International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Recent Advances in the Concept of Field Programmable Gate Array (FPGA)

Sunil Kumar P.[1], C. P. Rajendra[2], T. V. Vidya[2, 5], Dr. T. C. Manjunath[3], Pavithra G.[4]

[1]UG B.E. Student, USN: 1NH05EC407
Dept. of E & CE, New Horizon College of Engg., Bangalore, Karnataka
[2]Ex-faculties of NHCE, Dept. of E & CE, New Horizon College of Engg., Bangalore, Karnataka
[5]& currently, Faculty, CMRIT, Bangalore, Karnataka
[3]Ex-Prof. & HOD, ECE, NHCE, Bangalore & currently, Prof. & HOD, ECE, DSCE, Bangalore, Karnataka
[4]VTU RRC Ph.D. Full Time Research Scholar, Belagavi, Karnataka

Abstract: ***In this paper, some of the recent advances in the concept of FPGA is presented, which is a presentation made as a part of the UG seminar work. Designing hardware with Field-Programmable Gate Arrays (FPGAs) avoids the higher non-recurring engineering costs and longer development times of Mask-Programmed Gate Array (MPGA) solutions. These two key advantages, shorter design cycles and lower development costs, have made FPGAs an extremely popular technology for prototyping and low-volume production runs. The work presented in this paper is the seminar related work of the under-graduate student & it is just a survey paper & as such there is no novelty in it & is presented as a part & parcel of the UG curriculum seminar work EC85.***

Keywords: Hardware, FPGA, HDL, VHDL, Code

1. Introduction

In this part, a quick review of the related work is presented. Field-programmable gate array (FPGA) is a semiconductor device that can be configured by the customer or designer after manufacturing—hence the name "field-programmable". FPGAs are programmed using a logic circuit diagram or a source code in a hardware description language (HDL) to specify how the chip will work. They can be used to implement any logical function that an application-specific integrated circuit (ASIC) could perform, but the ability to update the functionality after shipping offers advantages for many applications.

FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together"—somewhat like a one-chip programmable breadboard. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

2. History

The FPGA industry sprouted from programmable read only memory (PROM) and programmable logic devices (PLDs). PROMs and PLDs both had the option of being programmed in batches in a factory or in the field (field programmable), however programmable logic was hard-wired between logic gates.

Xilinx Co-Founders, Ross Freeman and Bernard Vonderschmitt, invented the first commercially viable field programmable gate array in 1985 – the XC2064. The XC2064 had programmable gates and programmable interconnects between gates, the beginnings of a new technology and market. The XC2064 boasted a mere 64 configurable logic blocks (CLBs), with two 3-input lookup tables (LUTs). More than 20 years later, Ross was entered into the National Inventors Hall of Fame for his invention.

Some of the industry's foundational concepts and technologies for programmable logic arrays, gates, and logic blocks are founded in patents awarded to David W. Page and LuVerne R. Peterson in 1985. In the late 1980s the Naval Surface Warfare Department funded an experiment proposed by Steve Casselman to develop a computer that would implement 600,000 reprogrammable gates. Casselman was successful and the system was awarded a patent in 1992.

Xilinx continued unchallenged and quickly growing from 1985 to the mid-1990s, when competitors sprouted up, eroding significant market-share. By 1993, Actel was serving about 18 percent of the market. The 1990s were an explosive period of time for FPGAs, both in sophistication and the volume of production. In the early 1990s, FPGAs were primarily used in telecommunications and networking. By the end of the decade, FPGAs found their way into consumer, automotive, and industrial applications.

FPGAs got a glimpse of fame in 1997, when Adrian Thompson merged genetic algorithm technology and FPGAs to create a sound recognition device. Thomson's algorithm allowed an array of 64 x 64 cells in a Xilinx FPGA chip to decide the configuration needed to accomplish a sound recognition task.

