International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6S2, August 2019

Design and Implementation of 6-Stage 64-bit MIPS Pipelined Architecture

P. Indira, M. Kamaraju

Abstract*: Pipelining is the concept of overlapping of multiple instructions to perform their operations to optimize the time and ability of hardware units. This paper presents the design and implementation of 6 stage pipelined architecture for High performance 64-bit Microprocessor without Interlocked Pipeline Stages (MIPS) based Reduced Instruction set computing (RISC) processor. In this work, combining efforts of pre-fetching unit, forwarding unit, Branch and Jump predicting unit, Hazard unit are used to reduce the hazards. Low power unit is used to minimize the power. Cache Memories, other devices and especially balancing pipeline stages optimize the Speed in this work. DDR4 SDRAM (Double Data Rate type4 Synchronous Dynamic Random Access Memory) controller is employed in this pipeline to achieve high-speed data transfers and to manage the entire system efficiently. Low power, Low delay Flip flops are used in pipeline registers that implicitly enhance the performance of the system. The proposed method provides better results compared to the existing models. The simulation and synthesis results of the proposed Architecture are evaluated by Xilinx 14.7 software and supporting graphs are plotted through MATLAB tool

Keywords : DDR4 SDRAM Controller, MIPS, RISC processor, Xilinx Tools

I. INTRODUCTION

Power, Speed and Area are the Major design constraints of any VLSI system design [1]. Battery life is sole important for portable devices, as it emphasizes the significance of power. According to the customer demand, speed and smaller devices (lesser area) are also equally important. Faster the operations, power consumption will be more. Similarly, sizing down leads to denser device of transistors in the devices and increases the power consumption. The limitations to achieve optimal design need the further Technology Innovation [2].

"Instruction pipelining" is a Technique applied in the design of recent microprocessors, microcontrollers and CPUs to attain the high performance design criteria [3], [4]. It is used to save the power while maintaining the higher speed.

Processors are the heart of the computer. There are many processors in the market to meet customers' specific applications to fulfill the design criteria. Pipelining is harder for older complex instruction set computing (CISC). As various instructions had various lengths (or) formats, the fetch and decode stages would require more time to find out the actual length of instructions and the position of the fields.

Reduced Instruction Set Computer (RISC) is smart computer architecture; it uses the most required 20% of the instructions; makes the pipelining much faster; and reduces the power consumption [5]. The first successful and classical MIPS RISC Processor contains 32-bit, 5-stage Pipelining with load and store memory access instructions [6]. RISC Processor nurtures the complexity at software level rather than at hardware level [7]. But conventional RISC processor is always busy with their reduced Instruction set, which leads to system delay. Hence, MIPS (Microprocessor without Interlocked Pipelining Stages) became a better alternative to it by exploiting all the advantages of conventional RISC [8]. Any processor when executes millions of instructions per second, a concern arises called "interlocking in the pipeline stages"; and the solution is MIPS, which will take care of such issues. For Power reduction, dual edge triggering types of Flip-Flop designed registers are used in the pipeline. Thus, the low power Flip-flops are implicitly involved in the low power pipelining design [9]. Similarly, the DDR4 SDRAM ("Double data rate type 4 synchronous Dynamic Random Access Memory") controller bridges the gap between SDRAM memory devices and processors subsystem. It not only accepts large data with easy data transfers, but also optimizes the power of the system [10]. In this work, 64 bit, 6 stage MIPS RISC Processor is designed and implemented with high performance features

II.PROPOSED METHODOLOGY

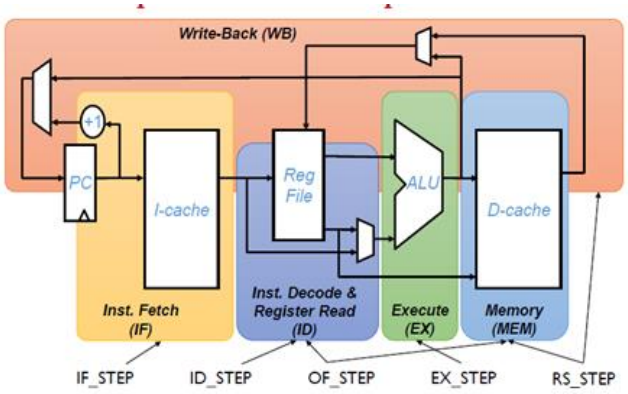

Fig.1. A Simple MIPS Data path Unit

A. MIPS RISC Processor

Microprocessor without interlocked pipeline stage (MIPS) is a type of "RISC based processor" widely used in embedded systems.

