Xilinx Application Note Summaries

Application Notes

XAPP004 Loadable Binary Counters

The design strategies for loadable and non-loadable binary counters are significantly different. This application note discusses the differences, and describes the design of a loadable binary counter. Up, down and up/down counters are described, with lengths of 16 and 32 bits. Design files are available for all six versions.

XAPP005 Register-Based FIFO

While XC3000 series FPGA devices do not provide RAM, it is possible to construct small register-based FIFOs. A basic synchronous FIFO requires one CLB for each two bits of FIFO capacity, plus one CLB for each word in the FIFO. Optional asynchronous input and output circuits are provided. Design files are available for two implementations of this design. The fastest of the two implementations uses a constraints file to achieve better placement.

XAPP007 Boundary Scan Emulator for XC3000

CLBs are used to emulate IEEE 1149.1 Boundary Scan. The FPGA device is configured to test the board interconnect, and then reconfigured for operation.

XAPP008 Complex Digital Waveform Generator

Complex digital waveforms are generated without the need for complex decoding. Instead, fast loadable counters are used to time individual High and Low periods.

XAPP009 Harmonic Frequency Synthesizer and FSK Modulator

Harmonic Frequency Synthesizer:
Uses an accumulator technique to generate frequencies that are evenly spaced harmonics of some minimum frequency. Extensive pipelining is employed to permit high clock rates.

FSK Modulator:
A modification of the Harmonic Frequency Synthesizer that automatically switches between two frequencies in accordance with an NRZ input.

XAPP010 Bus Structured Serial Input/Output Device

Simple shift registers are used to illustrate how 3-state busses may be used within an FPGA device Dedicated wide decoders are used to decode an I/O address range and enable the internal registers.

XAPP011 LCA Speed Estimation: Asking the Right Question

A simple algorithm is described for determining the depth of logic, in CLBs, that can be supported at

XAPP012 **Quadrature Phase Detector**

A simple state machine is used to adapt the output of two photo-cells to control an up/down counter. The state machine provides hysteresis for counting parts correctly, regardless of changes in direction.

XAPP013 **Using the Dedicated Carry Logic in XC4000E**

This Application Note describes the operation of the XC4000/Spartan dedicated carry logic, the standard configurations provided for its use, and how these are combined into arithmetic functions and counters.

XAPP014 **Ultra-Fast Synchronous Counters**

This fully synchronous, non-loadable, binary counter uses a traditional prescaler technique to achieve high performance. Typically, the speed of a synchronous prescaler counter is limited by the delay incurred distributing the parallel Count Enable. This design minimizes that delay by replicatin the LSB of the counter. In this way even the small longline delay is eliminated, resulting in the fastes possible synchronous counter.

XAPP015 **Using the XC4000 Readback Capability**

This Application Note describes the XC4000/Spartan Readback capability and its use. Topics include: initialization of the Readback feature, format of the configuration and Readback bitstreams, timing considerations, software support for reading back FPGA devices, and Cyclic Redundancy Check (CRC).

XAPP017 **Boundary Scan in XC4000 and XC5200 Series Devices** v3.0 (11/99)

XC4000/XC5200/Spartan FPGA devices contain boundary scan facilities that are compatible with IEEE Standard 1149.1. This Application Note describes those facilities in detail, and explains how boundary scan is incorporated into an FPGA design.

XAPP018 **Estimating the Performance of XC4000E Adders and Counters**

Using the XC4000/Spartan dedicated carry logic, the performance of adders and counters can easily be predicted. This Application Note provides formulae for estimating the performance of such adder and counters.

XAPP022 **Adders, Subtracters and Accumulators in XC3000**

This Application Note surveys the different adder techniques that are available for XC3000 designs. Examples are shown, and a speed/size comparison is made.

XAPP023 **Accelerating Loadable Counters in XC4000**

The XC4000/Spartan dedicated carry logic provides for very compact, high-performance counters. This Application Note describes a technique for increasing the performance of these counters usin minimum additional logic. Using this technique, the counters remain loadable.

XAPP024 **XC3000 Series Technical Information**

This Application Note contains additional information that may be of use when designing with the XC3000 series of FPGA devices. This information supplements the data sheets, and is provided for guidance only.

XAPP026 Multiplexers and Barrel Shifters in XC3000 Series

This Application Note provides guidance for implementing high performance multiplexers and barre shifters in XC3000 Series FPGA devices.

XAPP027 Implementing State Machines in LCA Devices

This Application Note discusses various approaches that are available for implementing state machines in FPGA devices. In particular, the one-hot-encoding scheme for medium-sized state machines is discussed.

XAPP028 Frequency/Phase Comparator for Phase Locked Loops

The phase comparator described in this Application Note permits phase-locked loops to be constructed using FPGA devices that only require an external voltage-controlled oscillator and integrating amplifier.

XAPP029 Serial Code Conversion Between BCD and Binary

Binary-to-BCD and BCD-to-binary conversions are performed between serial binary values and parallel BCD values.

XAPP030 Megabit FIFO in Two Chips: One LCA Device and One DRAM

This Application Note describes the use of an FPGA device as an address controller that permits a standard DRAM to be used as deep FIFO.

XAPP043 Improving XC4000 Design Performance

This Application Note describes XC4000 architectural features that can be exploited in high-performance designs, and software techniques that improve placement, routing and timing. It also contains information necessary for advanced design techniques, such as floor planning, locking down I/Os, and critical path optimization.

XAPP045 XC4000 Series Technical Information

This Application Note contains additional information that may be of use when designing with XC4000 Series devices. This information supplements the product descriptions and specifications, and is provided for guidance only.

XAPP051 Synchronous and Asynchronous FIFO Designs

This application note describes RAM-based FIFO designs using the dual-port RAM in XC4000 Serie devices. Synchronous designs with a common read/write clock are described, as well as asynchronous designs with independent read and write clocks. Emphasis is on the fast, efficient an reliable generation of the handshake signals FULL and EMPTY, which determine design performance.

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

Shift registers longer than eight bits can be implemented most efficiently in XC4000 or Spartan Series SelectRAM memory. Using Linear Feedback Shift Register (LFSR) counters to address the RAM makes the design even simpler. This application note describes 4- and 5-bit universal LFSR counters, very efficient RAM-based 32-bit and 100-bit shift registers, and pseudo-random sequence generators with repetition rates of thousands and even trillions of years, useful for testing and encryption purposes. The appropriate taps for maximum-length LFSR counters of up to 168 bits are listed.

XAPP053 Implementing FIFOs in XC4000 Series RAM

This Application Note demonstrates how to use the various RAM modes in XC4000 and Spartan Series logic blocks. A simple FIFO is implemented in several different ways, using combinations of level-sensitive (asynchronous) and edge-triggered (synchronous), single-port and dual-port RAM.

XAPP054 Constant Coefficient Multipliers for the XC4000E

This paper identifies two points at which constant coefficient multipliers become the optimum choic in DSP, and implements constant (k) coefficient multipliers (KCMs) in the XC4000E. It also reveals the solution to an interesting design problem which emerges.

XAPP055 Block Adaptive Filter

This application note describes a specific design for implementing a high-speed, full-precision, adaptive filter in the XC4000E/X family of FPGAs. The design may be easily modified, and demonstrates the suitability of using FPGAs in digital signal processing applications. This application note is based on a 12-bit data, 12-bit coefficient, full-precision, block adaptive filter design. This design can be modified to accommodate different data and coefficient sizes, as well a lesser precision. The application note covers how to modify the design including the trade-offs involved. The filter is engineered for use in the XC4000 Series.

XAPP056 System Design with New XC4000X I/O Features

The XC4000X FPGA family (XC4000EX, XC4000XL, XC4000XLA, XC4000XV) provides several new I/ features, including an additional latch on each input and an output multiplexer on each output. The output multiplexer can also be configured as a two-input function generator. Two different types of clock buffers allow system timing flexibility. These features are discussed, and examples show how to use them.

XAPP057 Using SelectRAM Memory in XC4000 Series FPGAs

XC4000 and Spartan Series FPGAs include SelectRAM memory, which can be configured as ROM o as single- or dual-port RAM, with edge-triggered or level-sensitive timing. This application note describes how to implement SelectRAM memory in a design: in schematic entry, using LogiBLOX synthesis, and HDL synthesis environments. Specifying timing requirements, evaluating performance, and floorplanning are also described.

XAPP058 Xilinx In-System Programming Using an Embedded Microcontroller v3.0 (01/15/01)

The Xilinx high-performance CPLD, FPGA, and configuration PROM families provide in-system programmability, reliable pin locking, and JTAG boundary-scan test capability. This powerful combination of features allows designers to make significant changes and yet keep the original device pinouts, thus, eliminating the need to re-tool PC boards. By using an embedded controller to program these CPLDs and FPGAs from an on-board RAM or EPROM, designers can easily upgrade modify, and test designs, even in the field.

XAPP059 Gate Count Capacity Metrics for FPGAs

Three metrics are defined to describe FPGA device capacity: Maximum Logic Gates, Maximum Memory Bits, and Typical Gate Range. The methodology used to determine these values is described.

XAPP060 Design Migration from XC4000 to XC5200

This Application Note reviews the differences between the XC5200 and XC4000 families, recommends approaches for converting XC4000 designs to the XC5200 architecture, and provides methodology to migrate designs easily in multiple CAE environments.

XAPP061 Design Migration from XC2000/XC3000 to XC5200

This Application Note reviews the differences between the XC5200 and XC2000/XC3000 families, recommends approaches for converting XC2000/XC3000 designs to the XC5200 architecture, and provides a methodology to migrate designs easily in multiple CAE environments.

XAPP062 Design Migration from XC4000 to XC4000E

The XC4000E is an enhanced architecture based on the XC4000 family, but offers many new features, particularly SelectRAM memory. When converting XC4000, XC4000A, XC4000D, and XC4000H designs, the XC4000E is an excellent choice. The conversion process may be as simple as downloading the same bitstream into the XC4000E device (XC4000 and XC4000D bitstreams only), or it may involve changes to the schematic or HDL code. This Application Note describes techniques that should be employed to convert from any of the XC4000, XC4000A, XC4000D, or XC4000H families to the XC4000E family.

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

The XC4000E/X and Spartan FPGA families provide distributed on-chip RAM. SelectRAM memory ca be configured as level-sensitive or edge-triggered, single-port or dual-port RAM. The edge-triggered capability simplifies system timing and provides better performance for RAM-based designs. The dual-port mode offers new capabilities and simplifies FIFO designs.

