Heaven's Light is Our Guide

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

Computer System Design Using FPGA for Educational Purpose

Author

Nahin Ul Sadad

Roll No. 123063

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

Supervised by

Dr. Md. Nazrul Islam Mondal

Professor

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

ACKNOWLEDGEMENT

At first, I would like to thank to God to give me the scope and enthusiasm for successful completion of my thesis work.

Firstly, I want to express my sincere appreciation and deepest sense of gratitude and respect to my supervisor Dr. Md. Nazrul Islam Mondal, Professor of department of Computer Science and Engineering, Rajshahi University of Engineering and Technology, Rajshahi, who not only supervised me but also his advice, guidance, continuous encouraging, kind help and thoughts have enriched my thesis without which this work would not have been materialized in the present form.

I am very grateful to all respective teachers of Computer Science and Engineering, Rajshahi University of Engineering and Technology, Rajshahi for good & valuable suggestions and inspiration from time to time.

Finally, I convey my thanks to my parents, friends and well-wishers for their constant inspiration and endeavor throughout the work.

Date: **Nahin Ul Sadad**

RUET, Rajshahi

Heaven's Light is Our Guide

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology

CERTIFICATE

This to certify that the thesis paper entitled "**Computer System Design Using FPGA for Educational Purpose**" submitted by **Nahin Ul Sadad, Roll: 123063**, has been carried out under our supervision. This thesis has been accomplished in partial fulfillment of requirement for degree of Bachelor of Science in Computer Science and Engineering.

Supervisor	**External Examiner**
_____	_____
Dr. Md. Nazrul Islam Mondal	**Dr. Md. Nazrul Islam Mondal**
Professor	Professor
Department of Computer Science & Engineering	Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology	Rajshahi University of Engineering & Technology
Rajshahi-6204	Rajshahi-6204

Abstract

As Computer Science (CS) grows rapidly, computer system has become more complex with many levels of abstraction. As a result, traditional computer system courses like digital electronics, computer architecture, interfacing, compiler etc. fail to show their interconnection in computer system as a whole. In this thesis, a computer system is built which combines digital electronics, computer architecture, assembler and interfacing together using modular/building block approach. Computer written in Verilog consists of digital electronics building blocks which combine logic gates, then computer architecture building blocks which combine digital electronics building blocks, finally computer architecture building blocks are combined to build computer which can be run on FPGA. Then assembler written in C or any programming language can be built on the top of computer to support assembly programming to make programming task easier. Finally, computer is interfaced with LED, seven segment display of FPGA board and VGA monitor to show output of the computer program running on computer in FPGA.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

This chapter starts with motivation where area of problems and literature review are discussed. Then in proposed methodology section, an methodology is proposed to solve the problem. Then, in thesis contribution, contributions of thesis are outlined. Finally, chapter ends with conclusion.

1.2 Motivation

Computer Science is a highly technical and practical field. As each individual discipline in computer science grows rapidly, their courses has become more broader and specialized in scope and complexity. Since only small portion of individual discipline can be taught within short time-span of undergraduate studies, these courses focus more on theoretical foundations with many higher levels of abstraction [1].

Several courses are offered focusing on computer system design like digital electronics, computer architecture, interfacing, compiler etc. An important purpose of these courses is to provide understanding on how computer works as a whole. These courses teach details of each level of abstraction well but they fail to teach how they are interconnected with each other in computer system as a whole [2, 3].

A major problem with typical courses on computer hardware is a failure to "connect the dots": various details of each level of abstraction are covered, but the crucial connections between the levels are largely ignored. To understand computer systems properly, it is necessary

to have a working understanding of each level of abstraction, and also to understand the connections between the levels. To achieve this goal, it is essential to select the topics to be covered in each level carefully, so the techniques presented in one level are sufficient for supporting the next level up.

Lorem ipsum [4] dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1.3 Proposed Methodology

Lorem ipsum [4] dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1.4 Thesis Contributions

Lorem ipsum [4] dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1.5 Why using FPGA and Verilog?

Computer Science is a highly technical and practical field. As each individual discipline in computer science becomes more specialized,

Lorem ipsum [4] dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation

ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1.6 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2 - Digital Electronics Building Block

This chapter describes digital electronics building blocks which can be built by logical gates. It includes combinational circuits like adder, multiplier etc. and sequential circuits like D flip-flop, register etc.

