IJRD International Journal of Research & Development Organisation

ISSN: 2455-6653

Journal of Applied Science

A proposed FPGA Architecture Design

Yelkal Mulualem #1(*)

#1(* corresponding author)

Department of Information Technology, CNCS, University of Gondar Ethiopia.

{ ye_mu79@yahoo.com }

M.Mohamed Sirajudeen#2

Department of Information Technology, CNCS, University of Gondar, Ethiopia.{mdsirajudeen1@gmail.com}

ABSTRACT

This research paper discuss with a general over view of FPGAs particularly, Architectural Description of Virtex-4 FPGA and Verilog Hardware Description Language of the proposed work. It has also depth description of design of the new developed FPGA Architectural. FPGA architecture design and development approach used in this paper is followed top-down approach. Later the design and implementation of the four modules, namely, Top-Module, Read-Code Word Module, ReadAveraged Sample Module and Decoder-Module will be discussed with the necessary information's.

Keywords: Architectural, FPGA, Virtex and Verilog.

I. INTRODUCTION

Field-programmable gate arrays are chips that can be programmed to perform virtually any logic operation. They can be used in place of multiple smaller components such as glue logic, or they can contain large designs such as processors or graphics controllers. Many FPGAs can be reconfigured any number of times making them ideal for design prototyping, and they have recently been replacing ASICs and MPGAs in low-volume productions due to the high initial cost and long turnaround time of these custom manufactured chips. Their high logic capacity and abundance of flip flops distinguish FPGAs from other kinds of programmable logic devices. Much different architecture exists, but Figure 1.1 shows the basic structure of a typical FPGA: a matrix of configurable logic blocks (CLBs) and interconnection resources surrounded by I/O blocks.

The CLBs are often complex but are likely to contain one or more function generators followed by flip flops. Made using either look-up tables (LUTs) or multiplexers, function generators are capable of producing any k-input Boolean function where k is usually four. LUTs are 1-bit wide memories and essentially store the truth table of the Boolean function they generate.

They often can be used for general storage when not acting as a function genera- tor. The output of a function generator can serve as part of combinational logic or can be directed to a flip flop to create a latched signal. Interconnection resources are composed of horizontal and vertical wires that can form connections with each other through the programmable switches. There are also programmable switches that connect wires to CLBs. I/O blocks can be programmed to allow their associated pin to operate as either an input or an output. Current FPGAs often include additional components such as clock managers, RAM, and dedicated circuitry for common arithmetic operations.

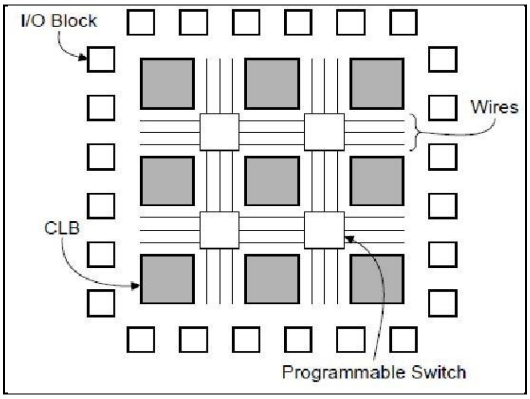

Figure 1.1 the basic structure of a typical FPGA

Due to the size and complexity of FPGAs, CAD tools must be used to take a design from its initial description to a bit stream that can be programmed on a device. Three processing steps are required: mapping, placement, and routing. During mapping, a designs description in terms of logic gates is translated to a form suitable for function generators. During placement, CLB sites on the chip are chosen, and during routing, the necessary connections between CLBs are formed. A programming le can be generated and downloaded into the device once these steps are completed, which can take several hours for large, highly-constrained designs. The physical mechanism for programmability varies for different FPGAs with the main methods being static RAMs and antifuses. System performance is a major concern while designing complicated systems. The maximum performance can be achieved when circuits are optimized for single problem.

As a new problem arises with minor changes to the old one, the whole system has to be redesigned and re optimized. Reconfigurable computing addresses this problem by allowing dedicated circuits to be built on to FPGAs and modifying these circuits by reprogramming the chips again. This greatly improves system flexibility and functional density. The circuits are loaded into the hardware and unloaded from it dynamically during the operation of the system. The high throughput computational requirements of real-time digital signal pro- cessing (DSP) systems typically dictate hardware intensive solutions. By increasing the system density configurable computing can deliver functionality of a device many times more than its size by dynamically reconfiguring the system.

