RESEARCH ARTICLE OPEN ACCESS

High Performance 6-Stage MIPS RISC Pipelined Processor Architecture Design

P. Indira*, M. Kamaraju**
** Research Scholar, Department of ECE, CU Shah University, Wadhwan, Gujarat*
*** Professor, Department of ECE, JNT University, Andhra Pradesh*
Corresponding Author : P. Indira

ABSTRACT
Pipelining is a process of concurrent execution of instructions in sections overlapped for effective utilization of resources. In this paper, high performance 64 bit, 6 stage MIPS RISC processor is designed to enhance the speed and for better handling of the criticality of the pipelining process. The paper consists of several optimizing devices and methods which not only concentrate on low power and high speed but also curtail the hazards in a critical manner. In the comparison study, our proposed architecture has reduced power by 19% and enhanced speed by 12%, when compared to our nearest counterpart. The simulation results are carried out with Xilinx platform and proved the superiority of our model. The 3-D graphic representation employed through MATLAB tool.

***Keywords* -** MIPS RISC Processor, Xilinx platform, MATLAB, DDR3 SDRAM.

Date Of Submission:27-12-2018 Date Of Acceptance:11-01-2019

I. INTRODUCTION

In VLSI circuit design, generally complexities arise due to trade off between critical parameters such as power, delay and area occupied. As the portable device's speed enhancement chuck the designers into difficulties in controlling power for minimizing cooling cost, battery durability etc. One of the important ways out is pipelining implementation which not only reduces the delay in the long path but also increases speed with optimal power without overhead of throughput [1].

Pipelining is a well constructed linear structure, where implementation of instructions is carried out stage wise to deploy the ease of the operation. Moreover, instructions implementation synchronization optimizes the time requirements and resources.

Processors are the crucial units in a computer, as many of the important critical activities can be performed through them [2]. Microprocessor without interlocked pipelining stages (MIPS) is an RISC (Reduced Instruction Set Computer) based instruction set architecture which not only enhances speed by RISC set, but also efficient in controlling Hazards.

The pipeline Registers in this work are employed with Dual Edge Implicit pulse triggered Flip-flop with an embedded Clock Gating Scheme (DIFF-CGS) flip-flops. These are low power Flip-flops, implicitly contribute in the reduction of overall power of the pipelining process [3].

DDR3 SDRAM (Double Data Rate type 3 Synchronous Dynamic Random Access Memory) controller is an efficient controller that not only reduces power and enhances speed of operations, but also has got robust features like Auto Refresh, Master reset, Larger densities and modules for all applications than DDR2 SDRAM [4].

Hazard unit is an important section in the Pipelining process as millions of instructions concurrently perform the tasks causing problems, sometimes lead to flushing of the pipeline. These are effectively handled by Forwarding unit, Branch & jump predicting unit, Pre-fetching unit besides Software controlling. [5][6][7].

In this research, 64-bit 6 stage MIPS RISC Processor is designed for low power and high speed pipelining.

II. PROPOSED METHODOLOGY:

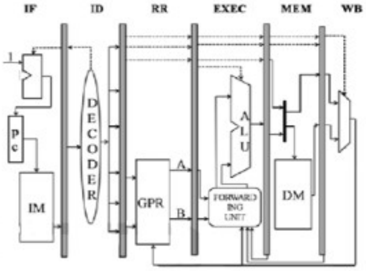


Fig. 1: Simple MIPS pipeline stages.

2.1 MIPS RISC Processor

This processor has 64-bit instruction set and general purpose registers. Optional floating point unit with sleep, suspend Mode/ Instructions.

Built-in High Performance/ Low power 28 nm, high-k-metal gate (HKMG) process technology. It has 52 Gb/s I/O bandwidth, 100,000 logic cell capacity, 264 GMAC/s DSP and flexible built-in DDR3 memory interfaces enable a new class of high-throughput, low-cost automotive applications. It has many high-end features such as integrated advanced Analog mixed signal (AMS) technology, Analog next level integration through the seamless implementation of independent dual 12-bit, 1 SPS, 17-channel analog-to-digital converters and meets the maximum temperature of 125deg.C.