Chapter 3 - Computer Architecture Building Block

This chapter first describes instruction set architecture (ISA) of computer itself. Then computer architecture building blocks are described which can be built by combining digital electronics building block developed in previous chapter. It includes arithmetic and logic unit (ALU), register file, control unit (CU) and instruction memory.

Chapter 4 - Computer Design

This chapter combines computer architecture building blocks to build computer. It includes central processing unit (CPU) and its data path design.

Chapter 5 - Interfacing and Assembler

This chapter describes interfacing of computer with LED, seven segment display and VGA monitor. Then assembler is described which is built on the top of computer to make programming task easier.

Chapter 6 - Simulation and Results

This chapter shows simulation of assembler and computer on software. Finally, it describes computer system running on fpga to verify the simulation results.

Chapter 7 - Conclusion and Future works

This chapter concludes the thesis and shows direction of future work.

1.7 Conclusion

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Chapter 2

Digital Electronics Building Block

2.1 Introduction

This chapter focuses on digital electronics building blocks. This chapter begins with digital circuits. Then two types of digital circuits, combinational and sequential circuits with their respective components are discussed. Finally, chapter ends with conclusion.

2.2 Digital Circuits

Digital electronics building blocks are built by using logic gates. These blocks can be written in Verilog. To make block design easier, dataflow and behavioral style can also be used. There are two types of digital circuits that are used in computer. They are: combinational and sequential circuits. These circuits can be found in any traditional digital electronics books like [?].

2.3 Combinational Circuit

Combinational circuits are mainly used for arithmetic, datapath selection operations etc.

2.3.1 Adder/Subtracter

In this computer design, specification of adder/subtracter is:

1. **Input:** 32-bit a (input 1)
2. **Input:** 32-bit b (input 2)

3. **Input:** 1-bit addorsub [if addorsub == 1 then a - b = a + (-b) = a + not(b) + 1 = a + (b xor addorsub) + addorsub] [if addorsub == 0 then a + b = a + b = a + b + 0 = a + (b xor addorsub) + addorsub]
4. **Output:** 32-bit s (result)

Example of 4-bit carry look ahead adder is shown below.

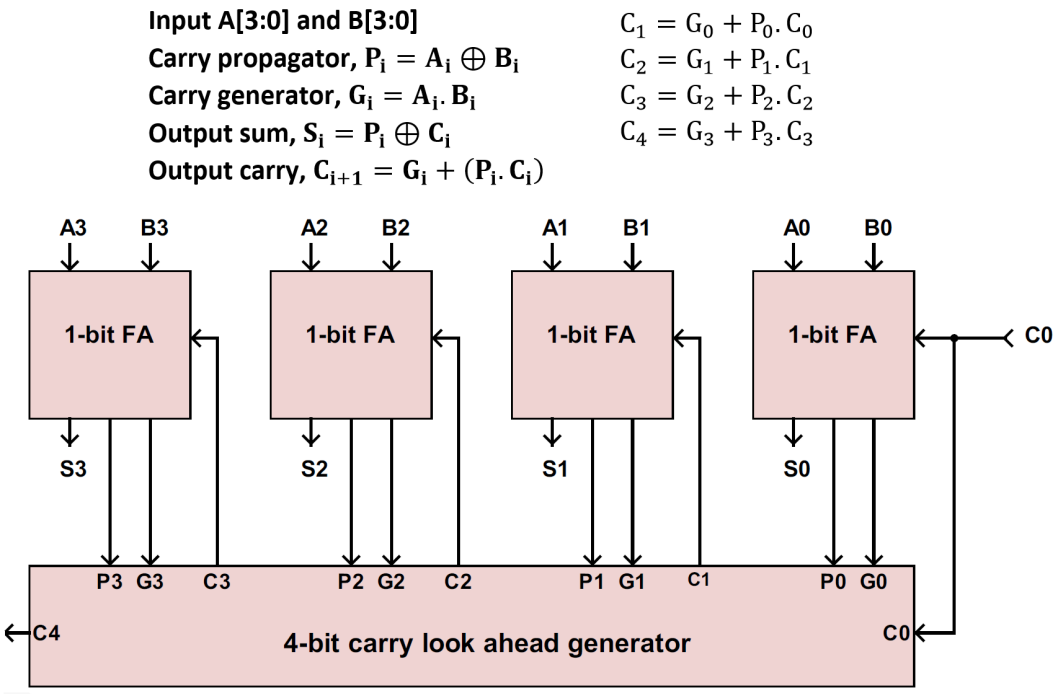

Figure 2.1: 4-bit carry look ahead adder

As shown in figure, adder takes A[3:0], B[3:0] and C[0] as input and produces S[3:0] and C[4] as output. Each input A[i] and B[i] go through 1-bit full adder to produce sum S[i]. Logic diagram of full adder [5] is shown in figure.

