Simulation, Design and Hardware Implementation of a Low Cost Processor

Ateeq A. Khan

***Abstract*-Embedded microprocessor has been widely used as a tool for technological innovations and cost reduction for the last forty years. The main characteristic determining its performance are its speed and programmability. Therefore, for a design to be competitive its processor has to fit for the characteristics like: relative inexpensiveness, flexibility, adaptation, speed and re-configurability. A solution to this is the use of Field Programmable Gate Arrays (FPGA). This paper describes the realization of a 4-bit FPGA based simple low cost processor. The system is implemented on the Xilinx Spartan 3 xc3s200FT256 using ISE foundation 9.2i and VHDL. 88(4%) of the slices are used. A maximum frequency of 61.451 MHz was reached with a maximum delay of 10.145 ns.**

Keywords: Processor, FPGA, VHDL, Simulation, Synthesis

I. INTRODUCTION

Microprocessor is the main building block for technological components and functions as the heart of smart devices. It has a wide range of uses and a huge market. These days embedded microprocessors are used in wireless communication, computing and robotic devices. They comprise thousands of electronic components and use a collection of machine instructions which enables them to perform mathematical operations and transfer information from one memory location to another. The hardware implementation of a simple low cost 4-bit microprocessor on a Spartan 3 FPGA (field programmable gate array) is developed which performs programming of some simple operations using VHDL.

Manuscript received February 11, 2012. This work was supported by Salman Bin Abdul Aziz University, Al Kharj, Saudi Arabia.

Dr. Ateeq Ahmad Khan is an Assistant Professor at the Department of Electrical Engineering at Salman Bin Abdul Aziz University. Phone-+966508840934 , Email ateeq@ksu.edu.sa

II. DIGITAL CIRCUIT

Figure 1 shows how the different parts and components fit together to form the microprocessor. From transistors, the basic logic gates are built. Logic gates are combined together to form either combinational circuits or sequential circuits. The difference between these two types of circuits is only in the way the logic gates are connected together. Latches and flip-flops are the simplest forms of sequential circuits, and they provide the basic building blocks for more complex sequential circuits. Certain combinational circuits and sequential circuits are used as standard building blocks for larger circuits, such as the microprocessor. These standard combinational and sequential components usually are found in standard libraries and serve as larger building blocks for the microprocessor. Different combinational components and sequential components are connected together to form either the datapath or the control unit of a microprocessor. The circuit for either a dedicated or a general microprocessor in the end will be produced by combining the datapath and the control unit together.