XAPP067 Using Automatic Test Equipment to Program XC9500 Devices In-System

This application note describes how to program XC9500 devices in-system, using standard Serial Vector Format (SVF) stimulus files.

XAPP068 In-System Programming Times

This application note discusses the in-system programming speed of the XC9500 devices.

XAPP069 Using the XC9500 JTAG Boundary Scan Interface

This application note explains the XC9500 boundary scan interface and demonstrates the software available for programming and testing XC9500 CPLDs. An appendix summarizes the JTAG programmer operations and overviews the additional operations supported by XC9500 CPLDs for ir system programming.

XAPP070 Using In-System Programmability in Boundary Scan Systems

This application note discusses basic design considerations for in-system programming of multiple XC9500 devices in a boundary scan chain, and shows how to design systems that contain multiple XC9500 devices as well as other IEEE 1149.1-compatible devices.

XAPP071 Using the XC9500 Timing Model

This application note describes how to use the XC9500 timing model. All XC9500 CPLDs have a uniform architecture and an identical timing model, making them very easy to use and understand. To determine specific timing details, users need only compare their paths of interest to the architectural diagrams and, using the timing model presented here, perform a simple addition of incremental time delays.

XAPP073 Designing with XC9500 CPLDs

This application note will help designers understand the XC9500 architecture and how to get the best performance from these devices.

XAPP074 Pin Preassigning with XC9500 CPLDs

This application note describes the planning required for successful pin preassigning and gives a detailed example.

XAPP076 Embedded Instrumentation Using XC9500 CPLDs

This application note shows how to build embedded test instruments into XC9500 CPLDs.

XAPP078 XC9536 ISP Demo Board

The demo board described in this application note is a tool for demonstrating the In-System Programming (ISP) capabilities of the XC9500 CPLD family.

XAPP079 Configuring Xilinx FPGAs Using an XC9500 CPLD and Parallel PROM v1.1 (07/27/00)

All Xilinx FPGA families can be configured through a serial interface. This application note describes a simple, low cost design to configure any Xilinx FPGA in a serial configuration mode using a Xilinx XC9500 CPLD and any parallel PROM.

XAPP080 Supply Voltage Migration, 5V to 3.3V

Mixed voltage environments could create a variety of design challenges. The new 3.3-V XC4000XL, XC4000XLA, and Spartan-XL FPGA families are immune to all power sequencing problems and car be interfaced directly with older technology 5-V devices, making them an ideal solution for many mixed voltage systems.

XAPP088 I/O Characteristics of the 'XL FPGAs

Data sheets describe I/O parameters in digital terms, providing tested and guaranteed worst case values. This application note describes XC4000XL/XLA and Spartan-XL I/O parameters in analog terms, giving the designer a better understanding of the circuit behavior. Such parameters are, however, not production tested and are, therefore, not guaranteed.

XAPP090 FPGA Configuration Guidelines

These guidelines describe the configuration process for all members of the XC3000, XC4000, XC5200, and Spartan FPGA devices and their derivatives. The average user need not understand or remember all these details, but should refer to the debugging hints when problems occur.

XAPP091 Configuring Mixed FPGA Daisy Chains

Xilinx FPGAs can be configured in a common daisy chain structure, where the lead device generates CCLK pulses and feeds serial configuration information into the next downstream device, which in turn feeds data into the next downstream device, etc. There is no limit to the number of devices in a daisy chain, and XC3000, XC4000, Spartan, and XC5200 series devices can be mixed freely with onl one constraint: the lead device must be a member of the highest order family used in the chain.

XAPP092 Configuration Issues: Power-up, Volatility, Security, Battery Back-up

This application note covers several related subjects: How does a Xilinx FPGA power up, and how does it react to power supply glitches? Is there any danger of picking up erroneous data and configuration? What can be done to maintain configuration during loss of primary power? What can be done to secure a design against illegal reverse engineering?

XAPP093 Dynamic Reconfiguration

All Xilinx SRAM-based FPGAs can be in-system configured and re-configured an unlimited number c times. This application note describes the procedures for reconfiguring the more traditional Xilinx FPGAs.

XAPP094 Metastable Recovery

Whenever a clocked flip-flop synchronizes an asynchronous input, there is a small probability that th flip-flop output will exhibit an unpredictable delay. The flip-flop can enter a symmetrically balanced transitory state, called metastable (meta = between). Xilinx evaluated the XC4000 and XC3000 serie flip-flops. The result of this evaluation shows the Xilinx flip-flop to be superior in metastable performance to many popular MSI and PLD devices.

XAPP095 Setup and Hold Times

Beware of hold time problems, because they can lead to unreliable, temperature-sensitive designs that can fail even at low clock rates.

XAPP096 Overshoot and Undershoot

When users put modern CMOS devices on PC boards, and interconnect them with unterminated lines, there are reflections, commonly called "ringing", that cause overshoots and undershoots of substantial amplitude.

XAPP097 Xilinx FPGAs: A Technical Overview for the First Time User

In the Spartan, XC3000, XC4000, and XC5200 device families, Xilinx offers several evolutionary and compatible generations of Field Programmable Gate Arrays (FPGAs). Here is a short description of their common features. This overview describes two aspects of Xilinx FPGAs:

- What logic resources are available to the user
- How the devices are programmed

XAPP098 The Low-Cost, Efficient Serial Configuration of Spartan FPGAs

This application note shows how to achieve low-cost, efficient serial configuration for Spartan FPGA designs. The approach takes advantage of unused resources in a design, thereby reducing the cost, part count, memory size, and board space associated with the serial configuration circuitry. As a result, neither processor nor PROM needs to be fully dedicated to performing configuration. Information is provided on how the idle processing time of an onboard controller can be used to load configuration data from an off-board source. As a result, it is possible to upgrade a Spartan design in the field by sending the bitstream over a network. A brief summary of Spartan slave serial configuration, its protocol and signals, lays the groundwork for a discussion on ways to reduce bitstream storage and processing requirements. A detailed example illustrates how these techniques can be put into practice. Finally, different formats for configuration data are described along with instructions for their use.

XAPP100 Choosing a Xilinx Product Family

This Application Note describes the various Xilinx product families. Differences between the familie are highlighted. The focus of the discussion is how to choose the appropriate family for a particular application. Covers the Spartan, XC3000, XC4000, XC5200, and XC9500 families.

XAPP102 XC9500 Remote Field Upgrade

This application note describes the concept and design of a remote field upgrade subsystem for an in-system programmable XC9500 CPLD. The description of the subsystem is given along with guidelines that should help with variations on it. Additional VHDL files are available for direct use of this design. Specifically, the VHDL files include a complete IRDA receiver design fitting into an XC95108 CPLD.

XAPP103 The Tagalyzer - A JTAG Boundary Scan Debug Tool

The Tagalyzer is a diagnostic tool that helps debug long JTAG boundary scan chains. It can be modified to adapt to a wide variety of different testing situations, and is made from a single XC9536 CPLD. It can be used to debug JTAG chains made up of any manufacturer's parts. The Tagalyzer ca be expanded to support arbitrarily long boundary scan chains and adapted to change its functionality as needed.

XAPP104 A Quick JTAG ISP Checklist

ISP circuitry is beneficial for fast prototype development. However, even the most robust circuitry needs minimal consideration to deliver the best in system programming results. This application brief describes a short list of considerations needed to get the best performance from your ISP designs.

XAPP105 A CPLD VHDL Introduction v2.0 (08/30/01) UPDATE!

This introduction covers the basics of VHDL as applied to CPLDs. Specifically included are those design practices that translate well to CPLDs, permitting designers to use the best features of this powerful language to extract the best performance from CPLD designs.

XAPP107 Synopsys/Xilinx High Density Design Methodology Using FPGA Compiler

This paper describes design practices to synthesize high density designs (i.e. over 100,000 gates), composed of large functional blocks, for today's larger Xilinx FPGA devices using the Synopsys FPG Compiler. The Synopsys FPGA Compiler version 1998.02, Alliance Series 1.5, and the XC4000X family were used in preparing the material for this application note.

XAPP108 Chip-Level HDL Simulation Using the Xilinx Alliance Series v2.0 (05/22/00)

This application note describes the basic flow and some of the issues to be aware of for HDL simulation with Alliance Series software. The goal of this document is to familiarize the user with some of the concepts, but should not be considered a replacement for the Xilinx or HDL simulator's documentation.

XAPP109 Hints, Tips and Tricks for using XABEL with Xilinx M1.5 Design and Implementation Tools

This application note summarizes the issues and design techniques specific to the Xilinx ABEL Interface, version M1.5.

XAPP110 XC9500 CPLD Power Sequencing

Mixed signal systems require logic parts that can operate with two power supplies. XC9500 CPLDs are designed to operate in either mixed 5V/3.3V systems or 5V only systems. To handle both conditions, care has been taken to ensure that designers need not introduce elaborate circuitry to guarantee that 5V and 3.3V power supplies rise or fall in any particular sequence. This application note describes the underlying XC9500 circuitry to give designers the understanding they need to bes use these powerful CPLDs.

XAPP111 Using the XC9500XL Timing Model v2.0 (08/20/01) UPDATE!

This application note describes how to use the XC9500XL timing model.

XAPP112 Designing With XC9500XL CPLDs

This application note will help designers get the best results from XC9500XL CPLDs. Included are practical details on such topics as pin migration, timing, mixed voltage interfacing, power management, PCB layout, high speed considerations and JTAG best practices.

XAPP113 Faster Erase Times for XC95216 and XC95108 Devices on HP 3070 Series Testers

This application note describes an enhanced procedure for utilizing the new faster bulk erase capability of the XC95216 and XC95108 devices on the HP 3070 tester.

XAPP114 Understanding XC9500XL CPLD Power

The goal of this application note is to discuss XC9500XL CPLD power estimation and optimization and provide the reader with an understanding of sense-amplifier based CPLD power dissipation. A brief discussion of the process for estimation is given. With this information, you can accurately assess the power dissipation for a design. You will also be given guidelines permitting you to make key choices to manage the power dissipation of your design and understand the package thermal limits.

XAPP115 Planning for High Speed XC9500XL Designs

Discovering electrical problems at debug is too late. The printed circuit board has been built and ma have to be significantly changed to debug. The best approach is to avoid the problem. By anticipatin common problems, designs can be substantially “bullet-proofed” before debug. This means planning for options at the outset is the best solution. A thorough but practical checklist is one aspec of planning for success. This application note provides a framework for checklisting a design early t eliminate problems.

