A Survey of Dynamically Reconfigurable FPGA Devices

Sumanth Donthi
Department of Electrical and Computer Engineering
Tennessee Technological University
Box. 5004
Cookeville, TN 38505 USA
srd6517@tntech.edu

Roger L. Haggard
Department of Electrical and Computer Engineering
Tennessee Technological University
Box. 5004
Cookeville, TN 38505 USA
rhaggard@tntech.edu

Key Words: FPGA, Dynamic Reconfiguration, Programmable Logic

Abstract—**The FPGA market has evolved at an extremely rapid pace, with larger and faster devices being released to the industry by different vendors. One has to select a target FPGA device that suits one's application. This paper presents a survey of currently available dynamically reconfigurable FPGA devices. The devices were: the Atmel AT40k family, the Xilinx Virtex family, the Lattice Semiconductors ORCA and ispXPGA families, and the Altera APEX20k family. The degree of dynamic reconfigurability of these devices was compared based on partial or full reconfiguration. The architectures of these devices were evaluated based on the granularity and reconfiguration time.**

I. INTRODUCTION

Field Programmable Gate Arrays (FPGAs) are one of the fastest growing parts of the digital integrated circuit market in recent times. They can be configured to implement complex hardware architectures. FPGA reconfiguration typically requires the whole chip to be reprogrammed even for the slightest circuit change [1]. Dynamic reconfiguration means modifying the system when it is under operation. Dynamically reconfigurable FPGA systems can adapt to various computational tasks through hardware reuse. Numerous architectures that are dynamically reconfigurable have been proposed in the recent past. Unlike designers of static non-reconfigurable systems, designers of dynamically reconfigurable systems are required to analyze various design characteristics simultaneously. The search for a good design solution requires a thorough analysis of how the system can be dynamically reconfigured. This includes system throughput, reconfiguration time, sharing of reconfigurable resources in different configurations, size of the configuration data, selecting the target FPGA device, etc.

Out of the aforementioned considerations, selecting a target FPGA device is important in all dynamically reconfigurable system applications. The chosen target FPGA device should provide large amounts of hardware resources and it should be highly flexible so as to yield optimal performance. In currently known dynamically reconfigurable devices, the main performance overhead is the speed of dynamic reconfiguration. The speed of dynamic reconfiguration is directly proportional to the number of configuration memory locations that need to be changed in order to implement the desired design modification.

In section II, we briefly discuss the architecture of the FPGA devices. In section III, we discuss the configuration of the FPGA devices. In section IV, we discuss the reconfiguration time of the FPGA devices. Section V gives possible future work. Section VI gives the conclusion and, finally, the references are listed in section VII.

II. ARCHITECTURE OF DEVICES

The basic architecture of any FPGA device consists of an array of logic blocks surrounded by programmable I/O blocks, and connected with a programmable interconnect. There are two main architectures for FPGAs, namely fine-grained and coarse grained. A fine-grained architecture consists of a large number of small logic blocks (e.g. sea of gates, transistors, smaller macro cells) whereas a coarse-grained architecture consists of a smaller number of larger, more powerful logic blocks (e.g. flip-flops and LUTs). The architecture of the surveyed devices is discussed briefly in the following subsections.

A. Xilinx Virtex Architecture [2]

The Xilinx Virtex architecture is coarse grained. The basic cell of the Virtex FPGA is configurable logic block (CLB). Virtex CLBs are organized in rows and columns. Each CLB consists of four Logic Cells (LCs) organized in two slices. In addition to the four basic LCs, the Virtex CLB contains logic that combines function generators that are implemented as 4-input look up tables (LUTs). Apart from operating as a function generator, each LUT can also provide a 16x1-bit single or dual port synchronous RAM. The Virtex

0-7803-7697-8/03/$17.00 ©2003 IEEE

FPGA uses Block SelectRAM memory blocks that are organized in columns. The Virtex XCV1000 device has 32 Blocks and a total of 131,072 Block SelectRAM bits with each Block SelectRAM cell containing 4096 bits. Each Virtex CLB has 4 storage elements that can be configured either as edge-triggered D-type flip-flops or as level-sensitive-latches. The D inputs can be driven either by the function generators within the slice or directly from the slice inputs, bypassing the function generators. The Virtex device also contains programmable input/output blocks (IOBs) interconnected to the CLBs by fast, versatile routing resources. The availability of these routing resources permits the Virtex family to accommodate large, complex designs.

