WP151 (v3.0.1) December 17, 2007

System ACE Configuration Solution for Xilinx FPGAs

By: Gary Knipper

Design techniques for electronic systems are constantly changing. In industries at the heart of the digital revolution, this change is especially acute. Functional integration, dramatic increases in complexity, new standards and protocols, cost constraints, and increased time-to-market pressures have bolstered both the design challenges and the opportunities to develop modern electronic systems. One trend driving these changes is the increased integration of core logic with previously discrete functions to achieve higher performance and more compact board designs. Traditionally, ASICs have been the vehicle for such integration but now, with their advanced system capabilities, programmable logic devices (PLDs), especially field programmable gate arrays (FPGAs), have begun to take on this role in system design.

Trends in FPGA Usage

Until recently, PLDs had been used primarily as *glue logic* tying various system functions together and acting as programmable high-speed interface logic. In these instances, FPGA usage was usually limited to one or two devices per system. Now, however, the new generation of FPGAs, with their integration of specialized functionality and expansion of performance and capabilities, are being used as the core of advanced electronic systems. This increasing role of FPGAs at the heart of system development has been spurred, in part, by Xilinx integrating specialized functions and high-performance application and interface circuitries into the fabric of Virtex™ and Spartan™ FPGAs. The flexibility and increase in capabilities of the FPGA relative to ASICs, especially given the pressures of greatly compressed product development cycles, has accelerated the use in systems of multiple FPGAs as their core logic. In addition, the average density of FPGAs designed into new systems is growing rapidly.

The increased usage of FPGAs has led to a growing focus on designing for FPGA configuration. When only one or two FPGAs were used in a system requiring only one configuration bitstream, a dedicated configuration PROM was a fast and simple configuration solution. In those cases, the board space cost taken up by the PROM was offset by the speed and ease of implementing a PROM-based solution. As the number of FPGAs per system, the size of bitstreams, and the need for flexibility in configuration grow, using multiple dedicated PROMs becomes unwieldy. In such a case, it is sometimes more efficient to have a centralized source for configuration of all FPGAs. Traditionally, designers have often used an embedded solution using on-board commodity flash memory controlled by a processor or PLD. When a processor is used, configuration data is pulled directly from system memory over the memory bus and fed to the FPGA chain via a Boundary-Scan (JTAG) interface. Alternatively, PLDs paired with commodity flash memory can be used to configure FPGA chains, supplying bitstream data either serially or in parallel using 8-, 16-, or 32-bit words for faster configuration speeds.

Such embedded FPGA configuration options are often complex design challenges requiring valuable development bandwidth. System engineers must devote design time and effort to developing and testing the microcode for having the processor control configuration. In addition, using the processor to manage general system startup and FPGA configuration simultaneously can delay startup times. There is also a danger of bus contention with the FPGAs competing with other resources for processor and memory access. Embedded solutions also require extra board space for the additional memory used for configuration storage. If Boundary-Scan (JTAG) is used for board test or FPGA programming, separate trace lines and scan-chain-control devices might also be required. In designs using FPGAs and CPLDs as configuration controllers, designers must still add an additional packaged part to the board simply to convert FPGA clocks to address increments and, in serial mode, to serialize the data. The FPGA also requires a separate PROM to configure it as the controller.

System ACE Technology

Given the options described in "Trends in FPGA Usage," page 2 and the growing need for configuration flexibility, designers using multiple FPGAs that need large numbers of configuration bits and enhanced configuration flexibility are faced with a need to make a trade-off: Use a self-contained, pre-engineered multi-PROM solution at the expense of board space or devote engineering development and debug time to design a customized, space-efficient, flexible configuration solution. To meet the need for a space-efficient, pre-engineered, high-density configuration solution for multiple-FPGA systems, Xilinx has developed the System ACE™ (Advanced Configuration Environment) CF product.

