ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 4, Issue 6, June 2016

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Design and Implementation of 8-Bit Pipelined Microprocessor using Verilog HDL

Manasa. V[1]
Department of Electronics and Communication Engineering
Auroras Engineering College
Hyderabad – India

Sai Prasad Goud. A[2]
Department of Electronics and Communication Engineering
Auroras Engineering College
Hyderabad – India

Abstract: This paper includes the design and implementation of 8-bit pipelined architecture of microprocessor. Pipelining is the process of overlapping of multiple instructions during execution time. It splits one task in to multiple subtasks. The top level design of this processor consists of modules like program counter, memory unit, ALU and control unit. The functional performance of the designed modules are tested using Xilinx tool. Different design steps such as verilog code, simulation, synthesis and implementation on fpga are performed using Xilinx 14.5 ISE and is implemented on Xilinx Spartan 3E. The implemented results subsequently shows the optimized circuit performance with a maximum frequency of 156.372 MHz and a minimum period of 6.395ns.

Keywords: pipelining; verilog HDL; xilinx; fpga; micro processor.

I. INTRODUCTION

Now a days microprocessors are involved in almost every sphere of human life. It is one of the most exciting technology invented in electronics since the beginning of transistors .This device has contributed immensely in revolutionizing the field of digital electronics. Microprocessor's applications and uses could range from very complex process controllers to simple game machines. Since their invention by Xilinx in1984, FPGAs have gone from being simple glue logic chips to actually replacing the custom application specific integrated circuits(ASICs) and processors for signal processing and control applications.

FPGA devices consists of a large number of logic blocks connected with large number of interconnection lines. Each logic block consists of logic function implemented in Look Up tables together with one or more flip flops. Each independent processing task is assigned to a dedicated section of the chip and can function autonomously without any influence from other logic blocks. The FPGA optimized processor can be used to enhance computer architecture education by programming selected CPU parts onto the processor. The processor debugging tool would provide us with complete control over the processor. FPGAs provide hardware timed speed and reliability, but they do not require high volumes to justify the large upfront expense of custom ASIC design. Any hardware design can be described in terms of its operations at different levels of abstraction, from system through to logic gates .At each level of this hierarchy the overall inputs and output remain the same but the functionality of distinct sections become clearer with the help of detailed schematics.

The microprocessor or CPU reads each instruction from the memory and executes it. It processes the data as required in the instructions. The processing is in the form of arithmetic and logical operations. The data is retrieved from memory or taken from an input device and the result of processing is stored in the memory or delivered to an appropriate output device, all as per the instructions. To perform all these functions, the microprocessor incorporates various functional units in an appropriate manner.

Research in microprocessor architecture investigates all possible ways to increase the speed at which the microprocessor executes programs. All approaches have in common the goal of exposing and exploiting parallelism hidden within programs. One of the approaches is pipelining.

Pipelining is a technique used in advanced microprocessors where the microprocessor begins executing a second instruction before the first has been completed. It is a series of stages, where some work is done at each stage. The work is not finished until it has passed through all stages. With pipelining, the computer architecture allows the next instructions to be fetched while the processor is performing arithmetic operations, holding them in a buffer close to the processor until each instruction operation can performed .The pipeline is divided into segments and each segment can execute its operation concurrently with the other segments. Once a segment completes an operation , it passes the result to the next segment in the pipeline and fetches the next operation from the preceding segment. Thus the speed and efficiency of the processor can be improved. The processor implemented here consists of fetch, store and execution operations which are performed using the process of pipelining. This processor is designed and implemented on Xilinx Spartan3E on FPGA.

II. DESIGN METHODOLOGY

The planning of the processor was classified into 4 stages. The testing steps are described in testing methodology below.

1. Create individual modules that are the components of pipelined processor using Verilog HDL code.
2. Test individual modules for functional verification.
3. Place all the individual modules together i.e., instantiating all the modules in a single processor.
4. Create a program to verify the functionality of the processor.

The pipelined implementation of the processor was subdivided into modules which were created before placing together. Once all the modules are designed, the final processor module was created to join all the individual modules together with logic. The external clock that we provide through test bench has to be controlling the modules at respective timings. All of the pipeline registers were clocked so that they would write and read only at the positive edge of the clock.

II.1 Processor Architecture

The processor presented in this paper consists of modules such as control unit, memory unit, program counter, arithmetic and logical unit and accumulator. All the blocks designed here are configured to perform operations on 8 bit data. The description of each block and their respective operations are as follows.

A. Control Unit

The control unit is designed in such a way that it allows each state to run at each clock cycle. Control unit performs its operations only when cu_en is enabled. The state of the processor changes continuously by means of a 3-bit counter which changes its state continuously on every posedge of the clock inside the control unit.