3. Modern Development

A recent trend has been to take the coarse-grained architectural approach a step further by combining the logic blocks and interconnects of traditional FPGAs with embedded microprocessors and related peripherals to form a complete "system on a programmable chip". This work mirrors the architecture by Ron Perlof and Hana Potash of Burroughs Advanced Systems Group which combined a

reconfigurable CPU architecture on a single chip called the SB24. That work was done in 1982. Examples of such hybrid technologies can be found in the Xilinx Virtex-II PRO and Virtex-4 devices, which include one or more PowerPC processors embedded within the FPGA's logic fabric. The Atmel FPSLIC is another such device, which uses an AVR processor in combination with Atmel's programmable logic architecture.

An alternate approach to using hard-macro processors is to make use of "soft" processor cores that are implemented within the FPGA logic. As previously mentioned, many modern FPGAs have the ability to be reprogrammed at "run time," and this is leading to the idea of reconfigurable computing or reconfigurable systems — CPUs that reconfigure themselves to suit the task at hand. The Mitrion Virtual Processor from Mitrionics is an example of a reconfigurable soft processor, implemented on FPGAs. However, it does not support dynamic reconfiguration at runtime, but instead adapts itself to a specific program.

Additionally, new, non-FPGA architectures are beginning to emerge. Software-configurable microprocessors such as the Stretch S5000 adopt a hybrid approach by providing an array of processor cores and FPGA-like programmable cores on the same chip.

4. FPGA Features

- Besides primitive logic elements and programmable routing, some FPGA families add other features
- Embedded memory-Many hardware applications need memory for data storage. Many FPGAs include blocks of RAM for this purpose
- Dedicated logic for carry generation, or other arithmetic functions
- Phase locked loops for clock synchronization, division, multiplication.
- System on chip-System on Chip is a skill that is in great demand. This design methodology is becoming increasingly popular as FPGA costs fall.
- FPGA technology allows you to embed a processor, ROM, RAM, DSP, and any other block onto a single chip
- This is replacing a lot of Application Specific Integrated Circuit chips
- This has major advantages for electronics companies in terms of cost, reliability, reusability of intellectual property, and time to market
- Embedded Application-Specific Blocks
- Almost all modern commercial FPGAs include application-specific blocks
- Embedded memory
- From some bits to several kilobytes / block
- Largest Altera Stratix III: 2.034 MB in 1,088 blocks
- Hardwired multipliers (DSP blocks)
- For example, 18x18-bit multiplier
- DSP applications
- Even hardwired processor cores
- 2 PowerPC cores on Xilinx Virtex-4
- Ethernet MACs, PLLs, bit-file encryption, AD/DA converters, etc.

5. FPGAs may have an edge

Architecture well-suited for large designs, in parallellized code, computational circuitry per computational abstraction can be

- Confined to a small area (locality of computation)
- Wire delays are less of a bottleneck less need for complex instruction issue logic or bypasses. Regular array structure builds in redundancy
- Increased yield - if a cell has a defect, simply don't use it!
- Possibly 100% yield per wafer -> lower cost per part even though each part has many
- Millions of transistors
- Can include distributed embedded dram blocks for extremely high bandwidth
- Operations (tens of gigabytes/second per die)

6. FPGA Architecture

FPGAs Structure:

The interior of FPGAs typically contain three elements that are programmable:

- Programmable Logic Blocks
- Programmable I/O Blocks
- Programmable Routing resources

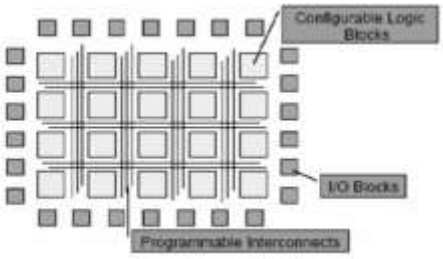

Figure 6.1: Shows the typical FPGA structure

In general an FPGA will have an array of uncommitted logic blocks, programmable wires and programmable switches to realize any function out of the logic blocks and to implement any interconnection topology.

