Optimization of Pipeline process using MIPS RISC Processor

P. Indira[1], Dr. M. Kamaraju[2], Dr. Ved Vyas Dwivedi[3]
[1]*CUShah University, Wadhwan, Gujarat*
[2]*Gudlavalleru Engg. College, JNT University, Kakinada*
[3]*CU Shah University, Wadhwan, Gujarat*
(indiradelhi@yahoo.co.in)

***Abstract*--**Pipelining is a method of elevating the performance of a processor by optimally utilizing the time and hardware to improve the speed and throughput. In this paper, a High Performance pipelined Architecture with MIPS RISC five- stage processor is implemented with Kintex7 family. The power gating technique is used to minimize the power and super pipelining method, enhances the speed. The pipelining hazards, and Exceptions are systematically treated in various ways for the smooth functioning of parallel pipeline operations. In the comparative study, it elevates the performance of our design with respect to the existing models with the reduction in power by 28% and rises in speed by 42% with robust control of advanced features. Xilinx vivado ISE suite is used for simulation results with Verilog HDL coding. The MATLAB graphic representation relates the performance of various parameters.

Keywords: *MATLAB, MIPS, RISC, Kintex7, Xilinx.*

I. INTRODUCTION

In the modern era, electronic gadgets are indivisible items to mankind [1]. And also customers are expecting multi-functionality with high performance features [2]. Low power VLSI design faces difficulties to compromise with the delay, speed and area occupied [3]. Pipelining process resolves this problem with critical factors to provide optimum output [4]. This architectural approach allows the simultaneous execution of several instructions [5].

In pipelining, the implementation of Reduced Instruction Set Computing (RISC) processor is better than the Complex Instruction Set Computing (CISC) processor. RISC processor instruction set elevates the overall performance of the device by enhancing the speed and minimizing the total power [6]. This simple RISC processor not only speeds up the operations but also consumes less power [7] [8] [9] [10] [11].

In this work, 5-stage 64-bit MIPS RISC processor pipelining connects millions of instructions to perform their operations. Kintex 7 family device is used to implement pipeline operations using Verilog HDL simulations.

II. PROPOSED PIPELINE STAGES

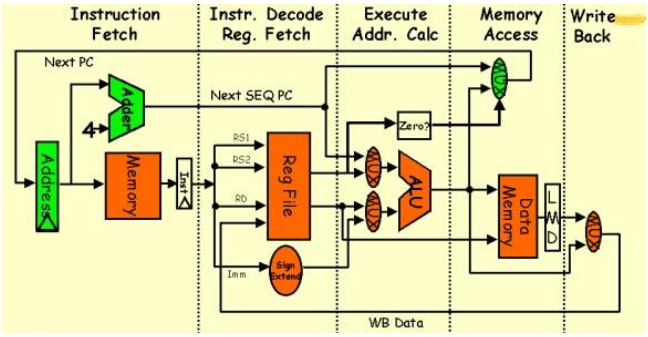

Fig 1. Five Pipeline stages

A. MIPS RISC Processor

For a new class of FPGA advanced processor cores have to balance the integration of price, performance, and power. The solution is the Xilinx Kintex7 family FPGAs. It has 28nm high-k metal gate (HKMG) process technology and high performance, Low-power (HPL) approach that drives up power efficiency. It is highly efficient, balanced design, optimized packaging, and performance. The flexibility and time-savings of programmability and targeted design platforms are for lower development times and costs. It has unified architecture to guard Internet protocols and can assimilate 6 series designs. The 72-bit, 1833 Mbps memory interface supports single memory buffer designs, can allow video signals through IP gate way support 123G channels over a 4-channel 10 gigabit Ethernet bridge. The unprecedented 144 MACs DSP Power, makes this processor is an excellent option for application for communications and ultrasound equipment.

B. Five stage pipelinling of an instruction:

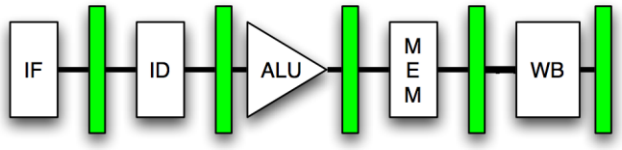

Fig 2. Pipelining stages

In the pipelining technique, number of instructions is executed simultaneously by various stages. Each stage has a specific task and for each instruction, the same task is performed. The execution time of each stage is fixed. But, the overall accomplishment of the task takes minimum time, because they are connected in a pipeline fashion. This increases the throughput of the device by utilizing the hardware optimally.

