

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

أَقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ﴿١﴾ خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ ﴿٢﴾

أَقْرَأْ وَرَبُّكَ الْأَكْرَمُ ﴿٣﴾ الَّذِي عَلَّمَ بِالْقَلَمِ ﴿٤﴾ عَلَّمَ الْإِنْسَانَ

مَا لَمْ يَعْلَمْ ﴿٥﴾

صدق الله العظيم

سورة العلق

## **DEDICATION**

With my worm feeling, I am dedicating this thesis to the wholly great father, great mother, to my family members god bless them.

# **ACKNOWLEDGEMENTS**

In the name of Allah, Most Gracious, And Most Merciful Praise be Almighty Allah (Subhanahu Wa Ta'la) who gave me the courage and patience to carry out this work.

I would like to express my deep gratitude to my supervisor, Dr. MARTINO OJWOK AJANG AJANGNAY for his kindness, constant endeavor, guidance and the numerous moments of attention he devoted through this work.

I would like to thank all the staff of electrical engineering department, Sudan University of Science and Technology for their help and advice.

Finally, my deepest gratitude owed to my friends, my husband and my family.

# ABSTRACT

Microprocessors are the devices in a computer that make things happen. They are capable of performing basic arithmetic operations, moving data from place to place, and making basic decisions based on the quantity of certain values. In this project a systematic step of designing of microprocessor from the ground up methodology has been described. To define the behavior of the Complex Programmable Logic Devices (CPLDs), a description of the hardware's structure and behavior was written in a High-level Hardware Description Language (VHDL) and that code was then compiled and downloaded prior to execution. The design entry step is followed or interspersed with periods of functional simulation using QuartusII software. That's where a simulator is used to execute the design and confirm that the correct outputs are produced for a given set of test inputs.

Compilation only begins after a functionally correct representation of the hardware exists. This hardware compilation consists of two distinct steps. First, an intermediate representation of the hardware design was produced. This step is called synthesis and the result of a representation called a netlist. The netlist is device independent, so its contents do not depend on the particulars of the FPGA or CPLD; it is usually stored in a standard format called the Electronic Design Interchange Format (EDIF). The second step in the translation process is called place & route. This step involves mapping the logical structures described in the netlist onto actual macrocells, interconnections, and input and output pins; this is the bitstream format.

Once the bitstream has been created a CPLD kit (ALTERA kit) was used to check the designed microprocessor performance. The generated bitstream was downloaded into max EPM7128SLC84-7.

## المستخلص

المعالجات الدقيقة هي الأجهزة داخل الكمبيوتر التي ينتج عنها حدوث الأشياء. وهي قادرة على تأدية العمليات الحسابية الأساسية ، نقل البيانات من مكان لآخر واتخاذ قرارات أساسية استنادا على مجموعة من القيم المعينة. في هذا البحث وصف لخطوة منظمة لتصميم المعالج الدقيق من أساسياته. لتعريف السلوك للأجهزة المنطقية المعقدة القابلة للبرمجة (CPLDs) فان وصف بنية المعدات الصلبة وسلوكها يكتب بلغة وصف المعدات الصلبة عالية المستوى (VHDL) وتلك الشفرة تترجم و تحمل قبل التنفيذ. الخطوة التالية عبارة عن محاكاة التصميم باستخدام برنامج Quartus II. وهنا يستخدم المحاكى لتنفيذ التصميم والتأكد على ان المخرجات الصحيحة انتجت من مجموعة المدخلات المختيرة.

الترجمة تبدأ بعد التأكد من التمثيل الصحيح للمعدات الصلبة. تتكون ترجمة المعدات الصلبة من خطوتين: أولا اعطاء تمثيل متوسط لتصميم المعدات الصلبة وهذه الخطوة تسمى التركيب (synthesis) والنتيجة تسمى netlist . netlist لا تعتمد على الجهاز ولذا فان مكوناته لا تعتمد على (FPGA) أو (CPLD) معين وعادة تخزن في صيغة قياسية تسمى صيغة تبادل التصميم الإلكتروني (EDIF).

الخطوة الثانية في عملية الترجمة تسمى place & route هذه الخطوة تشمل تخطيط للتركيب المنطقي الموصوف في netlist لل macrocells الحقيقية، التوصيلات ومسامير الدخل والخرج وهذه هي صيغة bitstream . عندما تكون bitstream تحمل في شريحة CPLD (ALTERA kit MAXEPM7128SLC84-7) للتأكد من أداء المعالج الدقيق المصمم.

