FPGA Implementation of Three Stage Pipelining

Deepak Sharma[1] Tejas H C[2] Amruta H Naik[3] Veena S Channappagoudar[4] Sujata N Bogur[5]
[1,2]Lecturer [3,4,5]Student
[1,2,3,4,5]Department of Electronics & Communication Engineering
[1,3,4,5]KLS's VDRIT, Haliyal, Karnataka, India. [2]PESITM, Shivamogga, Karnataka India.

Abstract— Pipeline is a technique used to increase the instruction throughput (the number of instructions that can be executed in a unit of time). Pipelining doesn't reduce the time it takes to complete an instruction; it increases the number of instructions that can be processed at once, thus reducing the delay between completed instructions. The fundamental idea is to split the processing of a computer instruction into a series of independent steps, with storage at the end of each step. This allows the computer's control circuitry to issue instructions at the processing rate of the slowest step, which is much faster than the time needed to perform all steps at once. The term pipeline refers to the fact that each step is carrying data at once (like water), and each step is connected to the next (like the links of a pipe.) The terms, "Fetch, Decode, and execute" that become common usage in pipelining. In fetch, the instruction is fetched from the memory. In decode, instruction is decoded and in execute, it is executed i.e. result is got from the ALU. Implementation of three stage pipelining has been done taking the ALU design.
Key words: pipelining, fetch, decode, execute, ALU

I. INTRODUCTION

A pipeline is a set of data processing elements connected in series, so that the output of one element is the input of the next one [1]. In most of the cases we create a pipeline by dividing a complex operation into simpler operations. We can also say that instead of taking a bulk thing and processing it at once, we break it into smaller pieces and process it one after another. process the data and finally write the result back to memory[2]. Without a pipeline a single instruction has to fully go through all these stages before the next instruction is fetched from the memory. But if we apply the concept of pipelining in this case, when an instruction is fetched from memory, the previous instruction must have already decoded.

A. Instruction Fetch (If):

Stage Is Responsible For Obtaining The Requested Instruction From Memory[4]. The Instruction And The Program Counter (Which Is Incremented To The Next Instruction) Are Stored In The If/Id Pipeline Register As Temporary Storage So That May Be Used In The Next Stage At The Start Of The Next Clock Cycle.

B. Instruction Decode:

The Instruction Decode (ID) stage is responsible for decoding the instruction and sending out the various control lines to the other parts of the processor. The instruction is sent to the control unit where it is decoded and the registers are fetched from the register file.

C. Execution:

The Execution (EX) stage is where any calculations are performed[3]. The main component in this stage is the ALU. The ALU is made up of arithmetic, logic and capabilities.

If you are aware of microprocessor architectures, then you may know about instruction pipelining. In microprocessors for executing an instruction there are many intermediate stages like getting instruction from memory, decode the instruction, get any other required data from memory,

II. FLOWCHART

The fig. 1 shows the three stages of pipeline mechanism. Fig. 2 shows the way how three stage non pipeline mechanism works and the Fig. 3 shows the way how three stage pipeline mechanism works. The figures clearly depicts the reduction of time which is shown on the x axis in all the figures.

III. RESULTS AND DISCUSSIONS

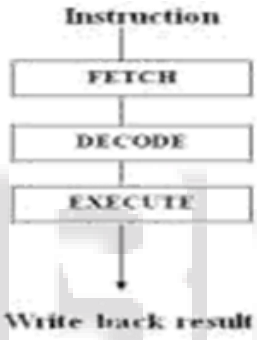

Fig. 1: stages of three stage pipelining

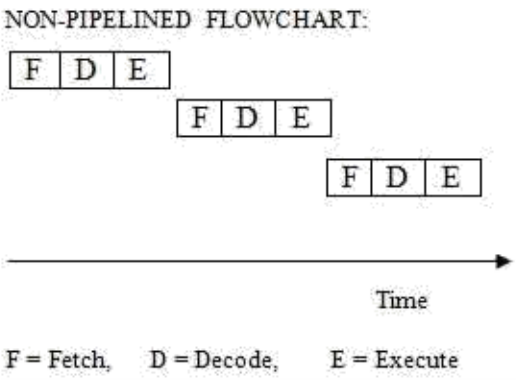

Fig. 2: Three stage non pipelining mechanism

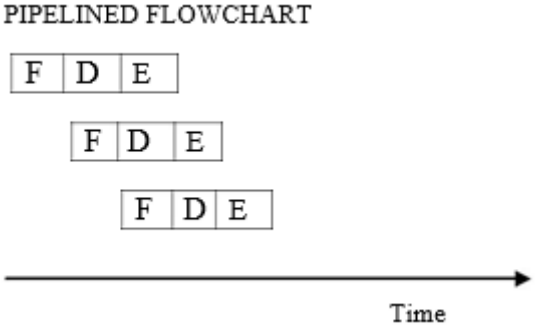

Fig. 3: Three stage pipelining mechanism

F = Fetch, D = Decode, E = Execute

IV. IMPLEMENTATION

Implementation of three stage pipelining has been done taking the ALU design. ALU is the fundamental building block of central processing unit of computer [4]. It is the fundamental unit of several combinational circuits which performs logical and arithmetic operations. Even the simplest microprocessors contain one ALU for purposes such as maintaining timers. Mathematician John von Neumann proposed the ALU concept in 1945, when he wrote a report on the foundations for a new computer called the EDVAC. Few arithmetic and logical instructions were chosen and all these were implemented by using both non pipelining mechanism and pipelining mechanism.

