International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

Qualitative Analysis of 32 Bit MIPS Pipelined Processor

1st Shobhit Shrivastav
Department of Electrical Engineering
Delhi Technological University

2nd Shubham Kumar
Department of Electrical Engineering
Delhi Technological University

3rd Sarthak Gupta
Department of Electrical Engineering
Delhi Technological University

4th Bharat Bhushan
Department of Electrical Engineering
Delhi Technological University

***Abstract*—A new generation of applications requires low power consumption without sacrificing performance. The main objective of this paper is to differentiate our proposed low power design 32 bit MIPS pipelined processor based on the simulation, timing and power it consumes with 32 bit Non-Pipelined processor. The comparative study elevates the proposed model in terms of Power, timing and frequency. From the architectural point of view, the prosposed pipelined processor has 5-stage pipeline, 32 bit register banks, 32-bit ALU. The power comparison result shows that the Pipelined Processor works on 3 times less power than Non Pipelined one.**

***Index Terms*—Verilog, Processor, Xillinx, Pipelined, Power, Timing,etc.**

I. INTRODUCTION

Various types of processors exist in today's market. Most of them are designed using Hardware Descriptive language like VHDL, Verilog, etc . Processors are of two types either RISC or CISC based. RISC architecture is one of the most efficient computer architecture, which is used for the high speed and low power application[1]. RISC processors can be pipelined compared to CISC due to the complex instruction set of CISC Processors. Pipelining enables us to increase the throughput and performance of the processor.

In order to enhance the performance of the RISC processor pipelining is used. In a CPU, there are so many processes that are executed and CPU has to process billions of instructions. When one instruction is executed, instead of waiting for the instruction to complete, pipelining let us run the new pro- cess simultaneously without affecting the ongoing instruction execution[2]. To achieve this, each part of the instruction is divided into 5 stages separated by latches. After each clock cycle the result of the stage is stored into the next stage, and the stage can take data from the previous stage to work upon. So no stage have to wait for the previous stage for data. They can perform independently. Hence all the stages can perform parallel operations .

The HDL design is done in Xilinx ISE software tool and they optimize the circuit performance to enhance their processors performance[4]. Author Supraj Gaonkar, Anitha M proposed a design of 16 bit RISC processor. Their architecture was based on Harvard structure using minimal functional units. Their processor was non pipelined single cycle 16 bit processor which can perform 11 instructions. They used RISC architecture because RISC architecture boosts computer speed by using simplified machine instructions for frequently used functions. They used finite state machine to implement their architecture[3]. Their processor had 4 states which are idle, fetch, decode and execute. Our processor is 32 bit MIPS based RISC Processor. We have included five stages in our design which includes IF (Instruction Fetch), ID (Instruction Decode), EX (Execution), Memory Access(MEM) and Write Back(WB). In the pipelined processor consecutive stages are separated by latches to store intermediate output of different stages [1].

Author Mohit Rane, Arjav Naik and Kalpan Mehta proposed a custom architecture of 8 bit processor. They implemented 9 instructions in their processor. They implemented non pipelined single cycle 8 bit processor. The difference in their processor architecture and conventional MIPS architecture is in the execution of jump type of instruction[7].In 2016 Sarika
U. Kadam, S.D. Mali, designed "Design of RISC Processor us- ing VHDL". The proposed 16-bit RISC processor is designed using a parallel programming language called VHDL. It is simulated and synthesized using Xilinx ISE 13.1i. Pipelining is used to make processor faster. In Pipelining instruction cycle is divided into parts so that more than one instruction can be operated in parallel[8]. Swati Joshi, Sandhya Shinde, Amruta Nikam designed '32 bit pipeline Risc Processor in VHDL using Booth Algorithm'. The aim of paper was to design instruction fetch unit and ALU which are part of RISC processor architecture[9]. Vishwas V.Balpande ,Vijendra P.Meshram,Ishan A. Patilm,Sukeshini N.Tamgadem,Prashant Wanjari proposed "Design and Implementation of RISC pro- cessor on FPGA". In proposed paper 16-bit RISC processor is designed using VHDL programming. Four stage (viz. instruc- tion fetch stage, instruction decode stage, execution stage and memory/IO - write back stage) pipelining is used to improve the overall CPI [10].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

Our main objective is to reduce the power consumption in our processor by applying pipelining and to improve the throughput and efficiency of our processor. We have compared the power consumption and timing analysis of pipelined and non pipelined processor. This paper helped us to analyze the performance gain using pipelined processor.

Through the different papers it is observed that the work is concentrated on techniques rather than changing the whole architecture of processors techniques like power gating, clock gating, pipelining are used for reduction in power and enhance- ment in speed. In our research paper we have used 5-stage pipelining with multi-phase clocks to avoid race condition and compared the results of our processor with a 32 bits RISC non pipelined multi-cycle processor to get to know the difference.