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.

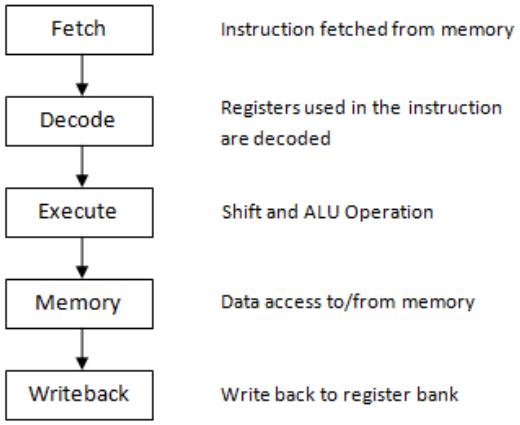

Fig 3. Flow chart of five Pipelining stages

1) Instruction Fetch: Instead of using the instruction memory, the instruction cache is used to obtain the fast fetching operation. Instruction Cache is small and it contains frequently used data. Fetching the opcode from the instruction cache is easy and delay is also condensed. Sometimes, along with the PC, predictor and buffer units are used to enhance the performance.

2) Instruction Decode: Once the instruction opcode is fetched from the instruction cache, it moves to the decode stage, where the decoder decodes the instruction opcode and at the same time, the control signals in the data path control the operations. For example, if the instruction is a branch instruction, then by taking the help of PC predictor, it jumps into the target rather than following the normal next instruction.

3) Execute: The execute stage is operated through an integer unit. The integer unit consists of the ALU, shifter and multiplier units. The arithmetic and logic unit performs Boolean operations (AND, OR, NOT, NOR, NAND, etc.), integer addition and subtraction. The bit shifter is responsible for the shift/rotate-operations. The multiple cycle instructions perform multiply/divide and floating point operations.

4) Memory access stage: Here also instead of Data memory, Data cache is used. It is also used for fast transactions. The data may be retrieved from this memory or the results of integer units are transferred into it. The load/store instructions are used for this purpose. The single cycle and two-cycle instruction write-read results/data can be stored in it.

5) Write Back Stage: During this stage, the results of integer unit are written into register bank.

C. Data path components of the processor:

1) Main components: Kinetix7 FPGA architecture consists of 5 main blocks, which includes main logic fabric, embedded memory, DSP resources, high - speed transceivers & memory interface, and integrated block for PCI express.

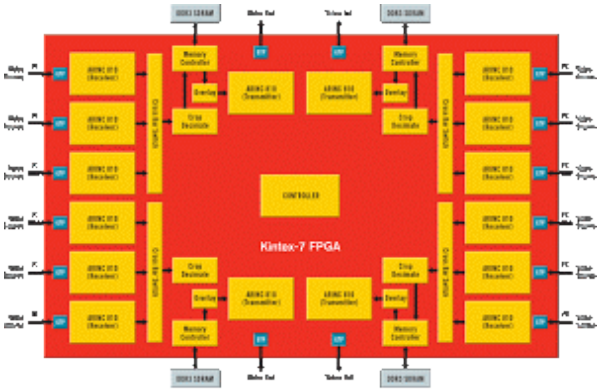

Fig 4. Kentex7 Processor components

a) Main logic fabric: The main configurable logic blocks (CLBs) accommodate two units and every unit consists of four 6-input LUTs and four flip-flops, carry chain logic and four additional chain flip-flops. For more efficient use of this main logic fabric, a dedicated multiplexer without LUTs is employed to enhance the speed.

b) Embedded memory: The embedded memory consists of 34Mb storage space allocates 477K logic cells. It has two dual - port block RAMs and each consists of 36 KB data storage and for parity bits, the 4 KB is allocated. It

accommodates a variety of sizes of word lengths and for flexibility, 6 input LUTs is there for small memory arrays. The special feature in this memory contains an error correcting codes (ECC) and an efficient power management system. Automatically, in power saving mode Xilinx vivado recognizes the unused RAMs and disable them efficiently to minimize the power.

c) DSP resources: Kinetix7 family consists of 1920 DSP slices and 2845 GMACs of performance with more than double the DSP bandwidth and each multiplier supports 18x25 bits to implement up to 35x25 multiply operations using a preadd block which runs up to 741 MHz.