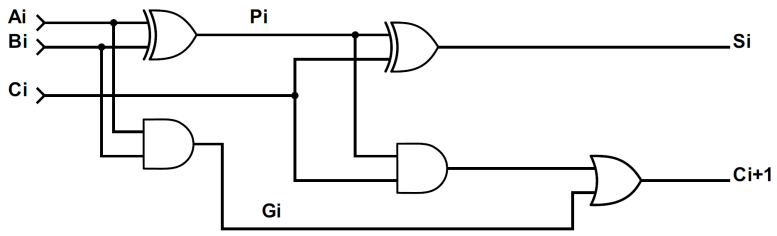

Figure 2.2: Logic diagram of full adder

Since all the carry C[3:0] will be generated at the same time, C[4] will not have to wait for C[3] and C[2] to propagate. Logic diagram of a look ahead carry generator [5] is shown in figure.

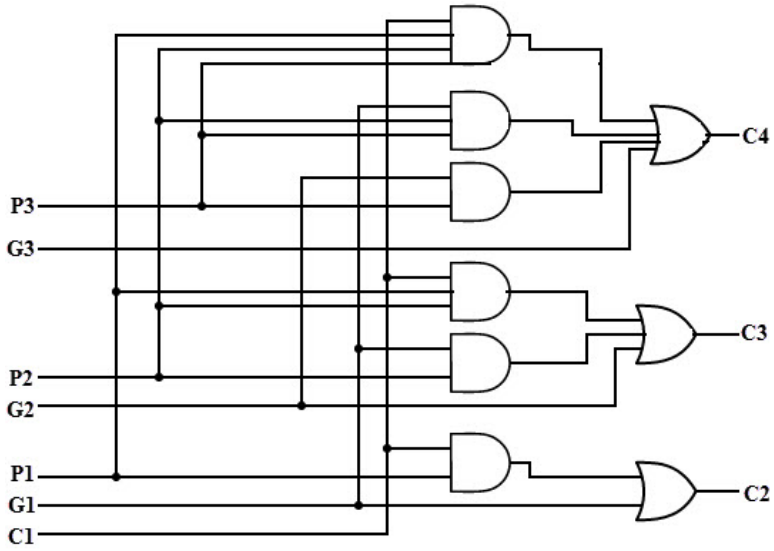

Figure 2.3: Logic diagram of 4-bit carry look ahead adder

2.3.2 Multiplier

In this computer design, specification of multiplier is:

1. **Input:** 16-bit a (input 1)
2. **Input:** 16-bit b (input 2)
3. **Output:** 32-bit r (result)

Example of multiplier using partial product accumulation method is described below. Multiplier takes A[3:0] and B[3:0] as input where A is multiplicand and B is multiplier. The multiplication of A and B is shown in figure. Each ANDed term is a partial product. The resulting product is accumulated as shown in figure.

				A3	A2	A1	A0
				B3	B2	B1	B0
				A2 B0	A2 B0	A1 B0	A0 B0
			A3 B1	A2 B1	A1 B1	A0 B1	
		A3 B2	A2 B2	A1 B2	A0 B2		
	A3 B3	A2 B3	A1 B3	A0 B3			
S7	S6	S5	S4	S3	S2	S1	S0

Figure 2.4: Product accumulation

The building block of multiplier [6] is shown in figure. Building block takes previous sum (Sum In), partial product X & Y and carry-in (Cin) as input. These inputs go through AND gate and full adder. Finally, building block produces carry-out (Cout) and next sum (Sum Out).

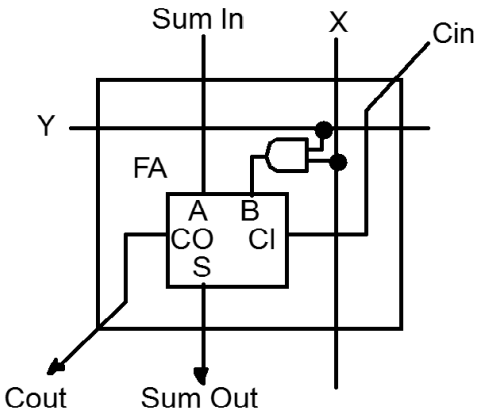

Figure 2.5: Building block of multiplier

4-by-4 combinational multiplier [6] is shown in figure.

