International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Functional Verification and the Physical Design of MIPS Architecture

B. Kiran Kumar, Anilkumar B

Abstract*: The Microprocessor with no Interlocked Pipeline Stages (MIPS) is Reduced Instruction Set Computation (RISC) based architecture. The RISC is composed of a smallest set of instructions for increasing the speed processor. In this work, functional verification and physical design of the 32-bit 5-stage pipelined MIPS processor is proposed to be done. Here, the physical design is aimed to improve the area, power and speed performance of the design. Cadence SOC encounter is proposed to use for the physical design of the MIPS architecture. Also, this work explores the complete the ASIC flow from RTL to GDSII for MIPS processor using TSMC (Taiwan semiconductor Manufacture Company) 90nm technology. Finally, performance of the proposed implementation will be compared with the existing implementations.

Index Terms*: Inter locked pipe line stages, ASIC flow, MIPS, RISC.

I. INTRODUCTION

The Present day's power is most important feature for the many applications. But in any IC, a tradeoff between the three factors there is area, power and delay. For achieving low power circuits we have to compromise with these factors.

MIPS (Microprocessor without Interlocked Pipelined Stages) is a RISC Architecture .It is established by MIPS Computer Systems. The architecture has evolved a new technology and developed a robust ecosystem and industry support. It was commercialized in 1984 by MIPS Computer Systems and acquired by Imagination Technologies in 2013.The MIPS with a 5-stage pipeline, was the first major commercial success. The MIPS architecture eventually expanded to serve low-power, low-cost markets including consumer electronics, networking and microcontrollers. The M4K family is based on the classic 5-stage 32-bit pipeline [1]. The MIPS family added the 16-bit micro MIPS instruction set to reduce code size for cost-sensitive embedded applications. MIPS Processor having 5-stage Pipelining process to be done by the enhancing the processor speed and performance is highly to be done by them.

Revised Manuscript Received on 30 July 2019.
* Correspondence Author
B. Kiran Kumar*, ECE , GMR Institute of Technology, GMR Nagar, Rajam, AP, India.
B. Anil Kumar, ECE , GMR Institute of Technology, GMR Nagar Rajam, AP, India.

II. RISC ARCHITECTURE AND PIPELINING CONCEPT

Microprocessors and Microcontrollers are programmed by two architectures namely, Complex Instruction Set Computing (CISC) and Reduced Instruction Set Computing (RISC). The instruction set used in CISC depends on different number of operands in different locations. To make individual instructions execute faster, RISC processors have less number of instructions, fixed instruction length, more general-purpose registers, load-store architecture and simplifies addressing modes to reach a net gain in performance and a total simple design with a reduced amount of silicon area consumption compared to CISC [2].

MIPS processor is depends on the RISC design which uses load/store architecture. The simplified instruction set of RISC processors have the performance value two to four times than that the RISC processor are taken into scenario. Speed can also be enhanced using suitable clocking methodology. The design is evaluated based on MIPS instruction set. The non-interlocked pipelining technique is performed for evaluation [4].

A. Pipelining Technique

We use pipeline design process to reduce the execution time of instruction effectively, which involves instruction fetch (IF), instruction decode (ID), execution (EXE), data memory (MEM), write back (WB) building block of 32-bit RISC processor [5]. Execution flow of MIPS shown in Fig. 1.

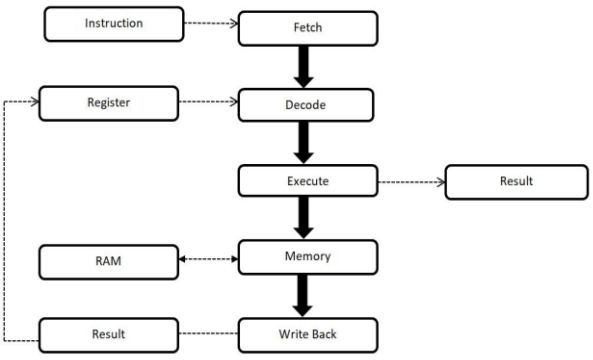

Fig.1 Execution flow of MIPS

In the beginning of the pipelining process the IF stage fetches the instruction. Then this moves to the next stage i.e., ID stage where the next instruction enters IF stage.