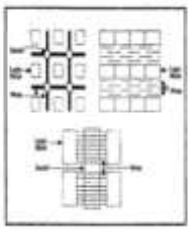

Figure 6.2: Illustrates the layouts of commercially available FPGAs

Several programmable logic blocks are distributed in the FPGA. These logic blocks are surrounded by I/O interface blocks. These I/O blocks can be considered to be on the periphery of the chip. They connect the logic signals to FPGA pins. The space between the logic blocks is used to route connections between the logic blocks. The "field"

programmability in FPGA is achieved by reconfigurable elements which can be programmed or reconfigured by the user. Programmable logic blocks are created by using multiplexers, look-up tables, AND-OR or NAND-NAND arrays. “Programming” them means changing the input or control signals to the multiplexers, changing the look-up table contains or selecting/not selecting particular gates in AND-OR gate blocks. For Programmable interconnect, “programming” means making or breaking specific connections. This is required to interconnect various blocks in the chip and to connect the specific I/O pins to specific logic blocks. Programmable I/O blocks denote blocks which can be programmed to be Input or output, or bi-directional lines. Typically they can also be “programmed” to adjust their properties of their buffers such as inverting or non-inverting, tristate, passive pull-up, or even adjust the slew rate, which is the rate of change of signals on that pin.

7. Configuration Memory Cell

The static memory cell used for the configuration memory in the FPGA has been designed especially for high reliability and noise immunity. Integrity of the device configuration memory based on this design is assured even under adverse conditions. As shown in figure, the basic memory cell consists of two CMOS (Complementary Metal Oxide Semiconductor) inverters plus a pass transistor used for writing and cell data. The cell is only written during configuration and only read during read back. During normal operation, the cell provides continuous control and the pass transistor is off and does not affect cell stability. This is quite different from the operation of conventional memory devices, in which the cells are frequently read and written. It is loaded with one bit of configuration program and controls one program selection in the FPGA.

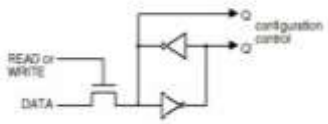

Figure 7.13: illustrates the Configuration Memory Cell

The memory cell outputs Q and Qbar use ground and VCC levels and provide continuous, direct control. The additional capacitive load together with the absence of address decoding and sense amplifiers provides high stability to the cell. Due to the structure of the configuration memory cells, they are not affected by extreme power-supply excursions or very high levels of alpha particle radiation. In reliability testing no soft errors have been observed even in the presence of very high doses of alpha radiation. The method of loading the configuration date is selectable. Two methods use serial data, while three use byte-wide data. The internal configuration logic utilizes framing information, embedded in the program data by the development system, to direct memory-cell loading. The serial-data framing and length-count preamble provide programming compatibility for mixes of various FPGA device devices in a synchronous, serial, daisy-chain fashion.

8. Different FPGA Architecture

FPGAs use a variety of architectures. The FPGA architecture refers to the manner or topologies in which the logic blocks and interconnect resources are distributed inside the FPGA. If one examines the various FPGAs that have been in the market since their inspection in late 80s, One could classify them into 4 different basic architectures:

- Matrix based (Symmetrical array) architectures
- Row based architectures
- Hierarchical PLD architectures
- Sea of gates architectures

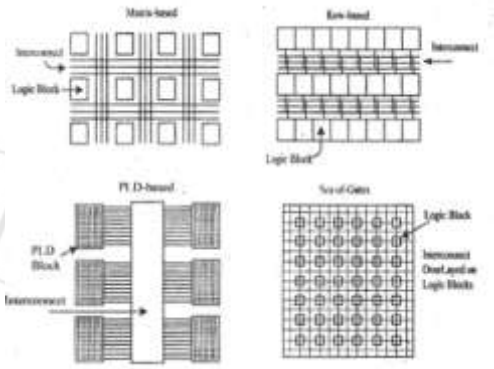

Figure 8.1: Illustrates the matrix based, row based, hierarchical PLD &SEA of gates architectures

9. Matrix Based (Symmetrical Array) Architectures

The logic blocks in the architecture are organized in a matrix-like fashion as illustrated in figure. Most Xilinx FPGAs belong to this category. The logic blocks in these architectures are typically fairly big (capable of implementing 4-variable functions or more). These architectures typically contain 8X8 arrays in the smaller chips and may contain more than 100X100 arrays in the larger chips. The routing resources are interspersed between the logic blocks. The routing in this architecture is often called 2-dimentional channeled routing since routing resources are generally available in horizontal and vertical directions.