Manuscript published on 30 August 2019.
* Correspondence Author (s)
P. Indira*, ECE Department, C.U. Shah University, Wadhwan, Gujarat, India.
Dr. M. Kamaraju, Professor, ECE Department, Gudlavalleru Engineering College, JNT University, Gudlavalleru, Andhra Pradesh, India

The MIPS simple 64-bit architecture is a "fixed-length instruction set" and it is a "load/store data model". It consists of improved implementation of high-level languages. It supports 4 integer data types of 8-bit bytes, 16-bit half words, 32-bit words and 64-bit double words. Flexibility of high performance caches and memory management schemes strengthen the MIPS architecture. Predominant 64-bit floating-point registers and execution bits accelerate the tasks of handling some DSP algorithms and computing graphics tasks in real-time. Paired-single instructions pack two 32-bit floating-point operands into a 64-bit register, permitting "single instruction multiple data operations (SIMD)". This delivers double fast performance compared to customary 32-bit floating-point elements. This MIPS architecture features are compatible to 32-bit and 64-bit addressing modes

i. Steps in processing an instruction:

The idea of pipelining processing can be successfully applied for instruction execution in the processors. The instruction cycle can be sub-divided into constituent operations.

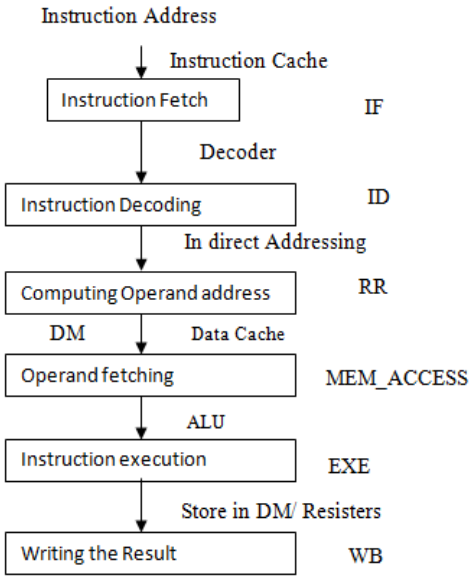

Fig.2. Flow chart of Pipelining stages

1) Instruction fetch (IF_STEP)
   In the IF stage, instructions are fetched from I-Cache Memory according to the PC address. The program counter arranges the next instruction after the first clock cycle.
2) Instruction Decode (ID_STEP)
   The fetched instruction code is decoded according to the operation by a Decoder Unit. Here the 64 bit instruction will be divided into several parts.
3) Operand Fetch (OF_STEP)
   The operand is fetched either from register or memory. In general, the **register bank** contains sufficient data space to store the required data. Also, for ease of operation, registers are preferred than Memory. When indirect addressing mode is used, it is easy to locate the address of the data by using registers and also it saves time.
4) Data Cache (RS_STEP)
   For data (operands) retrieval data cache memory unit is used. In MIPS architecture, it uses only loading and storing instruction as write and read operations. The results of ALU can be directly stored in data memory.
5) Execute (EX_STEP)
   The operation is performed on the operands with Arithmetic and Logic Unit.
6) Write back (WB_STEP)
   Write the execution result back to registers and memory

If any stage requires more clock cycles to complete the task, Hazard unit inserts stalls or buffer for a time lag.

ii. Data path components of 6-stage pipelining

1) Main components

a) Cache Memory (Instruction cache, Data cache):

Cache memory is a tiny and fast memory used to store most frequently used items of Main Memory. Because of its low storage space, immediate retrieval is possible, and the average access time to retrieve the items is very less compared to main memory. To speed up the process in our proposed architecture, separate data cache and instruction cache are used.

b) Registered bank:

This is a two port register file which can execute two concurrent read and write processes. These register files are used during the arithmetic, data commands and floating point operations. While the Reg_write signal is high a write operation is done to the register.

c) "Low Power Dual Edge-Triggered Flip-flop" (D-f/f):

Low power high speed Dual edge triggered Flip-flops (LPHSFF) [7] are used in the pipeline Registers. In this Flip-flop design explicit pulse triggering is used to control the clock activity. This Flip-flop employs dual edge triggering to reduce the delay and this conditional discharging saves the power. When compared to its counter parts these flip flops contain 25 transistors which is very less and hence gives an Area efficient design.

d) ALU:

Arithmetic Logic Unit is the Execution unit, to perform all the required operations. All the other units are to assist this unit to get the ultimate Result. It is not only responsible for high performance operations, but also performs in rapid manner. Here ALU is designed with IVC (Input Vector Control) technique to reduce leakage power by using particle swarm optimization algorithm.

e) DDR4 SDRAM Controller unit:

Presently, we used the most advanced version Controller DDR4 SDRAM meant for its data Reliability. The advantage of SDRAM over existing memories is its lower power consumption, lower cost, higher speed and allocation of high volume. High performance DDR4 SDRAM based controller [8] is used in this pipeline design to bridge the connectivity between SDRAM memory devices and processors subsystem

