Development of a 16-Bit Pipelined Processor: Architecture, Implementation, and Performance Analysis

G. Appala Naidu
Assistant Professor, ECE Department, JNTUGV, Vizianagaram-535003

ABSTRACT:

In this paper, the 16-bit four stage pipeline RISC CPU based on MIPS is implemented. Here, Develop design, implementation, and performance analysis of a 16-bit pipelined processor. The development of pipelined architectures aims to enhance processing speed by dividing the instruction execution into multiple stages, thereby achieving higher throughput and performance. the architectural design of the 16-bit pipelined processor, encompassing instruction fetch, decode, execute, memory access, and write-back stages. The instruction set architecture (ISA) is carefully crafted to accommodate a diverse range of operations while ensuring compatibility with the 16-bit data path. Overall, this paper provides valuable insights into the development of a 16-bit 4 stage pipelined processor, offering a deeper understanding of its architecture, implementation challenges, and performance characteristics. The findings contribute to the ongoing research and development in processor design, aiding in the creation of more powerful and energy-efficient computing systems for diverse applications in the modern digital era. The Xilinx ISE 10.1 tool is used for simulation.

1. LITERATURE REVIEW:

A processor which can perform multiple tasks with high speed by using differ techniques like parallel processing are proposed by many authors and a brief overview of their work is mentioned in this chapter. The Author in [1] presented the design and verification of 16-bit processor with three different instruction formats available. All the modules in [1] are designed using VHDL. The advantage of using VHDL for designing a processor was proposed in [2] with minimum clock period. This model consists of numerous modules which are assembled together and communicate through 16 bit tristate data bus. A processor with the feature of directly accessing the memory is presented in [3] which consists of 16-bit word address. The model described in [3] comprises 16 general-purpose registers (R0 - R15), along with a program counter and a condition code register. This processor is capable of executing 16 different instructions and operates at a frequency of 3 MHz for arithmetic and logical operations, while for operands, load, and store instructions, the frequency is approximately 1.5 MHz.

A 16-bit microprocessor with simple architecture comprising of ALU, Shift register, comparator, program controller, Address register and a output register was proposed in [4]. The processor design in [5] is a prototype for demonstrating the hazards in the Pipelining and techniques used for solving them. Pipelining is a method of organizing computer processing that allows multiple instructions to be executed concurrently, overlapping their execution stages for improved efficiency. Various problems associated with pipelined data referred to as Pipeline Hazards. The techniques used for reducing them are data forwarding, pipelined Datapath with stalling are stated in [5]. The basic RISC processor with high performance and fixed length instructions is stated in [6] which is inspired from the MIPS architecture where MIPS stands for microprocessor without interlocked pipelined stages This paper introduces a 32-bit pipelined processor, based on the RISC architecture, featuring a three-stage pipeline. The processor incorporates an extensive array of registers and supports a wide range of fundamental instructions for general computing purposes. These instructions encompass arithmetic, logical, rotate, jump, and load/store operations, employing various instruction formats like R format, I format, and J format. The designed processor, as described in [6], operates at a significantly improved speed, achieving a frequency of 310.878 MHz with a clock period of 3.217 ns. It includes a 128-bit data memory and a 128-bit memory size, along with 128 registers. The use of separate data and instruction memories ensures the absence of structural hazards in the processor's operation.

Design of a processor including one of the Parnellism techniques like pipelining is described in [7]. The concept of pipelining, latency, applications, efficiency and throughput are described in the design which included four stage pipelining by which the performance of the processor increased considerably, also by increasing the stages the speed can be increased a pretty more. In [8] the various hazards which occur due to multiple access of the resources and the data were ignored which is a leading issue.

Here, we designed a 16-bit four stage Pipelined Processor which can perform 32 instructions which include the arithmetic, logical, shift, load and store operations. The register file consists of 16 registers of 16-bit size, Program counter of 1 byte size and the instructions are of 1 byte. For execution of first instruction, it takes four cycles and the rest of the instructions get executed in the next cycle itself thus improving the speed of the processor.

2. PROCESSORS:

The processor is vital role in any processing unit. They are different types of processors are like CISC, RISC, Special Processors (Coprocessor, Input/Output Processor, Transputer, DSP processors). To improve the processor performance one of the best techniques is pipelining.