B. Lattice ORCA Architecture [3]

The ORCA series is a coarse grained architecture that consists of four basic elements: programmable logic cells (PLCs), programmable input/output cells (PIOs), embedded block RAMs (EBRs), and system-level features. The PLCs are arranged in an array of rows and columns, and the whole array is split into four quadrants. PIOs are located on all four sides of the FPGA. Every group of 4 PIOs on the device edge has an associated Programmable Interconnect (PIC). The PLC consists of a Programmable Function Unit (PFU), System Level Interconnect (SLIC), and routing resources. Each PFU within a PLC contains eight 4-input (16-bit) LUTs, eight latches/FFs and one additional flip-flop for arithmetic functions. The PFU is the basic logic element of the PLC, containing elements for both combinational and sequential logic. The PFU uses two sets of four LUTs and FFs that can be controlled independently. The twin-quad architecture of an LUT provides a facility to implement from one to eight independent combinational logic functions and a large number of complex logic functions using multiple LUTs. The flexibility of the LUT to handle wide input functions, as well as multiple smaller input functions, maximizes the gate count per PFU while increasing the speed. All the basic elements are interconnected with a routing fabric of both local and global wires.

C. Lattice ispXPGA Architecture [4]

The architecture of the ispXPGA device is symmetric and coarse grained. It consists of an array of Programmable Function Units (PFUs) enclosed by Programmable Interconnects (PICs) with columns of embedded block RAMs (EBRs) distributed throughout the array. Each PIC has two corresponding sysIO blocks, each of which includes one input and output buffer. On the two sides of the device, between PICs and the sysIO blocks, there are sysHSI high-speed interface blocks. The symmetrical architecture allows designers to easily implement their designs, since any logic function can be placed in any part of the device. The PFU is the basic building block of the ispXPGA architectures; they are arranged in rows and columns. Each PFU consists of four Configurable Logic Elements (CLEs), four Configurable Sequential Elements (CSEs), and a Wide Logic Generator (WLG). By utilizing these components, the PFU can implement arithmetic, memory and register functions quickly and efficiently. The CLE is made up of a four-input look-up table (LUT-4), a Carry Chain Generator (CCG) and a two-input AND gate. The LUT-4 creates combinational logic and memory elements, the CCG creates a single one-bit full adder, and the AND gate can expand the CCG to incorporate the Booth Multiplier capability by feeding the output of the AND gate to one of the inputs of the CCG.

The PICs interface the PFUs and EBRs to the external pins of the device. They allow connections directly to the different logic elements for fast access to the combinatorial functions. The sysMEM EBRs are large, fast access memory elements that can be configured as RAM, ROM, FIFO, and other storage types. They are designed to facilitate both single and dual-port memory for high-speed applications. These three components of the architecture are interconnected via a high-speed, flexible routing array. The routing array consists of variable length interconnect (VLI) lines between the PICs, PFUs and EBRs. There are additional routing lines available to each PFU for feedback and direct routing of signals to adjacent PFUs or PICs.

D. Atmel AT40K Architecture [5]

The Atmel AT40K architecture is a symmetrical array of identical cells and is fine grained. The array is continuous from one edge to the other, except for the bus repeaters spaced every four cells. At the intersection of each repeater row and column there is a 32x4 RAM block accessible by adjacent buses. The Atmel AT40K FPGA offers distributed 10 ns SRAM capability, where RAM can be used without losing logic resources. The RAM can be configured as either a single-port or dual-port RAM, with either synchronous or asynchronous operation. The AT40K FPGA offers 8-sided core cells with direct horizontal, vertical, and diagonal cell-to-cell connection that implements ultra fast array multipliers without using busing resources. The cells in the Atmel array are small and efficient, and they can implement any pair of Boolean functions of three inputs or any single Boolean function of four inputs. The cell's small size leads to arrays with large numbers of cells, greatly multiplying the functionality in each cell. A simple high-speed busing network provides fast, efficient communication over medium and long distances.