2) Auxilary components

a) Low Power Unit:

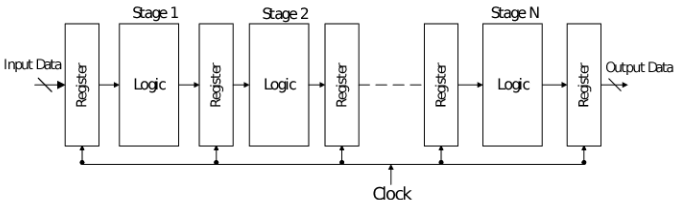

Fig.3. Clock Gating to Pipeline registers

The Low power unit is to reduce the unnecessary power, by connecting the entire pipeline Registers to Clock gating System. In this, non-working units are connected to NOP instruction and the only working registers utilize the required power. Clock gating method is used to reduce the switching activity (or) to minimize the dynamic power. The global clock operates the whole pipeline stages and the output is connected with clock gating. The gated clock blocks the main clock in the below circumstances:

1) When the halt Instruction is performed.
2) When there is a NOP operation for a long time.
3) When the increment to next Instruction of PC fails.

Dynamic Power Management (DPM) is a design method that connects the hardware units to get away from unnecessary power wastage. It concentrates on the devices which are in idle state. It turns the devices into standby mode wherever power is not required. In this way Total Power is optimized.

b) Hazard Unit:

An instruction pipeline may stall or be flushed for any of the following reasons:

1. Resource Conflicts:

This type of Hazard is also called as Structural Hazard. Pipelining is a parallel operations process while performing the functions sometimes needs the same hardware. In order to avoid stalls or flushing different units are allocated for different operations such as register bank, data memory, instruction memory etc. In this design, a separate Register Read stage is also taken to avoid this conflict even in indirect addressing mode also.

2. Data dependencies:

This type of hazard comes when an instruction depends on previous instruction result (or) any data which is not yet generated useful for present stage. This dependency is also called as Data Hazard. Potential Data dependencies are:

- RAW (Read-after-write): Read must wait until earlier write finishes.
- WAR (Anti dependence): Write must wait until earlier read finishes.
- WAW (Output dependence): Earlier write cannot be later write.

Check for data hazard each time whether any of the above conditions are met or not. Suppose RAW condition is met, compare Read register specifiers for new instructions with write register specifiers for older instructions.

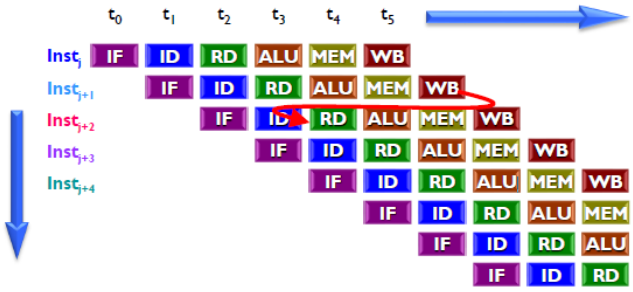

Fig. 4 .Data Hazard occur due to RAW condition

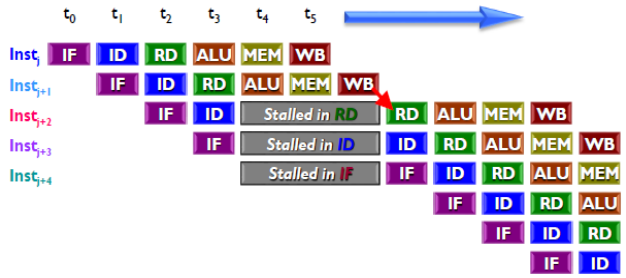

Fig. 5.Stalls in different stages due to RAW condition

Suppose, the 3rd instruction register_read stage requires the information from write back stage of the 2nd instruction, RAW dependency occurs. RD stage comes first (at 4th clock cycle) when compared to WB stage (at 6th clock cycle) in 2nd instruction. In this case, stalls occur in 3rd instruction in RD stage, in 4th instruction in ID stage and in 5th instruction in IF stage. Here, forwarding path unit transfers the information directly to ALU from WB stage to avoid stalls.

3. Conditional branching:

This Hazard mainly occurs when the branch instruction address is not known before the branching. This hazard is called as Control Hazard. In general, control dependencies occur due to

- Branch condition that must execute before branch target.
- Instruction of the branch that cannot run before branch.
  - due to multiple in-flight instructions

Here "Pre-fetching unit" and "Branch and Jump unit" help to rearrange stage and store the required address helps to control this hazard.