While the first instruction transfers to EXE stage, second instruction transfers to ID stage and the third instruction enters the IF stage and this process continues. In this way, pipelining technique flow proceeds on with the instructions that are to be executed [7]. Five stage Pipelining technique shown in Fig. 2.

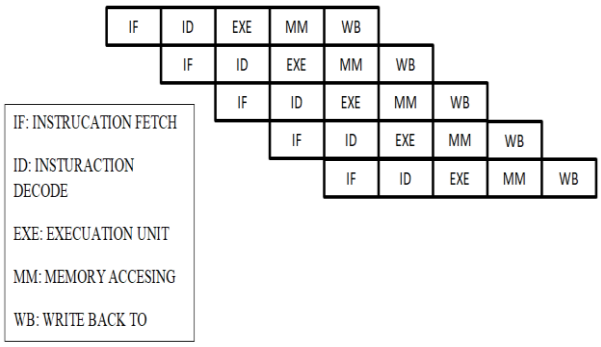

Fig.2 Five stage Pipelining technique

B. Five stage Pipelining Process

Instruction Fetch: In this phase the next instruction is fetched. From the beginning, it sends the information of the PC register to the instruction memory, which consist the following instruction address. The instruction memory responds by transferring the right instruction, this stage holds the instructions that have various modules. The IF phase fetches the following instruction, the next phase of PC is calculated and fed back to the IF phase. PC is updated by a multiplexer on a trigger occurrence. The selection line or control line for the multiplexer comes from the ID phase based on the type of instruction.

Instruction Decode: This phase have two main tasks. To calculate and fetch the next PC operands for the existing instruction, single phase is used. The possibilities for the next PC are the common instance for all the normal instructions is the value of the next phase PC+4. This is fetched to the next instruction. In a destination register, these two jump instructions needs to send the contents of the PC+4 to save the address of the next instruction. The authors of the accepted manuscripts will be given a copyright form and the form should accompany your final submission.

Execution: The calculations to be implemented in this phase execute the instruction. All calculations are performed by the Arithmetic and Logic Unit (ALU).From the register file, the two ALU operands are executed from ID phase. This stage includes the left shift by 2 and an adder for branch if equal (beq) operation. The selection line from the ID phase controls the multiplexer. The result is passed from the execution stage to the MEM stage.

Memory Access: The purpose of the memory access is to load operands from memory and store operands into memory. Except load and store, the MEM phase passes on the value to the WB stage from the ALU for all the instructions. To load and store, the address executed by the ALU passes the effective address to the memory. To load, the memory will return by passing the requested data. A control signal is produced in this stage that is sent to next WB phase. To store, the data is sent to the next phase including the actual address. The memory will respond by informing itself.

Write Back: It is a simple phase. The result is write back into the register file in this stage. The data which is loaded from the memory and write it on one of the register file is out of MEM phase .As there will be no output from previous stage i.e., MEM phase, this phase does anything to store instructions

III. MIPS ARCHITECTURE

The pipelined based architecture design implementation is basically known as MIPS. The five stages of pipeline are carried by MIPS architecture. The pipelined stages are Instruction Fetch (IF), Instruction Decode (ID), Execution (EX), Memory Access (MEM), and Write Block (WB). For the first program, fetches the operands. Later on, both IF and ID stays idle. And then the EX phase executes the instruction. The clock period is far extensive that is sufficient to propagate an instruction through all the five stages [3]. Fig.3 refers the 32-bit MIPS architecture.

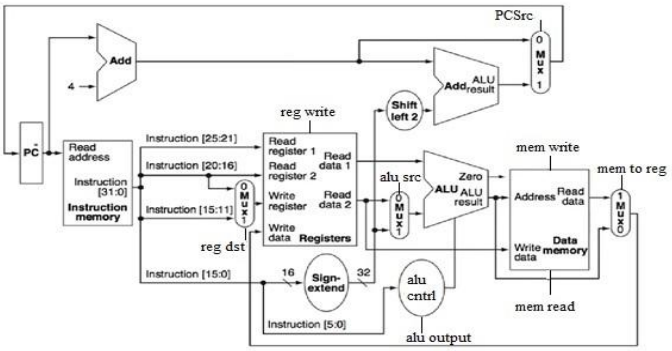

Fig.3 32-bit MIPS architecture

IV. RESULTS

The functionality of the 32-bit 5 stage MIPS processor with RISC architecture is verified. To check the outputs, the code for 32-bit MIPS based RISC Processor was implemented and simulated using Cadence NC launch tool. NC Launch helps to simulate Verilog or VHDL codes. The Verilog code for 32-bit MIPS based RISC processor was simulated using Cadence NC Launch to check the outputs. Fig.4 refers simulation waveforms of the module [8].