E. Altera APEX20K Architecture [6]

The Altera APEX20K architecture is coarse grained. These devices combine LUT-based logic, product-term-based logic, and memory into one device. Signal interconnections within APEX20K devices are provided by the Fast Track

Interconnect, which consists of fast, continuous row and column channels that run the entire length and width of the device. APEX20K devices are constructed from an array of MegaLAB structures. Each MegaLAB structure consists of a group of logic array blocks (LABs), one Embedded System Block (ESB), and a MegaLAB interconnect which routes signals within the MegaLAB structure. The largest device (EP20K1500E) has 216 MegaLABs. A MegaLAB consists of 24 LABs and each LAB consists of 10 Logic Elements (LEs), the associated carry and cascade chains, LAB control signals, and local interconnect. The local interconnect transfers signals between LEs in the same or adjacent LABs, input/output elements (IOEs), or ESBs. Each LE can drive 29 other LEs through the Fast Track Interconnect. Each LE contains a four-input LUT that can implement any function of four variables. Each LE's programmable register can be configured for D, T, JK, or SR operation. The ESB can implement a variety of memory functions, including RAM, dual-port RAM, ROM, and FIFO functions. Embedding the memory directly into the die improves performance and reduces the die size compared to distributed-RAM implementations.

Table 1 shows the characteristics of the surveyed FPGA devices, including the device family, the largest device available, the basic cell capacity, the typical number of gates, the maximum number of user I/O pins, the technology, and the granularity.

Table 1: Characteristics of FPGA Devices

Vendor	Xilinx	Lattice semiconductors		Atmel	Altera
Device Family	Virtex, Virtex-E	ORCA	ispXPGA	AT40K	Apex20K, Apex20KE
Largest Device	XCV812E	OR4E06	ispXPGA 1200	AT40K40LV	EP20K1500E
Capacity, Basic cell	21,168 LCs	16,192 LUTs	15,376 LUTs	2,304 cells	512,840 LEs, 24LABs
Typical Gates	254,016 logic gates	471K-899K	1200K	5K-50K	1500K
Max User IO pins	556	466	496	384	808
Technology	SRAM	SRAM	SRAM	SRAM	SRAM
Granularity	Coarse Grained	Coarse Grained	Coarse Grained	Fine Grained	Coarse Grained

III. DYNAMIC RECONFIGURATION OF DEVICES

FPGAs are hardware circuits that can be modified at almost any point during use. In FPGAs, both the logic functions performed within the logic blocks and the interconnections between the blocks can be altered by sending signals to the chip. In dynamically reconfigurable FPGAs, the configurable logic blocks can be rewired and reprogrammed repeatedly even when a portion of the application is active.

Dynamic reconfiguration is classified into two categories. The first category is whether the chip is partially reconfigurable or fully reconfigurable. If only a portion of the chip is modified and the remaining logic operates normally without any disruption, then it is partially reconfigurable. If the whole chip is modified at once, with a total loss of the previous configuration and the state of the internal flip-flops, then it is fully reconfigurable.

The second category reflects how the dynamic reconfiguration is achieved. If the reconfiguration is achieved via a serial or a parallel interface, where the configuration data is moved from an external memory device into the chip configuration memory, then it is called off-chip (serial or parallel) reconfiguration. If the reconfiguration is achieved by swapping the full or partial contents of the configuration memory with the contents of an on-chip context memory, then it is called context reconfiguration. Reconfiguration of the devices will be discussed in the following subsections.

A. Xilinx Virtex Configuration [7]

The Virtex FPGA can be either fully configurable or partially reconfigurable. The partial reconfiguration is useful for applications that require the loading of different designs into the same area of the device or the flexibility to change portions of a design without having to either reset or completely reconfigure the device. Partial reconfiguration of a Virtex FPGA device is accomplished in one of two modes, either Slave SelectMAP mode or Boundary Scan mode (JTAG). For current FPGA devices, data is loaded on a column basis, with the smallest load unit being a configuration bitstream "frame," which varies in size based on the target device.

There are two styles of partial reconfiguration in the Virtex FPGA, namely Module-Based partial reconfiguration and Small-Bit Manipulations. In Module-Based partial reconfiguration, distinct portions of an FPGA are referred to as reconfigurable modules. The height of the reconfigurable module is always the full height of the device. The width of the reconfigurable module ranges from a minimum of four slices to a maximum of the full-device width. The number of reconfigurable modules should be minimal (i.e. a single reconfigurable module) to reduce problems in complex designs. This type of partial reconfiguration is used for independent design applications and for modules that communicate with each other using a special bus macro. Each time the partial reconfiguration is performed, the bus macro is used to establish unchanging routing channels between the modules, guaranteeing correct connections.