*c) Instruction pre-fetching unit[*11]*:*

Instruction Pre-fetching Unit is to eliminate stall situation by reorder the sequence of the instruction. For avoiding the stalls, data forwarding Unit directly sends the data to concerned unit by avoiding the proper channel. This is one way to evade the stalls. The other way is to reorganize the stages in such a way that data dependencies can be eliminated.

d) Branch and Jump prediction unit [11]:

A 2-bit buffer register is used to store the Branch Return address (or) Branch status for ease of operation and to avoid the stalls. Whenever Branch (or) Jump instruction is performed, it has to know the Return address. It is stored in the buffer and the other time it can be retrieved easily and can be avoid flushing also.

We can minimize the hazards at hard level (dynamic hazard) and at software level (static hazard). Complier guarantees correctness by inserting No-ops (or) independent stage between dependent stages. Hardware checks at runtime for stalls and chooses the required hazard elimination unit to control this hazard. Pipeline Interlock is the mechanism to eliminate the dynamic hazard resolution.
Though care has been taken (Software and hardware level) to reduce the Hazard, totally it cannot be eliminated.

III. BALANCING THE PIPELINE STAGES

Pipelining is a concept used in many fields as well as in CPU architectures to accomplish a task. It doesn't lessen the time to complete an instruction; instead, it rises the magnitude of instructions that can be handled simultaneously and decreases the delay among the completed instructions. Reduction in the rotation time of processor increases the instruction throughput.

Suppose each clock cycle allotted of 10 sec time. If instruction fetch phase takes 6 seconds time, instruction decode phase takes 2 seconds, etc. as indicated in Fig .6
Without pipelining, total time taken to complete the task

$T_{cyc} = T_{IF} + T_{ID} + T_{OF} + T_{ES} + T_{OS}$

$T_{cyc} = 6 + 2 + 9 + 5 + 9 = 31$sec.

With pipeline :

$T_{cyc} = \max \{T_{IF}, T_{ID}, T_{OF}, T_{ES}, T_{OS}\}$

$= 9$ sec.

Speed up = 31/9 = 3.44

The system increases speed by 3.44 times.

Suppose, combine IF & ID stage into 8 sec.

Time, comprise of 4 machine cycles.

$T_{cyc} = 9$ sec.

Suppose machine cycle time reduces to 3 seconds – produce more number of clock cycles, but time can be optimized - minimize the wastage

$T_{cyc} = 3$ sec.

Speed up = 31/3 = 10.34

The system increases speed by 10.34 times.

4 machine cyc/ins **11 machine cyc/ins**

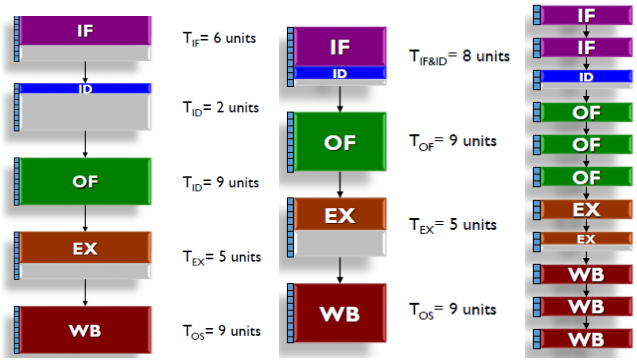

Fig. 6.Balancing Pipeline stages

IV. RESULTS AND ANALYSIS

MIPS consist of 6 stage pipeline techniques. Here, each stage simulation and RTL synthetic diagrams and results are tabulated.

A. Instruction Fetch (IF)

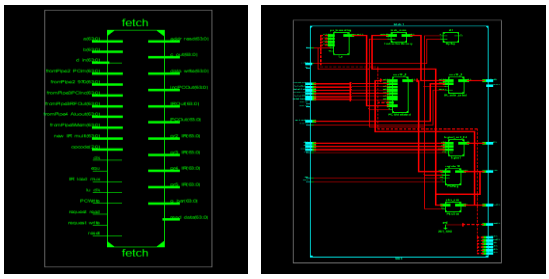

Fig.7.RTL schematic of Fetch Stage

The instructions of the program are fetched from Instruction Cache, according to PC address and it is the first step of the instruction pipeline stages. Based on the preceding instruction fetched, a PC will be incremented after each clock cycle.