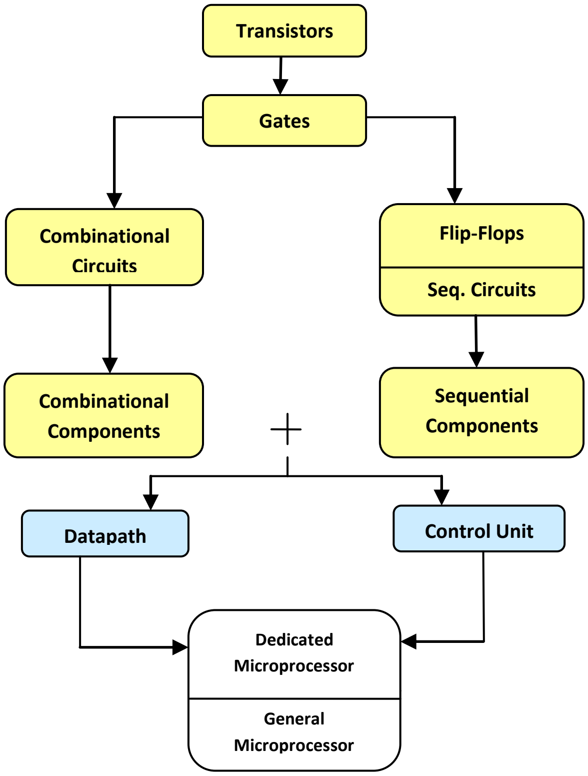

Figure 1: Microprocessor Components

III. MICROPROCESSOR DESIGN

This microprocessor performs the following instructions

General forma	Object Instruction code	Operation	Comment
LDA<addr>	8<addr>	A □□M<addr>	load Accumulator direct
ADD<addr>	A<addr>	A □□A+M<addr>	Add Accumulator direct
SUB<addr>	B<addr>	A □□A-M<addr>	Sub Accumulator direct
JZ<addr>	C<addr>	If Z=1 then PC □□<addr>	Jump on Zero flag set
JC<addr>	D<addr>	If C=1 then PC □□<addr>	Jump on Carry flag set
AND<addr>	E<addr>	A □□A and M<addr>	logic AND Accumulator direct
CMA	0	A □□not A	Complement Operation
INCA	2	A □□A+1	Increment Accumulator
DCRA	4	A □□A-1	Decrement Accumulator
HLT	6	Halt	Halt CPU

Table 1: Instruction Set

A=Accumulator

M= Memory

ad**dr**= Address

Z=Zero Flag

C= Carry Flag

PC= Program Counter

The details about the design and actual architecture of the control unit, what operation code (opcode) is assigned to each of the instructions and how many bits are to be encoded in an instruction and what are the control words used in micro program are given in control unit of microprocessor.

III.(i) MEMORY

At first the memory was created as Random Accesses Memory (RAM) with 16 words of 4-bits, in order to support the store and load instructions, but since our design has no actual inputs the memory was turned into a Read Only Memory (ROM) with same size so we can store our program that we want the microprocessor to execute. As a result the store operation was canceled and subsequently the connection between the accumulator and the memory was eliminated.

III.(ii) CONTROL UNIT DESIGN

The purpose of the control unit is to initiate a series of sequential steps of micro-operations. During any given time, certain operations are to be initiated while all others remain idle. Thus, the control variables at any given time can be represented by a string of l's and 0's called a control word. As such, control words can be programmed to initiate the various components in the system in an organized manner. A control unit whose control variables are stored in a memory is called a micro program control unit. Each control word of memory is called a microinstruction, and a sequence of microinstructions is called a micro program since alteration of the micro program is seldom needed, the control memory can be a ROM. The use of a micro program involves placing all control variables in words of the ROM for use by the control unit through successive read operations. The content of the word in the ROM at a given address determines the micro operations for the system. Figure 2 illustrates the general configuration of the micro program control unit. The control memory is assumed to be a ROM, within which all control information is permanently stored. The control memory address register specifies the control word read from control memory. It must be realized that a ROM operates as a combinational circuit, with the address value as the input and the corresponding word as the output. The content of the specified word remains on the output wires as long as the address value remains in die address register, no read signal is needed as in a random-access memory. The word out of the ROM should be transferred to a buffer register if the address register changes while the ROM word is still in use. If the change in address and ROM word can occur simultaneously, no buffer register is needed.

The word read from control memory represents a microinstruction. The microinstruction specifies one or more micro operations for the components of the system. Once these operations are executed, the control unit must determine the next address. The location of the next microinstruction may be the next one in sequence, or it may be located somewhere else in the control memory. For this reason, or it is necessary to use some bits of the microinstruction to control the generation of the address for the next microinstruction. The next address may also be a function of external input conditions. While the micro operations are being executed, the next address is computed in the next-address generator circuit and then transferred (with the next clock pulse) into the control address register to read the next microinstruction.