d) High speed transceivers: The high speed serial transceivers (GTX) are used to catch up 12.5 GB per second, which is the highest data rate transfer in a mid- range FPGAs. The special features of transceivers consist of lowest jitter ability and high signal quality with high performance packaging integrity at the lowest price.

e) Memory interface: The Memory interface in this processor improves the interface speeds at 28nm by using the advanced clocking technology and critical data path components in Xilinx vivado software. Different speed rates are operated up to 1866 mb/s. DDR3 and DDR2 SDRAM devices flexibly connected through it. External memory devices can also be connected to it to support video and data storage.

f) PCI Express: It features both IP and PCI express GEN3 and integrated hard IP for PCIe GEN2 with full support for end point and root port configurations. The integrated hard block supports to 8 GEN1 and GEN2 channels while the soft IP supports up to 8 GEN3 channels with 8GB/s performance. All PCI express solutions are designed to AMBA 4AX14 specification.

g) Balanced Design: Its DSP resources are: wireless communication infrastructure, software design, serial connectivity, memory and logic performance that all are combined with power efficiency and ideal price make suitable for high- performance applications.

h) DDR3 Memory Module: DDR3 memory is the next version of DDR2 memory. It is a volatile memory used for storing the code and the data. It is implemented through I/O banks and every bank consumes 1.5 V supply voltage. The VRP/VRN DCI resistor is connected to the Bank 33 to cascade to 32 and 34 data interface banks by

Set DCI_CASCADE

Set_property slave_banks (32, 34) [get_iobanks32]

The Data path width is of 64 bits with Data rate maximum of 1600 MT/s. Additional 0.75 reference voltage is needed for this purpose.

III. OPTIMIZATION OF PERFORMANCE OF PIPELINING

A. Power gating:

Power gating is a well known technique where a sleep transistor is added between actual ground rail and circuit ground (called virtual ground) to reduce the power. This device turns off in the sleep mode to cut-off the leakage path. This technique provides a substantial reduction in the leakage at a minimal impact on the performance.

Power gating technique uses high V_t sleep transistors, whose cut-off V_{DD} is from a circuit block, when the block is not switching. The sleep transistor sizing is an important design parameter. This is known as MTCMOS (multi threshold CMOS) reduces stand by or leakage power and also enables I_{ddq} testing.

Power gating affects the design architecture more than that of a clock gating technique. It increases the time delays as power gated modes have to be safely entered and exited.

The power reduction must be achieved without trade off between performance and power; which makes harder to reduce leakage during normal (runtime) operation.

B. Super pipelining:

Super-Pipelining is a better pipeline process than normal pipelining in attaining high performance. Generally, all the pipeline stages do not utilize full time allocated for it in completing their task. In other words, much of the clock time is getting wasted due to allotment of equal time to all the stages.

Super pipelining deeply analyzes the pipeline stages and fragmant them into tiny stages for optimum utilization of time.

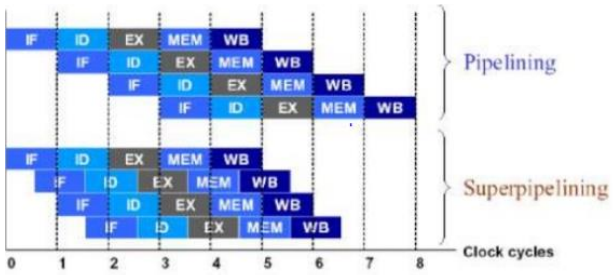

Fig. 5 Pipelining and Superpipelining stages

The above figure represents the saving of CPI through super pipelining than normal pipelining stages.

The major advantage of this super pipelining is to accommodate more stages of an instruction in less time. The disadvantage of this is by minimizing each stage time, potential data hazards may occur which introduces stalls. Then, automatically, delay of the pipeline may increase.

In this process, superpipelining is used and the precautions are taken at software and hardware level to prevent flushing of the pipeline.

IV. THE CONFLICTS IN PIPELINE STAGES

A. HAZARDS:

Hazards are the troubles encountered in the pipeline process by creating the obstacles to the next stage from being executed during its allocated clock time. By introducing stalls, flushing in the pipeline can be avoided; however, they make the system run with delay.

1) Structural Hazards: The insufficient hardware units cause problems in simultaneous operations of instructions. For example, if there is only one memory unit for instruction code storage and data storage, execution of concurrent results storage and instruction fetching are not possible.