Table-I: Power Consumption of Fetch Instruction

Frequency	Clock power µW	Leakage power µW	Dynamic power µW	Total µW
At 0	0	0.042	0	0.042
At 10 MHZ	0.00014	0.042	0.003	0.046
At 20 MHZ	0.00029	0.042	0.006	0.049
At 30 MHZ	0.00043	0.042	0.01	0.052

B. Instruction Decode (ID)

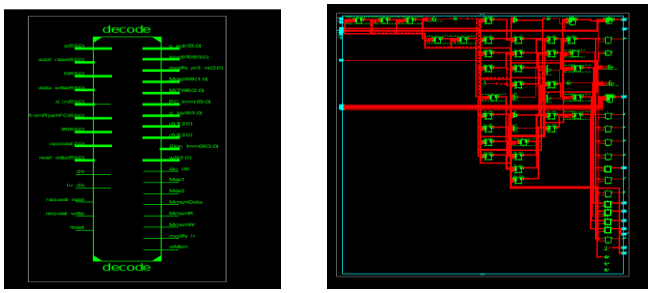

Fig. 8.RTL Schematics of Decode Stage

The next stage of the pipeline is Decode Stage, where a decoder decodes the instructions which are fetched from the instruction cache.

Table-II: Power Consumption of Decode Instruction

Frequency	Clock power µW	Leakage power µW	Dynamic power µW	Total µW
At 0	0	0.042	0	0.042
At 10 MHZ	0.00009	0.042	0.003	0.045
At 20 MHZ	0.00018	0.042	0.005	0.048
At 30 MHZ	0.00027	0.042	0.008	0.05

C. Register Read (RR)

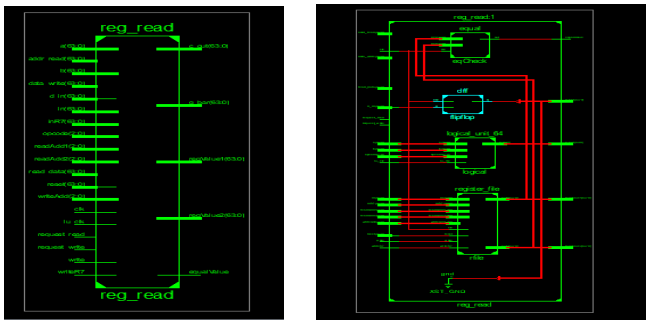

Fig. 9.RTL schematic of Register Read

Table-III:Power Consumption of Reg_Read Instruction

Frequency	Clock power µW	Leakage power µW	Dynamic power µW	Total µW
At 0	0	0.042	0	0.042
At 10 MHZ	0.00006	0.042	0.002	0.044
At 20 MHZ	0.00013	0.042	0.003	0.046
At 30 MHZ	0.00019	0.042	0.005	0.047

D. Execute

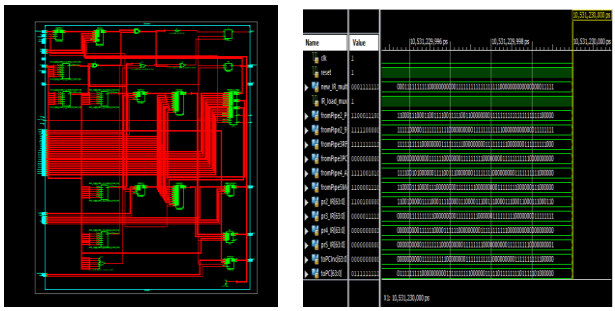

Fig.10. RTL schematic of Executive Stage

In this stage, the instructions are executed using ALU unit. The result is stored in data memory as well as in register bank.

Table-IV: Power Consumption of Execution of an Instruction

Frequency	Clock power µW	Leakage power µW	Dynamic power µW	Total µW
At 0	0	0.042	0	0.042
At 10 MHZ	0.00006	0.042	0.002	0.044
At 20 MHZ	0.00013	0.042	0.003	0.046
At 30 MHZ	0.00019	0.042	0.005	0.047

E. Memory –Access

The purpose of "Memory Access" is to scan from and copy to the data memory. The control signals pass in the EX/ MEM pipeline register to decide which of the tasks is to be performed. The outcome of the memory is printed in the MEM/WB register together with the WB control.

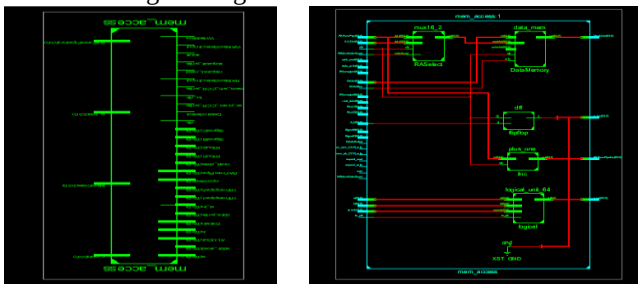

Fig. 11. RTL Schematic of Memory Access stage

Table-V: Power consumption of Execution of an Instruction

Frequency	Clock power µW	Leakage power µW	Dynamic power µW	Total µW
At 0	0	0.042	0	0.042
At 10 MHZ	0.00006	0.042	0.002	0.044
At 20 MHZ	0.00013	0.042	0.003	0.046
At 30 MHZ	0.00019	0.042	0.005	0.047