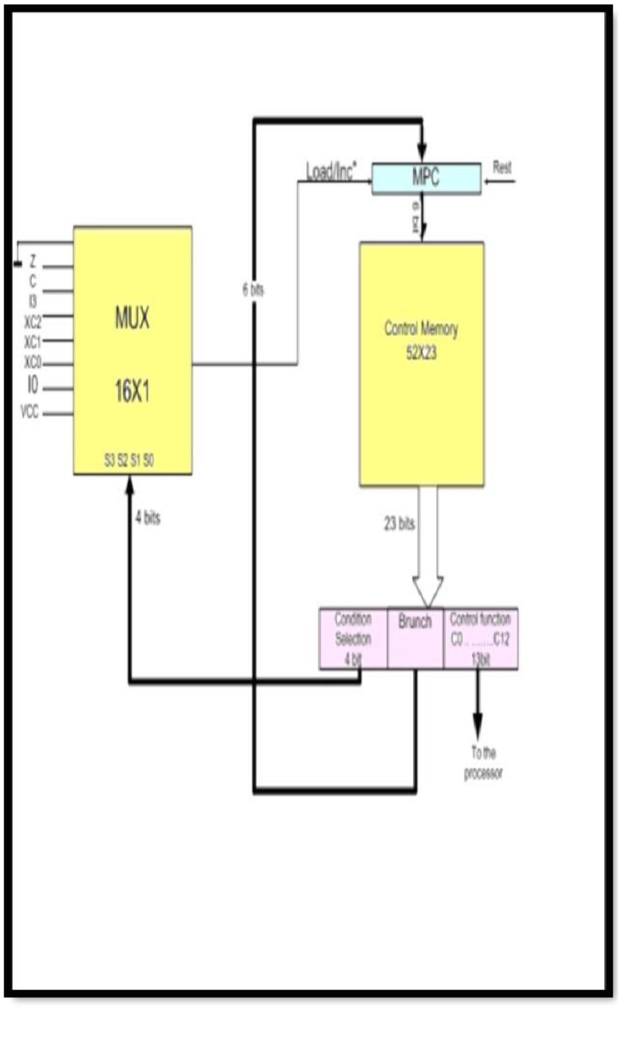

Figure 2: Control Unit Architecture

III.(iii) CONTROL MEMORY

Control memory is a ROM with 6-bit address given as 64 words, but actually only the first 48 are used. Each word is 23-bit long of which the most significant 13-bit are sent to the datapath as controlling signals determining how each component in the datapath behaves subject to what kind of operation is being preformed. The 6-bits after the control bits in the control word are the brunch bits which make the ROM jump from one word to another if needed depending on the status conditions coming from the datapath particularly the zero and carry flags as well as the signals from the IR . The 4 least significant bits in the control word are the condition selectors which select one of the status conditions to be checked if needed and do so by serving as the selecting signals for a 16X1 multiplexer where the output of the multiplexer is the control signal for the MPC. Based on this way of operating, the controlling words were created. After that the VHDL code was written for it and simulated. A simple behavioral model was used for creating the ROM. Since it is a read-only memory, no clock signal or write-enable pin is necessary. The circuit contains a pile of pre-stored words, being the one selected by the address input (addr) presented at the output (data).

III.(iv) MICROPROCESSOR CONNECTIONS

The microprocessor is built by connecting the datapath with the control unit. This is done by using the same method in creating the datapath and control unit which is port mapping of the structural model, with the inputs being a clock signal to help regulate the transfer of data inside the microprocessor, and a reset signal to make the microprocessor go to the start of the control words stored in the ROM by resetting the MPC which holds the address of the ROM. The microprocessor has a 4-bit output which is the result of the instructions executed by it.

The program stored in the memory of the datapath which defines the instructions that will be executed and the order they are executed in, can be written in any manner seen appropriate for a specific function(s) that the microprocessor is intended to perform. The only limitation is that the space in which the instructions can be stored is limited to 16 word, bearing in mind that some instructions need two words while other need only one word of space. The program written is designed to test all of the