XAPP119 Adapting ASIC Designs for Use with Spartan FPGAs

Spartan FPGAs are an exciting, new alternative for implementing digital designs that, previously, would have employed ASIC technology. Pre-existing ASIC intellectual property can be adapted for us with Spartan devices by following a straightforward procedure. Each step of the procedure is explained in detail. Guidelines show how an ASIC design, in the form of an RTL-level HDL file, can be revised to take full advantage of the Spartan series’ capabilities, thereby achieving efficient, high-performance implementations.

XAPP120 Spartan FPGAs -- The Gate Array Solution

This application note discusses the enormous strides made by Spartan series FPGAs in terms of density and performance and how it should be viewed as the Gate Array replacement. The Spartan family from Xilinx offers many of the features that are desired by Gate Array designers with one majo advantage — programmability, which can prove to be the key factor in the success of the product.

XAPP122 The Express Configuration of Spartan-XL FPGAs

Express Mode uses an eight-bit-wide bus path for fast configuration of Xilinx FPGAs. This applicatio note provides information on how to perform Express configuration specifically for the Spartan-XL family. The Express mode signals and their associated timing are defined. The steps of Express configuration are described in detail, followed by detailed instructions that show how to implement the configuration circuit.

XAPP123 Using Three-State Enable Registers in XLA, XV, and Spartan-XL FPGAs

The use of the internal IOB three-state control register can significantly improve output enable and disable time. This application note shows you how to use hard macros to implement this register in both HDL and schematic based designs.

XAPP124 Using Manual Power Down Mode With Spartan-XL FPGAs

Spartan-XL FPGAs come equipped with a Power Down mode that permits an exceptionally low level of power consumption (ICCO = 100 µA typical), making the family ideal for portable battery-powered applications. This application note provides all the information the designer needs to use Power Down mode effectively, including descriptions of the mode’s common applications, internal functioning and electrical characteristics.

XAPP125 Conserving Power With Auto Power Down Mode in Spartan-XL FPGAs

Power consumption plays an important role in battery-powered applications. Spartan-XL FPGAs are designed with segmented routing, 3.3-V operation and advanced process technology to meet the needs for low power and high performance. This application note shows how to reduce power consumption by selectively disabling portions of the design that are not required all the time. Considerable amount of power is saved by disabling the non-critical user logic. This approach is particularly useful for the devices which must be operating all the time. This application note discusses different strategies for reducing the supply current incrementally for an operating device.

XAPP126 Data Generation and Configuration for Spartan Series FPGAs

This application note describes various methods to configure Spartan series FPGAs. Each configuration method is described in detail. Information on necessary software programs to run with input files required, output files produced, download cables used, and other hardware necessary to accomplish the task is discussed. This application note targets users who are new to Xilinx devices and Alliance/Foundation series software tools and is intended to make the configuration and debugging flows easy to understand.

XAPP130 Using the Virtex Block SelectRAM+ Features v1.4 (12/18/00)

The Virtex series provides dedicated blocks of on-chip, true dual-read/write port synchronous RAM, with 4096 memory cells. Each port of the block SelectRAM+ memory can be independently configured as a read/write port, a read port, or a write port, and each port can be configured to a specific data width. The block SelectRAM+ memory offers new capabilities allowing the FPGA designer to simplify designs.

XAPP131 170 MHz FIFOs Using the Virtex Block SelectRAM+ Feature v1.6 (06/06/01)

The Virtex FPGA series provides dedicated on-chip blocks of 4096 bit dual-port synchronous RAM, which are ideal for use in FIFO applications. This application note describes a way to create a common-clock (synchronous) version and an independent-clock (asynchronous) version of a 511 x FIFO, with the depth and width being adjustable within the Verilog or VHDL code. A hand-placed version of the design runs at 170 MHz in the -6 speed grade.

XAPP132 Using the Virtex Delay-Locked Loop v2.3 (09/20/00)

The Virtex FPGA series offers up to eight fully digital dedicated on-chip Delay-Locked Loop (DLL) circuits providing zero propagation delay, low clock skew between output clock signals distributed throughout the device, and advanced clock domain control. These dedicated DLLs can be used to implement several circuits which improve and simplify system level design.

XAPP133 Using the Virtex SelectI/O Resource v2.4 (04/17/00)

The Virtex FPGA series includes a highly configurable, high-performance SelectI/O resource to provide support for a wide variety of I/O standards. The SelectI/O resource is a robust set of features including programmable control of output drive strength, slew rate, and input delay and hold time. Taking advantage of the flexibility and SelectI/O features and the design considerations described in this document can improve and simplify system level design. Appendix A is a SelectI/O update for both the Virtex-E and Virtex-E Extended Memory (Virtex-EM) product families. Appendix B is the Virtex-E and the Virtex-EM LVDS and LVPECL SelectI/O design guide.

XAPP134 **Virtex Synthesizable High Performance SDRAM Controller** v3.1 (02/01/00)

Synchronous DRAMs are becoming available in speed grades above 100 MHz using LVTTL IOs. Th Virtex FPGA family has many features, such as the SelectIO and the Clock Delay Lock Loop, that make it easy to interface to high speed Synchronous DRAMs. This application note describes the design and implementation of a synthesizable, parameterizable, flexible, auto-placed-and-routed synchronous DRAM controller in the Virtex FPGA family. A 32-bit wide data interface version can run up to 125 MHz when automatically placed and routed in a Virtex -6 speed grade. Hand placed versions of the design can run even faster.

XAPP135 **Virtex I/V Curves for Various Output Options**

These typical curves describe the output sink and source current for average processing, nominal supply voltage and room temperature. For the other families see XAPP150. For additional data see the Xilinx IBIS files.

XAPP136 **Synthesizable 143 MHz ZBT SRAM Interface** v2.0 (01/00)

The Virtex Series FPGAs provide access to a variety of on-chip and off-chip RAM resources. In addition to the on-chip SelectRAM and Block SelectRAM+ memory, a Virtex design can interface to megabytes of external high-speed SRAM and DRAM. The combination of high speed SelectIO levels and on-chip Clock Delay-Locked Loop enables the interface to operate at maximum RAM speeds. A Virtex interface to ZBT (Zero Bus Turnaround) SRAM provides interleaved Read/Write without wastef turnaround cycles.

XAPP137 **Configuring Virtex FPGAs from Parallel EPROMs with a CPLD**

Previous generations of Xilinx FPGAs supported a Master Parallel Configuration Mode which allowe the FPGA to configure itself directly from a parallel (byte wide) PROM. The Virtex family of Xilinx FPGA does not utilize a Master Parallel mode. This application note describes a simple interface design to configure a Virtex device from a parallel EPROM using the SelectMAP configuration mode.

XAPP138 **Virtex Configuration and Readback** v2.4 (07/25/01)

This application note is offered as complementary text to the Configuration section of the Virtex Data Sheet. It is strongly recommended that the Virtex Data Sheet be reviewed prior to reading this application note. This application note first provides a comparison of how Virtex configuration and readback is different from previous Xilinx FPGAs, followed by a complete description of the configuration process and flow. Each of the configuration modes are outlined and discussed in detail, concluding with a complete description of data stream formats, and readback functions and operations.

XAPP139 **Configuration and Readback of Virtex FPGAs Using (JTAG) Boundary-Scan** v1.0 (11/99)

This application note demonstrates using a boundary-scan (JTAG) interface to configure and readback Virtex FPGA devices. Virtex devices have boundary-scan features that are compatible with the IEEE Standard 1149.1. This application note is a complement to the configuration section in the Virtex Data Sheet and application note XAPP138: "Virtex Configuration and Readback". Review of bot the Virtex Data Sheet and XAPP138 is recommended prior to reading this document.

XAPP141 **In-System Programming Times for XC9500XL**

This application note discusses the in-system programming speed of the XC9500XL devices.

XAPP144 **Designing CPLD Multi-voltage Systems** v1.1 (02/00)

This application note discusses XC9500XL use in multi-voltage systems.

XAPP146 **Designing an Eight Channel Digital Volt Meter with the Insight Springboard Kit** (v1.0) 4/30/01

This Application Note is intended for all Springboard designers. While it is true that the accompanying design files are intended for an Insight Springboard Development Kit, the concepts presented here are applicable to any other Springboard design. This 8-channel Digital Volt Meter design builds upon the TI ADS7870 Data Aquisition System Interface described in XAPP355. This Application Note will not discuss the ADS7870 Interface in detail. Readers should familiarize themselves wiith XAPP355 before proceeding.

XAPP147 **Low Power Handspring Springboard Module Design with CoolRunner CPLDs** (v1.0) 1/25/01

This application note presents development aids to help designers successfully and easily create Handspring TM Springboard TM Module designs. It includes a general discussion of the overall process, a summary of available software tools and an introduction to Xilinx CoolRunner CPLDs as used on the Insight Electronics Springboard Development Card. The appendices include additional details that developers will find helpful for both code creation and hardware development. Examples of hardware code (VHDL or Verilog) as well as "C" code are provided to augment the development o Handspring Springboard cards.

XAPP150 **I/V Curves for Various Device Families**

These typical curves describe the output sink and source current for average processing, nominal supply voltage and room temperature. For the Virtex FPGAs see XAPP135. For additional data se the Xilinx IBIS files.

XAPP151 **Virtex Series Configuration Architecture User Guide** v1.4 (08/03/00)

The Virtex architecture supports powerful new configuration modes, including partial reconfiguration These mechanisms are designed to give advanced applications access to and manipulation of on-chip data through the configuration interfaces. This document is an overview of the Virtex architecture emphasizing data bit location in the configuration bitstream. Knowing bit locations is the basis for accessing and altering on-chip data. FPGA applications can be built that change or examine the functionality of the operating circuit without stopping the circuit loaded in the device. A glossary is included to explain some of the terminology used in this application note.

XAPP152 **Virtex Power Estimator User Guide**

This application note is complementary to the Virtex power estimator worksheet. To use the worksheet, users should have completed a Virtex design with a successful functional simulation.

XAPP153 **Status and Control Semaphore Registers Using Partial Reconfiguration**

The Virtex FPGA Series supports partial reconfiguration of a cross-section of data while the rest of th circuit is still in operation. This enables a system to read and write specific bits within a LUT configured as RAM, through the configuration port. This application note demonstrates how to lock the LUT SelectRAM to specific locations, determine the corresponding frame of data in the .RBT (Rawbits) file, modify the LUT memory as desired, and re-write this frame into the chip. This provide a microprocessor/FPGA interface through the configuration port with a minimum of IOs.