II. RELATED WORK

In digital signal processing (DSP) applications, the emphasis is most often on performance, in terms of real-time requirements as well as throughput. For radio signal processing in particular, the high throughput requirements dictate that a hardware solution is almost always required. Currently available DSP microprocessors are suited for baseband waveform processing. Conventional DSP microprocessors are implemented using inherently serial architectures. A DSP chip operating at 40 million instruction per second (MIPS) has a useful bandwidth of less than 500 kHz. DSP microprocessors are thus inadequate for the implementation of most stages of a radio system. Field Programmable Gate Arrays (FPGAs) provide a rapid prototyping platform, which can be reprogrammed for different hardware functions without incurring the non-recurring engineering costs typically associated with custom IC fabrication.

Implementing DSP functions in FPGA provides the several advantages over conventional DSP hardware: Reconfigurability and Parallelism. Typical FPGA board device is shown in Figure 1.2.

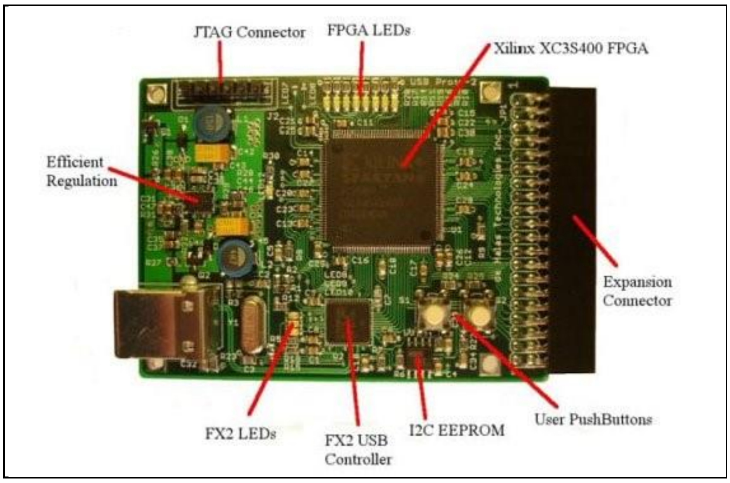

Figure 1.2 Typical FPGA board device

The FPGA device is configured by using the in-system programming (ISP) method, which means that the FPGA can be programmed while the chip is still attached to its circuit board. The storage cells in the LUTs in an FPGA are volatile, which means that they lose their stored contents whenever the power supply for the Hence the FPGA has to be programmed every time power is applied. For this reason, a small memory chip that holds its data permanently, called a programmable read-only memory (PROM) is included on the circuit board that houses the FPGA. The storage cells in the FPGA are loaded automatically from the PROM when power is applied to the chips.

III. PROPOSED ARCHITECTURE

The Verilog language is a hardware description language (HDL) that is designed to model digital logic circuits. It provides a means of specifying a digital system at a wide range of levels of abstraction. The language supports the early conceptual stages of design with its behavioural level of abstraction, and the later implementation stages with its structural abstractions. The language includes hierarchical constructs, allowing the designer to control a description complexity. The Verilog language provides the digital system designer with a means of de- scribing a digital system at a wide range of levels of abstraction, and, at the same time, provides access to computer-aided design tools to aid in the design process at these levels. During the design process, behavioural and structural constructs may be mixed as the logical structure of portions of the design is designed.

The description may be simulated to determine correctness, and some synthesis tools exist for automatic design. Indeed, the Verilog language provides the designer entry into the world of large, complex digital systems design. The Verilog language describes a digital system as a set of modules. Each of these modules has an interface to other modules as well as a description of its contents. A module represents a logical unit that can be described either by specifying its internal logical structure for instance describing the actual logic gates it is comprised of, or by describing its behaviour in a program-like manner in this case focusing on what the module does rather than on its logical implementation. These modules are then interconnected with nets, allowing them to communicate. For this thesis, the whole FPGA architecture has been developed using the Verilog hardware description language.
The trace acquisition board is based on a Xilinx Virtex-4 XC4VFX20 FPGA. Virtex-4 devices are user-programmable gate arrays with various configurable elements and embedded cores optimized for high-

density and high-performance system designs. Besides the standard logic and I/O blocks, it embeds some specific resources to extend the conventional data processing and storing capabilities, as well as to improve the clock management. The main features of these resources, which have been used within the Distributed Temperature Sensing (DTS) system application [16].

The tasks performed by the FPGA and the hardware implemented to carry out these tasks. Only the parts of the FPGA design common to each architecture are discussed in this section. At the top of hierarchy level, the FPGA architecture is represented by the block diagram of Figure 1.3.