F. 'Write_Back

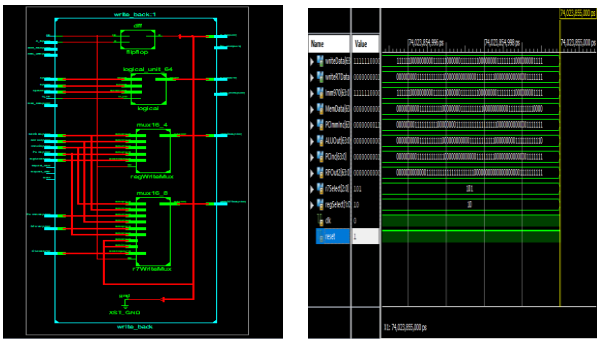

Fig. 12.Output of Write-back

Write_back is the last stage, where the data (results) is stored in the registers. Simultaneously, the same is stored in data cache for further use. Whenever a change occurs, that data is stored in register bank. However, for any permanent information retrieval, generally we use data memory, as it is having all the information. It is related with the same data stored in data memory and cache memory.

Table-VI: Power Consumption of Write_Back Instruction

Frequency	Clock power µW	Leakage power µW	Dynamic power µW	Total µW
0	0	0.042	0	0.042

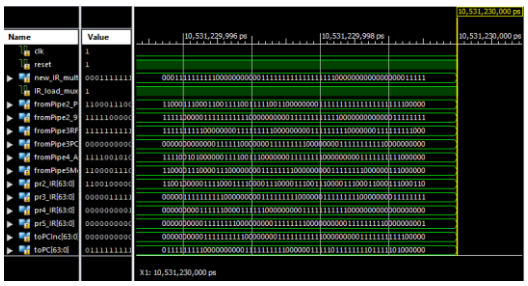

Fig. 13. Output of Pipelining

V. SOFTWARE TOOL

Xilinx ISE14.7 software tool is used in this work for design and analysis of the pipelining architecture. Verilog HDL ("Hardware Description Language") is used smartly for coding to reduce the software level hazards. MATLAB tool is used to evaluate relation between various parameters by plotting 3D graphs as pointed out in Fig. 15 & Fig. 16.

```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 3.908ns (Maximum Frequency: 255.886MHz)
   Minimum input arrival time before clock: 4.458ns
   Maximum output required time after clock: 0.663ns
   Maximum combinational path delay: 1.000ns
```

Fig.14. Time and delay summary report

Figure 14 gives information of execution time and delay. Less time and less delay increase the speed parameter. Due to this, speed of the system is also increased.

Table-VII: Device Utilization Summary

Logic Utilization	Used	Available	Utilization
Number of slice LUTs	151	63400	0%
Number of fully used LUT-FF pairs	0	151	0%
Number of bonded IOBs	210	265	74%
Number of BUFG/ BUFGCTRL/ BUFHCEs	1	128	0%

The figure 19 shows that the number of LUTs used for proposed architecture is 151 out of available 63400 LUTs. Hence, less number of LUTs is used and thereby, less area is required to build the architecture.

Based on the Artix7 Family and XC7A100T Device, the voltage and load range has been represented from the datasheet. For leakage power and dynamic power, 3D graph is plotted with 10, 20, 30 MHz. frequencies; and variations are shown in the figure below.

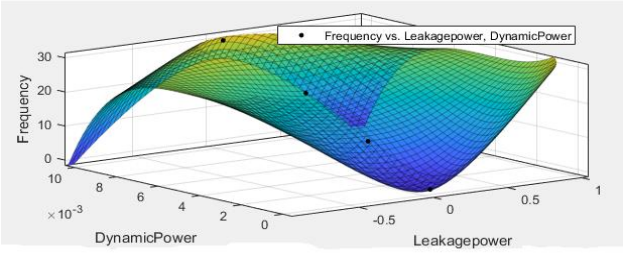

Fig. 15. Frequency vs. Leakage Power & Dynamic Power

Characteristics of the auxiliary supply voltage have a minimum of -0.5V to the maximum of 2V. The capacitive load used in the Artix 7 Family and it ranges from 0 pF to 8 pF. According to this range the graph has been represented.

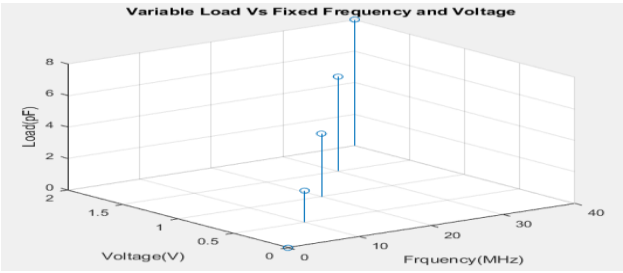

Fig. 16.Variable load vs. fixed frequency and voltage.