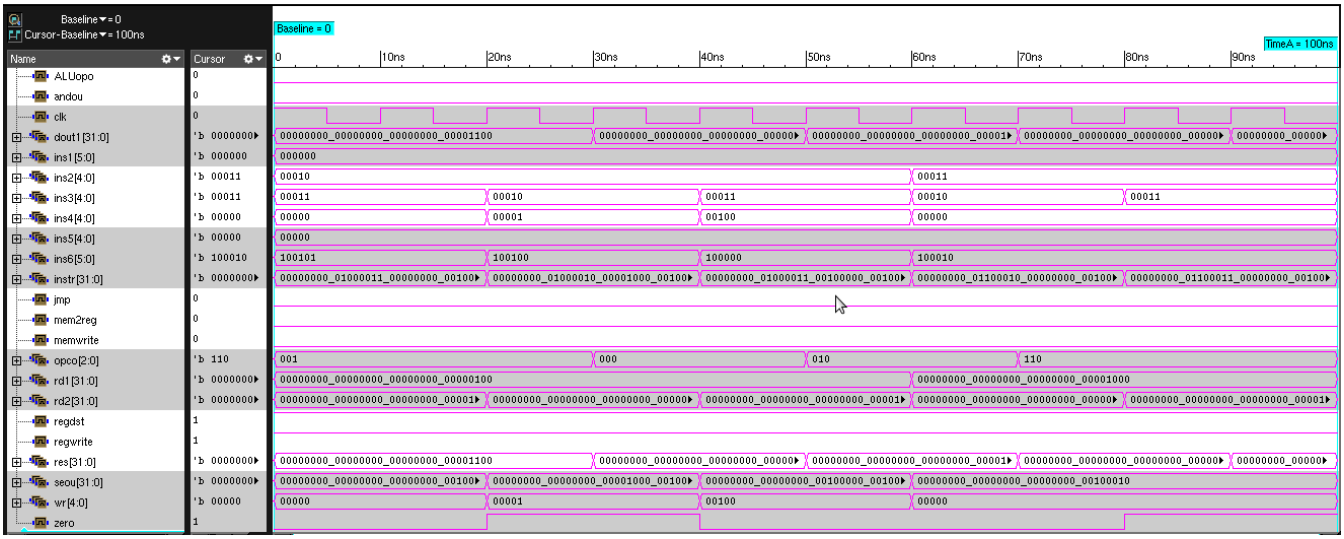

Fig.4 Simulation waveforms of the module

We implement the test bench for the MIPS design. Some series of instructions were given in the module instruction memory that are read from the module and are performed.

Along with the simulation, schematic is also seen. The below figure indicates the schematic of the 32-bit MIPS using NC launch in Cadence. Fig.5 refers schematic of the module.