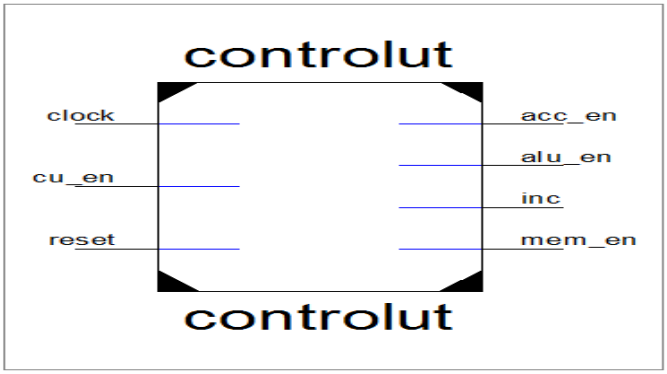

Figure 1

The first stage is the reset state. When the reset input of control unit is high, all the blocks inside the processor are disabled. Following the reset operation, fetching and execution operations are performed inside the processor. The outputs obtained from the control unit are given to the memory unit, accumulator and ALU designed to perform the processor operations.

B. Program Counter

The program counter implemented here is a 8 bit counter.. The output of the program counter is given to the memory unit to specify the address of the register into which required operation has to be performed. It is used to store the address of the next memory location to or from which read and write operations are to be performed respectively inside the processor . It is incremented before the data is fetched from the memory in the processor.

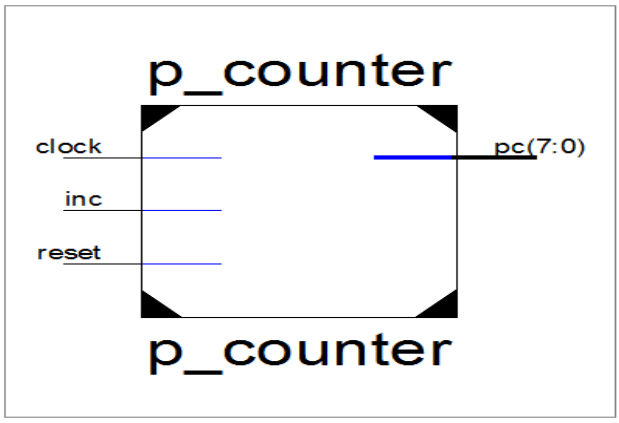

Figure 2

C. Memory Unit

Memory unit designed here consists of 8-bit registers which are used to store the 8-bit data. Based on the control signals from the control unit, memory write and read operations are performed in the processor .The addresses of the memory locations into which the read and write operations are to be performed are obtained from the program counter. The data from memory is written into the accumulator and into the ALU.

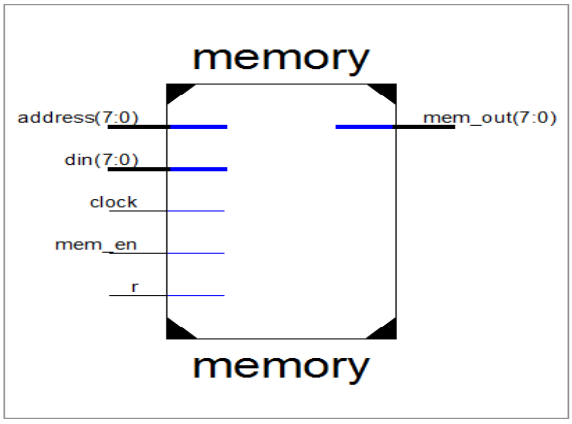

Figure 3

D. Accumulator

Accumulator designed here is a single 8-bit register. The input to the accumulator is taken from the memory and the output is given to the ALU. It stores the data from the memory and transfers it to the ALU to perform the required arithmetic or logical operations.

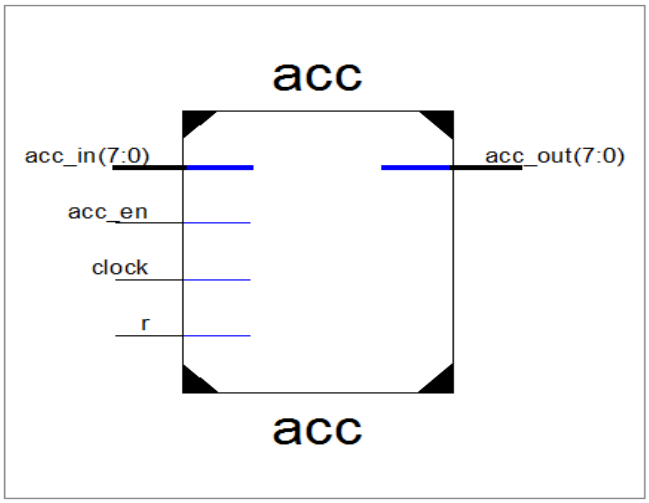

Figure 4

E. Arithmetic And Logical Unit

The ALU designed here is to perform operations on 8-bit data. ALU is used to perform the arithmetic and logical operations in the processor. It performs four arithmetic and four logical operations. The process of execution is performed inside ALU. It receives the data from both the memory and accumulator blocks and performs the operation specified by the particular opcode which is provided as input to the ALU externally so as to perform the required arithmetic or logical operation. The output from the ALU is stored into the accumulator or obtained as the output of the processor.