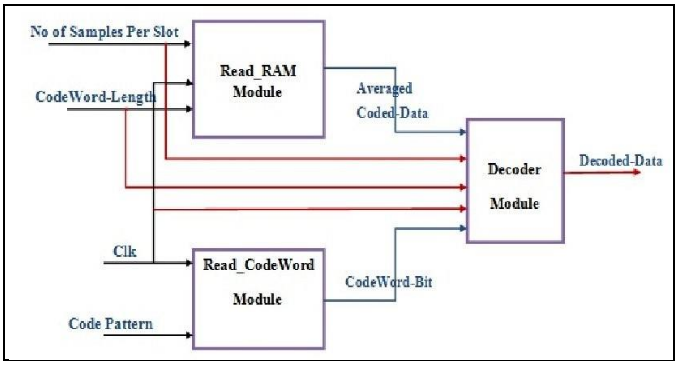

Figure 1.3 the top of hierarchy level, the FPGA architecture

The core of the system is the Decoder-Module, whose task is acquiring Averaged coded Stokes and Anti-Stokes sample trace data from the Read-Sample Module and Codeword bits from the ReadCodeword Module, and then build the Decoded traces data and finally make them available to the RAM. Read-RAM Module is important to acquire (read) the Averaged coded Stokes and Anti-Stokes sample trace data from the Dual port RAM according to the particular acquisition mode selected. In the other way, Read-Codeword Module has been used to read codeword bit patterns from the Register (REG) and then it will return one bit of codeword at a single clock cycle, in our experiment we would like to implement with clock frequency of 150 MHz(6 ns).

The total operation of the system is performed in a single clock cycle. It is referred to as Top Module because it is the outer interface interacts with FPGA board. It takes averaged coded Stoke and Antistoke trace data and codeword bit pattern, and then it returns the decoded Stoke and Anti-Stoke sampled trace data.

IV. CONCLUSION

In this thesis work, Verilog hardware description languages (HDLs) has been used to design digital systems. These hardware description language allows designing digital systems by writing a program that describes the behaviour of the digital circuit. The program can then be used to both simulate the operation of the circuit and synthesize an actual implementation of the circuit in an FPGA. Once hardware has been described the functional simulators used to produce waveforms that will verify the design. This hardware description can then be synthesized to logic equations and implemented or mapped to the FPGA architecture.

REFERENCES

[1]. J. M. Lopez-Higuera Ed., Handbook of Optical Fiber Sensing Technology, Chichester,U.K., John Wiley and Sons Ltd., 2001.

[2]. A. Rogers, Distributed optical fiber sensing, in Handbook of Optical Fiber Sensing Technology", J. M. Lopez-Higuera Ed. Chichester, U.K., John Wiley and Sons Ltd., 2001, ch. 14, p.271-312.

[3]. J. P. Dakin, Distributed optical fiber systems, in Optical Fiber Sensors: Systems and Applications", B. Culshaw, J. Dakin, Eds. Norwood, MA, Artech House, 1988, vol.2, ch. 15, p. 575598.

[4]. S. Adachi, "Distributed optical fiber sensors and their applications," in Proc.SICE Annual Conference, 2008, pp. 329333.

[5]. A. H. Hartog, A distributed temperature sensor based on liquid-core optical bers, J. Lightwave Technol. 1(3), pp. 498 - 509 , 1983.

[6]. K. Aoyama, K. Nakagawa, and T. Itoh, "Optical time domain Reectometry in a single-mode fiber," IEEE Journal of Quantum Electronics, vol. 17, no. 6, pp. 862-868, 1981.

[7]. M. A. Farahani and T. Gogolla, "Spontaneous Raman scattering in optical fibers with modulated probe light for distributed temperature Raman remote sensing," IEEE/OSA Journal of Light wave Technology, vol. 17, no. 8, pp. 1379-1391, 1999.

[8]. R. Singh, "Raman and the Discovery of the Raman Effect," Physics in Perspective (PIP), vol. 4, pp. 399-420, 2002, 10.1007/s000160200002.

[9]. D. Long, "The Raman effect: a unified treatment of the theory of Raman scattering by molecules." John Wiley and Sons Inc, 2002.

[10]. "Wavelength Coded Optical Time-Domain Reectometry" Ning Hua Zhu, Member, IEEE, Jian Hong Ke, Hong Guang Zhang, Wei Chen, Jian Guo Liu, Ling Juan Zhao, and Wei Wang. [11] . "Virtex-4 FPGA user guides, Xilinx, Inc.", www.xilinx.com.