A. PIPELINING

The speed at which programs are executed can be influenced by multiple factors. Enhancing performance can be achieved through two approaches: employing faster circuit technology for constructing the processor and main memory, as well as designing the hardware in a manner that enables simultaneous execution of multiple operations. By adopting this strategy, the overall number of operations executed per second is increased, without altering the individual elapsed time required for each operation.

The below fig.2.1 elaborate each instruction execute as follows: Fetch(F), Decode(D), Execute(E), Write(W).

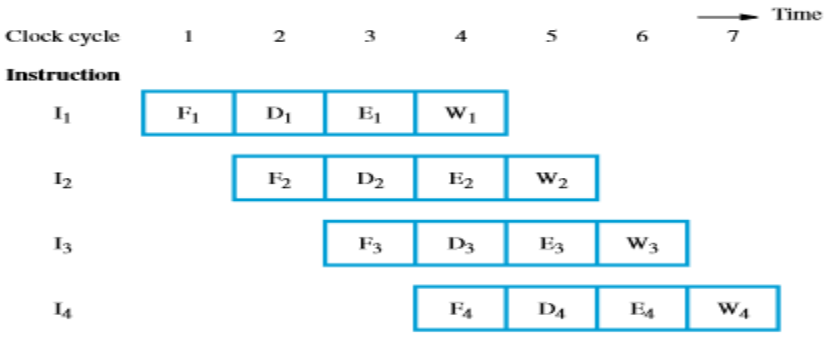

Figure. :2.1 Instruction execution divided into four steps

B. PERFORMANCE OF A PIPELINED PROCESSOR:

let's a pipeline with 'k' segments and a clock cycle time of 'Tp.' Within this pipelined processor, there are 'n' tasks to be completed. The initial instruction requires 'k' cycles to traverse the entire pipeline, while the subsequent 'n – 1' instructions only require '1' cycle each, resulting in a total of 'n – 1' cycles. Consequently, the time taken to execute 'n' instructions in this pipelined processor can be calculated as follows:

$$ET_{pipeline} = k + n - 1\ cycles = (k + n - 1)T_p$$

for a non-pipelined processor, the execution time of 'n' instructions is:

$$ET_{non-pipeline} = n * k * T_p$$

Therefore, the speedup (S) of the pipelined processor compared to the non-pipelined processor, when 'n' tasks are executed on the same processor, can be expressed as the ratio of the performance of the pipelined processor to that of the non-pipelined processor. Since a processor's performance is inversely related to its execution time, the relationship can be represented as follows:$S = ET_{non-pipeline}/ET_{pipeline}$

$$S = (n * k)/(n + k - 1)$$

Suppose, $n \gg k$, $S = k$,where 'k' is the number of stages in the pipeline.

$$\boldsymbol{Efficiency = given\ speed\ up/\max speed\ up = S/S_{max}}$$

We know that, $S_{max} = k$,So, $effeciency = S/k$

$$Throughput = \text{Number of instructions}/total\ time\ to\ complete\ instruction$$

$$\text{So, } Throughput = n/(k + n - 1) * T_p$$

Note: an ideal pipelined processor, the cycles per instruction (CPI) value is 1.

3. RESULTS:

XILINX ISE Software tool is used to The Pipelined processor and its RTL schematic is shown in fig. 3.1. This design consists of four stages each stage is confined for different functionality. The inputs for the processor are the clock signal and the interrupts (irq0 – irq3).