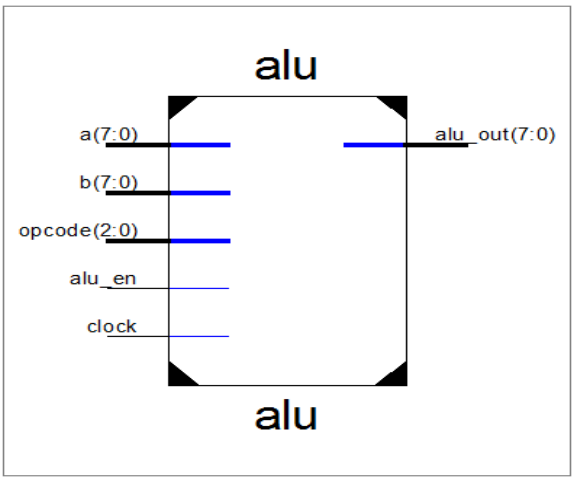

Figure 5

F. Testing Individual Modules

All the modules implemented in this processor are designed using Verilog HDL and are simulated using Xilinx tool. These modules are tested individually to attain their functionality. After ensuring that the individual blocks are functioning properly these units are placed together to build the processor using verilog code. Each unit implemented here individually shows the optimized circuit performance.

G. Processor Design

A verilog code is written to place all the individual blocks together inside a single processor. All these individual blocks are instantiated together in a single processor code. The designed circuit is implemented on Xilinx Spartan 3E on FPGA. The implemented circuit is synthesized and its architecture is obtained as shown below in figure 6.