XAPP154 **Virtex Synthesizable Delta-Sigma DAC**

Digital to analog converters (DACs) convert a binary number into a voltage directly proportional to the value of the binary number. A variety of applications use DACs including waveform generators and programmable voltage sources. This application note describes a Delta-Sigma DAC implemented in a Virtex FPGA. The only external circuitry required is a low pass filter comprised of just one resistor and one capacitor. Internal resource requirements are also minimal. For example, a 10-bit DAC uses only three Virtex CLBs. The speed and flexible output structure of the Virtex series FPGAs make them ideal for this application.

XAPP155 Virtex Analog to Digital Converter

When digital systems are used in real-world applications, it is often necessary to convert an analog voltage level to a binary number. The value of this number is directly or inversely proportional to the voltage. The analog to digital converter (ADC) described here uses a Virtex FPGA, an analog comparator, and a few resistors and capacitors. An 8-bit ADC can be implemented in about 16 Virtex CLBs, and a 10-bit ADC requires about 19 CLBs.

XAPP157 Board Routability Guidelines with Xilinx Fine-Pitch BGA Packages v1.1 (11/99)

Xilinx supplies full array fine-pitch BGA (Ball Grid Array) packages with 1.00 mm ball pitch. Successful and effective routing of these packages on PC boards is a significant challenge to designers. This application note provides board level routing guidelines for using Xilinx fine-pitch BGA packages. Specific examples are provided to choose appropriate routing schemes. These examples are based on package and board design rules for standard PCB technology and are not drawn to scale.

XAPP158 Powering Virtex FPGAs v1.4 (02/06/01)

Power consumption in Xilinx FPGAs depends upon the number of internal logic transitions and is proportional to the operating clock frequency. As device size increases, so does power consumption. It is common for a large, high-speed design to require several amperes of current. Without an accurate thermal analysis, the heat generated could easily exceed the maximum allowable junction temperature. Power supply requirements, including initial conditions, transient behavior, turn-on, and turn-off are also important. Bypassing or decoupling the power supplies at the device, in the context of the device’s application, requires careful attention. All these aspects of the power supply must be considered in order to achieve successful designs.

XAPP161 XC1700 and XC18V00 Design Migration Considerations v1.2 (2/00)

PROMs is advantageous because migration between XC1700 and XC1800 series devices is simple. This application note discusses two migration paths: XC1700 designs upgrading to XC18V00, and XC1800 designs migrating to XC1700 for production-stable cost reductions. The topics discussed are pinout compatibility, power and ground connections, and boundary-scan chain integrity.

XAPP164 Using Xilinx and Synplify for Incremental Designing (ECO)

Guided place and route (PAR) can help you reduce runtimes when incremental changes are made to a design, such as for an Engineering Change Order (ECO). By making only small changes to a design along with optimizing only the changed block(s), you allow guided PAR to perform at its best, preserving timing and reducing PAR runtimes. To localize the design changes without affecting the remainder of your design, either a top-down preserving hierarchy or a bottom-up methodology must be used.

XAPP165 Using Xilinx and Exemplar for Incremental Designing (ECO)

Guided place and route (PAR) can help you reduce runtimes when incremental changes are made to a design, such as for an Engineering Change Order (ECO). By making only small changes to a design along with optimizing only the changed block or blocks, you allow guided PAR to perform at its best, preserving timing and reducing PAR runtimes. To localize the design changes without affecting the remainder of your design, either a top-down preserving hierarchy or a bottom-up methodology must be used.

XAPP166 TAU/BLAST Support in 2.1i

The Xilinx 2.1i development system adds Stamp Model Generation. This feature supports the use of board level Static Timing Analysis tools, such as Mentor Graphics' Tau and Viewlogic's Blast. With these tools, users of Xilinx programmable logic products can accelerate board level design verification.

XAPP168 Getting Started With the MultiLINX Cable v1.2 (04/20/00)

This application note provides a quick introduction to the MultiLINX cable hardware. Topics covered are a description of the cable, how to order a MultiLINX system, a list of features, what the cable may be used for, the current software support, and how to integrate cable access into a user's board. For more information on the MultiLINX cable and other hardware products from Xilinx, please refer to the Hardware User's Guide.

XAPP169 MP3 NG: A Next Generation Consumer Platform v1.0 (01/00)

This application note illustrates the use of a Xilinx Spartan-II FPGA and an IDT RC32364 RISC controller in a handheld, consumer electronics platform. Specifically the target application is an MP3 audio player with advanced user interface features. In this application the Spartan device is used to implement the complex system level glue logic required to interface and manage the memory and I/O devices.

XAPP170 Implementing an ISDN PCMCIA Modem Using Spartan Devices v1.0 (7/99)

This application note illustrates the use of Spartan devices in an ISDN modem. The design example shows how cost effective a Spartan device can be in these applications. While the design is targeted at solving a specific problem, it illustrates solutions to a number of general technical issues.

XAPP171 Implementing an ADSL to USB Interface Using Spartan Devices v1.0 (3/99)

This application note illustrates the use of Spartan devices in an ADSL modem. The Spartan device is used to implement the complex system level glue logic required for the modem’s USB interface and manages DMA transfers of ATM cells. The design example shows how cost effective a Spartan device can be in these applications. While the design is targeted at solving a specific problem, it illustrates solutions to a number of general technical issues. These include implementing Utopia interfaces for ATM devices and remote configuration of Spartan devices.

XAPP172 The Design of a Video Capture Board Using the Spartan Series v1.0 (3/99)

This application note describes a reference design for a video capture board that acts as an interface between a video source such as a camcorder, VCR, CCD camera, etc. and a PC. The board captures and digitizes frames from a video source, which it then transfers to a PC for viewing. The main electronic components consist of a video pixel decoder, DRAM and a Spartan FPGA, all chosen to achieve a low overall cost, making the board suitable for high-volume, consumer-oriented products. To this end, the ability to implement all the interface and memory control logic in a single programmable Spartan device provides crucial benefits including low cost, reduced part count, a small form factor, low power, and easy field upgrades.

XAPP173 Using Block SelectRAM+ Memory in Spartan-II FPGAs v1.1 (12/11/00)

The Spartan-II FPGAs provide dedicated blocks of true dual-port RAM, known as Block SelectRAM+ memory. This dedicated memory provides a cost-effective use of resources without sacrificing the existing distributed SelectRAM memory or logic resources. The Block SelectRAM+ memory is fully synchronous for easy timing analysis and is easily initialized at configuration. This additional integration capability makes the Spartan-II family ideal for cost-sensitive applications.

XAPP174 Using Delay-Locked Loops in Spartan-II FPGAs v1.0 (01/00)

The Spartan-II series provides four fully digital dedicated on-chip Delay-Locked Loop (DLL) circuits, which provide zero propagation delay, low clock skew between output clock signals distributed throughout the device, and advanced clock domain control. These dedicated DLLs can be used to implement several circuits that improve and simplify system level design.

XAPP175 High Speed FIFOs In Spartan-II FPGAs v1.0 (01/00)

This application note describes how to build high-speed FIFOs using the Block SelectRAM+ memory in the Spartan-II FPGAs. Verilog and VHDL code is available for the design. The design is for a 512x8 FIFO, but each port structure can be changed if the control logic is changed accordingly. Both a common-clock version and an independent-clock version are described.

XAPP176 Spartan-II FPGA Family Configuration and Readback v1.0 (01/00)

This application note is offered as complementary text to the configuration section of the Spartan-II data sheet. It is strongly recommended that the Spartan-II data sheet be reviewed prior to reading this note. Spartan-II FPGAs offer a broader range of configuration and readback capabilities than previous generations of Xilinx FPGAs. This note first provides a comparison of how Spartan-II configuration is different from previous Xilinx FPGAs, followed by a complete description of the configuration process and flow. Each of the configuration modes are outlined and discussed in detail, concluding with a complete description of data stream formats, and readback functions and operations.

XAPP177 Spartan-Family I/V Curves for Various Output Options v1.0 (01/00)

These typical curves describe the output sink and source current for average processing, nominal supply voltage and room temperature for the Spartan-II family of FPGAs. These curves are graphical representations of IBIS models, which are traditionally used for system and board-level simulation.

XAPP178 Configuring Spartan-II FPGAs from Parallel EPROMs v1.0 (01/00)

This application note describes a simple CPLD-based interface design to configure a Spartan-II device from a parallel EPROM using the Slave Parallel configuration mode.

XAPP179 Using SelectI/O Interfaces in Spartan-II FPGAs v1.0 (01/00)

The Spartan-II FPGA family simplifies high-performance design by offering SelectI/O inputs and outputs. The Spartan-II devices can support 16 different I/O standards with different specifications for current, voltage, I/O buffering, and termination techniques. As a result, the Spartan-II FPGA can be used to integrate discrete translators and directly drive the most advanced backplanes, busses, and memories. This application note describes how to take full advantage of the flexibility of the SelectI/O features and the design considerations to improve and simplify system level design.

XAPP181 SEU Mitigation Design Techniques for the XQR4000XL v1.0 (03/15/00)

This Application Note discusses system and FPGA design techniques for applications that operate in space or in other environments exposed to heavy ion or charged particle radiation. Single Event Upset (SEU) detection, correction, and mitigation for the XQR4000XL are demonstrated.

XAPP188 Configuration and Readback of Spartan-II FPGAs Using Boundary Scanv2.0 (04/19/01)

This application note demonstrates using a boundary-scan (JTAG) interface to configure and read back Spartan™-II FPGA devices. Spartan-II devices have boundary-scan features that are compatible with the IEEE Standard 1149.1. This application note is a complement to the configuration section in the Spartan-II Data Sheet and application note XAPP176: “Spartan-II FPGA Family Configuration and Readback”.

XAPP189 Powering Xilinx Spartan-II FPGAs v1.1 (07/20/01)

Power consumption in Xilinx Spartan-II FPGAs depends upon the number of internal logic transitions and is proportional to the operating clock frequency. As device size increases, so does power consumption. It is common for a large, high-speed design to require one Ampere or more of current. Without an accurate thermal analysis, the heat generated could easily exceed the maximum allowable junction temperature. Power supply requirements, including initial conditions, transient behavior, turn-on, and turn-off are also important.