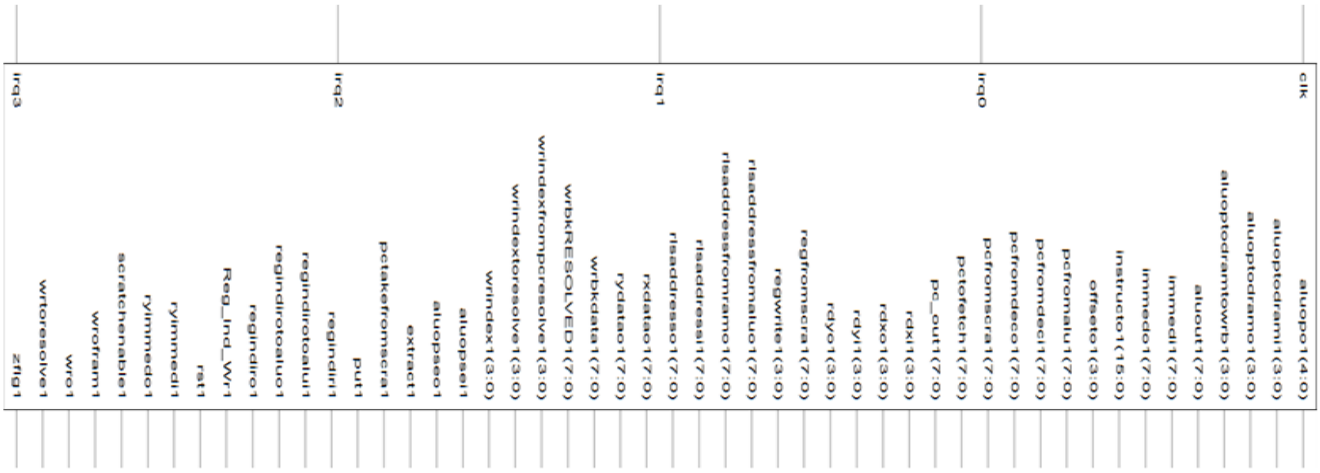

Figure. 3.1: RTL Schematic of Pipelined Processor

The figure 3.2 shows the internal architecture of the four-stage pipelined processor. The first stage is the Fetch stage which consists of the instruction memory which is of 32 x 16 size containing 32 instructions and a Program counter which is of 1 byte.

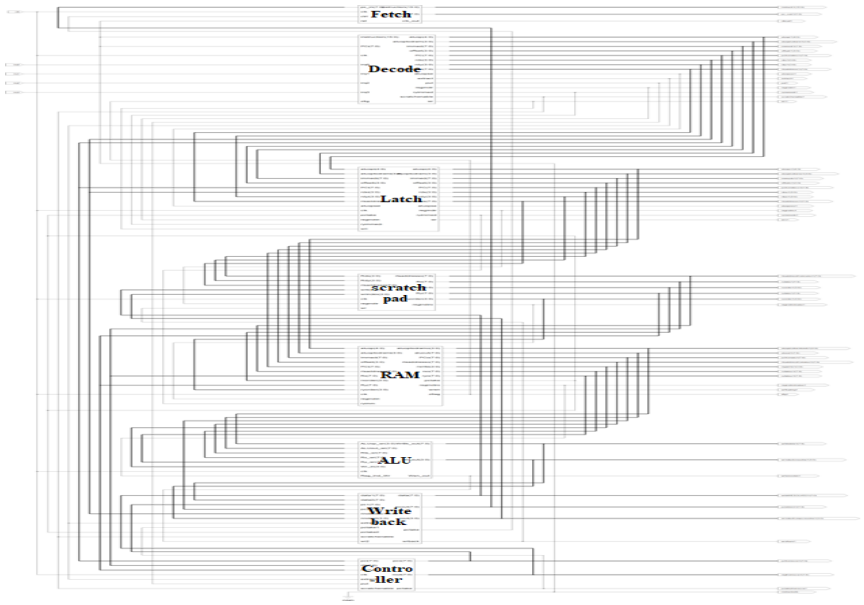

Figure. 3.2: Internal Architecture of the Processor

A. Program for adding the immediate data to the register

CLR R[0]

CLR R[4]

immed R[4], 1001

ADD R[0], R[4]

In the above program first we cleared the contents of the registers R[4] and R[0] and then the immediate data is stored in R[4] and it is added with the R[0]. The simulation results are shown below.