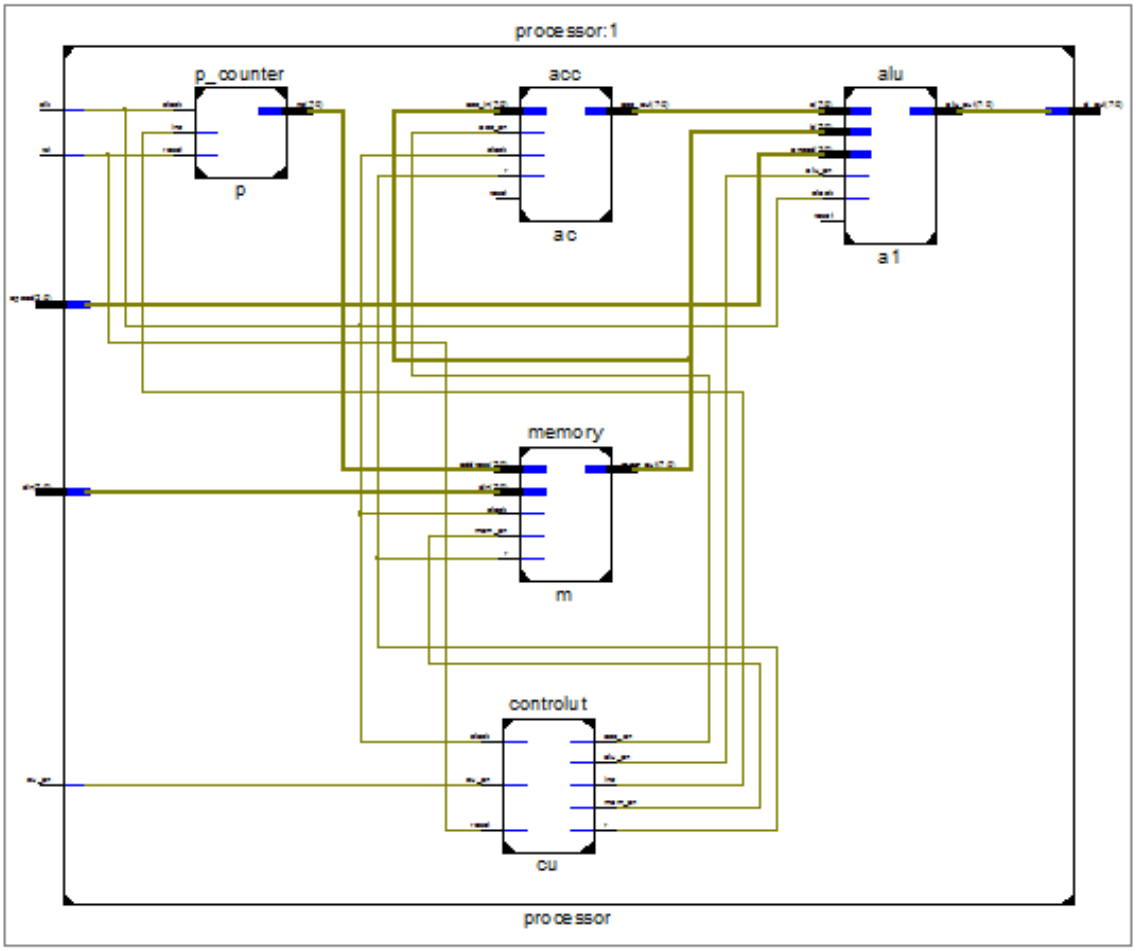

Figure 6

III. RESULTS

A. Simulation Results

Synthesis and simulation of the verilog code of the processor using Xilinx (version14.5) is presented here. The simulation results for the implemented processor are obtained as shown in figure 7.

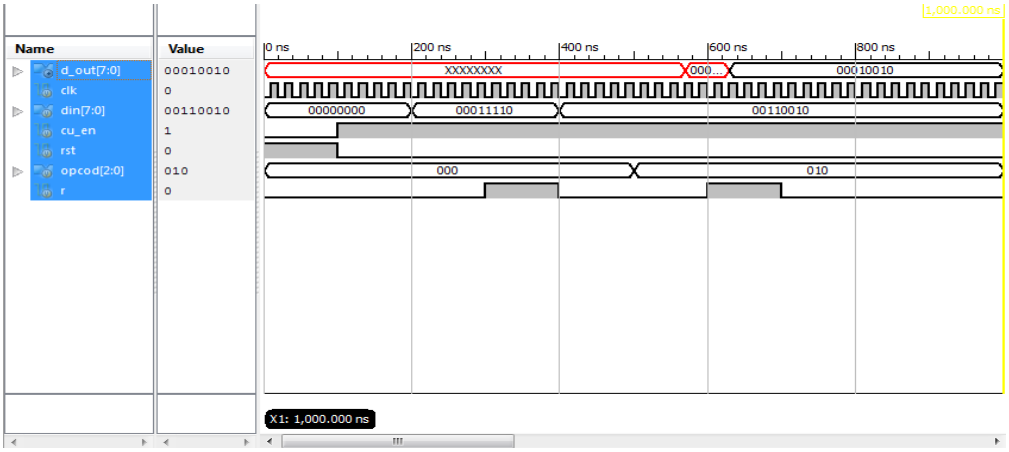

Figure 7

B. Synthesis Results

TABLE .1

Slices	58(1%)
Bells	141
Frequency	156.372Hz
Minimum period	6.395ns

This code is tested and verified using Xilinx 14.5 ISE software. The synthesized results justifies the proper connections among all the designed blocks and the simulation results verify the proper functionality of the processor implemented with all designed modules placed together.

IV. CONCLUSION

In this paper efforts have been made to optimize the circuit performance of the processor. The pipelined processor designed here is implemented on Xilinx 14.5 ISE on Spartan 3E on FPGA. Among all the processors implemented this processor has the maximum frequency of 156.372 MHz and minimum period of 6.395 ns. Most of the goals were achieved and simulation shows that processor is working perfectly. This processor can be used for mathematical computing in calculators.

References

1. Jayant Chowdhary(Corresponding Author) Vivek Garg Tushar Negi Shreya Jain "Realization Of An 8-bit Pipelined Microprocessor in Verilog HDL" Computer Engineering and Intelligent Systems ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online) Vol 3, No.7, 2012.
2. R Uma, Design and performance analysis of 8 bit RISC processor using Xilinx tools, March-April 2012," International Journal of Engineering Research and Applications (IJERA), pp 053-058.
3. E. Ayeh, K. Agbedanu, Y. Morita, O. Adamo, and P. Guturu"FPGA Implementation of an 8-bit Simple Processor"
4. Ramandeep Kaur1, Anuj2 "8 Bit RISC Processor Using Verilog HDL" Anuj et al Int. Journal of Engineering Research and Applications www.ijera.com ISSN : 2248- 9622, Vol. 4, Issue 3(Version 1), March 2014, pp.417-422.
5. Shankar Kumar Mishra, Dr. Nisha P Sarwade "Review of 5 stage Pipelined Architecture of 8 Bit Pico Processor" International Journal of Electronics, Communication & Soft Computing Science and Engineering ISSN: 2277-9477, Volume 3, Issue 4.
6. Aizup - A Pipelined Processor Design and Implementation on XILINX FPGA Chip, FCCM'96, IEEE Symposium on FPGAs for Custom Computing Machines, April 17 – 19, 1996, Napa, California, USA, pp.98–106.