In Small-Bit Manipulations, the partial reconfiguration is accomplished by making a small change to the design, and then generating a bitstream based on only the difference between the two designs. Switching the configuration of a module from one implementation to another is very fast, as the bitstream difference can be much smaller than the entire device bitstream. In the Virtex device, the configuration bitstreams are used in altering the on-chip data. This is off-chip reconfiguration.

B. Lattice ORCA Configuration [3]

The ORCA series FPGA family supports partial reconfiguration as well as full reconfiguration. The reconfiguration style in ORCA series FPGAs is off-chip. The function of an ORCA series FPGA depends on the state of the internal configuration RAM. Partial reconfiguration is done by setting a bitstream option in the previous configuration sequence that tells the FPGA to not reset the entire configuration RAM during a reconfiguration. Then only the configuration frames that are to be modified need to be rewritten, thereby reducing the reconfiguration time.

Other bitstream options are also available that allow one portion of the FPGA to remain in operation while a partial reconfiguration is done. If this is option is used, one must be careful to not cause disruption between the two configurations (the bitstream resident in the FPGA and the partial configuration bitstream) as the second reconfiguration bit stream is being loaded.

C. Lattice ispXPGA Configuration [4]

The ispXPGA device is either partially reconfigurable or fully reconfigurable. The reconfiguration style in the ispXPGA is both off-chip and context. The configuration method in ispXPGA is different from all the other devices. The ispXPGA device consists of two types of memory, static and non-volatile E^2CMOS cells. The static RAM is used to control the functionality of the device during the normal operation and the E^2CMOS memory cells are used to load the SRAM. The E^2CMOS module can be thought of as the hard drive for the ispXPGA configuration and the SRAM as the working configuration memory. The SRAM is configured either from the E^2CMOS memory or from an external source. There are two possible ports that can be used for the configuration of the SRAM memory; the IEEE standard 1149.1 Test Access Port (TAP) that accommodates bit-wide configuration and the Peripheral port that allows byte-wide configuration [4]. When programming the E^2CMOS memory, only the 1149.1 TAP can be used.

D. Atmel AT40K Configuration [8]

The AT40K device can be configured any number of times. The entire device or select portions can be configured. The reconfiguration data in Atmel devices is read from an external memory device (off-chip configuration). Configuration data is transferred to the AT40K device in one of six modes. There is one auto-configuring Master mode, four Slave modes, and a Synchronous RAM Mode for accessing SRAM-based configuration memory directly from a parallel microprocessor port.

The Master mode is auto-configuring; that is, after power-on-reset and the clearing of configuration memory, it self-initiates configuration. The Master Mode uses an internal oscillator to provide the configuration clock (CCLK) for clocking external EEPROMs that contain the configuration data. After auto-configuration is complete, the user can initiate reconfiguration manually. In the slave modes, configuration is always initiated by an external signal. In Slave Serial Mode the device receives serial configuration data and in Slave Parallel mode, the device receives either 8-bit wide or 16-bit wide parallel data. In Synchronous RAM Mode, the device receives a 32-bit wide bitstream composed of a 24-bit address and an 8-bit data word. The FPGA configuration SRAM is seen as a simple memory-mapped address-space. The user has full read and write access to the entire FPGA configuration SRAM. This mode helps in achieving faster reconfiguration.

E. Altera APEX20K Configuration [9]

The Altera APEX20K device supports only full configuration. During device operation, the APEX20K device stores its configuration data in SRAM cells. The SRAM cells must be loaded with configuration data each time the device powers up because SRAM is volatile. The configuration data for the APEX20K device can be loaded using active or passive configuration schemes. When using an active configuration scheme with a configuration device, both the target device and the configuration device generate control and synchronization signals. When both devices are ready to begin configuration, the configuration device sends data to the APEX20K device. When using a passive configuration scheme, the APEX20K device uses a microprocessor that controls the configuration process. The microprocessor supplies configuration data from a storage device such as hard disk, RAM, or other system memory. Thus, reconfiguration in the APEX20K FPGA is off-chip.