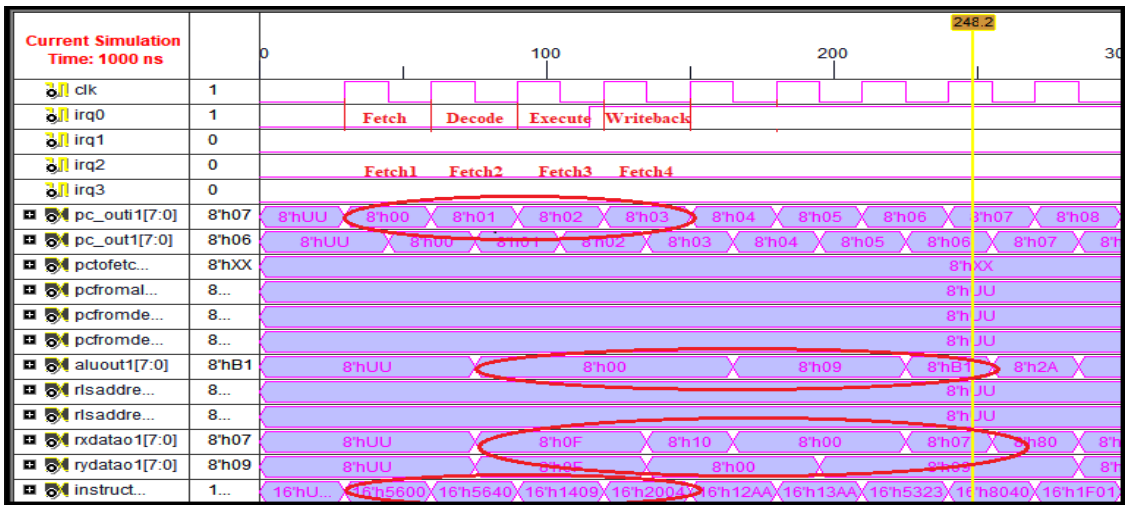

Figure. 3.3.a: showing the address of instruction, ALU output, register contents and the instructions

The above figure shows the clock cycles taken by each stage of the Pipeline processor to execute the above program. At the fourth clock cycle the result is stored and the clock cycles required for executing the program is '6'. The register contents on which the operations are performed and the instructions performed is shown in the figure 3.3.a.

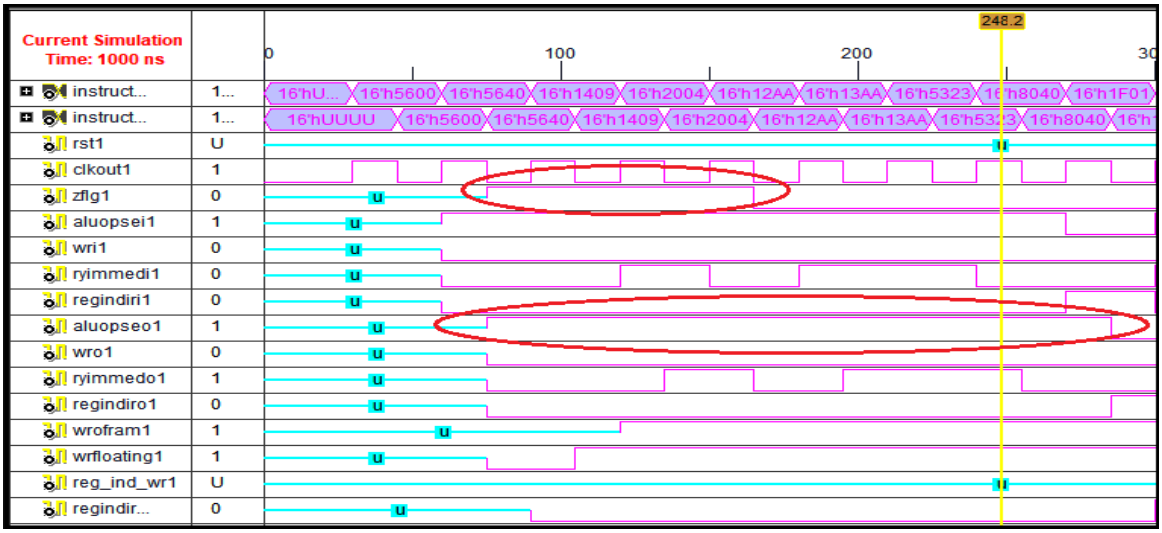

Figure. 3.3.b: showing the zero flag status, ALU output select signal

On executing the above program the zero flag get enabled on clearing the register contents and the ALU select signal gets enabled as shown in the figure 3.3.b.