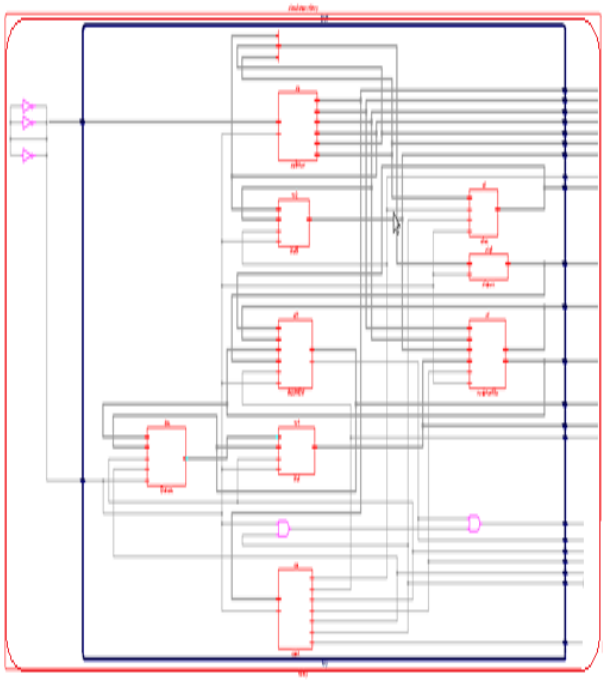

Fig.5 Schematic of the module

A. Cadence RTL compiler

In this, RTL compiler tool the Register transfer level file can be converted in to a gate level netlist. The 32-bit MIPS based RISC architecture, netlist will be generated in which the file will be in the form of Standard cell library. The RTL compiler tool can be generated by area, power and timing reports of the entire module. Fig. 6 refers RC Schematic of the module.

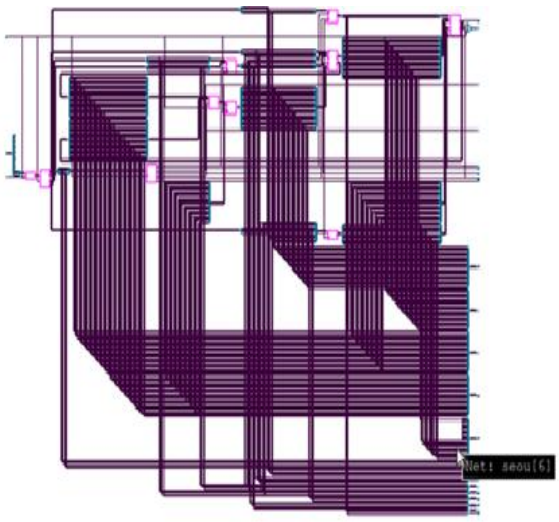

Fig.6 RC Schematic of the module

B. Area,Power and Delay Report:

The power report consists of some columns like Dynamic power (nW), Leakage power (nW) and Total power (nW) for the module written in verilog code. The power report will be generated as per the hierarchy given in the verilog code. Here, the modules used in the code are data RAM, register files, ALU unit, control unit etc., the leakage power and dynamic power consumed by the every module.

The timing report consists of fanout, slew rate, delay time, arrival time as per the hierarchy given in the verilog code. In arrival time the rising time data will be there in the timing report. The arrival time, delay time and slew rate consumed by the every module. The area report gives the summary of each component in the current design.With the help of specified technology library and the number of gates, the area report is generated. The hierarchy levels are indented in the report. RC report cell area as well as estimated net area is included. Table I refers Report for Area, Power and Delay in 90nm technology.

Table I: Report for Area, Power and Delay

S.no	Parameters	Values
1.	Area	54956 (mm^2)
2.	Power	400611 (nw)
3.	Delay	1048 (ps)

C. Layout

The automated general layout is generated by cadence SoC Encounter where the design constraints are imported and the design netlist is generated in pre-layout steps. Final layout of the design is shown after performing nano route shown in Fig. 7. And the Clock distribution in layout has shown in Fig.8.

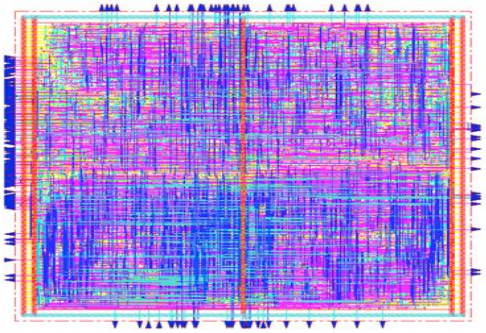

Fig. 7 Layout of the module using Cadence SoC Encounter tool

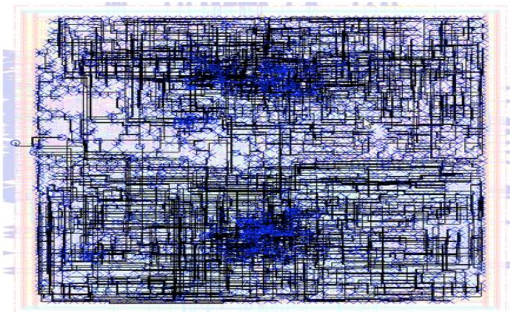

Fig. 8 Clock distribution in layout

D. Timing Signoff Analysis

After verifying the physical design that is implemented on the encounter tool, the analysis which is generated with timing is completed through the tempus file for the overall clock in the physical design. The tempus file implements the timing slack of the overall physical design [7].The results obtained are shown in Fig.9 below. The timing debug analysis results of encounter tool and tempus tool are shown in Fig.9. The generated timing slack from tempus tool gives better result in comparison with encounter tool.