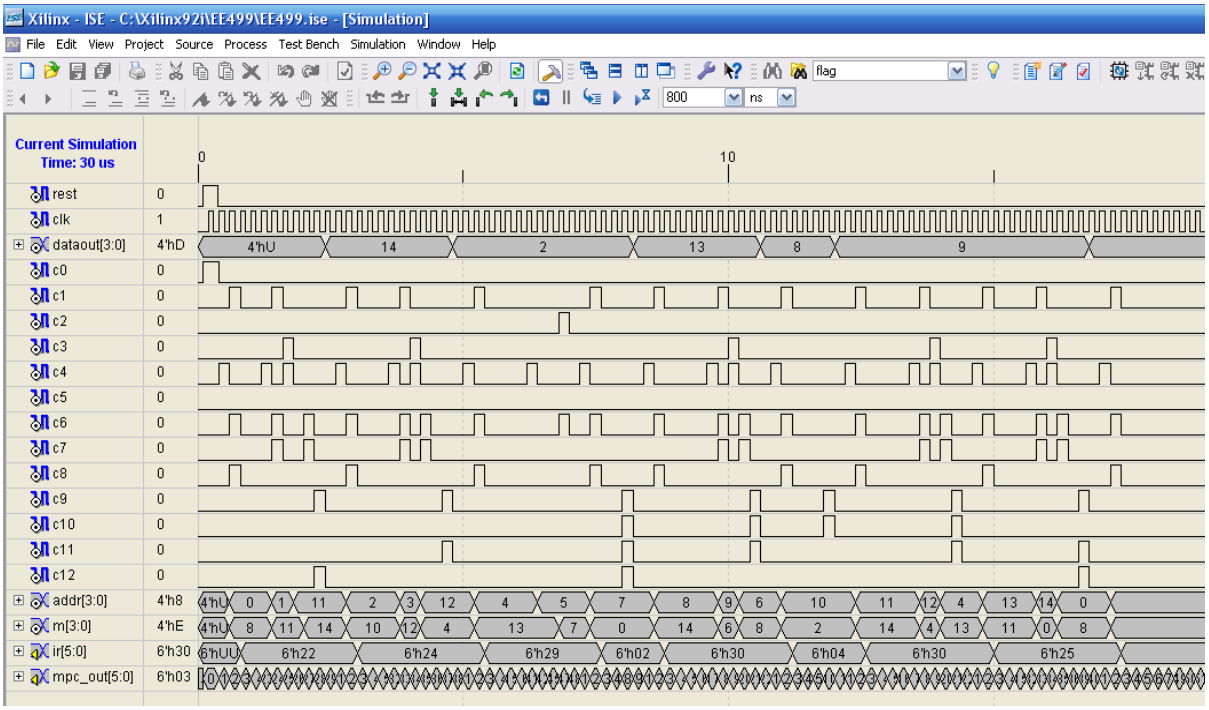
Xilinx - ISE - C:\Xilinx92i\EE499\EE499.ise - [Simulation]
File Edit View Project Source Process Test Bench Simulation Window Help
flag
800 ns
Current Simulation Time: 30 us
0
10
rest 0
clk 1
dataout[3:0] 4'hD
c0 0
c1 0
c2 0
c3 0
c4 0
c5 0
c6 0
c7 0
c8 0
c9 0
c10 0
c11 0
c12 0
addr[3:0] 4'h8
m[3:0] 4'hE
ir[5:0] 6'h30
mpc_out[5:0] 6'h03
4'hU 14 2 13 8 9
6'hUU 6'h22 6'h24 6'h29 6'h02 6'h30 6'h04 6'h30 6'h25