Row Based Architectures: These architectures were inspired by traditional gate arrays and standard cell designs. The logic blocks in this architecture are organized into rows as illustrated in the figure. Thus there are rows of logic blocks and routing resources. The routing resources interspersed between the rows can be used to interconnect the various logic blocks. Traditional mask programmable gate arrays and standard cell designs used very similar architectures. The routing in these architectures is often called 1- dimensional channeled routing. Actel FPGAs typically employee this architecture.

Hierarchical PLD Architectures: The architecture of these FPGAs is not very different from CPLDs. When Xilinx came up with the XC2000 FPGAs, many PLD manufactures felt that they could achieve similar capability if they integrate

several PLDs with a programmable interconnect. Altera immediately introduced devices by putting together several PLDs with a programmable interconnect. The logic blocks are of fairly large grain because they are essentially PLDs. The routing in these architectures is often via a few large scale switching networks instead of individual programmable points distributed throughout the chip.

SEA of Gates Architectures: The sea of gates architecture is yet another manner to organize the logic blocks and interconnect in an FPGA. Typically the logic blocks are fine-grain and there are a lot of them spread throughout the chip. Then there is an interconnect superimposed on the sea of gates as illustrated in the figure. Plessey, a manufacturer who was in the market in the mid-90s used to make FPGAs of this architecture. The basic cell used was a NAND gate, in contrast to the larger basic cells used by manufacturers like Xilinx.

10. Basic Process Technology Types

- SRAM - based on static memory technology. In-system programmable and re-programmable. Requires external boot devices. CMOS.
- Antifuse - One-time programmable. CMOS.
- EPROM - Erasable Programmable Read-Only Memory technology. Usually one-time programmable in production because of plastic packaging. Windowed devices can be erased with ultraviolet (UV) light. CMOS.
- EEPROM - Electrically Erasable Programmable Read-Only Memory technology. Can be erased, even in plastic packages. Some, but not all, EEPROM devices can be in-system programmed. CMOS.
- Flash - Flash-erase EPROM technology. Can be erased, even in plastic packages. Some, but not all, flash devices can be in-system programmed. Usually, a flash cell is smaller than an equivalent EEPROM cell and is therefore less expensive to manufacture. CMOS.
- Fuse - One-time programmable. Bipolar.

11. Virtex™-II Platform FPGAS

The Virtex-II family is a platform FPGA developed for high performance from low-density to high-density designs that are based on IP cores and customized modules. The family delivers complete solutions for telecommunication, wireless, networking, video, and DSP applications, including PCI, LVDS, and DDR interfaces. The leading-edge 0.15 µm / 0.12 µm CMOS 8-layer metal process and the Virtex-II architecture are optimized for high speed with low power consumption. Combining a wide variety of flexible features and a large range of densities up to 10 million system gates, the Virtex-II family enhances programmable logic design capabilities and is a powerful alternative to mask-programmed gates arrays. As shown in Table 11.1, the Virtex-II family comprises 11 members, ranging from 40K to 8M system gates.