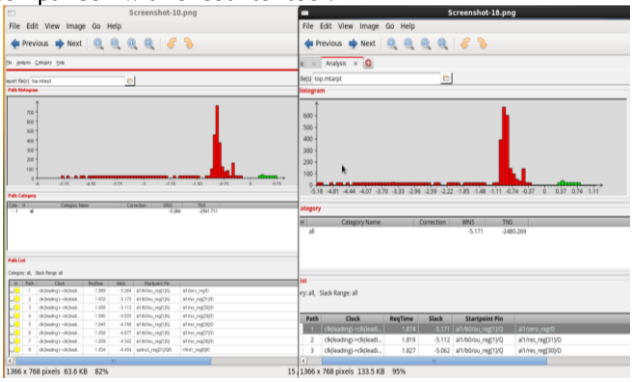

Fig. 9 The timinig debug analysis

V. CONCLUSION

The Five stage 32-bit MIPS processor based on the RISC architecture has been designed and implemented in this work using Cadence tool. The MIPS processor Functionality is verified using NC launch tool and synthesis is performed using Cadence work handles different instructions. Each category has a unique format to serve different functionalities. Power, timing and area parameters are optimized. The SoC encounter tool is used to build a layout from particular design and more over to calculate a static timing analysis of the design.

REFERENCES

1. S. L. Harris et al., "MIPS fpga: using a commercial MIPS soft-core in Systems, vol. 11, no. 4, pp. 283-291, 7 2017.
2. A. Ashok and V. Ravi, "ASIC design of MIPS based RISC processor for high performance," 2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2), Chennai, 2017, pp. 263-269computer architecture education," in IET Circuits, Devices.
3. M. N. Topiwala and N. Saraswathi, "Implementation of a 32-bit MIPS based RISC processor using Cadence," IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, 979-983.B. Smith, “An approach to graphs of linear forms (Unpublished work style),” unpublished.
4. Neeraj Jain, "VLSI Design and Optimized Implementation of a MIPS RISC Processor using XILINX Toor', International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE) Volume 1, Issue 10, December 2012.
5. Sharda P. Katke, G.P. Jain, "Design and Implementation of 5 Stages Pipelined Architecture in 32 Bit RISC Processor", IJETAE, Volume 2. Issue 4. April 2012, pp. 340-346.
6. https://www.mips.com/. About MIPS internal structure and it’s register formats and it’s types. Pipelining technique in MIPS.
7. T.R.Padmanabham, B.Bala Tripura Sundari, " Design Through Verilog HDL",Text book in 2004.
8. Pipeline Hazards and their effects on the execution flow of any architecture which contains pipelining techniques in them. https://www3.nd.edu/~mniemier/teaching/2011_B_Fall/lectures/14_PPT_1up.pdf.

AUTHORS PROFILE

B.Kiran Kumar received his B.Tech degree in Electronics and Communication Engineering from Sri Vani Group of Institutions which is affiliated under JNTUK, Andhra Pradesh, India in 2017. He is pursuing his M.Tech degree in VLSI and Embedded Systems Design specialization in GMRIT which is affiliated under JNTUK, Andhra Pradesh, India. His areas of interest are ASIC design implementations and low power VLSI.

Anilkumar B has completed Bachelor of Technology from JNTU, Hyderabad in 2005 and completed Master of Technology with Embedded systems specialization from JNTU, Hyderabad in 2008. He is working as an Assistant Professor in the department of ECE, GMRIT-Rajam, India since 2009 and pursuing PhD(Part-time) in Medical Image analysis from Andhra University, Visakhapatnam, India.