II. METHODOLOGY

A. STAGES

Fig. 1 shows the stages used in our processor.

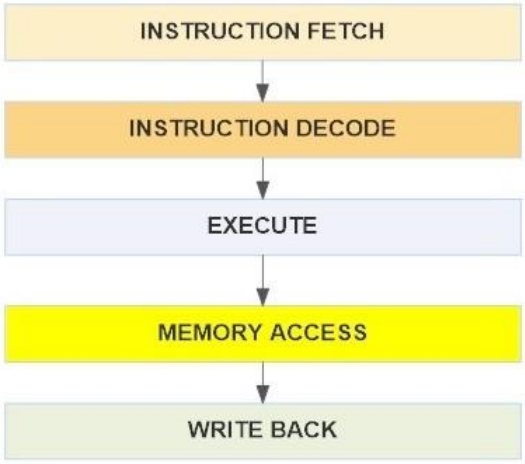

Fig. 1. Stages of Processor

- Instruction Fetch Stage : In this stage the instruction is retrieved from the instruction memory and retrieved instruction is stored in instruction register. The address of the instruction which has to be executed is given by program counter. In this stage Program Counter is also incremented by 1 to point to the address of next instruction.
- Instruction Decode Stage : In this stage the instruction retrieved in first stage is decoded. The opcode and operands are separated in this stage. In this stage register file is also accessed and values of different operands is retrieved from the register file.
- Execute Stage : This is the stage where actual compu- tation takes place. In this stage different operands are inputted to the ALU which different types of operation like addition, subtraction, logical operation, etc based on the value of the opcode.
- Memory Access Stage : In this stage data memory is accessed when required. Data is either written or stored in Data memory in this stage. It is used in load and store instructions.
- Write Back Stage : This is the last stage of the processor. In this stage the computed data is written back to the register file.

Datapath is the path that the input data follows in a processor to appear as an output. Fig. 2 shows the Pipelined Processor Data Path.

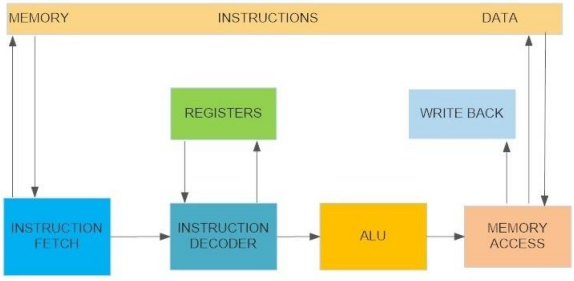

Fig. 2. Pipelined Data Path

B. HAZARDS

Hazards may occur in pipelining which may cause the processor to give incorrect output[5]. Some instructions when placed immediately after some other instruction can cause hazard. But they can always be resolved by waiting. They reduce the performance of the processor. There are 3 kinds of Hazards:

- STRUCTURAL HAZARD : This hazard occurs when two or more instruction wants to access the same component at the same time. For example when the same memory is used instruction fetch and data fetch this hazard may occur. This hazard can be prevented by using separate hardware resources or by replicating the resources but it may result in cost increase.

- DATA HAZARD : This hazard occurs when new instruc- tion uses the register data whose value has to be modified by previous instruction but it is retrieved by the new instruction before it gets modified. So this hazard may cause the new instruction to use wrong data. This hazard can be prevented by using.
  1) Dummy instruction
  2) Forwarding

- CONTROL HAZARDS : Control hazards may occur while using branching instruction. While using branch instruction branching takes place only when condition is met. And processor gets to know whether condition is met or not in Execute stage. So it may happen that

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

while deciding whether the condition is met or not wrong instruction may enter pipeline. This hazard can be prevented by not loading new instruction till the result comes.

Pipeline follows the steps as given in the flowchart in case of branch instruction as shown in Fig 3.

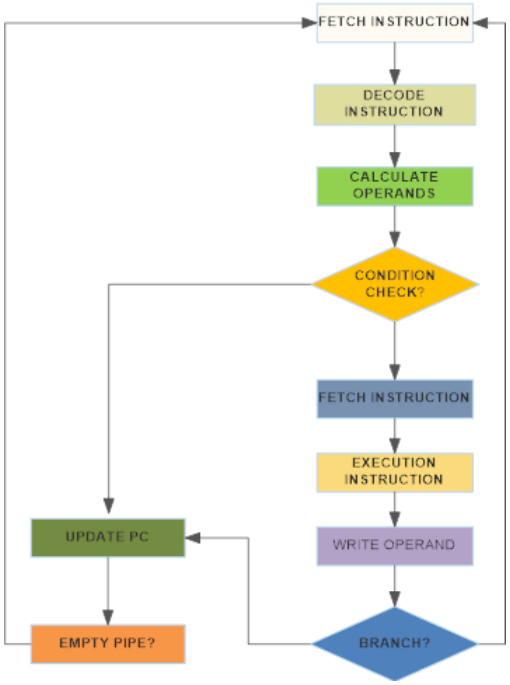

Fig. 3. Flow Chart

C. *Architecture*

Fig. 4 and 5 shows the architecture of Non Pipelined and Pipelined Architecture respectively.