Fig. 4: Result of 4-bit pipelining

Pipelined output of eight instructions for 4-bit input is shown above in fig. no. 4. The fetching, decoding and execution of all the instructions happen as per the pipelining concept saving the time

Fig. 5: Result of 8-bit pipelining

Pipelined output of eight instructions for 8-bit input is shown above in fig. no. 5. The fetching, decoding and execution of all the instructions happen as per the pipelining concept saving the time.

Fig. 6: Result of 4 bit non pipelining

Non Pipelined output of eight instructions for 4-bit input is shown above in fig. no. 6. The fetching, decoding and execution of all the instructions happen as per the pipelining concept saving the time.

Fig. 7: Result of 8 bit non pipelining

Non Pipelined output of eight instructions for 4-bit input is shown above in fig. no. 7. The fetching, decoding and execution of all the instructions happen as per the concept of non-pipelining concept taking more time to execute.

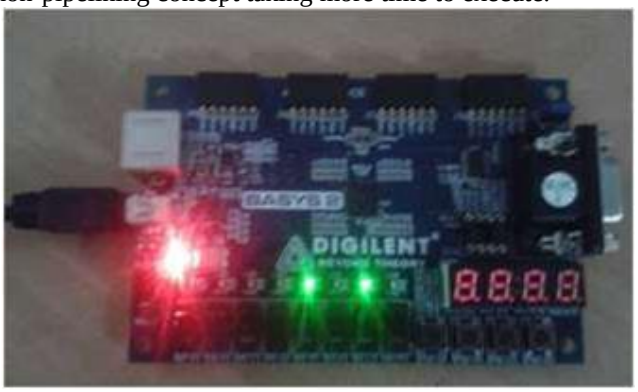

Fig. 8:

Final output of 4-bit Pipelining on Spartan-3E The final output of 4-bit Pipelining on Spartan-3E is shown above in fig. no. 8. The details of power consumption and time consumption are discussed in the conclusion part. The fig. 9 shows the delay comparison between Pipelining and Non-pipelining mechanism.

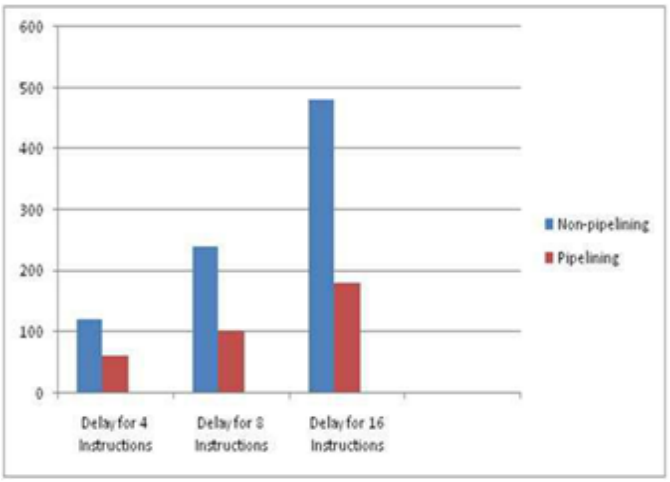

Fig. 9: Graph showing delay between Pipelining and Non-pipelining

V. CONCLUSION

The pipelined mechanism as well as non-pipelined mechanism was simulated in xilinx ise 14.7. The delay for 4 instructions is 600ps in pipelined mechanism whereas in non-pipelined mechanism it is 1200ps, the delay for 8 instructions is 1000ps in pipelined mechanism whereas in non-pipelined mechanism it is 2400ps, the delay for 16 instructions is 1800ps in pipelined mechanism whereas in non-pipelined mechanism it is 4800ps. The pipeline mechanism is implemented on FPGA SPARTAN-3E. POWER CONSUMPTION: The power consumption on FPGA is 0.034watts.

REFERENCES

[1] Y. Li and W. Chu, "Aizup - A Pipelined Processor Design and Its Implementation on XILINXFPGA Chip," Lecture Notes, The University of Aizu, 1995.
[2] Y. Li and W. Chu, "Design and Performance of pipelined architecture", IEEE conference, 2009.
[3] Analysis of A Multiple-threaded Multiple Pipelined Architecture," Proc. of the Second International Conference on High Performance Computing, New Delhi, India, December 1995.
[4] R. Iwanczuk, "Using the XC4000 RAM Capability,"The Programmable Logic Data Book. Xilinx,1994.
[5] XACT Development System - XACT User Guide,Xilinx, April, 1994.
[6] http://www.cs.sjsu.edu/~lee/cs147/fall2003/23147L25Pipelining.ppt