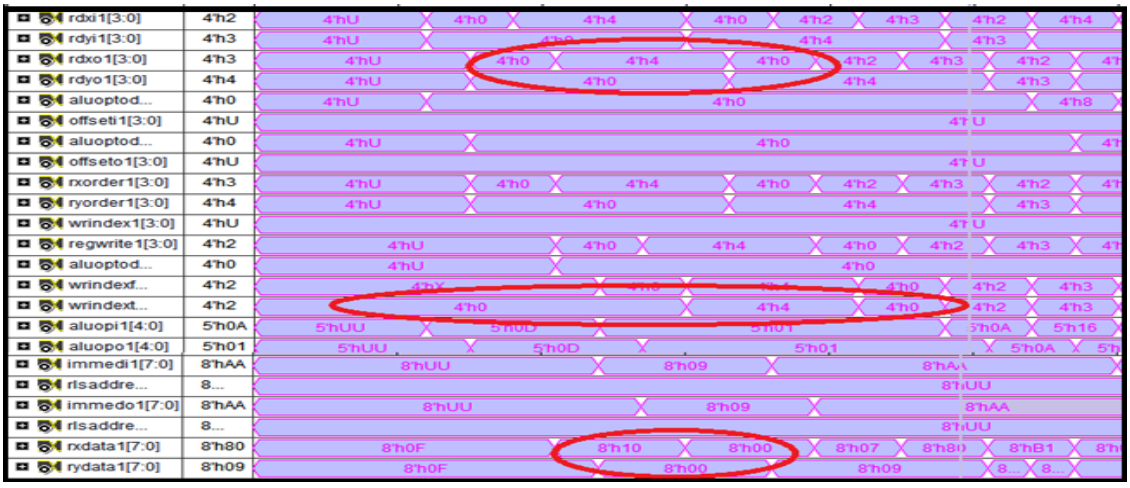

Figure. 3.3.c: showing the registers used, destination register address, data in registers

The registers which are used in the above program for performing the operations, the registers which are used for storing the results and their contents are shown in the figure 3.3.c

B. Program for performing OR operation on the immediate data

CLR R[1]

CLR R[15]

immed R[1], 1001

immed R[15],0001

OR R[1], R[15]

In the above program first we cleared the contents of the registers R[1] and R[15] and then the immediate data is stored in R[15] and R[1]. OR operation is performed on the two registers and the result is stored in R[1] .The simulation results are shown below.

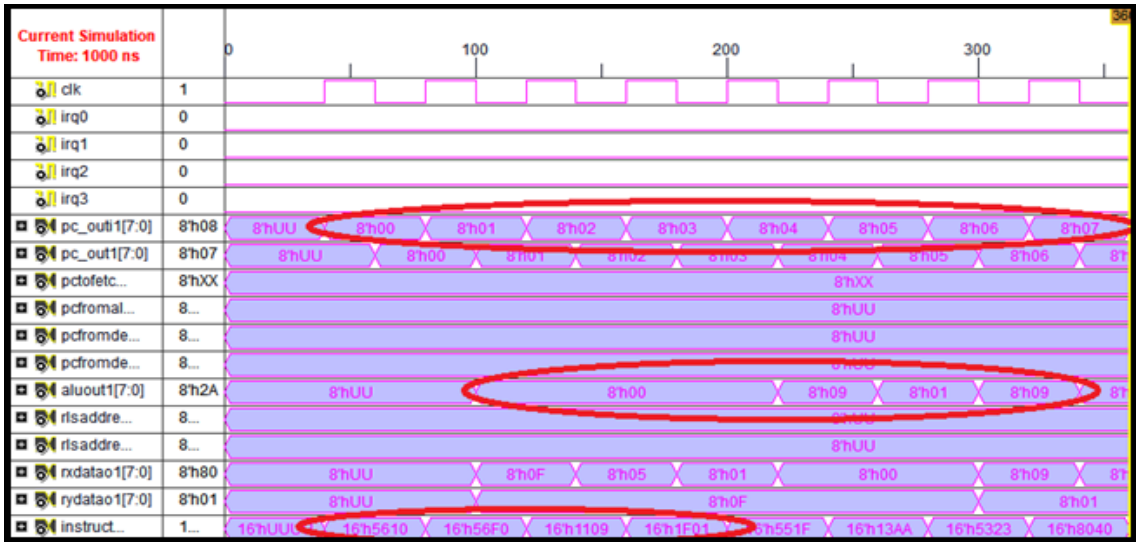

Figure. 3.4. a: showing the address of instruction, ALU output, register contents and the instructions

The above figure shows the clock cycles taken by each stage of the Pipeline processor to execute the above program. At the fourth clock cycle the result is stored and the clock cycles required for executing the program is ‘8’. The register contents on which the operations are performed and the instructions performed is shown in the figure 3.4.a.