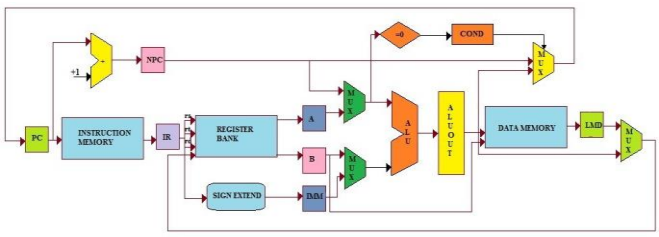

Fig. 4. Non Pipelined Architecture

III. PIPELINING PROCESS

Pipelining technique is generally used to save time and power so that hardware is properly utilized to its maximum extent and by that, we achieve maximum processing speed[2]. Generally power reduction through pipelining is because of reduction in the critical path delay by introducing a latch between the combination blocks and for more power reduction asynchronous blocks connected to the synchronous blocks

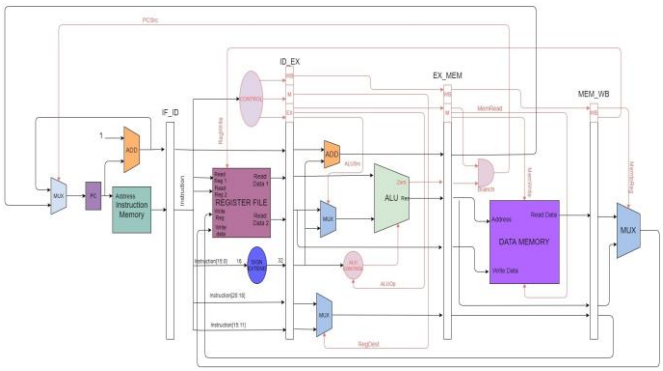

Fig. 5. Pipelined Architecture

with handshake signals to get high speed and good reduction in power. For more multi-cycled clocks are used from outside just to get the next stage ready as soon as possible and it's just save some time and control minute hazards. We have used 2 clock cycles in our processor which generally makes the next stage ready. We can use more but it increases its complexity and then results into breaking the parallelism.

IV. RESULTS AND ANALYSIS

The simulation is given below of 32 bit pipelined processor and 32 bit Non Pipelined Processor. We have analyzed the difference in the timing of execution of the same instruction in both the processors i.e the same instruction is executing at 95 ns in the Non Pipelined Processor (Fig. 6) and 65 ns in the Pipelined Processor(Fig. 7).

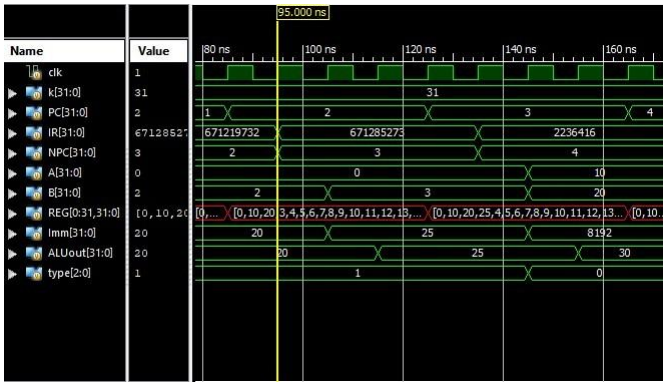

Fig. 6. SIMULATION RESULT OF NON PIPELINED PROCESSOR

Table I shows the power consumption comparison between both the Processors. We have analyzed the see power consumption is reduced by approximately 3 times in pipelined processor as compared to 32 bit non pipelined processor. The information regarding its timing summary of 32 bit pipelined processor is concerned is given in the Table II.

