ANALYSIS OF FPGA DESIGN METHODS USING AN 8 BIT ALU

ROSNAH BINTI MOHD ZIN

"This project report is presented in partial fulfilment of the requirement for the Award of the Degree of Master of Electrical Engineering"

Faculty of Electrical and Electronic Engineering
Kolej Universiti Teknologi Tun Hussein Onn

OCTOBER, 2005

For my beloved husband (Mohd Nizam Katimon), daughter (Husna),
mother and late father

ACKNOWLEDGEMENT

I would like to take this opportunity to acknowledge Assoc. Prof. Awtar Singh a/l Karnail Singh and Prof. Dr. J. Baesig as my supervisor and Mr. Hinz-Hemmers my lab supervisor for helping me in carrying out this project.

To all of those that I could not mention by name, accept my apologies and rest assured I appreciate everything you all have done for me.

ABSTRACT

Field Programmable Logic Arrays (FPGAs) have been growing at a rapid rate in the past few years. FPGA is a type of logic chip that can be programmed which supports thousand of gates and provide flexibility and low cost which is suitable for implementing a prototype system. The existence of CAD software to support FPGAs has grown in sophistication and it makes most user designs are now complete system and go to production as an FPGA. This thesis will discuss on FPGA design style using an 8-bit ALU as design hardware. Generally, the methods that used to implement the 8-bit ALU are using schematic based entry and VHDL based entry. Implementation of 8-bit ALU using VHDL includes a behavioral and structural description. The top level of design is using a schematic based entry. Finally this thesis will discuss the methods that are used in designing the 8-bit ALU in term of the flexibility, area consumption and timing analysis. The analysis will give the user more understanding in designing digital system using FPGA design style and give them a choice which depends on the design requirements.

ABSTRAK

FPGA telah berkembang dengan pesatnya sejak beberapa tahun kebelakangan ini. *FPGA* merupakan cip logik yang boleh deprogram, menampung ribuan get dan menyediakan kebolehpelbagaian dan mempunyai kos yang rendah dimana membolehkan ia sesuai untuk mengimplementasi sistem prototaip. Kewujudan perisian CAD yang menyokong perkembangan FPGA telah berkembang secara sofistikated dan membolehkan kebanyakan rekabentuk pengguna menjadi sistem lengkap dan menjadikan *FPGA* sebagai produk keluaran. Tesis ini membincangkan rekabentuk 8 bit ALU menggunakan gaya FPGA. Secara umumnya, kaedah yang digunakan untuk mengimplementasi 8 bit ALU adalah menggunakan rajah skematik dan VHDL. Penggunaan VHDL terdiri dari *behavioral* dan *structural description*. Paras atas (*top level*) rekabentuk menggunakan rajah skematik. Akhir sekali tesis ini membincangkan kaedah yang digunakan didalam merekabentuk 8 bit ALU darisegi kebolehpelbagaian, penggunaan ruang dan analisis pemasaan . Analisa ini akan membolehkan pengguna lebih memahami bagaimana merekabentuk sistem digital menggunakan kaedah *FPGA* dan dapat memberikan pilihan kaedah yang akan diguna pakai berdasarkan keperluan rekebentuk.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

AHDL	-	Altera Hardware Description Language
ALU	-	Arithmetic Logic Unit
CAD	-	Computer Aided Design
CPLD	-	Complex Programmable Logic Device
CPU	-	Central Processing Unit
DSP	-	Digital Signal Processing
EAB	-	Embedded Array Block
EEPROM	-	Electrically Erasable Programmable Read Only Memory
EPROM	-	Erasable Programmable Read Only Memory
FIFO	-	First In First Out
FPGA	-	Field Programmable Gate Array
HDL	-	Hardware Description Language
I/O	-	Input/Output
IOE	-	Input Output Element
IP	-	Intellectual Property
LE	-	logic Element
LED	-	Light Emitting Diode
LUT	-	Look Up Table
MUX	-	Multiplexer
PC	-	Personal Computer
PCB	-	Printed Circuit Board
PLA	-	Programmable Logic Array
PLD	-	Programmable Logic Device
RAM	-	Random Access Memory