XAPP192 Interfacing a Virtex-E Device to a MIPS Processor v1.0 (12/15/00)

This application note describes a reference design for a Virtex-E FPGA interface to a MIPS processor. The interface connections are shown while discussing techniques for running the design at the fastest data throughput speed available from a MIPS processor.

XAPP196 Interfacing a Virtex-E Device to a Pentium Processor v1.0 (12/15/00)

This application note describes a reference design for a Virtex-E FPGA interface to an Intel Pentium processor. The Pentium I system bus, design concerns, and possible applications of this design are discussed. Additionally, the differences between the Pentium I, II, and III busses are discussed. For more information specific to the Intel Pentium family of processors, see the Intel developer website (http://developer.intel.com/).

XAPP198 Synthesizable FPGA Interface for Retrieving ROM Number from 1-Wire Devices v1.0 (05/08/01) >

This application note describes the design and implementation of a simple, low-cost interface to the Dallas Semiconductor's 1-Wire devices in Virtex and Spartan-II families to acquire the 64-bit ROM number. The number is available in either eight sequential byte transfers through an 8-bit data port, or a 48-bit latched parallel output. A typical application is to use the 48-bit serial number in the ROM number as the physical address of a network interface. This reference design is synthesizable and utilizes only 52 registers, 65 look-up tables (LUTs), and 55 slices of FPGA resource.

XAPP199 Writing Efficient Testbenches v1.0 (06/18/01)

This application note is written for logic designers who are new to HDL verification flows, and who do not have extensive testbench-writing experience. Testbenches are the primary means of verifying HDL designs. This application note provides guidelines for laying out and constructing efficient testbenches. It also provides an algorithm to develop a self-checking testbench for any design.

XAPP200 Virtex Synthesizable 1.6 Gbytes/s DDR SDRAM Controller. v2.3 (03/21/00)

The DLLs and the SelectI/O™ features in the Virtex™ architecture and Spartan™-II family make it the perfect choice for implementing a controller of a Double Data Rate (DDR) SDRAM. This application note describes the reference controller design for a 64-bit DDR SDRAM. At a clock rate of 100 MHz, and 64-bit data changing at both clock edges, a peak bandwidth of 1.6 GBytes/s is obtained. The reference design is synthesizable and achieves 100 MHz performance with auto place and route tools.

XAPP201 An Overview of Multiple CAM Designs in Virtex Devices

Flexible CAMs (Content Addressable Memory) are implemented in Virtex devices by taking advantage of the reprogrammability of the basic LUT as a Shift Register or a SelectRAM memory and the fast carry logic chain. Although Cams are also feasible in Spartan and XC4000X devices, this application note concentrates on Virtex devices. The flexibility of a Virtex device is a key advantage in designing a CAM. The application must decide the best implementation.

XAPP202 Content Addressable Memory (CAM) in ATM Applications v1.2 (01/06/01)

Content Addressable Memory (CAM) or associative memory, is a storage device, which can be addressed by its own contents. Each bit of CAM storage includes comparison logic. A data value input to the CAM is simultaneously compared with all the stored data. The match result is the corresponding address. A CAM operates as a data parallel processor. CAMs can be used to design Asynchronous Transfer Mode (ATM) switches. Implementing CAM in ATM applications are specifically described in this application note. As a reference, the application note XAPP201, “An Overview of Multiple CAM Designs in Virtex Devices,” presents diverse approaches to implement CAM in other designs.

XAPP203 **Designing Flexible, Fast CAMs with Virtex Slices**

Content Addressable Memories (CAM) allow a fast search for specific data in a memory. Each application has different CAM requirements. A CAM design implemented in Virtex slices offers a flexible approach to CAM depth and width based upon LUTs configured as Shift Registers. This application note describes a fast CAM design finding a match in a single clock cycle. The application note ~~XAPP201~~ “An Overview of Multiple CAM Designs in Virtex devices” discusses the diverse solutions available when implementing CAM and introduces the specific solution described in this application note.

XAPP204 **Using Block RAM for High-Performance Read/Write CAMs** v1.2 (05/02/00)

CAM (Content Addressable Memory) offers increased data search speed. In various applications based on CAM, there are differing requirements for data organinzatation and read/write performance. The innovative design described in this application note is suited for small embedded CAMs with high-speed match and write requirements. The reference design is built using the true Dual-Port block SelectRAM+ feature of Virtex FPGAs. Application note XAPP201, "An Overview of Multiple CAM Designs in Virtex Devices", discusses the diverse solutions available when implementing CAM while introducing the specific solution described in this application note.

XAPP205 **Data-Width Conversion FIFOs Using the Virtex Block SelectRAM Memory** v1.3 (08/10/00)

Virtex FPGAs provide dedicated on-chip blocks of 4096-bit dual-port synchronous RAM (block SelectRAM+ memory). The block SelectRAM feature is ideal for use in FIFO applications. This application note describes how to create a common-clock (synchronous) version and an independent-clock (asynchronous) version of a FIFO for data-width conversion with different width read and write data ports.

XAPP208 **An Inverse Discrete Cosine Transform (IDCT) Implementation in Virtex Devices for MPEG Video Applications** v1.1 (01/00)

This application note describes an implementation of IDCT in the Virtex family. DCT/IDCT are used in the MPEG video standard to reduce the bandwidth requirements. IDCT is one of the most computation-intensive parts of the MPEG decoding process. A fast, hardware based IDCT implementation is crucial to speed the MPEG decoding process. In this implementation, the inherent parallelism is exploited to achieve throughput as high as 3.28 Gbits/s, making it suitable for real time video applications. The implementation is synthesizable Verilog code at the RTL level.

XAPP209 **IEEE 802.3 Cyclic Redundancy Check** v1.0 (03/32/01)

Cyclic Redundancy Check (CRC) is an error-checking code that is widely used in data communication systems and other serial data transmission systems. CRC is based on polynomial manipulations using modulo arithmetic. Some of the common Cyclic Redundancy Check standards are CRC-8, CRC-12, CRC-16, CRC-32, and CRC-CCIT. This application note discusses the implementation of an IEEE 802.3 CRC in a Virtex device. The reference design provided with this application note provides Verilog point solutions for CRC-8, CRC-12, CRC-16, and CRC-32. The Perl script (crcgen.pl) used to generate this code is also included. The script generates Verilog source for CRC circuitry of any width (8, 12, 16, 32), any polynomial, and any data input width.

XAPP210 Linear Feedback Shift Registers in Virtex Devices v1.2 (01/09/01)

This application note describes the implementation of Linear Feedback Shift Registers (LFSR) using the SRL macroavailable in the Virtex and Virtex-II series of FPGAs. The optimal implementation of a 15-bit LFSR, a 52-bit LFSR, and a 118-bit LFSR are discussed in this application note.

XAPP211 PN Generators Using the SRL Macro v1.1 (01/09/01)

Pseudo-random Noise (PN) generators are at the heart of every spread spectrum system. Many PN generators are required within Code Division Multiple Access (CDMA) base stations. PN generators are used to implement synchronization and uniquely code individual user signals across the transmission interface. PN generators are based upon Linear Feedback Shift Registers (LFSRs). Every Look-Up-Table (LUT) in a Virtex series or Virtex-II series device can be configured as a 16-bit shift register (SRL16 macro). Hence, Virtex devices implement efficient LFSRs and deliver a significant reduction in resource utilization when compared with alternative flip-flop only PLD structures. For example, a 16-stage LFSR can be realized in just one LUT.

XAPP212 CDMA Matched Filter Implementation in Virtex Devices v1.1 (01/10/01)

Code Division Multiple Access (CDMA) is a rapidly expanding data transmission technique in the emerging Universal Mobile Telecommunications System (UMTS). This application note describes the implementation of a CDMA matched filter using the architectural features of the Virtex series, Virtex-II series, and Spartan-II devices.

XAPP213 8-Bit Microcontroller for Virtex Devices v1.0 (09/25/00)

The Constant (k) Coded Programmable State Machine (KCPSM) presented in this application note is a fully embedded 8-bit microcontroller macro for the Virtex and Spartan-II devices. The module is remarkably small at just 35 CLBs, less than half of the smallest Spartan XC2S15 device, and virtually free in an XCV2000 device by consuming less than 0.37% of the device CLBs.

XAPP215 Design Tips for HDL Implementation of Arithmetic Functions v1.0 (06/28/00)

This application note provides design advice for implementing arithmetic logic functions in two High-Level Design Languages (HDLs), VHDL and Verilog.

XAPP216 Correcting Single-Event Upsets Through Virtex Partial Configuration v1.0 (06/01/00)

This application note describes the use of partial reconfiguration in Virtex series FPGAs for the purpose of correcting Single Event Upsets to the configuration memory array induced by cosmic rays. It is essential for the reader to have a basic understanding of the Virtex SelectMAP interface as well as configuration and readback operations. An in-depth review of Xilinx Application Note XAPP138 is highly recommended.

XAPP217 Gold Code Generators in Virtex Devices v1.1 (01/10/01)

Gold code generators are used extensively in Code Division Multiple Access (CDMA) systems to generate code sequences with good correlation properties. This application note describes the implementation of Gold code generators in Virtex, Virtex-E, Virtex-EM, Virtex-II and Spartan-II devices. The Gold code generators use efficiently implemented Linear Feedback Shift Registers (LFSRs) in both the Virtex/Virtex-II series and Spartan-II family using the SRL16 macro.

XAPP219 Transposed Form FIR Filters v1.1 (01/10/01)

This application note describes a high-speed, reconfigurable, full-precision Transposed Form FIR filter design implemented in the Virtex and Virtex-II series and Spartan-II family of FPGAs. The VHDL reference design provided with this application note is easily modified to change filter parameters including coefficients and the number of taps. By illustrating a design methodology for digital filters, the advantages of using FPGAs for digital signal processing applications (DSP) are emphasized. The Core Generator tool provides a preoptimized alternative solution to this reference design (Core Generator Tool).

XAPP220 LFSRs as Functional Blocks in Wireless Applications v1.1 (01/11/01)

Linear Feedback Shift Registers (LFSRs) are commonly used in applications where pseudo-random bit streams are required. LFSRs are the functional building blocks of circuits like the pseudo-random noise (PN) code generator (XAPP211) and Gold code generators (XAPP217) commonly used in Code Division Multiple Access (CDMA) systems. This application note describes two implementations of an LFSR using the SRL16 (Shift Register Look-Up Table) primitive for area-efficient designs. The first LFSR implementation describes the parallel output access and parity calculation; the second describes the multi-cycle output access and sequential parity calculation. This application note covers the Virtex series, the Virtex-II series and the Spartan-II family of devices.