Maximum combinational path delay is 27.310 ns in Non Pipelined Processor and 11.180 ns in Pipelined Processor ,this is very important aspect as it is basically a factor that has huge impact on speed.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

FIG. 7. SIMULATION RESULT OF PIPELINED PROCESSOR

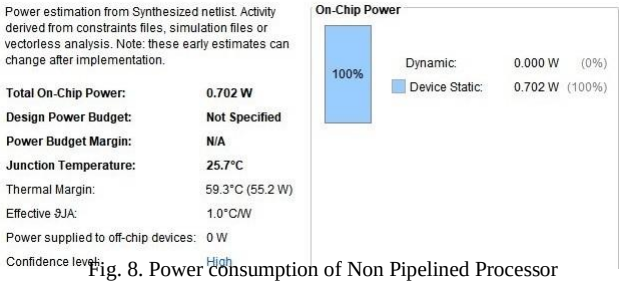

Fig. 8. Power consumption of Non Pipelined Processor

TABLE I
POWER CONSUMPTION COMPARISON

TYPE	POWER(W)
PIPELINED PROCESSOR	0.241
NON-PIPELINED PROCESSOR	0.702

TABLE II
TIMING SUMMARY COMPARISON

Timing Type	Non Pipelined	Pipelined
Minimum Period	18.27 nsec	3.502 nsec
Slack	1.73 nsec	0.796 nsec
Maximum Combinational Path Delay	27.310 nsec	11.180 nsec

V. CONCLUSION

In this research, Spartan 3E family device is used to implement both the processors. From the proposed model we concurred that Pipelined processor utilizes a total power of 0.241W whereas the non pipelined utilizes a total power of 0.702W. We found out that the pipelined processor takes less minimum period(max frequency), less slack and less combination path delay and the comparision in done in the Table 2.

VI. FUTURE SCOPE

As we used spartan 3E family FPGA device XC3S500E to get our information regarding all the parameters that we

Fig. 9. Power consumption of Pipelined Processor

required in our 5 stage pipelining with package FG320 and yields low power and high speed processor which utilizes
0.241 W which is less than its counterparts. But we have like virtex 7 processor which uses super scalar pipelining which enhances the speed and results in reduction in power and some other power gating and speed increases techniques are used so we can use other techniques as well which further reduces the power consumption an enhances the speed. Spartan 7 processors has more features and we can use it with more pipelining stages an with more use.

REFERENCES

[1] Muskan Saxena, Ojaswini Nimbalkar, Vidhi Jaiswal, Vishakha Pandey,
P. Sanjeevi. The survey of concepts of architecture in RISC and CISC computers, Volume-4, Issue-6, pp. 146-151, 2018.
[2] P. M. Kogge, The Architecture of Pipelined Computers, McGraw-Hill, 1981.
[3] M Design of 16-bit RISC Procesor International Journal of Scientific Research in Physics and Applied Sciences, 1(1), Feb 2017, pp 25-30.
[4] Ayeh, Agbedanu, Morita, Adamo, Guturu. FPGA Implementation of an 8-bit Simple Processor. 2008 IEEE Region 5 Conference. April, 2008, pp. 1-5.
[5] Lu J, Zhou X, Wang J. A novel dynamic scheduling algorithm of data hazard for embedded processor. International Conference on Asic. IEEE, 28-31 (2007).
[6] Pandey A.“ Study of data hazard and control hazard resolution tech- niques in a simulated five stage pipelined RISC processor”, IEEE Xplore, Inventive Computation Technologies (ICICT), International Conference on 26-27 Aug. 2016 at Coimbatore, India.
[7] Rane, Naik, Mehta. 8 Bit Custom MIPS Microprocessor. International Journal of Computer Engineering Technology, 8(5), Sep-Oct 2017, pp. 23-30.
[8] Sarika U. Kadam, S. D. Mali, “Design of Risc Processor using VHDL”, 2016 International Journal of Research Granthaalaya, Vol.4 (Iss.6): June, 2016, DOI:10.5281/zenodo.56647.
[9] Swati Joshi ,Sandhya Shinde,Amruta Nikam, “32-bit pipeline Risc Processor in VHDL using Booth Algorithm ,”International Research Journal of Engineering and Technology(IRJET), e-ISSN: 2395 -0056,
Volume: 03 Issue: 04 — April-2016 ,pp.2484-2487
[10] Vishwas V.Balpande ,Vijendra P.Meshram,Ishan A. Patilm,Sukeshini N.Tamgadem,Prashant Wanjari, “Design and Implementation of RISC processor on FPGA,”Indian Journal of Advanced Research in computer science and software Engineering, ISSN 2277 128xVol 9(8),Volume 5,Issue 3,March 2015,pp.1161- 1165.
[11] Mano, Ciletti. Digital Design with an Introduction to the Verilog HDL, 5th Edition, Pearson Education, 2013.
[12] Palnitkar. Verilog HDL A Guide to Digital Design and Synthesis, 2nd Edition, Sun Microsystems, 2003.
[13] Mano. Digital Logic and Computer Design, Pearson Education, 1979.