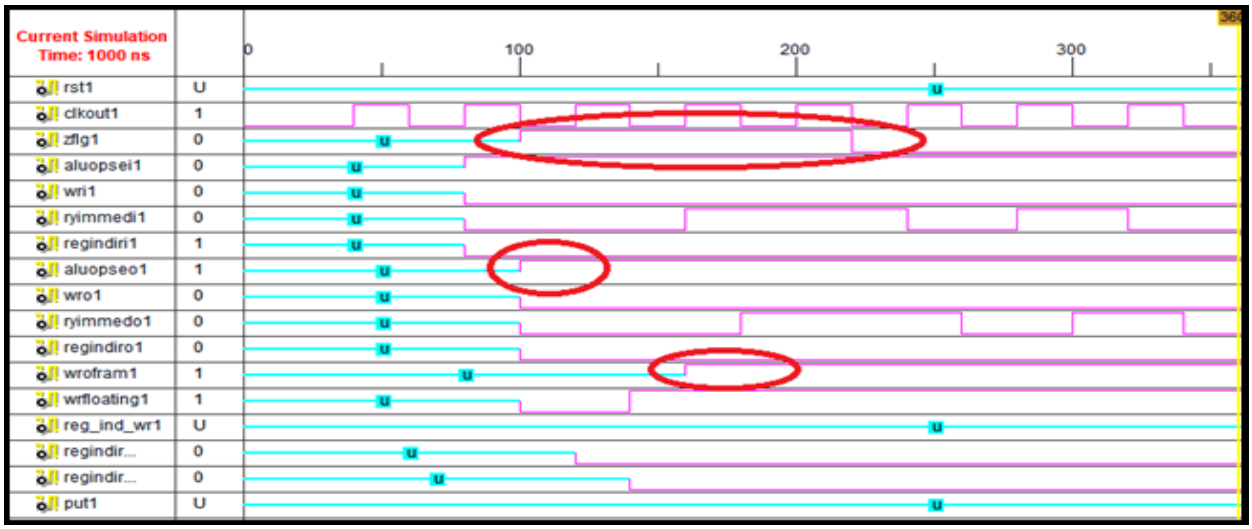

Figure. 3.4.b: showing the zero flag status, ALU output select signal and the write enable signal.

On executing the above program the zero flag get enabled on clearing the register contents, the ALU select signal and the write enable signal gets enabled as shown in the figure 3.4.b.