Figure 3: The Microprocessor Behavioral Simulation

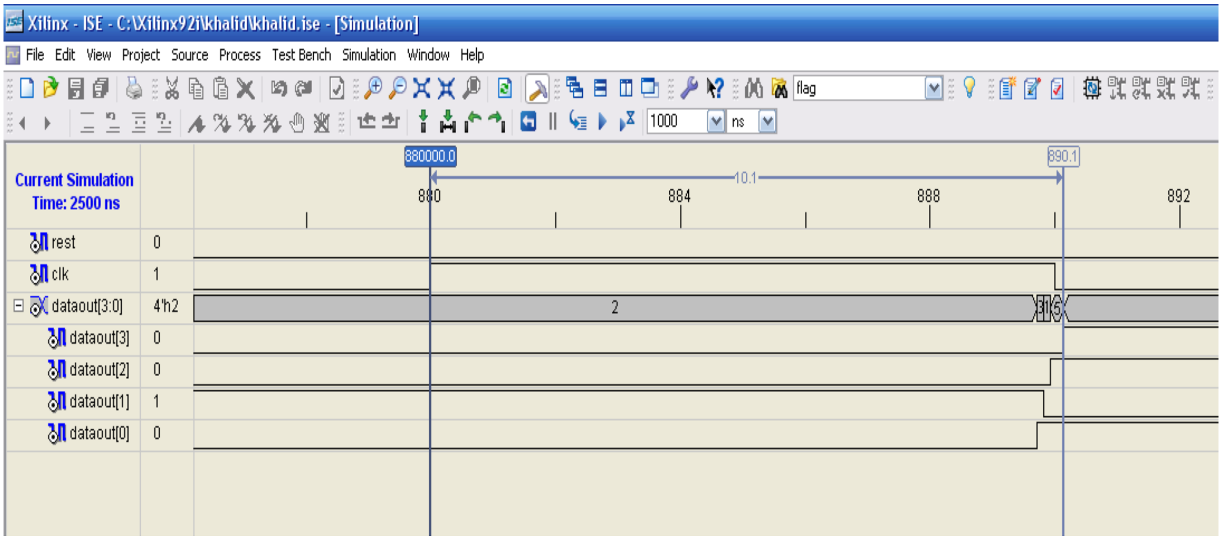
Xilinx - ISE - C:\Xilinx92i\khalid\khalid.ise - [Simulation]
File Edit View Project Source Process Test Bench Simulation Window Help
flag
1000 ns
880000.0
890.1
10.1
Current Simulation Time: 2500 ns
880
884
888
892
rest 0
clk 1
dataout[3:0] 4'h2
dataout[3] 0
dataout[2] 0
dataout[1] 1
dataout[0] 0
2

Figure 4: The Microprocessor Timing Simulation

instructions the microprocessor can perform in order to determine the success of this paper.

The behavioral simulation has been done to ensure that the microprocessor performs the program stored in its memory, after that the place and route simulation (timing simulation) is done to find out the delay and if it is too much that it could hinder the performance of the microprocessor. Both simulations are shown in figure 3 and figure 4.

IV. Conclusion

A large number of low cost prototype boards available for various FPGAs. This is certainly not true for ASICs as the tool cost alone can be prohibitive in many cases. The other reason for the low setup cost is that the use of an FPGA means that the mask costs associated with an ASIC are avoided which can be a significant saving for a modern technology.

The design of the microprocessor is analyzed carefully to ensure that any failure would be corrected, this is done at each step making sure that every component is functioning accurately on its own and also when it is connected to other parts. The measure of accuracy is the simulation which could give a clear picture on how the component performs as well as helping in finding what causes failures when they occur.

FPGA-based 4-bit microprocessor on Xilinx Spartan 3 board is successfully implemented using VHDL language. The microprocessor is made of the control unit, an arithmetic logic unit (ALU), a memory unit and registers. The processor has a maximum frequency of 61.451 MHz and 88 slices are utilized. It can perform ten different instructions as described in the specifications of the microprocessor. Simulations are done to ensure its functionality.

The microprocessor can be upgraded by adding the store instruction to it by changing the datapath memory into a RAM, also by increasing the number of bits the microprocessor can process; it could be used in many applications.

As far as the future work is concerned, latest Xilinx with new family of FPGA can be used to for speed improvement and area reduction. Altera can be tried for comparison of two technologies.

References

[1] Enoch O. Hwang, “Digital Logic and Microprocessor Design with VHDL”, *Brooks Cole, 2005*.

[2] Volnei. A. Pedroni, “Circuit Design with VHDL*”, Massachusetts, MIT Press, 2004.*

[3] Pierre-Emanuel Gaillardon, M.Haykel Ben Jamma, Giovanni Betti Benevetti, Fabien Clermidy, Luca Perniola, “Emerging Memory Technologies for Reconfigurable Routing in FPGA architecture”; *IEEE International Conference on Electronics, Circuits and Systems (ICECS)-invited paper, December 2010.*

[4] J.Rose and D.Hill, “ Architectural and Physical Design Challenges for One-Million Gate FPGAs and Beyond”, *Proceedings of the ACM Fifth International Symposium on Field-Programmable Gate arrays, pp.129-132, Feb.1997.*