ROM	-	Read Only Memory
SRAM	-	Static Random Access Memory
UP1	-	University Programmer 1
VHDL	-	VHSIC Hardware Description Language
VHSIC	-	Very High Speed Integrated Chip

LIST OF SYMBOLS

%	-	percent
Hz	-	Hertz
MHz	-	Mega Hertz
ns	-	nano second

LIST OF APPENDIX

CHAPTER I

INTRODUCTION

1.1 Overview

Field Programmable Gate Arrays (FPGAs) is one type of logic chip that can be programmed and its internal functional operation is defined by the user. An FPGA supports a more flexible block structure and through more flexible interconnects as the routing resources. The number of gates and features has increased dramatically to compete with capabilities that have traditionally only been offered through ASIC devices. The applications of FPGA have led to higher density devices, intellectual property (IP) integration, and high-speed I/O interconnects technology. All of these elements have allowed FPGAs to play a central role in digital systems implementations.

Current FPGA architectures are heterogeneous, containing thousands of logic elements and hundreds of embedded multipliers and memory units.(K. J. Alex, H. Raymond, S. K. Ivan, F. Joshua , D.K, J. Foster, S. B., and A. Muaydh, 2004). Internally, FPGA typically contain multiple copies of a basic programmable logic element (LE) or cell. Logic elements are arranged in a column or matrix on the chip. To perform more complex operations, logic elements can automatically connected to other logic elements on the chip using a programmable interconnection network. The basic methodology for FPGA design consists of three inter- related steps: entry, implementation, and verification (W.S. Cater,1994) as shown on Figure 1.1.

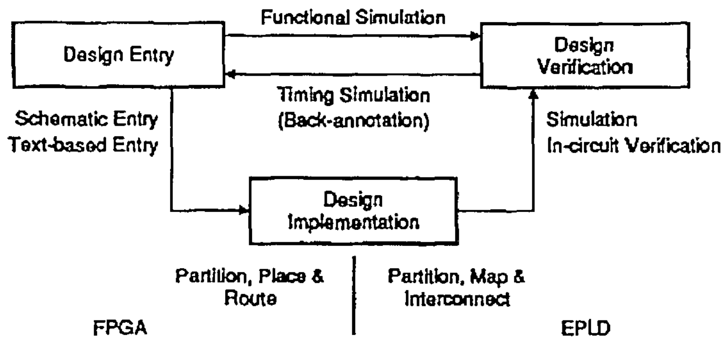

Figure 1.1: The basic FPGA design methodology consists of three steps: entry, implementation, and verification.

1.2 Project Aim

The project aim was to analyze the methods that are being used to design FPGA based hardware which provide a flexibility for the implementation.

1.3 Objective

The objectives of the project are:

(a) To design an 8-bit ALU using the ALTERA MAX+PLUS II Software and implement the hardware onto the UP2 board.

(b) Use a schematic and VHSIC Hardware Description Language (VHDL) for design entry.

(c) To analyze the methods that are used to design the 8-bit ALU in terms of the flexibility, area consumed and timing analysis.

1.4 Scope Of The Project

The scope of the project covers the study of the architecture, the FPGA design methodology and the targeted chip, FLEX10K70. The FLEX10K70 chip was used, since it is the by far the larger chip and is packaged with many more in/out pins available. The Altera's Max+PLUS II Version 10.1 software is used to draw the schematic diagram and write VHDL code for the design of 8-bit ALU. The design is then downloaded to the chip and the outputs are displayed on the 7 segment display. The inputs are controlled by the switches. The methods that are used to design the 8-bit ALU are analyzed in terms of the flexibility, area consumed and timing analysis.