VI. COMPARATIVE ANALYSIS

Table-VIII: Power Comparison of Various Pipeline Models

Technology (6)	Baseline (generic)	180 nm	Clock gating at 180 nm	Clock gating + Multi Vt	Our proposed Model
Leakage Power (µW)	1.1	1.463	2.317	1.904	0.252
Dynamic Power (µW)	317.3	28.823	2.967	2.707	3.348
Total Power (µW)	318.4	30.286	5.284	4.611	3.600

Comparison of Power parameters (Leakage Power, Dynamic Power and Total Power) of our model is done with various pipelining models as shown in Table 8. Our proposed model utilized least Total Power 3.6 µW, when compared to the other counterparts; same is shown in the plot (Fig. 17).

Table-IX: Frequency comparison of various pipeline models

Parameters	Our Model	Low Power MIPS [12]	MIPS Core [13]	Tiny CPU [14]
Max. Frequency	255.88 MHz	205.7 MHz	95.5 MHz	89 MHz
LUT	151	1890	2340	336

Similarly, speed analysis (Maximum Frequency analysis) is carried out with various pipelining models. Maximum frequency of 255.88 MHz. is achieved through our model when compared to the other pipeline models; and same is indicated with plot (Fig. 18). Also, number of device utilization is less compared to other pipelining models as shown in Table 9.

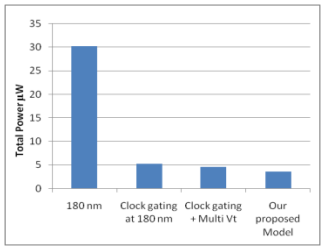

Fig.17.Power

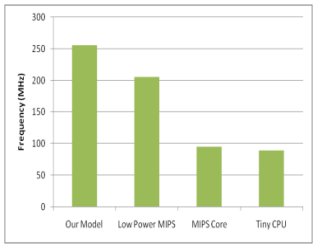

Fig.18. Frequency

VII. CONCLUSION

In this research work, we presented a 6-stage-64-bit MIPS RISC based Instruction structure Architecture. Here, we have used the hardware: Pre-fetching unit, forwarding unit, branch and jump predicting Unit used judicially to eliminate the hazards and proved it as high-performance architecture design. Every care is taken to reduce the hazards (eliminated 99% hazards) by using hardware and software levels. The devices used for various operations consume low power, high-speed. Additionally, low power unit controls the unnecessary wastage of power.

We compared our work with various other existing counterparts with respect to speed and power; and proved our architecture consumes less power and operates at higher speed. Our proposed architecture can be simulated and synthesized by Xilinx platform and 3D graphs support the results with MATLAB tool.

REFERENCES

1. P. Trivedi, and R. P. Tripathi, "Design & analysis of 16 bit RISC processor using low power pipelining" International Conference on Computing, Communication and Automation (ICCCA), May 2015. International Conference on (pp. 1294-1297). IEEE. (2015, May).
2. R. E. Bryant, K. T. Cheng, A. B. Kahng, W. Maly, R. Newton, and Sangiovanni-Vincentelli. "Limitations and Challenges of Computer Aided Design Technology for CMOS VLSI", IEEE Proceedings, Vol. 89, No. 3, March, 2001.
3. C. Sharma, and G. S. Saini, "Design And Analysis Of High-Performance RISC Processor Using Hyperpipelining Technique", IJASRE, vol. 3, 2017
4. A. Ashok, and V. Ravi, "ASIC design of MIPS based RISC processor for high performance" International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2), IEEE Proceedings, pp. 263-269, March 2017
5. J. Vijay Kumar, B. Nagaraju, C. Swapna and T. Ramanjappa, "Design and Implementation of Low Power Pipelined 64- bit RISC Processor using FPGA", International Journal of Advanced Research in Engineering and Technology, vol. 5, no. 2, Feb. 2014
6. Mangalwedhe, S. Roopa Kulkarni, and Y. Kulkarni, "Low Power Implementation of 32-Bit RISC Processor with Pipelining," Proceeding of the Second International Conference on Microelectronics, Computing & Communication Systems, 2017, pp 307-320.
7. S. Husainali Bhimani, N. Hitesh Patel and A. Abhishek 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 (IJAIS), May 2016.
8. P. V. S. R Bharadwaja, K. R. Teja, M. N. Babu, and K. Neelima, "Advanced low power RISC processor design using MIPS instruction set," 2nd International Conference on Electronics and Communication Systems (ICECS), IEEE proceedings, Feb. 2015, pp. 1252-1258.
9. R. Shandilya, and R. K. Sharma, R. K., "High-speed, low power dual-edge triggered D flip-flop. "International conference on Intelligent Computing and Control (I2C2)," IEEE proceedings, Jun. 2017, pp. 1-5.
10. M. A. Islam, M. Y. Arafath, and M. J. Hasan, "Design of DDR4 SDRAM controller," International conference on Electrical and Computer Engineering (ICECE), pp. 148-151, Dec. 2014.
11. M. Zulkifli, Y. P. Yudhanto, N. A. Soetharyo, and T. Adiono, "Reduced Stall MIPS architecture using pre-fetching accelerator," 2009 International conference on Electiral engineering and Informatics.
12. Mamum Bin Ibne Reaz, Shabiul Islam and S. Mohd. Sulaiman, "A single Clock Cycle MIPS RISC Processor Design using VHDL," ICSE 2002 Proceedings, Penang, Malaysia.
13. P. Gautham, R. Parthasarathy, and Karthi Balasubramanian, "Low Power Pipelined MIPS Processor Design," 2009 ISIC proceedings.
14. Koji Nakano, Kensuke Kawakami, Koji Shigemoto, Yuki Kamada and Yasuaki Ito, "A Tiny Processing System for Education and Small embedded Systems on the FPGAs," 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing.