Device	System Gates	CLB (1 CLB = 4 slices = Max 128 bits) Array Row x Col.	CLB Slices	CLB Maximum Distributed RAM Kbits	Multiplier Blocks	SelectRAM Blocks 18 Kbit Blocks	SelectRAM Blocks Max RAM (Kbits)	DCMs	Max I/O Pads[1]
XC2V40	40K	8 x 8	256	8	4	4	72	4	88
XC2V80	80K	16 x 8	512	16	8	8	144	4	120
XC2V250	250K	24 x 16	1,536	48	24	24	432	8	200
XC2V500	500K	32 x 24	3,072	96	32	32	576	8	264
XC2V1000	1M	40 x 32	5,120	160	40	40	720	8	432
XC2V1500	1.5M	48 x 40	7,680	240	48	48	864	8	528
XC2V2000	2M	56 x 48	10,752	336	56	56	1,008	8	624
XC2V3000	3M	64 x 56	14,336	448	96	96	1,728	12	720
XC2V4000	4M	80 x 72	23,040	720	120	120	2,160	12	912
XC2V6000	6M	96 x 88	33,792	1,056	144	144	2,592	12	1,104
XC2V8000	8M	112 x 104	46,592	1,456	168	168	3,024	12	1,108

Table 11.1: Virtex-II Field-Programmable Gate Array Family Members

Packaging

Offerings include ball grid array (BGA) packages with 0.80 mm, 1.00 mm, and 1.27 mm pitches. In addition to traditional wire-bond interconnects, flip-chip interconnect is used in some of the BGA offerings. The use of flip-chip interconnect offers more I/Os than is possible in wire-bond versions of the similar packages. Flip-chip construction offers the combination of high pin count with high thermal capacity.

Table 8.2 shows the maximum number of user I/Os available. The Virtex-II device/package combination table (Table 6 at the end of this section) details the maximum number of I/Os for each device and package using wire-bond or flip-chip technology.

Device	Wire-Bond	Flip-Chip
XC2V40	88	-
XC2V80	120	-
XC2V250	200	-
XC2V500	264	-
XC2V1000	328	432
XC2V1500	392	528
XC2V2000	-	624
XC2V3000	516	720
XC2V4000	-	912
XC2V6000	-	1,104
XC2V8000	-	1,108

Table 11.2: Maximum Number of User I/O Pads

Architecture

Virtex-II Array Overview: Virtex-II devices are user-programmable gate arrays with various configurable elements. The Virtex-II architecture is optimized for high-density and high-performance logic designs. As shown in Figure 1, the programmable device is comprised of input/output blocks (IOBs) and internal configurable logic blocks (CLBs). Programmable I/O blocks provide the interface between package pins and the internal configurable logic. Most popular and leading-edge I/O standards are supported by the programmable IOBs.

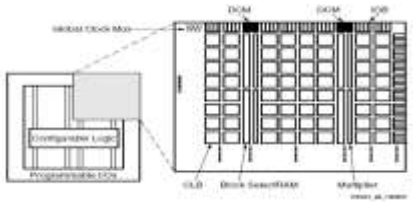

Figure 11.1: Virtex-II Architecture Overview

The internal configurable logic includes four major elements organized in a regular array.

- Configurable Logic Blocks (CLBs) provide functional elements for combinatorial and synchronous logic, including basic storage elements. BUFTs (3-state buffers) associated with each CLB element drive dedicated segmentable horizontal routing resources.
- Block SelectRAM memory modules provide large 18 Kbit storage elements of dual-port RAM.
- Multiplier blocks are 18-bit x 18-bit dedicated multipliers.
- DCM (Digital Clock Manager) blocks provide self-calibrating, fully digital solutions
- For clock distribution delay compensation, clock multiplication and division, coarse- and fine-grained clock phase shifting.

A new generation of programmable routing resources called Active Interconnect Technology interconnects all of these elements. The general routing matrix (GRM) is an array of routing switches. Each programmable element is tied to a switch matrix, allowing multiple connections to the general routing matrix. The overall programmable interconnection is hierarchical and designed to support high-speed designs. All programmable elements, including the routing resources, are controlled by values stored in static memory cells. These values are loaded in the memory cells during configuration and can be reloaded to change the functions of the programmable elements.