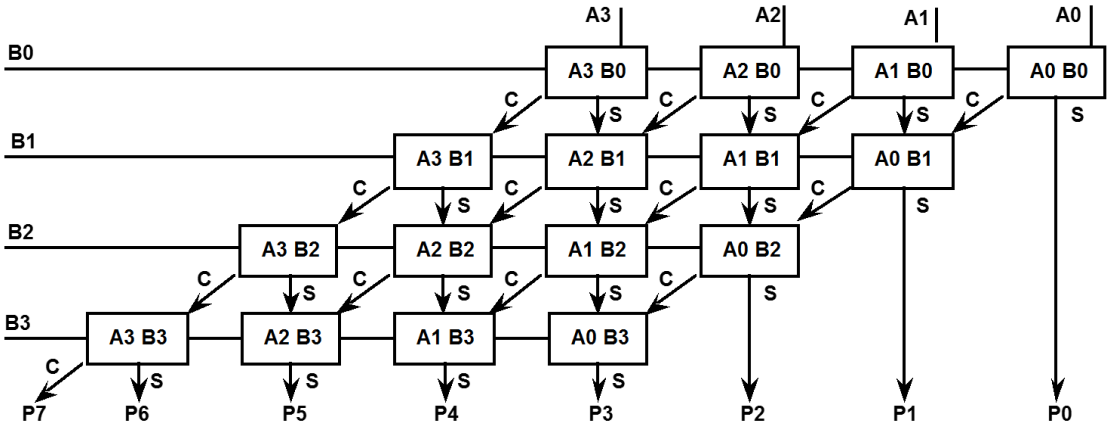

Figure 2.6: Multiplier

2.3.3 Shifter

In this computer design, specification of shifter is:

1. **Input:** 32-bit d (data)
2. **Input:** 2-bit sa (shift amount)
3. **Input:** 1-bit r (shift right) [shift right:1 and shift left:0]
4. **Input:** 1-bit a (arithmetic shift) [arithmetic:1 and logical:0]
5. **Output:** 32-bit sh (shifted data)

Example of 4-bit shifter [7] is shown below:

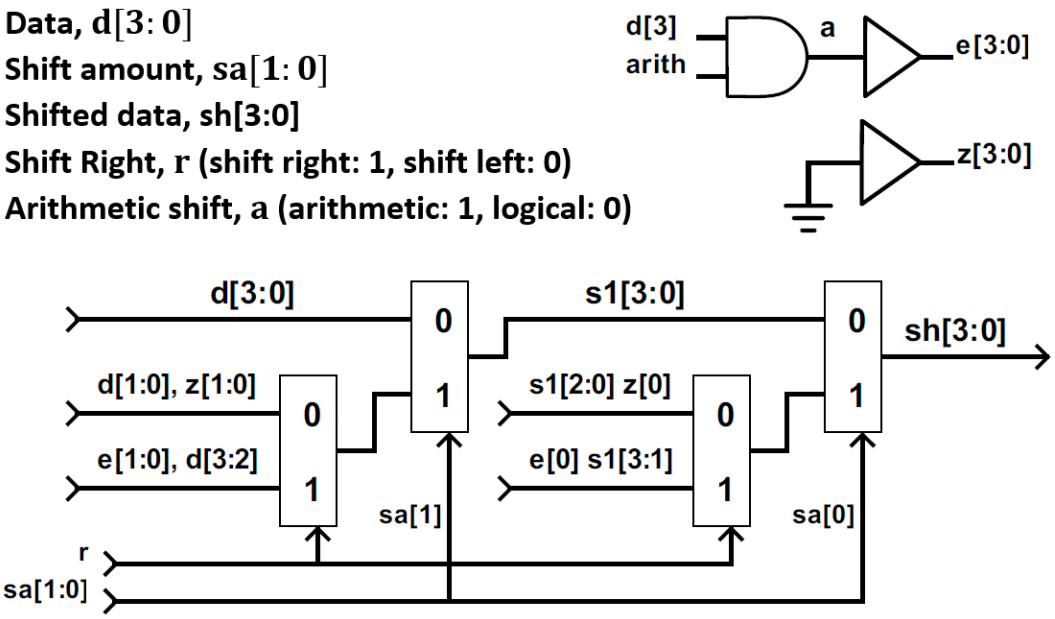

Figure 2.7: 4-bit shifter

As shown in figure, If arith is a 0 (logical shift), e will be zero which is generated by connecting each bit of e to ground. If arith is a 1 (arithmetic shift), every bit of e is equal to the sign bit of d which is generated by an AND gate. Two 2-to-1 multiplexers are used to select path depending on right or left shift bit, r. Other 2-to-1 multiplexers are used to select path depending on shift amount, sa.