XAPP223 200 MHz UART with Internal 16-Byte Buffer v1.1 (07/10/01)

This application note describes highly optimized UART transmitter and receiver macros for Xilinx Virtex, Virtex-E, and Spartan-II devices. The UART_TX and UART_RX macros not only communicate with each other, but they are also fully compatible with the standard Universal Asynchronous Receiver Transmitter (UART) communication protocols used for connecting to devices, such as PCs or microcontrollers.

XAPP224 Data Recovery in Virtex and Virtex-II Devices v1.1 (01/10/01)

Data recovery is a mechanism that allows a receiver to extract embedded clock data from an incoming data stream. The receiver usually extracts this information from the data stream concerned, but sometimes the receiver's clock is used for data transmission. The circuit described in this application note provides a partial solution at data rates up to 160 Mb/s in a Virtex-E, -7 device, and up to 210 Mb/s in a Virtex-II device. The solution is partial in the sense that no clock is actually recovered, but the data arriving is fully extracted. The speed is limited by the maximum frequency that can be accepted by the Data Locked Loop (DLL), in a mode where the DLL is capable of providing both a new clock, and another clock shifted by 90 degrees.

XAPP225 Data to Clock Phase Alignment v1.0 (09/18/01) NEW!

When designing digital systems, there is often a requirement to synchronize incoming data and clock signals with an internal system clock, i.e., the internal and external clock are at exactly the same frequency, but due to variable backplane, board, or application-specific standard product (ASSP) delays, the phase relationship is not known. The circuit described in this application note addresses this issue for both single traces and data busses up to 160 MHz in a Virtex™-E, -7 device. The speed limitation is imposed by the maximum frequency that can be accepted by the Data Locked Loop (DLL), in a mode where it is capable of providing both a new clock and a new clock shifted by 90 degrees.

XAPP230 The LVDS I/O Standard

This application note describes the LVDS I/O standard. LVDS provides higher noise immunity than single-ended techniques, allowing for higher transmission speeds, smaller signal swings, lower power consumption, and less electro-magnetic interference than single-ended signaling. Differential data can be transmitted at these rates using inexpensive connectors and cables. LVDS provides robust signaling for high-speed data transmission between chassis, boards, and peripherals using standard ribbon cables and IDC connectors with 100 mil header pins. Point-to-point LVDS signaling is possible at speeds of up to 622 Mb/s.

XAPP231 Multi-Drop LVDS with Virtex-E FPGAs

This application note describes how to use LVDS signaling for high-performance multi-drop applications with Virtex-E FPGAs. Multi-drop LVDS allows many receivers to be driven by one Virtex-E LVDS driver. Simulation results indicate that the reference design described here will operate from DC up to 311 Mbits/s. This application note includes DC specifications, microstrip and layout guidelines. With simple source and differential termination, Virtex-E FPGAs drive multi-drop LVDS directly, replacing costly TTL-LVDS drivers and receivers, reducing board area and skew for high-performance applications. The Virtex-E driver actually improves signal integrity over other LVDS drivers by absorbing any reflected energy at the source instead of passing it on down the line. This innovation enables 311 Mb/s signaling on multi-drop lines with as many as 20 LVDS receivers, spanning distances of over four feet in the reference design, with high signal integrity and noise immunity.

XAPP232 Virtex-E LVDS Drivers & Receivers: Interface Guidelines v1.0 (11/99)

This application note describes how to use the new Virtex-E LVDS (low-voltage differential signaling) drivers and receivers for high-performance LVDS interfaces to industry-standard LVDS devices. LVDS provides higher noise immunity than single-ended techniques, allowing for higher transmission speeds, smaller signal swings, lower power consumption, and less electro-magnetic interference than single-ended signaling. Differential data can be transmitted at these rates using inexpensive connectors and cables. Virtex-E LVDS drivers offer improved signal integrity over other LVDS drivers because they absorb reflected signals unlike other LVDS drivers.

XAPP233 Multi-Channel 622 Mb/s LVDS Data Transfer for Virtex-E Devices v1.2 (01/06/01)

Virtex-E devices provide dedicated on-chip differential receivers between adjacent user I/O pins, which are ideal for receiving LVDS signals at speeds of up to 622 Mb/s in the -7 speed grade. This application note describes how to design a high-speed, low-voltage differential signaling (LVDS) transmitter and receiver in a Virtex-E FPGA suitable for point-to-point data transmission at a data rate of 622 Mb/s.

XAPP234 Virtex SelectLink Communications Channel v1.0 (12/99)

Systems that include two or more FPGAs often require high-bandwidth data paths between devices. As the clock period and switching times of digital circuits become shorter, straightforward methods of transferring data between devices are often inadequate. At high frequencies, signal propagation delay and reflections that occur in conductors just a few centimeters long must be taken into account. The SelectLink™ communications channel utilizes special features of the Virtex family, including Delay Locked Loops, Block SelectRAM+, and SelectI/O, to create a system that can move large amounts of data between FPGAs at very high speeds. A code generation tool available at www.xilinx.com allows logic designers everywhere to instantly create customized SelectLink Verilog source code. The modules are easily instantiated in the designers top level code for a complete system solution.

XAPP235 Virtex Package Compatibility Guide v1.3 (06/20/00)

This package compatibility guide describes the pin-outs and established guidelines for package compatibility between the Virtex family and the Virtex-E and Virtex-E Extended Memory (Virtex-EM) devices. For the latest information regarding the Virtex-E families, see the Xilinx web site at http://www.xilinx.com.

XAPP238 LVDS System Data Framing v1.0 (12/18/00)

This document describes an implementation of a low-overhead data synchronization and framing method to use with the LVDS capability of Virtex-E devices described in XAPP233.

XAPP240 High-Speed Buffered Crossbar Switch Design Using Virtex-EM Devices v1.0 (3/14/00)

High-speed switches are increasingly required in high-bandwidth applications. In the face of constantly changing networking standards, FPGAs offer switch designers flexibility and adaptability. FPGAs with expanded memory capacity, such as Virtex-E Extended Memory (Virtex-EM) devices, are ideally suited for scalable, fast switches. This document discusses a high-speed buffered crossbar switch that effectively addresses each of these concerns.

XAPP241 Virtex-EM FIR Filter for Video Applications v1.0 (3/14/00)

Virtex-E Extended Memory (Virtex-EM) FPGA devices offer over a million bits of block RAM and up to 300 Kb of distributed RAM in a single high-performance device. This is ideal for high-bandwidth video applications where complex digital filtering logic can operate on several lines of pixel data on-chip. The reconfigurable nature of Virtex-EM devices offers designers a flexible platform for optimizing Digital Signal Processor (DSP) parameters and algorithms throughout the design and preproduction cycle, as well as when the devices are in the field. This reprogrammability allows periodic optimization of proprietary algorithms in such applications as MPEG compression.

XAPP242 Interfacing to Lara Networks Search Engine Using Virtex Devices v1.0 (6/08/00)

Due to rapidly expanding networking industry demands, there is a corresponding need for faster and faster search capabilities within Content Addressable Memory (CAM) devices. Every year new CAM devices emerge on the market. These devices have excellent capabilities and options, but they require an accompanying interface. Virtex devices have all the necessary features to interface with high-speed CAMs. This document describes a Virtex CAM controller for the Search Engine (a type of CAM device) from Lara Networks.

XAPP245 **Eight Channel, One Clock, One Frame LVDS Transmitter/Receiver** v1.0 (03/25/01)

This application note describes a 5.12 Gbps transmitter and receiver interface using ten Low-Voltage Differential Signalling (LVDS) pairs (one clock, eight data channels, one frame) implemented in a Virtex-E FPGA. The accompanying library of designs targets Virtex-E devices. The design is implemented as a EDIF netlist with embedded location constraints and VHDL and Verilog simulation files. The design does not rely on guide files for successful performance.

XAPP246 PowerPC 60X Bus Interface to a Virtex-E Device v1.0 (12/15/00)

This application note describes a reference design using a PowerPC 60X bus interface with interfaces to Synchronous Static RAM (SSRAM) and flash memory. The design supports two PowerPC 60X bus microprocessors (PowerPC 750 and 750CX) and implements a pipelined address bus and split address/data transactions on the 60X bus. This reference design uses a processor bus functional model to verify the 60X bus interface to a memory system. Having the capability to generate bus traffic and look inside the Virtex-E device, in a simulation environment, resolves system issues during the course of a complex system development. Design approaches using Virtex-E FPGAs accommodate evolutionary changes in microprocessor bus protocol, memory, and I/O standards through the ability to reuse and reprogram the design.

XAPP251 Hot-Swapping Virtex-II Devices v1.1 (087/15/01) UPDATE!

Hot-swapping or hot insertion describes a potentially dangerous method of inserting an unpowered board into a power-on (hot) running system. There are several concerns: the insertion must not cause physical harm or permanent damage to the system or the inserted board, and the insertion must not cause data corruption or any transient system upsets. This application note describes the physical aspects of hot-inserting a Virtex-II based card into a system or system backplane, using sequenced connectors, where VCC and GND mate well before any signal pins can mate. The dangers of using normal non-sequenced connectors are described in Hot Plug-In. Not addressed in this application note are system issues including detecting the presence or absence of a card, or how the card is accepted in the system.

XAPP253 Synthesizable 266 MBits/s DDR SDRAM Controller v1.0 (01/12/01)

The DDR, DCM, and SelectI/O features in the Virtex-II architecture make it the perfect choice for implementing a controller of a Double Data Rate (DDR) SDRAM. The Digital Clock Manager (DCM) provides the required Delay Locked Loop (DLL), Digital Phase Shift (DPS), and Digital Frequency Synthesis (DFS) functions. This application note describes a controller design for a 16-bit DDR SDRAM. The application note and reference design are enhanced versions of XAPP200 targeted to the Virtex-II series of FPGAs. At a clock rate of 133 MHz, 16-bit data changes at both clock edges. The reference design is fully synthesizable and achieves 133 MHz performance with automatic place and route tools.