The System ACE CF product is designed to meet the growing need for flexible, high-density storage and configuration control. This product combines Xilinx expertise in configuration control logic with industry expertise in high-density, low-cost data storage to provide a complete platform for meeting any configuration need. System ACE CF supports multiple-bitstream management and system reconfiguration/update over a network, providing:

- An easily scalable and reusable configuration platform
- A built-in interface to a system processor, such as Xilinx MicroBlaze™, or Virtex embedded IBM PowerPC™ processors
- Centralized configuration for the entire system
- Simplified debug
- Minimized board space

System ACE CF

System ACE CF uses memory based on the CompactFlash Association standard, providing a very flexible, high-density, two-piece configuration solution comprised of a CF card and the System ACE CF controller. The CF card interface can accommodate removable CompactFlash (CF) cards, currently ranging from 32 MB to 4 GB, or Hitachi Microdrives, currently ranging from 2 GB to 6 GB, all available in the CF form factors and board space requirements. For perspective, individual Virtex-4 FPGAs require from 4.8 Mb to 51 Mb of configuration data, meaning over 900 of the largest members of the Virtex-4 family can be configured with one System ACE CF solution. The use of CF card technology gives system designers access to low-cost, high-density flash memory in a very efficient footprint that does not change with density or new product generations. Thus designers have the flexibility to change the density of CF card memory as needed without any board redesign. Because the CompactFlash memory card technology is a removable medium, making changes or upgrades to the memory contents or density can be done simply by exchanging removable modules or by in-system programming.

The System ACE CF controller has built-in control logic with a variety of specialized interfaces (Figure 1). This device is the interface to the CF card, the JTAG chain containing one or more FPGA devices, an external board test environment, and a system processor. The default configuration mode takes bitstream data from the memory module and configures a chain of FPGAs via JTAG. There is also a Boundary-Scan test and programming interface to aid with system prototyping, testing, and debugging.

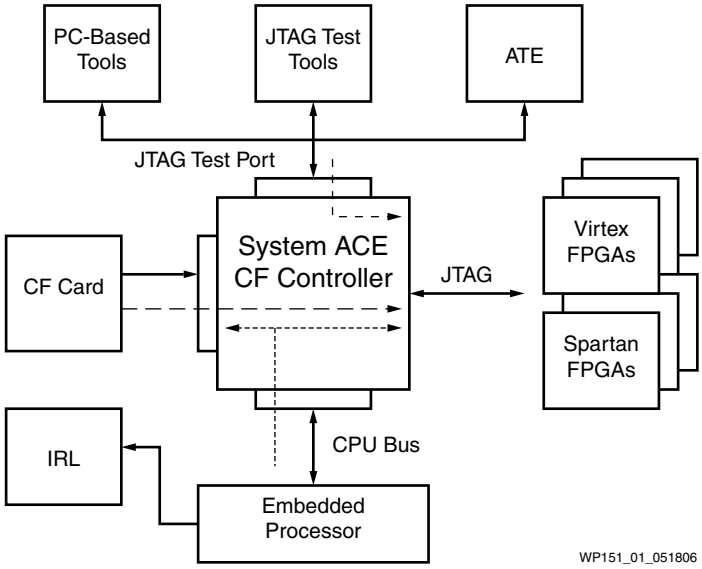


Figure 1: **System ACE Configuration**

With unprecedented density ranging to over 6 GB, one System ACE CF controller can configure hundreds of FPGAs and replace large arrays of configuration PROMs. The designer can also store a large number of different designs for a given array of FPGAs all on the same CF card. Because the System ACE CF controller uses a CF card formatted with the standard FAT file system, non-bitstream files (for example, release notes, technical schematics, documentation) can also be stored on the CF card or excess memory can be allocated for use by the application as system memory.

The three main advantages to the System ACE CF solution are System Configuration Management, Upgrade Management, and Flexible File Management.

System Configuration Management

System ACE CF is the first pre-engineered configuration solution to provide both the bit density and the control logic to manage configuration for all FPGAs within a system from one centralized location (Figure 2). Centralizing configuration management minimizes board space, simplifies design modifications (either during prototyping or in the field), and opens the door for system processors to take a more interactive role in managing re-configuration to increase system flexibility. In systems with multiple boards connected through a backplane, one System ACE CF controller can be used per board to manage each board's FPGA configuration. If one Boundary-Scan chain connects all FPGAs across multiple boards through the backplane, just one System ACE CF controller can be used to configure all FPGAs across these boards.