CHAPTER II

LITERATURE REVIEW

2.1 Field Programmable Gate Array

In recent years, field programmable gate arrays (FPGAs) have become the dominant form of programmable logic. In comparison to previous programmable logics such as PLAs, FPGAs can implement far larger logic functions. Rather than being used merely as 'glue logic', FPGAs have sufficient logic resources to implement complete systems and subsystems. Another difference of FPGA from previous programmable logics is the use of SRAM, instead of fuses, for the configuration memory. SRAM based FPGAs can act as 'soft-hardware', a configuration file can be loaded into a SRAM based FPGA's configuration memory and then run in a similar way to software. This reconfigurability is being used in a new class of FPGA based system, which are often called transformable systems.(R. Payne,1996)

A field-programmable gate array or FPGA is a semiconductor device containing programmable logic components and programmable interconnects. The programmable logic components can be programmed to duplicate the functionality of basic logic gates (such as AND, OR, XOR, INVERT) or more complex combinatorial functions such as decoders or simple math functions. In most FPGAs, these programmable logic components (or logic blocks, in FPGA parlance) also include memory elements, which may be simple flip-flops or more complete blocks of memories.

A hierarchy of programmable interconnects allows the logic blocks of an FPGA to be interconnected as needed by the system designer, somewhat like a one-chip programmable breadboard. These logic blocks and interconnects can be programmed after the manufacturing process by the customer/designer (hence the term "field-programmable") so that the FPGA can perform whatever logical function is needed.

Applications of FPGAs include DSP, software-defined radio, aerospace and defense systems, ASIC prototyping, medical imaging, computer vision, speech recognition, cryptography, bioinformatics, and a growing range of other areas. FPGAs originally began as competitors to CPLDs and competed in a similar space, that of glue logic for PCBs. As their size, capabilities and speed increased they began to take over larger and larger functions to the state where they are now marketed as competitors for full systems on chips. They now find applications in any area or algorithm that can make use of the massive parallelism offered by their architecture.

2.1.1 General Structure of FPGA

Field Programmable Gate Array (FPGA) is one of the programmable logic devices (PLD) that can be used to implement just about any hardware design. FPGA provides logic blocks for implementation of the required functions. The general structure of FPGA is illustrated in Figure 2.1. It contains three main types of resources : logic blocks, I/O block for connecting the pins of the package and interconnection wires and switches.

The typical basic architecture consists of an array of logic blocks and routing channels. Multiple I/O pads may fit into the height of one row or the width of one column. Generally, all the routing channels have the same width (number of wires). An application circuit must be mapped into an FPGA with adequate resources.

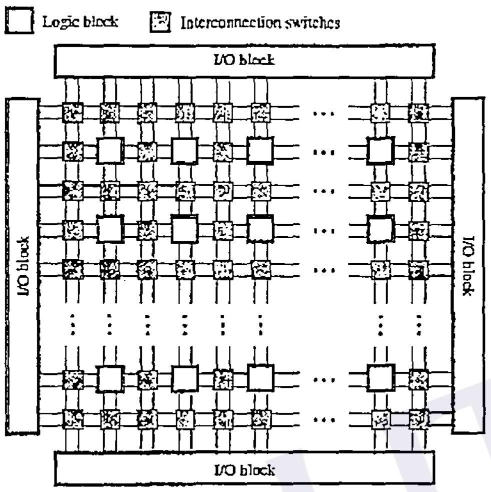

Figure 2.1: General Structure of FPGA

The typical FPGA logic block consists of a 4-input lookup table (LUT), and a flip-flop, as shown in Figure 2.2.

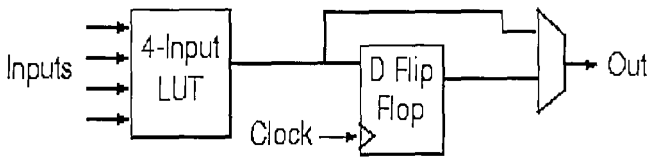

Figure 2.2: A typical Logic block

There is only one output, which can be either the registered or the unregistered LUT output. The logic block has four inputs for the LUT and a clock input. Since clock signals (and often other high-fanout signals) are normally routed via special-purpose dedicated routing networks in commercial FPGAs, they are accounted for separately from other signals.