# TABLE OF CONTENTS

		page
	الاية	i
	DEDICATION	ii
	ACKNOWLEDGEMENT	iii
	ABSTRACT	iv
	المستخلص	v
	TABLE OF CONTENTS	vi
	LIST OF TABLES	ix
	LIST OF FIGURES	x
	LIST OF ABBREVIATIONS	xii
<b>CHAPTER ONE</b>		
<b>INTRODUCTION</b>		
1.1	General Concepts	1
1.2	Objectives	4
1.3	Methodology	4
1.4	Thesis Structure	5
<b>CHAPTER TWO</b>		
<b>MICROPROCESSOR HARDWARE AND SOFTWARE</b>		
2.1	Hardware	6
2.1.1	History	7
2.1.2	Development	7
2.1.3	Applications	8
2.1.4	Architecture	8

2.2	Software	11
2.2.1	VHDL code structure	12
2.2.2	Design methodology	15
2.2.3	VHDL advantages	15
<b>CHAPTER THREE</b>		
<b>COMPONENTS OF MICROPROCESSORS</b>		
3.1	Introduction	18
3.2	The Datapath	18
3.2.1	Functional units	18
3.2.2	Registers and memories	30
3.2.3	Multiplexer	34
3.2.4	Dedicated datapath	36
3.2.5	General datapaths	42
3.3	The Control Unit	44
3.3.1	Constructing the control units	49
<b>CHAPTER FOUR</b>		
<b>MICROPROCESSOR DESIGN</b>		
4.1	Introduction	56
4.2	The EC General Purpose Microprocessor	57
4.2.1	Instruction set	57
4.2.2	Datapath	58
4.2.3	Control unit	60
4.2.4	Complete circuit	66
4.3	VHDL for EC General- Purpose Microprocessor	66
4.4	Assembly Program	70

4.5	Block Diagram	70
4.6	Simulation	71
4.7	Hardware Implementation	71
<b>CHAPTER FIVE</b>		
<b>CONCLUSIONS AND RECOMMENDATIONS</b>		
6.1	Conclusions	72
6.2	Recommendations	72
	References	73



## LIST OF TABLES

<b>TABLE</b>	<b>TITLE</b>	<b>PAGE</b>
3.1	Shifter and rotator Operation	26
3.2	Control Words for Solving the Count 0's and 1's Problem	42
3.3	Control words to generate and sum the numbers from n down to 1 using the datapath in Figure 3.18 part a	44
4.1	Instruction set for the EC	57

## LIST OF FIGURES

FIGURE	TITLE	PAGE
2.1	FPGA structure	9
2.2	Typical logic block	9
2.3	Configurable logic block	10
2.4	A Lookup table	10
2.5	The fundamental units of VHDL	12
3.1	Full adder	19
3.2	Ripple-carry adder	20
3.3	Adder-subtractor combination	21
3.4	4-bit ALU	23
3.5	ALU operations	23
3.6	K-maps, equations, and schematics	25
3.7	A 4-bit shifter	26
3.8	A 3-to-8 decoder	27
3.9	An 8-to-3 encoder	28
3.10	Comparator for $x > y$	29
3.11	A 4-bit register with parallel load and asynchronous clear	31
3.12	A $2^n * m$ RAM chip	32
3.13	Internal circuit for a $16 * 4$ ROM with no connections made	33
3.14	A 2-to-1 multiplexer	34
3.15	Complete model of datapath	35
3.16	Comparator for generating the status signal (A is an odd number)	39
3.17	Dedicated datapath for solving the count 0's and 1's problem	42
3.18	General datapath with a register file	44

3.19	Finite-state machine models	45
3.20	Complete circuit model of the control unit	47
3.21	The complete circuit of the microprocessor	47
3.22	Summary of how the parts of a microprocessor fit together.	48
3.23	The summation problem	51
3.24	State diagram for the summation problem for the example	51
3.25	Generating the status signal ( $n \neq 0$ )	52
3.26	Construction of the control unit	55
4.1	Datapath for the EC	60
4.2	Control unit for the EC	65
4.3	Complete circuit for the EC general-purpose microprocessor	66
4.4	Block diagram for EC microprocessor.	70
4.5	A sample simulation trace of the countdown program running on the EC starting at the input 3.	71
4.6	Layout of the ALTERA kit	71

# List of Abbreviations

CPU	Central Processing Unit
IC	Integrated Circuit
ALU	Arithmetic and Logic Unit
I/O	Input/Output
BCD	Binary-Coded Decimal
TI	Texas Instruments
CADC	Central Air Data Computer
DSP	Digital Signal Processors
SOC	System-On-a-Chip
GPU	Graphics Processing Units
ASIC	Application-Specific Integrated Circuits
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
CPLD	Complex Programmable Logic Devices
PLA	Programmable Logic Array
PAL	Programmable Array Logic
GAL	Generic Array Logic
FFT	Fast Fourier Transform
CLB	Configurable Logic Block
LUT	Lookup Table
SRAM	Static Random Access Memory
CRAM	Configuration Random Access Memory
VHDL	Very High Level Hardware Description Language
IEEE	Institute of Electrical and Electronics Engineers
RTL	Register Transfer Level

FA	Full Adder
LE	Logic Extender
AE	Arithmetic Extender
CE	Carry Extender
RAM	Random Access Memory
ROM	Read Only Memory
PROM	Programmable Read Only Memory
EPROM	Erasable Programmable Read Only Memory
EEPROM	Electrically Erasable Programmable Read Only Memory
MUX	Multiplexer
FSM	Finite State Machine
EC	Enoch's Computer