IV. RECONFIGURATION TIMES OF DEVICES

The reconfiguration time is inversely proportional to the speed of the dynamic reconfiguration. An FPGA device with shorter reconfiguration and execution times will do more work faster than an FPGA device with longer reconfiguration and execution times. The reconfiguration times will be reduced significantly for partial reconfiguration when compared to full reconfiguration of the device. If the device

is divided into smaller blocks and all these blocks configured in parallel with a common configuration clock, the reconfiguration time can also be reduced.

In the Virtex FPGAs each device has a different frame size. For example a Virtex 300K gate chip has a frame size of 672 bits, and a configuration clock (CCLK) of 50 MHz. Loading just one frame of data in the Select MAP Mode with an 8-bit bus at 50 MHz will take 1.68μsec. The full bit stream size in this device is 218,976 bytes, so, using the same CCLK, a full reconfiguration of the device takes 4.37 milliseconds [2].

In the Atmel AT40K05 device, the size of an 8-bit configuration bitstream for the AT40K05 device is 5263 bytes, and with a configuration clock of 1 MHz, the configuration time is 5.263 milliseconds [5].

In the Lattice ORCA OR4E06 device, using the slave parallel mode 8-bit interface and a CCLK of 100MHz, partial reconfiguration takes 1.9μsec/frame (one frame = 193 bytes). The OR4E06 device needs a total of 3076 frames or 5,933,668 bytes. So full reconfiguration takes 5.94 milliseconds .The Lattice ispXPGA FPGAs with only SRAM will take 300milliseconds to configure the device from an external memory source [3].

In the Altera APEX20K family, the EP20K100 device has a data size of 123 Kbytes, and with a CCLK of 10 MHz, the configuration time is 12.3 milliseconds [9].

V. FUTURE WORK

All dynamically reconfigurable devices that are discussed in this paper are of the same technology, and the majority of them have a coarse-grained architecture. In the future, a wide variety of FPGA devices that have different architectures and different technologies will be surveyed. An application will be implemented using both partial and full reconfiguration on two or more devices and the system performance will be compared.

VI. CONCLUSION

Among the surveyed dynamically reconfigurable devices, all except the Atmel AT40K device have a coarse-grained architecture. The Xilinx Virtex, Lattice ORCA series, Lattice ispXPGA and Atmel AT40K FPGAs can be either partially or fully reconfigured, while the Altera APEX20K FPGA is only fully reconfigurable. The reconfiguration in the Xilinx Virtex, Lattice ORCA series, Atmel AT40K, and Altera APEX20K FPGAs is off-chip, while that in the Lattice ispXPGA is both off-chip and context. The partial reconfiguration of all the devices takes less time than full configuration. Typical configuration times for most of the devices were given in the previous section. However, the partial reconfiguration time of the Atmel AT40K05 and the reconfiguration time of the Lattice ispXPGA devices were unknown. Thus, we have shown that there are a wide variety of dynamically reconfigurable FPGA devices available today that are suitable for many current applications.

VII. REFERENCES

[1] Raghavan Venugopal and R. L. Haggard, "Techniques to Enhance the Performance of Dynamically Reconfigurable FPGA-based Systems," *Proceedings of the 31st Southeastern Symposium on Systems Theory,* 2001.

[2] *Virtex™ 2.5 V Field Programmable Gate Arrays: Product Specification,* Xilinx, San Jose, CA., 2002.

[3] *ORCA Series 4 FPGAs: Datasheet,* Lattice Semiconductor, Hillsboro, OR., 2002.

[4] *IspXPGA™ Family: Advanced Datasheet,* Lattice Semiconductor, Hillsboro, OR., 2002

[5] *AT40K Family: Datasheet,* Atmel, San Jose, CA., 2002.

[6] *APEX20K Family: APEX20K Programmable Logic Device Family Datasheet,* Altera, San Jose, CA., 2002.

[7] *Virtex Families: Two Flows for Partial Reconfiguration: Module Based or Small Bit Manipulations,* Xilinx, San Jose, CA., 2002.

[8] *AT40K Family: AT40K series Configuration Datasheet,* Atmel, San Jose, CA., 2002.

[9] *APEX20K Family: Configuration Devices for SRAM-based LUT Devices,* Altera, San Jose, CA., 2002.