Modern FPGA families expand upon the above capabilities to include higher level functionality fixed into the silicon. Having these common function embedded into the silicon reduces the area required and gives those functions increased speed compared to building them from primitives. Example of these include multipliers, generic DSP blocks, embedded processors, high speed IO logic and embedded memories.

2.2 FPGA design and programming

To define the behavior of the FPGA the user provides a hardware description language (HDL) or a schematic design. Common HDLs are VHDL and Verilog. Then, using an electronic design automation tool, a technology-mapped netlist is generated. The netlist can then be fitted to the actual FPGA architecture using a process called place-and-route, usually performed by the FPGA company's proprietary place-and-route software. The user will validate the map, place and route results via timing analysis, simulation, and other verification methodologies. Once the design and validation process is complete, the binary file generated (also using the FPGA company's proprietary software) is used to (re)configure the FPGA device.

In an attempt to reduce the complexity of designing in HDLs, which have been compared to the equivalent of assembly languages, there are moves to raise the abstraction level of the design.

In a typical design flow, an FPGA application developer will simulate the design at multiple stages throughout the design process. Initially the RTL description in VHDL or Verilog is simulated by creating test benches to stimulate the system and observe results. Then after the synthesis engine has mapped the design to a netlist the netlist is translated to a gate level description where simulation is repeated to confirm the synthesis proceeded without errors. Finally the design is laid out in the FPGA at which point propagation delays can be added and the simulation run again with these values back-annotated onto the netlist.

2.2.1 Programming Technologies

A high performances FPGA requires a programmable interconnect switch that occupies a small area and at the same time has a low parasitic capacitance. Several different programming technologies are used to implement the programmable switches in FPGAs. The commonly used programming technologies are as follows:

(i) Antifuse switch of dielectric or amorphous silicon composition, which on electrical programming forms a low-resistance interconnect path.

(ii) SRAM-based technology in which the switch is a pass transistor controlled by the state of a RAM bit in a look-up-table (LUT).

(iii) EPROM-based technology (either EPROM or EEPROM) in which the switch is a floating gate transistor that can be turned of by injecting charge on the floating gate.

However majority of the FPGA being used, use antifuse and SRAM as programming technologies.

REFERENCES

Payne R. (1996). "Asynchronous FPGA Architecture." *Comput. Digit. Tech.* 5. 282 – 286

Buijs F. (1992). " ALU Synthesis from VHDL Descriptions to Optimized Multi-Level Logic" *IEEE.* **8**. 175-180

Jones A. K., Hoare R., Ivan S.K., Joshua F., D.R.,Foster Jn, S.B., Muaydh A. (2004)." A 64-way VLIW/SIMD FPGA Architecture and Design Flow" *Electrical and Computer Engineering,***5.** 499 -502.

Stephen B. and Zvonko V. (1995). " Fundamentals of Digital logic with VHDL Design" 3rd .ed. Mc Graw Hill.197 -204

Hwang E. (2003)."Where's the Hardware?" Issue 150 .Circuit Cellar. 32-37

Dueck, Robert K.(2004) " Digital Design with CPLD Applications and VHDL".2nd .ed. Canada Thomson Delmar Learning .375 – 377

Sharma, Ashok K. (1998). " Programmable logic Handbook". 1st. ed. United State of America. Mc Graw Hill. 99-171.

Barr. M (1999). "How Programmable logic Work".Miller Freeman. 77 – 84.

Perry, Douglas L. (1998). " VHDL" 3rd .ed. Singapore. Mc Graw Hill. 6 – 72.

Salcic Z., Smailagic A. (2000). " Digital Systems Design and Prototyping" 2nd .ed. USA. Kluwer Academic Publisher. 115 – 142.

Makhijani H, Meier S. "A High Level Design Solution for FPGA's" Synopsys, Inc. 596-603

Ma, Xiaqiwi, Tong, Jiaroiig (2003). "Design and Implementation of A New FPGA Architecture". *IEEE*. 03. 816-819