XAPP254 The Virtex-II SiberBridge v1.0 (01/12/01)

Designed to be implemented in a Virtex-II FPGA, the Virtex-II SiberBridge is a register transfer logic (RTL) design example demonstrating a reference interface between a 32-bit host (typically a network processor) and the SiberCAM device, or a cascade of SiberCAM devices. The SiberCAM device is a large capacity content addressable memory (CAM) product of SiberCore Technologies. The SiberBridge provides a way to initiate searches, obtain search results, and perform table maintenance operations for the SiberCAM, all using a single 32-bit synchronous SRAM or a ZBT SRAM interface. The SiberBridge is intended as a reference design having a low-gate count.

XAPP256 FIFOs Using Virtex-II Shift Registers v1.0 (01/15/01)

The shift registers available in Virtex-II devices are ideal when building synchronous FIFOs. By using the flexibility of the shift register LUT primitive (SRL16), FIFOs can be built with any width while producing a 1-bit resolution. With cascaded SRL16 shift registers (SRLC16), a flexible depth in multiples of 16 is available.

XAPP258 FIFOs Using Virtex-II Block RAM v1.2 (06/05/01)

The Virtex-II FPGA series provides dedicated on-chip blocks of 18 Kbit True Dual-Port synchronous RAM for use in FIFO applications. This application note describes a way to create a common-clock (synchronous) version and an independent-clock (asynchronous) version of a 511 t 36 FIFO, with the depth and width being adjustable within the Verilog or VHDL code.

XAPP261 Data-Width Conversion FIFOs Using the Virtex-II Block RAM Memory v1.0 (01/10/01)

Virtex-II FPGAs provide dedicated on-chip blocks of 18 Kb dual-port synchronous RAM (block RAM). The block RAM feature is ideal for use in FIFO applications. This application note describes how to create a common-clock (synchronous) version and an independent-clock (asynchronous) version of a FIFO for data-width conversion with different width read and write data ports.

XAPP262 Quad DataRate (QDR) SRAM Interface for Virtex-II Devices v1.0 (01/15/01)

The Virtex-II family of FPGAs provides access to a variety of on-chip and off-chip RAM resources. In addition to the on-chip distributed RAM and block RAM features, Virtex-II FPGAs interface to a variety of external high-speed memory devices. The combination of the high-speed SelectI/O resources and on-chip Digital Clock Manager (DCM) circuits enables a high-bandwidth interface to Quad DataRate (QDR) architecture SRAMs. This application note describes the implementation of an interface using the Cypress CY17C1302V25 QDR SRAM.

XAPP267 Parity Generation and Validation in Virtex-II Devices v1.0 (01/15/01)

In data transmission systems the transmission channel itself is a source of data error. Hence the need to determine the validity of transmitted and received data. Parity generation and validation is a scheme to provide single bit error detection capabilites. This application note describes how to generate and validate parity in a design using the Virtex-II architectural features including block RAM.

XAPP270 **High-Speed DES and Triple DES Encryptor/Decryptor** v1.0 (10/01/01)

The future of network security depends on encryption provided in the crucial building blocks, like switches, routers, bridges, and other communication equipment. All broadband applications need high-speed cryptosystems to speed up high-bandwidth data transfers and to protect privacy. DES cryptographic hardware is used to protect civilian satellite communications, gateway servers, set-top boxes, Virtual Private Networks (VPN), video transmissions, and numerous other data transfer applications.

XAPP283 **Color Space Converter** v1.0 (07/11/01)

This application note describes three ways to implement the YCrCb Color Space to RGB Color Space conversion necessary in many video designs. The first implementation shows how one might simply write Behavioral Verilog to describe the conversion equations and then synthesize to a silicon target. The second implementation uses the Xilinx feature of embedded RAM functioning as a Look-up Table (LUT), or ROM, to store all possible intermediate results for the terms in the three equations. Since three of the seven total terms are identical, only five ROMs are needed. The third implementation makes use of the embedded multiplier in the Virtex-II device to do the color space conversion. Again, only five multipliers are used. The Verilog model using the embedded multiplier is synthesized, placed, and routed. The design has a clock performance of 185 MHz after place and route, using simple constraints.

XAPP284 **3 x 3 Matrix Multiplier for 3D Graphics and Video** v1.0 (07/11/01)

This application note describes the implementation of 3 x 3 matrix multipliers in Virtex-II devices. Many pipelined functions in the fields of computer graphics and video can be expressed in matrix mathematics. The example given here is color space conversion, which can be viewed as a subset of matrix multiplication. However, the technique can be extended to other matrix math functions as well.

XAPP289 **Common Switch Interface CSIX-L1 Reference Design** v1.0 (08/10/01) NEW!

This application note describes a Virtex-II device implementation of a CSIX-L1 common switch interface between a network processor's traffic manager and the switching fabric for ATM, IP, MPLS, Ethernet, and similar data communications applications. This design uses a pipeline implementation to achieve a low clock period (approximately 166 MHz), and uses the 32-bit interface CSIX scheme.

XAPP300 **In-System Programming (ISP)** v1.1 (2/00)

Describes JTAG and ISP background information, and specific features in the CoolRunner CPLDs.

XAPP310 **Power Up Reset Characteristics of CoolRunner CPLDs** v1.1 (2/00)

Depending upon where and how CoolRunner CPLDs are used, the power up characteristics may b of interest.

XAPP311 **Five Volt Tolerance and PCI** v1.1 (2/00)

The purpose of this application note is to investigate the PCI (Peripheral Component Interface) environment when using 5 volt tolerant, 3.3 volt supply integrated circuits. In particular, we will examine the meaning of the statement "PCI compliant" when used in CPLD or FPGA data sheets.

XAPP312 **Differences In ABEL and PHDL** v1.0 (11/99)

This document highlights the few major differences between ABEL and PHDL. All other PHDL constructs and syntax not discussed in this document are supported in ABEL. Most PHDL designs will be accepted in Xilinx Project Navigator with just a modification to the file extension.

XAPP328 Design of an MP3 Portable Player Using a CoolRunner CPLD v1.1 (12/99)

MP3 portable players are the trend in music-listening technology. These players do not include any mechanical movements, thereby making them ideal for listening to music during any type of activity. MP3 is a digital compression technique based on MPEG Layer 3 which stores music in a lot less space than current CD technology.

XAPP329 Understanding True CMOS Outputs v1.0 (02/00)

This document provides a description of the CMOS output structures of the CoolRunner CPLDs and details some advantages of using true CMOS (rail-to-rail capable) output drivers.

XAPP332 Pin Locking in CoolRunner XPLA3 CPLDs v1.0 (01/07/00)

This document highlights the architectural features provided with CoolRunner CPLDs that enable pi assignments to be maintained through many design iterations.

XAPP333 CoolRunner XPLA3 I^{2}C Bus Controller Implementation v1.4 (07/21/00)

This document details the VHDL implementation of an I^{2}C controller in a Xilinx CoolRunner XPLA3 256 macrocell CPLD. CoolRunner CPLDs are the lowest power CPLDs available, making this the perfect target device for an I^{2}C controller. To obtain the VHDL code described in this document, go tc section "VHDL Code Download" on page 18 for instructions.

XAPP334 Utilizing XPLA3 Universal Control Terms v1.0 (02/00)

This document highlights the advantages of utilizing the universal control terms provided in the CoolRunner XPLA3 CPLD architecture. Design examples showing the efficiency of these universal control terms are discussed.

XAPP335 Macrocell Configurations in CoolRunner XPLA3 CPLDs v1.0 (04/17/00)

This document describes the macrocell configurations of Xilinx CoolRunner XPLA CPLDs.

XAPP336 Design of a 16b/20b Encoder/Decoder Using a CoolRunner CPLD v1.0 (06/19/00)

This document details the VHDL implementation of a fibre channel byte-oriented transmission encoder and decoder in a Xilinx CoolRunner CPLD. CoolRunner CPLDs are the lowest power CPLDs available today and can be utilized in any network design where reliable point-to-point transceivers are required. CoolRunner CPLDs utilize the patented Fast Zero Power (FZP) design technique to simultaneously deliver high performance and low power consumption. These devices offer pin-to-pin delays of 5.0 ns, and less than 100 µA of standby current (approximately 1/3 of the power consumed by other competing CPLDs at f_{MAX}). To obtain the complete VHDL 16b/20b encoder or decoder design code described in this document, refer to section "VHDL Code Download" on page 25 for instructions.

XAPP338 Using Xilinx WebPACK and ModelTech ModelSim Xilinx Edition (MXE) v1.0 (04/12/00)

Xilinx WebPACK software is now more powerful than ever with the addition of Model Technology, Inc. (MTI) to this popular EDA tool suite. This application note is designed to quickly show WebPACK users who are not familiar with MTI how to utilize this powerful new tool within the WebPACK environment.

XAPP339 Manchester Encoder-Decoder for Xilinx CPLDs v1.1 (04/17/00)

This application note provides a functional description of VHDL and Verilog source code for a Manchester Encoder Decoder. The reasons to use Manchester code are discussed. The code can be compiled into either the Xilinx XC9572 or XCR3064XL CPLD. To obtain the VHDL (or Verilog) source code described in this document, see section "VHDL (or Verilog) Code Download" on page (for instructions.

XAPP341 UARTs in Xilinx CPLDs v1.2 (11/28/00)

This application note provides a functional description of VHDL and Verilog source code for a UART. The code is used to target the XC95144 and XCR3128XL CPLDs. The functionality of the UART is discussed. To obtain the VHDL (or Verilog) source code described in this document, go to section "VHDL (or Verilog) Code Download" on page 3 for instructions.

XAPP342 XPLA3 I/O Cell Characteristics v1.0 (04/06/01)

This document describes the features and benefits of the I/O cells provided by Xilinx CoolRunner ® XPLA3 CPLDs.

XAPP343 In-System Programming of XPLA3 Devices v1.0 (08/30/00)

This document provides a brief description of how to perform ISP operations with XPLA3 CPLDs.

XAPP345 IrDA and UART Design in a CoolRunner CPLD v1.0 (08/08/01) NEW!

This application note illustrates the implementation of an IrDA and UART system using a CoolRunner XPLA3 CPLD. The fundamental building blocks required to create a half-duplex IrDA an full-duplex UART interface design is described.

XAPP346 Low Power Tips for CoolRunner Design v1.0 (10/16/00)

This document details specific implementation techniques which may be used to decrease power consumption in CPLD designs.