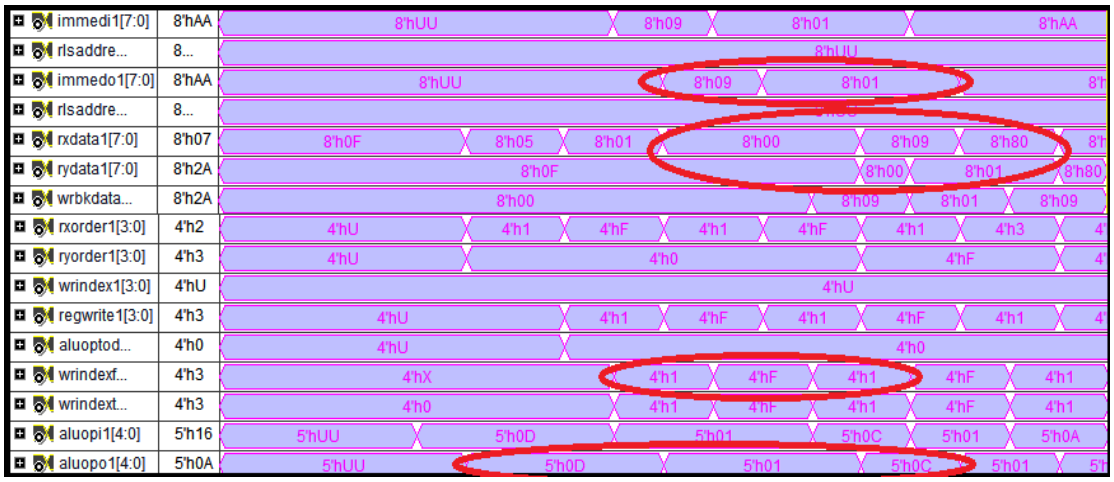

Figure. 3.4.c: showing the registers used, destination register address, data in registers and the opcode of instructions.

The registers which are used in the above program for performing the operations, the opcode of the instructions the registers which are used for storing the results and their contents are shown in the figure 3.4.cc

C. Program for obtaining the 2's complement of the given number

CLR R[4]

immed R[4], 1001

NOT R[4]

ADD R[4], #00000001

The simulation results are shown below.

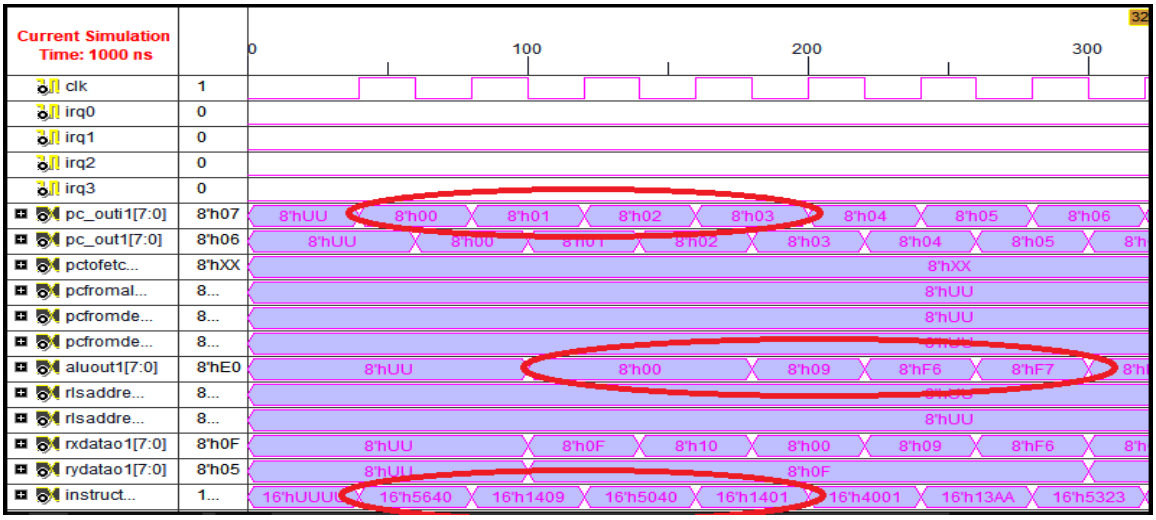

Figure. 3.5.a: showing the address of instruction, ALU output, register contents and the instructions

The above figure shows the clock cycles taken by each stage of the Pipeline processor to execute the above program. At the fourth clock cycle the result is stored and the clock cycles required for executing the program is '6'. The register contents on which the operations are performed and the instructions performed is shown in the figure 3.5.a.

Figure. 3.5.b: showing the zero flag status, ALU output select signal and the write enable signal.

On executing the above program the zero flag get enabled on clearing the register contents, the ALU select signal and the write enable signal gets enabled as shown in the figure 3.5.b.

Figure. 3.5.c: showing the registers used, destination register address, data in registers and the opcode of instructions.

The registers which are used in the above program for performing the operations, the opcode of the instructions the registers which are used for storing the results and their contents are shown in the figure 3.5.c

FEATURES OF THE PROCESSOR:

Number of stages in Pipelining : 4 stage processor

Processor size : 16 bit processor

Architecture followed : Harward Architecture

Frequency : 33MHz

Clock Period **:** 30 ns

Speed of the Processor **:** 3 MHz

Register size **:** 1 byte

Program counter size **:** 1 byte

Instruction size **:** 2 bytes.