The clock distribution provides six different types of clock lines (BUFG, BUFR, BUF10, BUFH, BUFMR and high performance clock) to address the different clocking requirements of high fan-out, short propagation delay and extremely low skew.

2.2 DDR3 SDRAM Controller unit:

In our pipeline design, a novel robust controller (DDR3 SDRAM) is used. When compared to DDR2 SDRAM, it consumes less power, has large databank to store the data and faster burst access. It manages the memory very efficiently with bi-directional data flow.

Double Data Rate type 3 Synchronous Dynamic Random Access Memory (DDR3 SDRAM) controller is the interface between DDR3 memory and the user. It has higher speed levels than DDR2 and DD4 SDRAM chips. It can transfer 64 bit word (double word) at a time than SDR SDRAM with (64-bit word) single data rate. It consumes less power (operating voltage: 1.5max) and higher speed (with chip capacity: 512 Mbits-8 gigabits) than DDR2 SDRAM (operating voltage 1.8V max and 256Mbits-46gigabits). It manages 8 data banks with efficient bank management systems and with operating stacks FIFO. It has 3 types of FIFOs: Attribute FIFO sends the required data to the sender up on their request. Write/Read data FIFOs writes on memory and reads from memory.

Request Breaker analyze the data based on the length, manner in which the data needs etc. and sends the information based on the decision of arbiter. Multiple requests from the users received by arbiter which estimates the request importance, rules associated with that before passing the request to the memory device.

The AUTO Refresh option in DDR3 SDRAM controller refreshes continuously before Read and Write process.

2.3 Low Power ALU Unit:

ALU unit is the core device used in pipelining for execution of the instruction. Generally, it consumes more power. Low power design is necessary to evaluate the results. As supply voltages reduce in nanometer Technology, Leakage power consumption is more in comparison with the dynamic power, in denser VLSI models. For this purpose Genetic algorithm based IVC Technique is used to reduce the leakage power [12].

2.4 Low Power Unit:

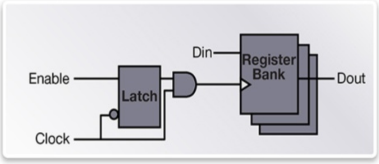


Fig. 2: Power gating to pipe line registers

Power gating is a Power management technique, which is better than clock gating technique. It has two modes of operations, Active mode and Low power mode. When it is in Active mode, it consumes the required power; when it is in power saving mode, power doesn't get wasted. This technique affects the communication and delay added to it

This can be applied to memory unit, I/O units, pipelining registers, etc.

The gated clock blocks the main clock in the following conditions:

1) When the halt Instruction is executed.
2) When NOP operation persist for a long duration.
3) When the increment to subsequent instruction of PC fails.

2.5 Hazard Unit:

When the instruction is prevented from being executed at any stage, it indicates pipelining is under problem, called Hazard, which is detected by Hazard unit.

There are broadly three types of Hazards.

2.5.1 Structural Hazard:

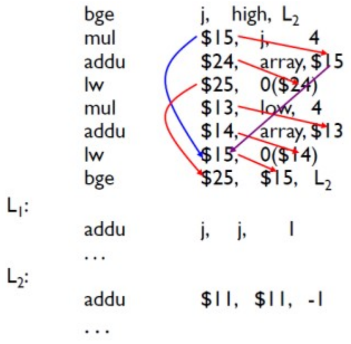
While performing various operations in pipelining stages with the same hardware units, scarcity of units, make the desired stage to wait for its turn. In order to protect from flushing, stalls (or) NOPs will be released.

For example, a single memory utilized as RAM and ROM, operand fetch and opcode fetch stages need the same hardware. In order to avoid this difficulty, we can either delay the later instruction or provide the separate memory units. In this pipeline stage, we are using two different memory units for operand fetch (instruction cache) and opcode fetch (data cache).

2.5.2 Data Hazard:

Data hazards may occur, if the present stage execution needs data (or) result from other stage, in which data is not yet available.