XAPP347 Decrease Processor Power Consumption Using a CoolRunner CPLD v1.0 (06/01/01)

This application note describes system design techniques using a low power CoolRunner CPLD to reduce overall system power consumption. Utilizing a CoolRunner CPLD to off load operations from the system microprocessor keeps the processor in a power saving mode longer and contributes to significant power savings.

XAPP348 CoolRunner XPLA3 Serial Peripheral Interface Master v1.0 (11/29/00)

This document details the VHDL implementation of a Serial Peripheral Interface (SPI) master in a Xilinx CoolRunner XPLA3 CPLD. CoolRunner CPLDs are the lowest power CPLDs available, makin this the perfect target device for an SPI Master.

XAPP349 CoolRunner CPLD 8051 Microcontroller Interface v1.0 (12/07/00)

This document details the VHDL implementation of an 8051 microcontroller interface in a Xilinx CoolRunner XPLA3 CPLD. CoolRunner CPLDs are the lowest power CPLDs available, making these CPLDs the perfect interface devices for many of today's popular microcontrollers.

XAPP350 Implementing HDL with WebPACK ECS Schematic Editor v1.0 (12/20/00)

This application note provides an introduction to the capabilities and functionality of the WebPACK™ ECS Schematic Editor for implementing Hardware Description Language (HDL) CPLD designs in WebPACK Project Navigator.

XAPP351 The CoolRunner CPLD IRL Demo: An Example of Using the Internet to Configure a CoolRunner CPLD v1.0 (11/07/00)

This document details the process used to demonstrate configuring a CoolRunner CPLD over the Internet. All of the VHDL source files and software files associated with this document can be downloaded.

XAPP352 Utilizing a User Constraint File for CoolRunner CPLDs v1.0 (06/20/01)

This application note provides an introduction to the capabilities and functionality of the User Constraint File (UCF) for CoolRunner™ XPLA3 CPLD designs in WebPACK™ Project Navigator.

XAPP353 CoolRunner XPLA3 SMBus Controller Implementation v1.0 (02/14/01)

This document details the VHDL implementation of an system Management Bus (SMBus) controller in a Xilinx CoolRunner XPLA3 256-macrocell CPLD. CoolRunner CPLDs are the lowest power CPLDs available, making this the perfect target device for an SMBus controller.

XAPP354 Using Xilinx CPLDs to Interface to a NAND Flash Memory Device v1.0 (08/30/01)

This application note describes the use of a Xilinx CoolRunner™ XPLA3 CPLD to implement a NAN Flash memory interface. CoolRunner CPLDs are the lowest power CPLD available and the ideal target device for memory interface applications.

XAPP355 Serial ADC Interface Using a CoolRunner CPLD v1.0 (04/30/01)

This document describes the design implementation for controlling a Texas Instruments ADS7870 Analog to Digital Converter (ADC) in a Xilinx CoolRunner™ XPLA3™ CPLD. CoolRunner CPLDs are the lowest power CPLD available and the ideal target device for controlling a serial ADC in a portabl handheld application. This document will provide an explanation of the VHDL code for the CoolRunner CPLD.

XAPP357 CoolRunner Visor Springboard LED Test v1.1 (06/25/01)

LED Test is a simple Springboard™ reference design that allows the Handspring Visor™ to control and blink each of the four LED's on the Insight Springboard™ Development Board. Specifically, four buttons displayed on the Visor are designed to output unique address and data values. The CoolRunner™ CPLD is then used to register (or latch) these values and blink, or turn off, the LEDs accordingly. This document will detail the C and VHDL code contained in the LED test reference Design. It is intended to help Springboard designers overcome the learning curve associated with the development flow of the CoolRunner XPLA3 CPLD.

XAPP359 Understanding the Insight Springboard Development Kit v1.0 (04/30/01) >

The Insight Springboard™ Development Card is designed such that the Xilinx CoolRunner™ CPLD serves as the central interface between the Handspring Visor™'s Springboard expansion slot and a other external devices (Flash, SRAM, and A/D Converter): see Figure 1. This means the CoolRunner is the only integrated circuit physically attached to the Springboard bus. The other components, namely the Flash, A/D, and SRAM, are also connected to the CoolRunner, but they do not have their data, address, and control lines connected directly to the Springboard expansion slot. Such a scheme is advantageous, but Springboard designers must be careful in order to avoid common pitfalls. This Application Note explains why such a design is preferable and will explain how to avoid common mistakes.

XAPP362 Using the XC9500XV Timing Model v1.0 (08/20/01)

This application note describes how to use the XC9500XV timing model.

XAPP400 Constraining Virtex Design in 2.1i v1.0 (10/01/99)

Constraining a Virtex Design is different in 2.1i compared to older versions of the software. There ar improvements in the Trace, Timing Analyzer, FloorPlanner, Constraints Editor, and other implementation tools to help make the designing procedure easier for Virtex. This paper is devoted to describing some of the simple steps necessary to constraining a Virtex design with the new 2.1i implementation tools. The major focus of this paper is to explain how to constrain with a CLKDLL in Virtex and the new look of the Timing Analyzer Reports.

XAPP401 2.1i FPGA Editor v1.0 (10/13/99)

This application note presents the new, easier to use FPGA Editor and how it differs from the previous version of EPIC. For general FPGA Editor usage, refer to the FPGA Editor Guide. This application note will also cover how to return to EPIC type actions for zoom and pan actions.

XAPP402 2.1i Floorplanner Support for Virtex FPGAs v1.0 (10/13/99)

With the release of M2.1i, the Floorplanner will support the Virtex family of FPGAs. This application note will show you how the major Virtex-specific architectural features such as BlockRAMs, global clock buffers, DLLs, and carry logic are represented within the Floorplanner GUI and how you can manipulate a design containing these elements. The general operation of the 2.1i Floorplanner is identical to that of the Floorplanner in the previous, 1.5i release.

XAPP403 Using the Version 2.1i Xilinx Design Manager and Flow Engine (DMFE) v1.0 (09/27/99)

Welcome to the version 2.1i Xilinx Design Manager (DM) and Flow Engine (FE). The functionality of both DM and FE has been significantly enhanced in this release. In 2.1i, the focus for DM/FE has been to improve "ease of use". A number of new features are provided including "self contained revisions" and the "Smart" Flow Engine, to name a few. These and many other new features are explained in the sections that follow.

XAPP404 Xilinx Alliance 3.1i Modular Design v1.2 (04/23/01)

This application note addresses the Modular Design feature of Xilinx Alliance Series 3.1i. Modular Design allows designers to partition large, complex FPGA projects into several independent pieces which are more easily managed and updated through the product’s lifetime. Design entry, design implementation, and simulation flows for such projects are discussed in detail, with specific instructions and screen graphics provided for using FPGA Express, LeonardoSpectrum, and Synplify software. Illustrative VHDL and Verilog code segments are provided.

XAPP406 Cross Probing to Synplicity and Exemplar v2.0 (12/01/00)

Xilinx Alliance software version 3.3.06i (3.1i Service Pack 6) or later has been enhanced to include logical and timing cross probing to Synplify/Synplify Pro and LeonardoSpectrum. The logical cross probing feature enables the user to select instances or nets in warning or error messages in the Error Viewer to cross probe back to the synthesis tool schematic view. This is useful for debugging a design with logical DRC errors/warnings. The timing cross probing feature enables the user to select a path, nets or instances to cross probe from the timing report within Timing Analyzer back to the synthesis tool schematic view. This feature is useful for analyzing timing problems. These functionalities can be used with Synplify / Synplify Pro version 6.0.0 or later from Synplicity and with LeonardoSpectrum version 2000.1b or later from Exemplar Logic.

XAPP408 Rethinking Your Verification Strategies for Multimillion-Gate FPGAs v1.0 (10/07/00)

Verification is an integral part of any FPGA design project. Many older verification models are no longer appropriate to the new multimillion-gate FPGAs, and more modern methods must be brough to bear if verification is to positively affect product time to market. The methodologies used for designing and implementing a good verification plan are discussed in detail, in the context of a real-world verification case study.

XAPP409 Simulating a Xilinx 3.1i CORE Generator VHDL Design v1.0 (06/18/01)

This application note provides an overview of the files that are generated from the Xilinx CORE Generator™ 3.1i for an HDL project and explains how and when each file is used. This application note briefly explains how to create simulator libraries, map the created libraries, and how to compile the XilinxCoreLib CORE Generator libraries to these user created libraries for the supported simulators.

DataSource CD-ROM Q4-01: Application Note Summaries

XAPP410 Simulating a Xilinx 3.1i CORE Generator Verilog Design v1.0 (06/18/01)

This application note begins with an overview of the steps necessary to include a Xilinx CORE Generator macro in a Verilog design. Next, the Input/Output Files, page 2 describes the Xilinx CORE Generator™ 3.1i files used in a Verilog HDL design project.

XAPP412 Architecting Systems for Upgradability with IRL (Internet Reconfigurable Logic) v1.0 (06/29/01)

Internet Reconfigurable Logic (IRL™) is a system design methodology to enable the remote upgrad of hardware, while insuring the reliability of the upgrade. FPGAs, which are “Field Programmable” ar inherently capable of changing their functionality with a new bitstream. IRL takes advantage of this capability by delivering new bitstreams and software drivers to the remote hardware. This applicatio note will describe the basic concepts of an IRL-enabled system, detail design considerations for building an IRL system and give a high level description of PAVE, the Xilinx API and development framework that enables embedded systems to be upgraded.

XAPP500 J Drive: In-System Programming of IEEE Standard 1532 Devices v1.1 (01/17/01)

The J Drive programming engine provides immediate and direct in-system configuration (ISC) support for IEEE Standard 1532 programmable logic devices (PLDs). To configure an in-system device, the programming engine uses the configuration algorithm information from a 1532 Boundary Scan Description Language (BSDL) file to apply configuration data from the 1532 data file through the IEEE Standard 1149.1 test access port (TAP). The J Drive executable, source code, and a programming example are available in a download package from the Xilinx website. The J Drive programming engine can be used for the following Xilinx families: XC18V00, XC9500XL, XC9500XV, Virtex, and Virtex-E.

XAPP501 Configuration Quick Start Guidelines v1.2 (08/02/01) NEW!

This application note discusses the configuration and programming options for Xilinx Complex Programmable Logic Device (CPLD), Field Programmable Gate Array (FPGA), and PROM families and demonstrates some of the most popular configuration methods used for each family. This document includes configuration quick start guidelines for the Virtex, Spartan, XPLA3, XC9500, XC17S00, and XC18V00 families.