Number of interrupts that can be handled **:** 4

Data Memory size **:** 32 bytes

Instruction Memory Size **:** 32 x 16 bits

Number of instructions	**:** 16
Instruction formats use	**:** R, I and J formats.
Registers	**:** 16 registers
Scratchpad size	: 16 x 8 bits
Data Bus size	**:** 8 bits
Address Bus size	: 8 bits
Operations that can be performed	: Arithmetic, Logical, Shift, Jump, Load and store operations.
Efficiency	: 0.725
Speedup	: 2.90
Throughput	**:** 24.24 * 10^6

4. CONCLUSION:

In this project we designed a 4 stage Pipelined Processor. Pipeline is implemented and every instruction is tested using the Test Bench. Proposed architecture supports 16 set of registers (R0 – R15). Design is simulated and synthesized using XILINX ISE 10.1 Design suit. The pipelining concept has lot of advantages in many of the systems. The pipelining has some hazards which are not considered in this design. The pipelining of instructions reduces cycles per instruction and also increases the overall throughput. The processor designed consists of 32 bytes of data memory, a scratchpad of 16 locations with a width of 8 bits for storing the initial values of program counter and the registers for servicing the interrupts. This processor operates at a clock rate of 625kHz with an instruction memory comprising 32 different instructions.

Introducing the concept of Pipelining has many advantages and the speed of the processor can be increased by increasing the number of stages which make the design complex as the number of stages increases the hazards also increases which require some control techniques to be included while designing a processor as stated by the author in [9]. The hazard can be controlled by using pipeline data path with data forwarding and stalling. There is a scope in this processor by increasing the number of functional units and the instructions with increased number of bits

REFERENCES:

[1] Nupur Gupta, Pragati Gupta, Himanshi Bajpai, Richa Singh and Shilpa Saxena."Analysis of 16 Bit Microprocessor Architecture on FPGA Using VHDL", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, No. 4, **(2014)**, pp. 8979-8986.

[2] Esma Alaer, Ali Tangel and Mehmet Yakut. "MIB-16 FPGA based design and implementation of a 16-bit microprocessor for educational use," 6th WSEAS International conference on circuits, Systems, Electronics, Control & Signal processing, Cairo, Egypt, **(2007)**, pp.326-330.

[3] Husainali S. Bhimani, Hitesh N. Patel Abhishek and A.Davda. "Design of 32-bit 3-Stage Pipelined Processor based on MIPS in Verilog HDL and Implementation on FPGA Virtex7", International Journal of Applied Information Systems, Vol. 10, No.9, **(2016)**, pp.26-37.

[4] Davidson, J "FPGA Implementation of a Reconfigurable Microprocessor" IEEE Custom Integrated Circuits Conference, **(1993)**, pp. 3.2.1- 3.2.4.

[5] Sueyoshi.T, Kuga.M, and Shibamura.H. "KITE Microprocessor and CAE for Computer Science", Systems and Computers in Japan, Vol. 33, No. 8, **(2002)**, pp.64-74.

[6] Mamun B, Shabiul.I and Sulaiman.S. "A Single Clock Cycle MIPS RISC Processor Design using VHDL". Penang, Malaysia, **(2002)**, pp.199- 203.

[7] Herman, H.S.Srihari and C.Matthew, M., "Pipeline Reconfigurable FPGAs", Journal of VLSI Signal Processing Systems", **(2000)**, pp. 129-146.

[8] Borgatti, M.Lertora, F.Foret, B and Cali L., "A Reconfigurable System Featuring Dynamically Extensible Embedded Microprocessor, FPGA and Customizable I/O", IEEE Custom Integrated Circuits Conference, **(2002)**, pp. 13-16.

[9]Jurado-Carmona, F.J., Tombs, J., Aguirre, M.A and Torralba, A., "Implementation of a fully pipelined ARM compatible microprocessor core," XVII Design on Circuits and Integrated Systems Conference (DCIS-02), **(2002)**, pp. 559-563.

[10] Ruchita Kawle and Shubhada Thakare. "Designing of 32-Bit Configurable Hack CPU On FPGA", 6th International Conference on Communication and Electronics Systems, **(2021)**, pp.233-236.