The data dependencies are shown as below:



```
        bge     j,   high, L2
        mul     $15,  j,    4
        addu    $24,  array, $15
        lw      $25,  0($24)
        mul     $13,  low,  4
        addu    $14,  array, $13
        lw      $15,  0($14)
        bge     $25,  $15,  L2
L1:
        addu    j,   j,     1
        . . .
L2:
        addu    $11, $11,  -1
        . . .
```

Fig. 3: Data dependencies

In the above data dependency model, Mul instruction is depending on the data from addu as well as on lw instruction.

To overcome this problem, not only the forwarding unit, but also the pre-fetching unit is employed.

2.5.3 Control Hazard

While performing the operations of original program, need arises for branching and hence the path jumps from the main program to sub program. If the destination address is not specified due to some reasons, it has the dilemma for return address. This hazard is called Control Hazard.

The remedy for control dependency is either re-ordering using pre-fetching unit or providing a location for return address, i.e., a branch and jump buffer unit is provided.

2.6 Forwarding Unit:

Hazard unit compares the addresses of the source registers of the current instruction decoded with the address of destination register of previous instruction, according to these addresses, hazard unit send control signal to the forwarding unit.

Current result of ALU and previous result of ALU, which are in next pipelined stage, are forwarded to data forwarding unit, so that according to control signal forwarding unit acts.

A forwarding unit is used to forward the result to Exe Stage from EX, MEM & WB stages.

2.7 Instruction pre-fetching unit:

This unit is also meant for Hazard elimination. This unit concentrates on the immediate 3 instruction sequence and judges which instruction has to be carried out after current instruction to evade the stall. This means it reorder (or) reorganizes the instruction without any rupture to pipelining flow.

The example:

```
0       lw      $1      [MEM]
4       addi    $2      $1,     100
8       bne     r6,     r7,     2
12      sub     r5,     r5,     r4
16
20      or      r1,     r2,     r4
```

In this case, pre-fetching unit will alternate the instruction sequence as below:

```
0       lw      r1,     256 (r0)
4       bne     r6,     r7,     2
8       addi    r3,     r1,     r2
```

LW instruction is taken first as PCs first address. Later, checks the registers r6 and r7 for equality. After that, addi immediate instruction is carried out. The BNE instruction depends upon the branch checking result. If the branch is taken, the next instruction is 'OR'. Else, the next instruction is SUB.

2.8 Branch and Jump prediction unit [8]:

Jump instructions seek the Return address, so that after finishing the subroutine task, the branch wants to link to the main program. In case, Return address is not known by the Jump instruction, it cannot attend to the sub-program call, due to dilemma, and then Hazard unit releases the stalls. To overcome this difficulty, the available 2-bit buffer storage is employed to avoid stalls and flushing.

However, the hazard cannot be totally eliminated though, sufficient care has been taken (software & hardware level)

III. BALANCING THE PIPELINE STAGES

Instruction pipelining is a smart way to perform several operations simultaneously; but they all are fixed stages linked in the same way for millions of instructions.

Suppose a Job (or) Instruction is completed by time (T) means latency (delay) is T.

Throughput (T_{seq}) = 1/T

N Jobs are completed in a time N*T. The Job is divided into smaller stages. Suppose, the number of stages is 6, then the throughput of the pipeline

$T_{pipeline}$ = K/T

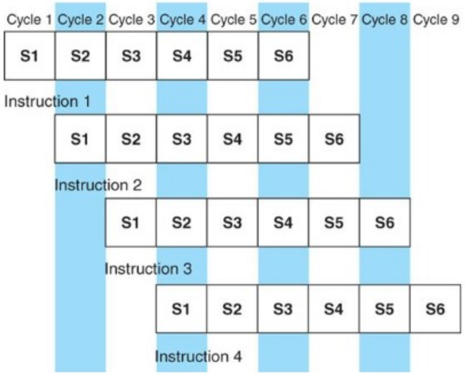


Fig.4: Six stage pipelining

Suppose the total time of all the clock cycles is T_{cyc}.
$T_{cyc} = T_{IF} + T_{ID} + T_{OF} + T_{ES} + T_{OS}$
$T_{cyc} = 6 + 2 + 9 + 5 + 9 = 31$sec.