2.3.4 Multiplexer

In this computer design, two types of multiplexer are used. They are: 2-to-1 multiplexer and 4-to-1 multiplexer. Specification of 2-to-1 multiplexer is:

1. **Input:** 32-bit d0 (data 0)
2. **Input:** 32-bit d1 (data 1)
3. **Input:** 1-bit s (select bit)
4. **Output:** 32-bit y (output)

Specification of 4-to-1 multiplexer is:

1. **Input:** 32-bit d0 (data 0)
2. **Input:** 32-bit d1 (data 1)

3. **Input:** 32-bit d2 (data 2)
4. **Input:** 32-bit d3 (data 3)
5. **Input:** 2-bit s (select bit)
6. **Output:** 32-bit y (output)

Example of 2-to-1 multiplexer is shown below:

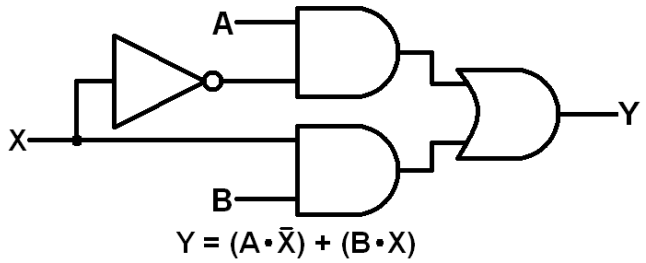

Figure 2.8: 2-to-1 multiplexer

As shown in figure, 2-to-1 multiplexer takes A and B as input and X as select bit. Depending on X, it connects A or B with Y.

2.3.5 Decoder

In this computer design, specification of 5-to-32 line decoder is:

1. **Input:** 5-bit a (input)
2. **Input:** 1-bit e (master enable)
3. **Output:** 32-bit y (output)

Example of 2-to-4 line decoder is shown below:

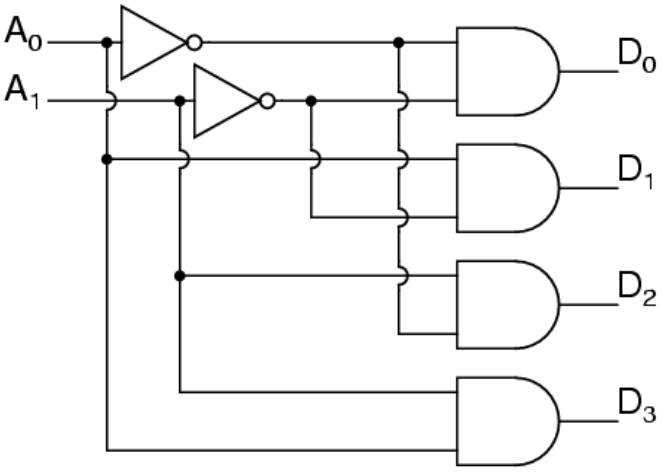

Figure 2.9: 2-to-4 line decoder

As shown in figure, 2-to-4 line decoder takes takes A[1:0] as input. Depending on A, One of y[3:0] outputs will be active High.

2.4 Sequential Circuit

Sequential circuits are mainly used to store data temporarily.

2.4.1 D Flip-flop

In this computer design, specification of D flip-flop is:

1. **Input:** 1-bit d (data)
2. **Input:** 1-bit en (enable)
3. **Input:** 1-bit c (clock input)
4. **Input:** 1-bit clrn (clear)
5. **Output:** 1-bit q (output)

Example of D Flip-flop is shown below:

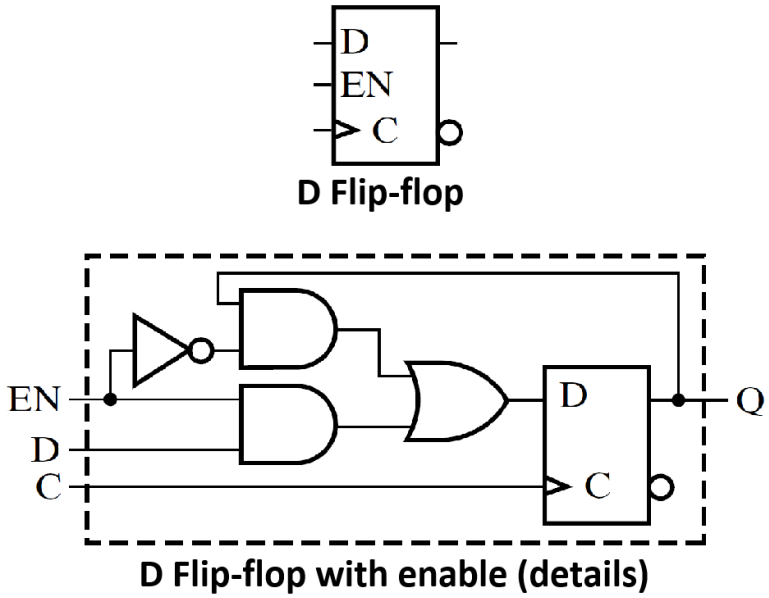

Figure 2.10: D Flip-flop with its detailed view

As shown in figure, D flip-flop takes 1-bit data (d), enable (en), clock input (c) and clear bit (clrn) as inputs. It stores the value d when clock is on positive edge. It can be thought of as a basic memory cell. If clear bit is on, then it will be reset. It generates q which is data stored in D flip-flop.

2.4.2 Register

In this computer design, specification of 32-bit register is:

1. **Input:** 32-bit d (data)
2. **Input:** 1-bit load (enable)
3. **Input:** 1-bit c (clock input)
4. **Input:** 1-bit clrn (clear)
5. **Output:** 32-bit q (output)

Example of D Flip-flop and register [5] are shown below:

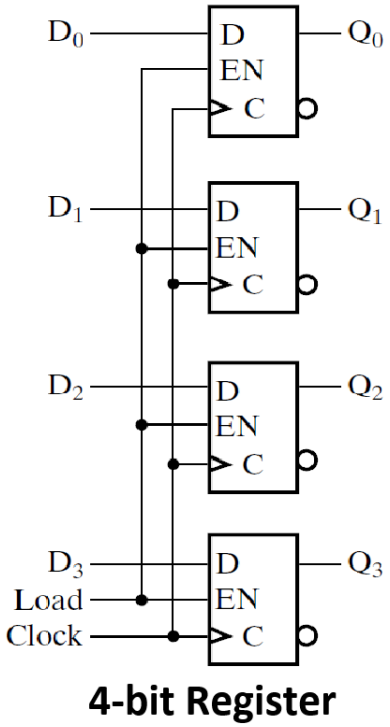

Figure 2.11: 4-bit register

As shown in figure, 4-bit register can be formed by connecting 4 D flip-flop in parallel. It takes 4-bit data d[3:0], enable load and clock as inputs. It stores 4-bit data when clock is on positive edge and load is enabled. It produces its stored data as output 4-bit q.

2.5 Conclusion

This chapter explained all necessary digital electronics building blocks needed to implement computer. These blocks can be described by using Verilog language. Necessary testing should be performed to verify those blocks. Next chapter will combine all these blocks to implement computer architecture building blocks.

Chapter 3

Computer Architecture Building Block

3.1 Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

3.2 Some Section

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Figure 3.1: A Camera

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

(a) A Typewriter (b) A Macbookpro

Figure 3.2: Typewriter and Macbookpro

References

[1] A. Clements, "Computer architecture education," *IEEE Micro*, vol. 20, no. 3, pp. 10–12, 2000.

[2] S. Schocken, N. Nisan, and M. Armoni, "A synthesis course in hardware architecture, compilers, and software engineering," in *ACM SIGCSE Bulletin*, vol. 41, pp. 443–447, ACM, 2009.

[3] J. T. O'Donnell, "Connecting the dots: Computer systems education using a functional hardware description language," *arXiv preprint arXiv:1301.5075*, 2013.

[4] M. M. Haque and D. M. A. Hossain, "Some awesome title," *Some awesome journal*, vol. 10, no. 20, pp. 30–40, 2016.

[5] M. M. Mano, C. R. Kime, T. Martin, *et al.*, *Logic and computer design fundamentals*, vol. 3. Prentice Hall, 2008.

[6] R. H. Katz and G. Borriello, "Contemporary logic design," 2005.

[7] Y. Li, *Computer Principles and Design in Verilog HDL.* John Wiley & Sons, 2015.