
EPF10K100B/E							25K	338	4992
XCV200	PQ240 BG256 BG352 FG256 FG456	166 180 260 176 284	64K	146K-237K	4704	2/24	130K	284	5292
EPF10K130E							65K	413	6656
XCV300	PQ240 BG352 BG342 FG456	166 260 316 312	83K	176K-323K	6144	2/24	160K	324	6912
EP20K200							104K	320	8320
EPF10K200E							98K	470	9984
XCV400	HQ240 BG432 BG560	166 316 404	130K	282K-468K	9600	2/24	230K	404	10800
XCV600	FG676 HQ240 BG432 BG560 FG676 FG680	404 166 316 404 444 512	187K	365K-661K	13824	2/24	312K	512	15552
EP20K400							208K	496	16640
XCV800	HQ240 BG432 PG560 FG676 FG680	166 316 404 444 512	254K	511K-888K	18816	2/24	406K	556	21168
XCV1000	BG560 FG680	404 512	332K	622K-1,124K	24576	2/24	512K	660	27648

* I/O Flip-Flops not counted.

High End FPGA Products & Benefits Snapshot	
Virtex-E • System-level integrated FPGA up to 3.2 million gates and 311MHz • Differential signaling: LVPECL, LVDS, Bus LVDS 36 I/O pairs at 622 MHZ • Up to 832K bits of True Dual-Port(™) Block RAM • Up to 1014K bits of Distributed RAM • Delivers Terabit memory bandwidth • 8 DLLs. Supports Double Data Rate Memory and backplane applications • 1.8V operation	**Virtex** • System-level integrated FPGA up to 1M gates and 200MHz • Up to 128K bits True Dual-Port RAM • Up to 384K bits Distributed RAM • 4 DLLs. Supports Double Data Rate Memory and backplane applications • Optimized for synthesis and IP-based design methodology • 2.5V operation
Total Solution	
Software & Cores • 50% faster compile times, increased performance, Internet-based design features. • Array of fully verified, predictable and parameterizable cores.	

Competitive Overview

Virtex Series FPGA
Competitive Cross Reference

Competitive Feature Comparison

Device	Timing Generation	5 Volt Tolerant I/O	LVTTL	LVCMOS	PCI33	PCI66	GTL/GTL+	SSTL2	SSTL3	AGP	HSTL Class I	HSTL Class III	HSTL Class IV	CTT	LVDS	Bus LVCS	LVPECL
						I/O Standards Supported									Differential Signaling		
Virtex-E	8 DLL	✓*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Virtex	4 DLL	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
APEX 20K	1 PLL		✓	✓	✓												
APEX 20KE	4 PLL		✓	✓	✓	✓	✓	✓	✓	✓	✓				✓		

* Requires 100 ohm external resistor

Competitive Feature Comparison

Device	Block	CAM	Distributed	SRAM	SSRAM	SGRAM	SDRAM	DDR	ZBT
	Internal			External					
			Memory Access						
Virtex-E	✓	**	✓	✓	✓	✓	✓	✓	✓
Virtex	✓	**	✓	✓	✓	✓	✓	✓	✓
APEX 20K	✓								
APEX 20KE	✓	✓		✓	✓	✓	✓		

** Supported with existing memory resources (see Application Notes on SPW)

Virtex-E is the Superior Programmable Solution

Superior Virtex-E LVDS Support

- Altera's APEX E LVDS only provides for 32 I/Os.
  Virtex-E supports LVDS, Bus LVDS, and LVPECL for up to 344 I/O pairs. Virtex-E Select I/O+ technology offers complete flexibility of choosing which standard is used on any given pin.

- Altera claims 622 MHz, but only for 16 inputs and 16 outputs, which is 10 Gbps.
  Virtex-E offers 36 pairs at 622 Mbps and delivers significantly higher LVDS Bandwidth by offering 344 I/O pairs at 311MHz per pair, which is a 107 Gbps (Gigabits per second) data rate.

- APEX E does not support backplane applications with Bus LVDS.
  Virtex-E offers Bus LVDS for all densities and speeds.

- APEX E does not support LVPECL
  Virtex-E offers LVDS and LVPECL up to 622MHz for all densities. LVPECL is the industry standard for transmission of on-board clocks at greater than 100MHz.

Superior Virtex Clock Management

- Altera's analog PLLs do not support precise clock management.
  Designers can achieve an exact 50/50 duty cycle using Virtex DLLs, which is crucial for Double Data Rate (DDR) applications. APEX PLLs can not offer this precision

Superior Virtex-E logic and RAM offering

- Virtex-E offers 3.2 million gate densities in Q2/00 and up to 832K bits of Block RAM, plus distributed RAM. Compare that to Altera's 2.5 million gates in the middle of Y2000 with only 456K Block RAM bits.

- APEX provides only block RAM and an interface to external RAM. Virtex offers more RAM and more types of RAM.

- Virtex interfaces to the industry's highest performance memories, including:
  - 200 MHz ZBT SRAM
  - 266 MHz Double Data Rate (DDR) memory

- Altera claims the dedicated CAM in APEX E, but the Virtex and Virtex-E offers CAM built from either distributed RAM or Block RAM at equivalent densities to those offered in APEX E. Most designs will require higher CAM densities than those offered with on-board dedicated CAM from Altera. In those cases, Virtex-E offers a faster external memory interface than APEX's.

And the beat goes on.

VIRTEX™-E

www.xilinx.com

XILINX®

2100 Logic Drive
San Jose, CA 95124-3450

64