Assuming each clock cycle is allocated with 10 sec time. In the event IF (instruction fetch) takes 6 sec times; ID (Instruction Decode) takes 2 seconds, so on as shown in Fig. 6.
If we reorganize the pipeline stages with four machine cycles per instruction,
$T_{cyc} = \max \{T_{IF}, T_{ID}, T_{OF}, T_{ES}, T_{OS}\}$
$= 9$ sec.
Speed up = 31/9 = 3.44
The system increases speed by 3.44 times.
If we again re organize 11 machine cycles per instruction, the time reduces to 3 seconds.
$T_{cyc} = 3$ sec.
Speed up = 31/3 = 10.34
The system increases speed by 10.34 times.

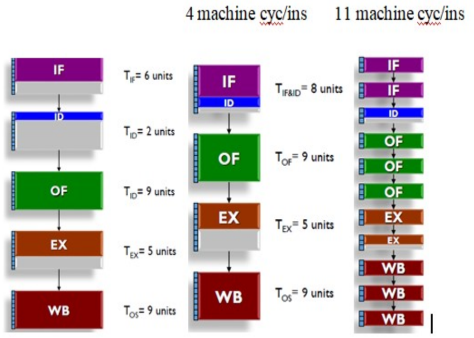


Fig. 5: Balancing Pipeline stages

IV. RESULTS AND ANALYSIS:

The simulation results with RTL schematic diagrams are evaluated in the each stage below

4.1 Instruction Fetch (IF) Stage:

Fig. 6: RTL schematic of Fetch Stage

In this phase, the instruction opcode is fetched from Instruction Memory. For ease of operation, memory unit is divided into two parts – one is Instruction Memory, which contain operation codes and Data Memory which contains data and Results.

TABLE 1. POWER CONSUMPTION OF FETCH STAGE

Frequency (MHz)	Time taken (ns)	Delay (ns)	Power (µW)
250	0.50	0.359	4.66
500	0.52	0.657	8.31
750	0.54	0.700	11.97
1000	0.57	0.736	14.63

4.2 Instruction Decode (ID) Stage:

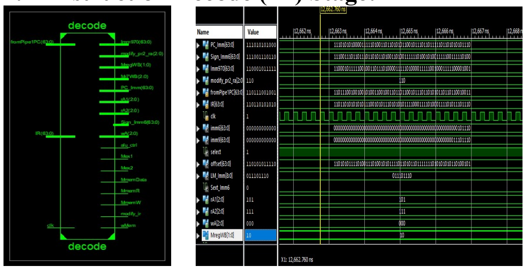


Fig. 7 RTL Schematics of Decode Stage

The retrieved operation code for the instruction is given to the Decode Stage. The decoded code is handed over to next stage.

TABLE 2. POWER CONSUMPTION OF DECODE STAGE

Frequency (MHz)	Time taken (ns)	Delay (ns)	Power (µW)
250	5.00	0.236	4.09
500	5.12	0.489	6.18
750	5.15	0.400	10.26
1000	5.20	0.458	12.35

4.3 Register Read (RR) Stage:

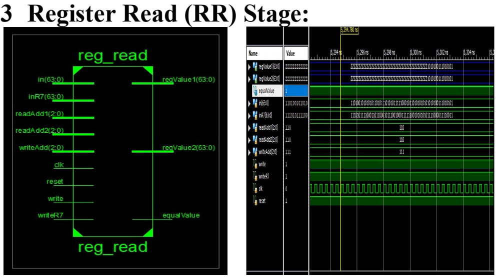


Fig. 8 RTL schematic of Register Read

There are various kinds of registers meant for different types of instructions. The required data is taken either from Registers (or) Main Memory.