[5] Singh, et.al. “A Novel High Throughput Reconfigurable FPGA Architecture”, *Proceedings of the ACM/SIGDA Symposium on Field-Programmable Gate arrays, pp. 22-29, Feb.2000.*

[6] M. Abd-El-Barr and H. El-Rewini, “Fundamentals of Computer Organization and Architecture”, *New Jersey, John Wiley & Sons. Inc, 2005.*

[7] E. Ayeh, K. Agbedanu, Y. Morita, O. Adamo, and P. Guturu, “FPGA Implementation of an 8-bit Simple Processor”, in the proceedings of *Region 5 Conference, 2008 IEEE held in Kansas City, MO, April 2008.*

[8] Baker, R. Jacob (2010). “CMOS: Circuit Design, Layout, and Simulation”, *Third Edition. Wiley-IEEE. pp. 1174. ISBN 978-0-470-88132-3. http://CMOSedu.com/*

[9] Wiśniewski,Remigiusz . “Synthesis of compositional microprogram control units for programmable devices”. *Zielona Góra: University of Zielona Góra. pp. 153. ISBN 978-83-7481-293-1.2009.*

[10] Ralph D. Wittig and Paul Chow. “OneChip: An FPGA Processor with Reconfigurable Logic”. *In The Fourth Annual IEEE Symposium on FPGAs for Custom Computing Machines FCCM'96, pages 126-135. IEEE, March 1996*

[11] P. A. Wilsey T. J. McBrayer, and D. Sims, "Towards A Formal Model of VLSI Systems Compatible with VHDL," *VLSI '91, A. Halaas and P. B. Denyer (eds), Elsevier Science Publishers B.V. (North-Holland), 225-236, 1991.*

[12] M. Kauppi and J-P. Soininen, "Functional Specification and Verification of Digital Systems using VHDL Combined with Graphical Structured Analysis," *EuroVHDL, 1991*

[13] D. Borrione and J. L. Paillet, "An approach to the formal verification of VHDL descriptions," *IMAG/Artemis, Grenoble, November 1987.*

[14] S. Hadjis, A. Canis, J.H. Anderson, J. Choi, K. Nam, S. Brown. T. Czajkowski, "Impact of FPGA architecture on resource sharing in high-level synthesis," to appear in the *ACM/SIGDA International Symposium on Field Programmable Gate Arrays, to be held at Monterey, CA, February 2012.*

[15] M. H. Ben Jamaa, D. Atienza, Y. Leblebici, and G. De Micheli. “Programmable Logic Circuits Based on Ambipolar CNFET”. *Design Automation Conference (DAC), Anaheim, California, USA, 2008.*

[16] Krishna Kant “Microprocessors and Microcontrollers: Architecture Programming And System Desig n”. *PHI Learning Pvt. Ltd., 2007 ISBN 8120331915 page 61, describing the iAPX 432*

[17] Gerry Kane, Adam Osborne “16 Bit Microprocessor Handbook”. *ISBN 0079310435 (0-07-931043-5)*

[18] L.Sterpone, M. Violante. “Hardening FPGA-based systems against SEUs: A new design methodology”. *Academy Publisher Journal of Computers, Vol.1, No.1, April 2006, pp.22-30.*

[19] JD Plummer, M.D. Deal, P.B. Griffin, “Silicon VLSI Technology”, Prentice Hall Inc., USA, 2000

[20] A.Singh, et al. “ A Novel High Throughput Reconfigurable FPGA Architecture”, Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, pp. 22-29. Feb 2000.

Ateeq Ahmad Khan was born in India in 1955. He obtained his PhD in Electronics Engineering from Aligarh Muslim University in 1999.
He joined as Assistant Professor in AMU in 1981 and was promoted as Associate Professor in 1987. Later, he joined Electrical Engineering Department, College of Engineering, King Saud University, Riyadh in 2003 and then joined Salman Bin Abdul Aziz University at Al Kharj and is presently continuing there. Dr. Khan is a fellow of Institute of Electronic and Telecommunication Engineering IETE and senior member of Institution of Engineers, India.