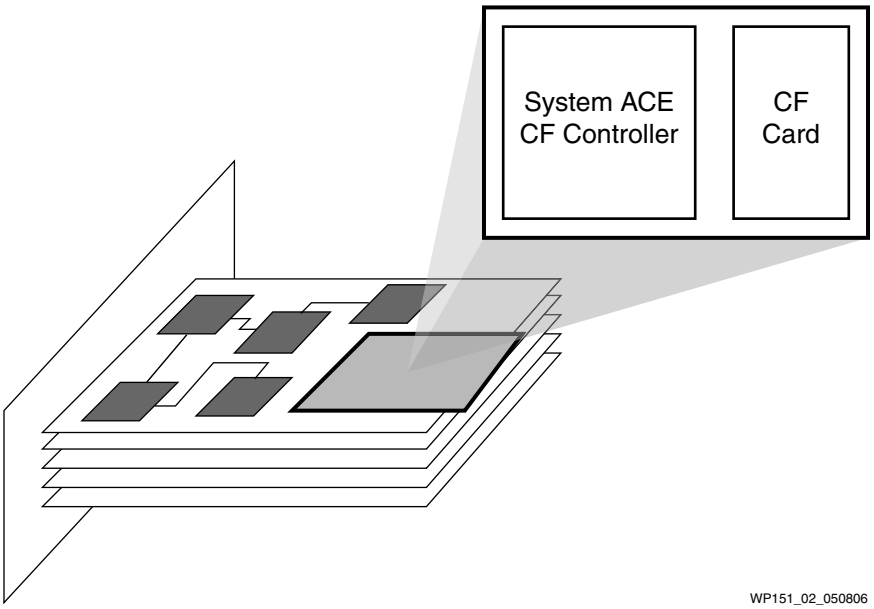


Figure 2: **System ACE Board Management Configuration**

The System ACE CF product handles a variety of configuration management needs. Its flexibility and capacity allow one System ACE CF to configure a board full of FPGAs or multiple boards connected through a backplane. Centralization greatly simplifies configuration management and upgrades. To change or upgrade the configuration of a system, remove the CF card and make the necessary alterations on the desktop PC, adjusting the CF card contents in-system through the processor port, or download a new configuration bitstream over a network using Internet reconfigurable logic (IRL).

The System ACE CF controller also allows for storage of multiple bitstreams at one time in one location. This allows one board design to serve multiple purposes. For example, if slight variations of an FPGA-based system are being shipped to different markets (for example, to accommodate different interface, broadcast, or electrical standards), designers can design a single system for all these markets with the only difference being which bitstream on the CF card is used for system configuration (Figure 3).

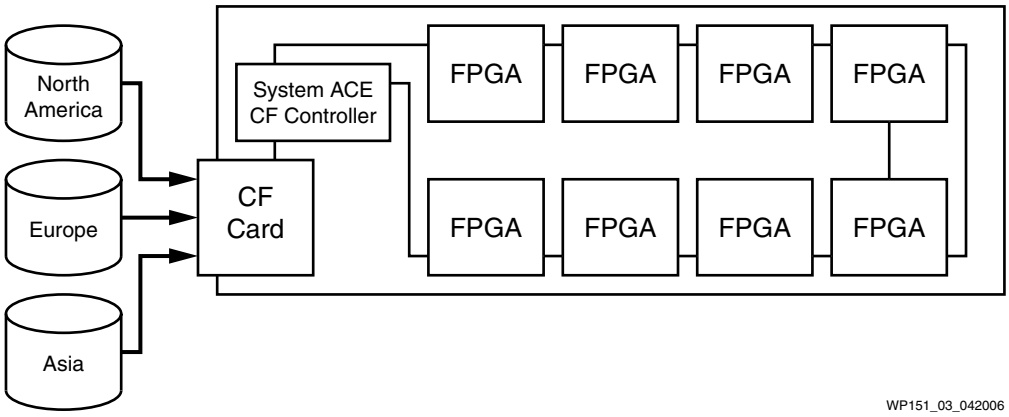


Figure 3: **Example of System ACE and Flash Card Configuration**

The CF card can store multiple versions or configurations of a single design that can be selected at run-time by the System ACE CF controller. For example, during prototyping, operational, test, and debug configurations can be stored in the CF card and different configurations for the design then selected.

Upgrade Management

The System ACE CF solution also greatly simplifies upgrading FPGA-based systems; all that is required is that a new or changed bitstream be stored in the CF card. Because it is centralized, System ACE CF controller allows designers to update their entire system simply by changing the contents of one CF card, either through physical removal or in-system re-programming, making updating and debugging much easier. Whether for a prototyping board in a lab or an installed system in the field, manually reconfiguring an FPGA-based system using the removable CF card requires little effort.

In-system programming can be accomplished either by using a programming cable or through a network interface. Network reconfiguration of the CF card through the System ACE CF controller eliminates the need for a direct interface by enabling users to update or debug their systems remotely by transmitting a new bitstream over a network (for example, internet or wireless WAN). In addition, the ability to store multiple bitstreams and have the processor activate any bitstream at any time allows system administrators to maintain access to previous versions of system configuration for use in the event there is a problem with a newly transmitted configuration bitstream.