TABLE 3. POWER CONSUMPTION OF REG_READ STAGE

Frequency (MHz)	Time taken (ns)	Delay (ns)	Power (µW)
250	0.678	0.424	4.67
500	0.726	0.610	9.99
750	0.846	0.730	12.23
1000	0.928	0.880	14.97

4.4 Execute Stage:

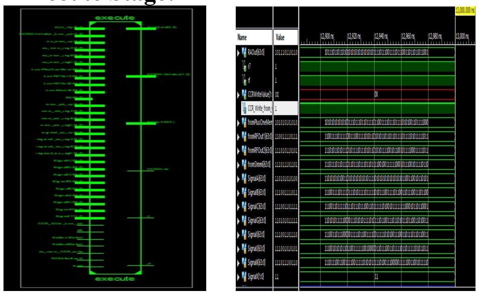


Fig. 9: RTL schematic of Executive Stage

This is the main stage to accomplish the task. The ALU unit performs not only Arithmetic and Logical Operations, but also does the Data transfer and Branch operations. As it is a critical stage, Hazards may occur. All the supporting units are used to assist ALU such as Forwarding Unit, Pre-fetching unit, Buffer unit, etc.

TABLE 4. POWER CONSUMPTION OF EXECUTION STAGE

Frequency (MHz)	Time taken (ns)	Delay (ns)	Power (µW)
250	1.14	0.604	5.49
500	1.89	1.276	9.88
750	4.57	2.230	12.48
1000	4.89	3.430	16.97

4.5 Memory –Access Stage:

This stage is used either before execution stage (or) after execution stage; Because, the results are transferred to memory (or) taken out from this memory to execution unit to perform the operation. Load and store architecture perform 'read from' and 'write to' operations.

mem_access
mem_access

Fig. 10: RTL Schematic of Memory Access stage

TABLE 5. POWER CONSUMPTION OF MEMORY ACCESS STAGE

Frequency (MHz)	Time taken (ns)	Delay (ns)	Power (µW)
250	4.01	0.329	4.97
500	5.83	0.623	9.55
750	6.03	0.815	12.82
1000	6.54	1.002	15.09

4.6 Write_Back Stage:

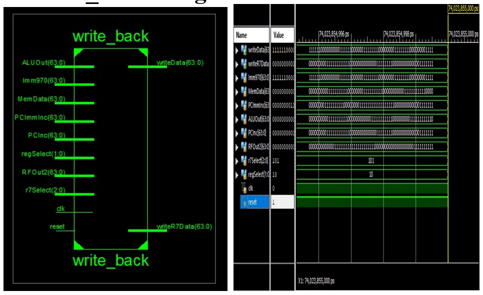


Fig. 11 Output of Write-back

The last stage is the Write-back stage where the storage of the result from ALU into register bank takes place.

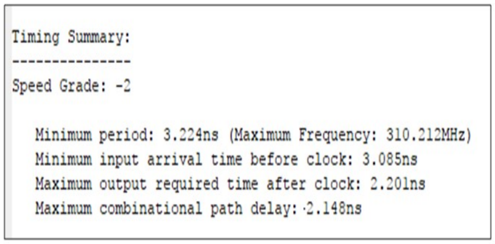
TABLE 6. POWER CONSUMPTION OF WRITE_BACK STAGE

Frequency (MHz)	Time taken (ns)	Delay (ns)	Power (µW)
250	5.02	0.196	3.56
500	5.10	0.477	5.00
750	5.17	0.410	8.23
1000	5.22	0.466	9.34

V. SOFTWARE TOOL

Xilinx ISE14.7 software tool is used in this pipelining work in power, frequency, Time, delay analysis. VHDL hardware description language is used judicially not only for optimal performance of operations, but also to support Hazard minimization.

MATLAB tool is used for graphical analysis and to show the relationship between different parameters, such as frequency, voltage, load, etc.