AUTHORS PROFILE

P. Indira has obtained B.E. (ECE) from Andhra University, Visakhapatnam, M. Tech. (Instrumentation & Control) from Jawaharlal Nehru Technological University (JNTU), Hyderabad. She has obtained non-engineering degrees like M.B.A. (Production, Planning and Maintenance and Quality Management) with Hotel Management as an additional elective in 2004 from Annamalai University; and pursued M.Sc. (Applied Psychology) from the same University, Tamil Nadu in 2016. And presently pursuing Ph.D. ("Low Power VLSI Design") from CU Shah University, Wadhwan, Gujarat, India. She has got an experience of 13 years as faculty and has 4 years of research experience (VLSI). She has published 12 Research papers in different journals: and participated. She has participated and organized a number of National/International conferences and workshops/seminars. Presently working as Assistant Professor in the Department of Electronics and Instrumentation Engineering (NBA Accredited), GITAM University, Hyderabad campus (an Autonomous and NAAC accredited Institute). She is a permanent member in ISOI technical body. Presently she is pursuing M.A. (Public Administration) from Indira Gandhi Open University (IGNOU). Reviewed Technical papers of VLSI design in Scopus Indexed Journals. Her areas of interest in research are Low Power VLSI Design and Architecture and Control Systems.

Dr. M. Kamaraju has obtained B.E. (ECE) and ME (EI) from Andhra University, Ph.D. (Low Power VLSI Design) from JNTUH, Hyderabad. He has an experience of 25 years in the field of teaching, and research experience of 5 years in the field of VLSI design. He has been published 112 Research/technical papers in various journals: National and International, and participated in national and international conferences. Editorial board member of International Journal of VLSI Design and Communication Systems (IJVLSICS) and Reviewer for number of International Journals and various IEEE international conferences organized outside INDIA. JNTUK, Kakinada awarded "Certificate of Appreciation" towards teaching methodologies, Research and transfer of Knowledge to the society on 21st August 2012. Best Lecturer Award of SGD and PG College, Visakhapatnam for the academic year 1996–1997. A total of 55 technical guest lecturers are delivered in the field of VLSI design, embedded system Design and IoT at various workshops. Member of Board of studies (ECE) (2012–2015), UCOE, JNTUK, Kakinada and Swarandhra Institute of Engineering and Technology, Narasapur and Chairman, BoS (ECE), GEC (2014–2017). The Govt. of AP nominated him as an Executive Council Member for JNTUK, Kakinada. His Professional Memberships are FIETE, FIE and LISTE, also MVSI and Member of IEEE. He served as Chairman of Institution of Electronics and Telecommunication Engineers, Vijayawada Centre for the years 2012–2014 and as Chairman of Institution of Engineers (India), Vijayawada Local Centre for the years 2014–2016 and served as First Chairman of Institution of Engineers (India), Andhra Pradesh State Centre, Vijayawada 2016–2018. He was appointed as Technical Evaluation member for Alakananda Hydraulic Project, GVK Industries, and Hyderabad at Uttaranchal. He received Research fund of Rs 1 Lakh for the development of Wallace Tree Network Implementations by the IE (I), Kolkata. Granted with an amount of Rs. 25 Lakhs by the AICTE for modernizing Labs under MODROB scheme. Presently working as Professor and Mentor (AS&A), Department of Electronics and Communication Engineering (NBA Accredited), Gudlavalleru Engineering College (an Autonomous and NAAC accredited Institute), Gudlavalleru.