Virtex-II Features:

Input / Output Blocks (IOBs):

IOBs are programmable and can be categorized as follows:

- Input block with an optional single-data-rate or double-data-rate (DDR) register
- Output block with an optional single-data-rate or DDR register, and an optional 3-state
- Buffer, to be driven directly or through a single or DDR register
- Bidirectional block (any combination of input and output configurations)
- These registers are either edge-triggered D-type flip-flops or level-sensitive latches.
- IOBs support the following single-ended I/O standards:

- LVTTL, LVCMOS (3.3V, 2.5V, 1.8V, and 1.5V)
- PCI-X compatible (133 MHz and 66 MHz) at 3.3V
- PCI compliant (66 MHz and 33 MHz) at 3.3V
- CardBus compliant (33 MHz) at 3.3V
- GTL and GTLP
- HSTL (Class I, II, III, and IV)
- SSTL (3.3V and 2.5V, Class I and II)
- AGP-2X

The digitally controlled impedance (DCI) I/O feature automatically provides on-chip termination for each I/O element. The IOB elements also support the following differential signaling

I/O standards:

- LVDS
- BLVDS (Bus LVDS)
- ULVDS
- LDT
- LVPECL

Two adjacent pads are used for each differential pair. Two or four IOB blocks connect to one switch matrix to access the routing resources.

Configurable Logic Blocks (CLBs):

CLB resources include four slices and two 3-state buffers.

Each slice is equivalent and contains:

- Two function generators (F & G)
- Two storage elements
- Arithmetic logic gates
- Large multiplexers
- Wide function capability
- Fast carry look-ahead chain
- Horizontal cascade chain (OR gate)

The function generators F & G are configurable as 4-input look-up tables (LUTs), as 16-bit shift registers, or as 16-bit distributed SelectRAM memory. In addition, the two storage elements are either edge-triggered D-type flip-flops or level-sensitive latches. Each CLB has internal fast interconnect and connects to a switch matrix to access general routing resources.

Block Select Ram Memory:

The block Select RAM memory resources are 18 Kb of dual-port RAM, programmable from 16K x 1 bit to 512 x 36 bits, in various depth and width configurations. Each port is totally synchronous and independent, offering three "read-during-write" modes. Block Select RAM memory is cascadable to implement large embedded storage blocks. Supported memory configurations for dual-port and single-port modes are shown in Table 11. 3.

16K x 1 bit	2K x 9 bits
8K x 2 bits	1K x 18 bits
4K x 4 bits	512 x 36 bits

Table 11.3: Dual-Port and Single-Port Configurations

A multiplier block is associated with each Select RAM memory block. The multiplier block is a dedicated 18 x 18-bit multiplier and is optimized for operations based on the

block SelectRAM content on one port. Both the SelectRAM memory and the multiplier resource are connected to four switch matrices to access the general routing resources.

GLOBAL CLOCKING: The DCM and global clock multiplexer buffers provide a complete solution for designing high-speed clocking schemes. Up to 12 DCM blocks are available. To generate de-skewed internal or external clocks, each DCM can be used to eliminate clock distribution delay. The DCM also provides 90-,180-, and 270-degree phase-shifted versions of its output clocks. Fine-grained phase shifting offers high-resolution phase adjustments in increments of 1/256 of the clock period. Very flexible frequency synthesis provides a clock output frequency equal to any M/D ratio of the input clock frequency, where M and D are two integers. Virtex-II devices have 16 global clock MUX buffers, with up to eight clock nets per quadrant. Each global clock MUX buffer can select one of the two clock inputs and switch glitch-free from one clock to the other. Each DCM block is able to drive up to four of the 16 global clock MUX buffers.

Routing Resources:

The IOB, CLB, block SelectRAM, multiplier, and DCM elements all use the same interconnect scheme and the same access to the global routing matrix. Timing models are shared, greatly improving the predictability of the performance of high-speed designs. There are a total of 16 global clock lines, with eight available per quadrant. In addition, 24 vertical and horizontal long lines per row or column as well as massive secondary and local routing resources provide fast interconnect. Virtex-II buffered interconnects are relatively unaffected by net fanout and the interconnect layout is designed to minimize crosstalk. Horizontal and vertical routing resources for each row or column include:

- 24 long lines
- 120 hex lines
- 40 double lines
- 16 direct connect lines (total in all four directions)

Boundary Scan:

Boundary scan instructions and associated data registers support a standard methodology for accessing and configuring Virtex-II devices that complies with IEEE standards 1149.1 — 1993 and 1532. A system mode and a test mode are implemented. In system mode, a Virtex-II device performs its intended mission even while executing non-test boundary-scan instructions. In test mode, boundary-scan test instructions control the I/O pins for testing purposes. The Virtex-II Test Access Port (TAP) supports BYPASS, PRELOAD, SAMPLE, IDCODE, and USERCODE non-test instructions. The EXTEST, INTEST, and HIGHZ test instructions are also supported.

Configuration:

Virtex-II devices are configured by loading data into internal configuration memory, using the following five modes:

- Slave-serial mode
- Master-serial mode
- Slave SelectMAP mode
- Master SelectMAP mode
- Boundary-Scan mode (IEEE 1532)

A Data Encryption Standard (DES) decryptor is available on-chip to secure the bitstreams. One or two triple-DES key sets can be used to optionally encrypt the configuration information..

Virtex-II Device/Package Combinations and Maximum I/O:

Wire-bond and flip-chip packages are available. Table 11.4 and Table8.5 show the maximum possible number of user I/Os in wire-bond and flip-chip packages, respectively. Table 11.6 shows the number of available user I/Os for all device/package combinations.

- CS denotes wire-bond chip-scale ball grid array (BGA) (0.80 mm pitch).
- FG denotes wire-bond fine-pitch BGA (1.00 mm pitch).
- FF denotes flip-chip fine-pitch BGA (1.00 mm pitch).
- BG denotes standard BGA (1.27 mm pitch).
- BF denotes flip-chip BGA (1.27 mm pitch).

The number of I/Os per package include all user I/Os except the 15 control pins (CCLK, DONE, M0, M1, M2, PROG_B, WRDWN_B, TCK, TDI, TDO, TMS, HSWAP_EN, DXN, DXP, and RSVD) and VBATT.

12. FPGA Comparisons

Historically, FPGAs have been slower, less energy efficient and generally achieved less functionality than their fixed ASIC counterparts. A combination of volume, fabrication improvements, research and development, and the I/O capabilities of new supercomputers have largely closed the performance gap between ASICs and FPGAs. Advantages include a shorter time to market, ability to re-program in the field to fix bugs, and lower non-recurring engineering costs. Vendors can also take a middle road by developing their hardware on ordinary FPGAs, but manufacture their final version so it can no longer be modified after the design has been committed. Xilinx claims that several market and technology dynamics are changing the ASIC/FPGA paradigm:

- IC costs are rising aggressively
- ASIC complexity has bolstered development time and costs
- R&D resources and headcount is decreasing
- Revenue losses for slow time-to-market are increasing
- Financial constraints in a poor economy are driving low-cost technologies

These trends make FPGAs a better alternative than ASICs for a growing number of higher-volume applications than they have been historically used for, which the company blames for the growing number of FPGA design starts. The primary differences between CPLDs and FPGAs are architectural. A

CPLD has a somewhat restrictive structure consisting of one or more programmable sum-of-products logic arrays feeding a relatively small number of clocked registers. The result of this is less flexibility, with the advantage of more predictable timing delays and a higher logic-to-interconnect ratio. The FPGA architectures, on the other hand, are dominated by interconnect.

13. Conclusions

The Electronics field is becoming more and more competitive and also becoming more and complex. So it is necessary now to increase the efficiency of the logic design, to reduce design cost, to reduce the NRE cost and most important to reduce the time-to-market the designed product, hence FPGA is a great asset to the design. But inspire of all these efficient design it is the responsibility of the designer to obtain best from the FPGA methodology.

References

[1]. http://www.wikipedia.com
[2]. Xilinx, Virtex-II Data Sheets
[3]. WWW.VLSIBANK.COM
[4]. Digital System Design Using VHDL: Hanumantha Raju M.C.