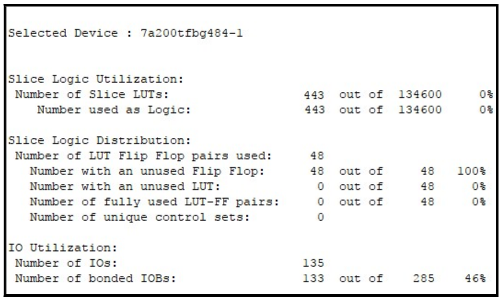


```
Timing Summary:
---------------
Speed Grade: -2

   Minimum period: 3.224ns (Maximum Frequency: 310.212MHz)
   Minimum input arrival time before clock: 3.085ns
   Maximum output required time after clock: 2.201ns
   Maximum combinational path delay: 2.148ns
```

Fig. 12: Time and delay summary report

Time and Delay summary report gives maximum operating frequency, that is 310.212 MHz which is high when compared to the existing models, and path delay found to be approximately 2.148 nanoseconds, which is also less.



```
Selected Device : 7a200tfbg484-1


Slice Logic Utilization:
 Number of Slice LUTs:                  443  out of  134600     0%
    Number used as Logic:               443  out of  134600     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:     48
   Number with an unused Flip Flop:      48  out of      48   100%
   Number with an unused LUT:             0  out of      48     0%
   Number of fully used LUT-FF pairs:     0  out of      48     0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                         135
 Number of bonded IOBs:                 133  out of     285    46%
```

Fig. 13 Device Utilization Summary

The above device utility summary shows that only 443 slice LUTs are utilized out of 204000 LUTs, which is quite less, and is an area efficient architecture.

Based on the Artix-7 family with XC7A200T device of FBG484 Package, the voltage, load and frequency plot is graphically represented its relation through 3D animation with 0 to 40 MHz frequency with varying capacitive load from 0 to 10 pf variation and 0 to 2V voltage change.

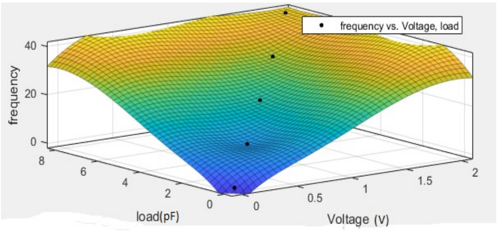


Fig. 14: Fixed Voltage and load vs. Variable frequency

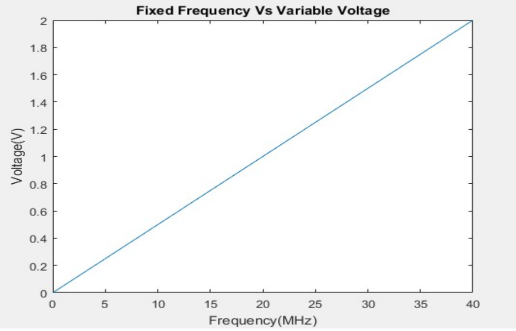


Fig 15: Fixed frequency vs. variable voltage

Fig 15 is a representation of fixed frequency versus variable voltage. This graph shows that frequency increases then voltage also increases linearly.

VI. COMPARATIVE ANALYSIS.

TABLE 7. POWER COMPARISON OF VARIOUS PIPELINE MODELS

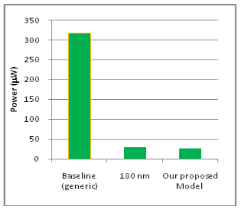
Technology (2)	Baseline (generic)	180 nm	Our Model
Frequency (GHz)	1.2	1.16	0.310
Area of Cells	558	560	443
Total Power (μW)	318.4	30.286	25.44

In the above comparative study, it is observed that our model at 0.310 GHz frequency, the power consumption is 25.44μW, which is 18.6% less when compared to the nearest counterpart.

TABLE 8. FREQUENCY COMPARISON OF VARIOUS PIPELINE MODELS

Parameters	Our Model	Single core Processor [13]	Low Power MIPS [10]	MIPS Core [9]	Tiny CPU [11]
Max. Frequency (MHz)	310.212	277.9	205.7	95.5	89
LUT	443	1168	1890	2340	336

Similarly the maximum operating frequency achieved in our model is 310.212 MHz which is 12% higher speed than the nearest counterpart.