Flexible File Management

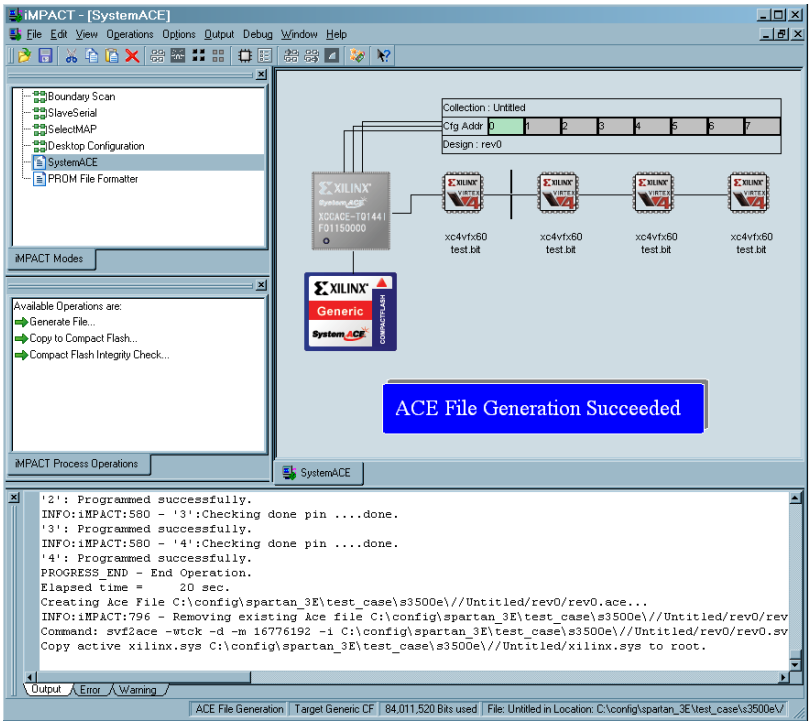
The file structure of the CF card simplifies the storage and management of multiple system configurations. This allows designers to use a single CF card to run multiple BIST patterns, run PCI applications, or store multiple bitstream variations on a single design; for example, versions for North America, Europe, Japan, and China (Figure 3, page 6).

Designers using Xilinx FPGAs with embedded processors can store the FPGA configuration data and the processor application in the same configuration stream, with the System ACE CF controller handling the initialization of both the FPGA logic and embedded processor. System ACE CF can also store multiple applications that are used by the processor core and deliver them as needed. Designers have the flexibility to store related files in the CF card, such as release notes, revision history, user guides, FAQs, or any other supporting files. The processor interface enables the utilization of unused CF card capacity for purposes other than bitstream storage; for example, generic system scratchpad memory.

Designers can use most CF cards available from a variety of third-party suppliers. Please refer to RPT036, *Notification Information System ACE CompactFlash Compatibility* for additional information. Hitachi Microdrives can also be used. The System ACE CF controller comes in a 144-pin TQFP package.

Xilinx Programming Software – iMPACT

The Xilinx iMPACT software tool allows designers to easily perform device configuration and programming either as a batch operation or through a convenient graphical user interface. This software tool can be downloaded from the Xilinx website at www.xilinx.com.



WP151_04_042606

Figure 4: **iMPACT Configuration**

The Xilinx iMPACT software is a full-featured software tool used for configuring and programming all Xilinx PLDs (FPGAs and CPLDs), System ACE CFs, and PROMs. This software tool features a series of design wizards that easily guide the user through each step of the configuration process.

The iMPACT software integrates seamlessly with existing Xilinx programming software and uses a standard file management structure allowing for drag-and-drop file manipulation on the CF card from any Windows PC environment.

Conclusion

With increased usage of multiple FPGAs as the core logic of modern electronic systems, having a pre-engineered, flexible, and robust configuration solution is of growing importance. System ACE technology frees systems engineers from the need to design an FPGA configuration system and allows them to focus their design effort on maximizing system performance and achieving faster time to market.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/25/01	1.0	Initial Xilinx release.
05/18/06	2.0	Updated content and figures throughout.
07/16/07	3.0	Removed the obsolete System ACE SC member from the System ACE family of products.
12/17/07	3.0.1	• Updated document template. • Updated URLs.