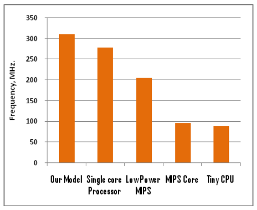


Fig. 16: Power comparison **Fig. 17**: Frequency comparison

VII. CONCLUSION

A 6-stage, 64-bit MIPS RISC Processor based ISA is designed with pre-fetching unit, forwarding unit, branch and Jump buffer unit etc. for controlling hazards. These precautionary measures with hardware and software level care almost eliminate all the hazards.

Power gating and Balancing Clock period methods with other implicit measures reduce power and enhance speed with no overhead performance and throughput.

The comparative analysis with other existing models with respect to power, frequency and device utility shows our model superiority. Our proposed model simulated and synthesized by Xilinx platform and 3D graphs are plotted for various parameters evaluation with MATLAB tool.

REFERENCES

[1]. Saranya Krishnamurthy, Ramani Kannan, Erman Azwanand & Kishore Bingi, Design of FIR filter using novel pipelined bypass multiplier, Third International Symposium on Robotics and manufacturing automation. IEEE, 2017.

[2]. Bharadwaja, P. V. S. R., Teja, K. R., Babu, M. N., & Neelima, K., Advanced low power RISC processor design using MIPS instruction set. In Electronics and Communication Systems (ICECS), 2nd International Conference, IEEE. 2015, 252-1258.

[3]. Liang Geng, Ji-Zhong Shen & Cong-Yuan Xu, Power efficient dual edge implicit pulse triggered flipflop with an embedded clock gating scheme, Frontiers of Technology & Electronic Engineering. 9, 2016, 962-972

[4]. Komala, M., Suvarna, D. & Nataraj, K.R, Design and Implementation of High Performance DDR3 SDRAM controller, Int. Journal of Engineering Research & Technology (IJERT), 2015, 4 (5), 857-861.

[5]. Kirat Pal Singh, Shivani Parmar (2012). VHDL implementation of MIPS 32 pipeline processor. International Journal Of Applied Engineering Research 7(11), 2012.

[6]. Christiensen C. Arandilla, Joseph Bernard A. Constantino, Alvin Oliver, M. Glova Anstacia, Balleswari-Alvarez P., & Joy Alinda P., Reyes High level implementation of the five stage pipeline ARM9TDM core, TENCON 2010 - IEEE Region 10 Conference, 2010.

[7]. Nishant Kumar & Ekta Aggrawal, General purpose 6 stage pipeline processor, International Journal of Scientific and Engineering Research, 4(9), 2013

[8]. Zulkifli. M., Yudhanto. Y. P., Soetharyo, N. A. & Adiono, T., Reduced Stall MIPS architecture using pre-fetching accelerator, International conference on Electrical engineering and Informatics, 2009.

[9]. Mamum Bin Ibne Reaz, Shabiul Islam & Mohd. S. Sulaiman, A single Clock Cycle MIPS RISC Processor Design using VHDL. ICSE 2002 Proceedings, Penang, Malaysia, 2002.

[10]. Gautham, P., Parthasarathy, R. & Karthi Balasubramanian, Low Power Pipelined MIPS Processor Design. ISIC proceedings, 2009.

[11]. Koji Nakano, Kensuke Kawakami, Koji Shigemoto, Yuki Kamada & Yasuaki Ito, A Tiny Processing System for Education and Small embedded Systems on the FPGAs. International Conference on Embedded and Ubiquitous Computing. IEEE, 2008.

[12]. Meera, S. & Umamaheswari, D., Genetic Algorithm for leakage Reduction through IV using VHDL, International Journal of Microelectronics Engineering (IJME), 1(1), 2015.

[13]. Nishant Kumar &and Ekta Aggrawal, General Purpose Six-Stage Pipelined Processor,. International Journal of Scientific & Engineering Research, 4(9), 2013.

P. Indira" High Performance 6-Stage MIPS RISC Pipelined Processor Architecture Design " International Journal of Engineering Research and Applications (IJERA), vol. 9, no.1, 2019, pp 17-23