Remedy:

- Sufficient hardware units are introduced in the pipeline stages.
- Whenever the need arises, we introduce a delay to avoid this hazard.

2) Data hazards: Whenever the data is required for a stage in a pipeline, and the data is not available at that moment, data hazard occurs.

Remedy:

- The Pre-fetching unit is employed to arrange the necessary data at that stage.
- An organizer organizes the independent and dependent data stages so that this conflict can be avoided.
- The Forwarding unit directly forwards the necessary data to that stage to evade this hazard.

3) Control Hazards: When the control signals effectively control the normal program with branching requirement by changing PC, Control Hazard occurs. The return address or the necessary data must be available for a branch to attend to its sub-routine. If it is not available, control hazard occurs.

Remedy: A two-bit branch prediction unit must be arranged to the branching to fulfill the return address location requirement.

B. A Cache miss:

Both in instruction cache or data cache, fast transactions are performed. The Possibility of missing data is called Cache miss. Before and after cache miss, stalls are introduced which are unavoidable and cause delay.

C. Exception:

In pipeline process, stage wise multiple operations are performed. Due to simultaneous execution of different operations at the same time, interrupts may occur. These are called as Exceptions. There are five types of exceptions.

1) Synchronous versus Asynchronous: In general, all the pipeline stages perform the same function for different instructions. In case it repeats with the same data and at the same location, with same memory allotment – that event is called a synchronous event.

When the event occurs haphazardly and abruptly due to malfunction of the hardware unit or external device or any other cause, it is called an Asynchronous event. To handle this problem easily, the event has to be accepted and completed after the present instruction.

2) User requested versus coerced: There are two types of situations the program has to attend. One is the user requested exception, the other one is coerced exception. In general, the user request is not considered as exception as it is predictable. But it comes under the exception as it is saved and stored, and needs to time allocation to attend it later.

Whereas, the coerced exception is caused due to the urgency of hardware breakup or any of that sort. These are very hard to correct because these are sudden events.

3) User maskable versus user nonmaskable

Some events are masked as per users' request. Some events are non maskable because they corrupt the hardware or software.

4) Within versus between instructions

The exception events may arise in the middle of the instruction or after/before the instruction. If the event occurs in the middle of the instruction, then it is called as 'within exception' and it is very hard to attend to them.

If the events occur between the instructions, they are called as 'between exceptions'. These are easy to attend when compared to the within exceptions because the instruction can be stopped and restarted. The within exceptions are always synchronous as the instruction itself triggers the exception.

5) Resume versus terminate

The other category of exception events is terminating event and Resume event. When the program is incessant in its operation after attending the interrupt, then it is called as a Resume exceptional event.

On the other side, if the program terminates for a while and restarts to connect with the original program, after attending the interrupt is called as Terminating exception. It is easy for the machine to attend such events.

Synchronous and coerced exceptions can be resumed in a more difficult way.

In general, almost all the advanced machines in the processor have the skill to attend to the exceptions in the

pipelining models, resulting to save the time and improve the performance of the core. The modern machines are capable of serving to virtual memory. For that, they have to conquer the exceptions.

V. RESULTS AND ANALYSIS:

There are five stages in the pipelining, consisting of IF, ID, EX, MEM, and WB stages. Each stage performs the part of the instruction operation.

A. Instruction Fetch (IF) stage:

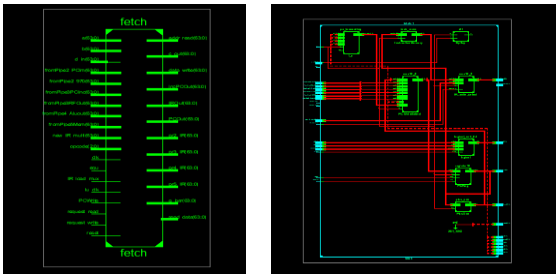

Fig. 6 RTL Schematic of Fetch Stage

The instruction fetch phase starts with the program counter showing the first instruction address. Based on that, first instruction opcode is fetched from the instruction memory. To speed up the operation, the Instruction cache is used instead of Instruction memory. Then, PC updates to the next instruction address.

TABLE 1. POWER CONSUMPTION OF FETCH STAGE

Frequency (MHz)	Time ns	Delay ns	Total Power mW
250	3.64	1.015	199
500	4.23	1.867	398
750	4.41	2.35	545
1000	5.121	3.68	786

B. Instruction Decode (ID) stage:

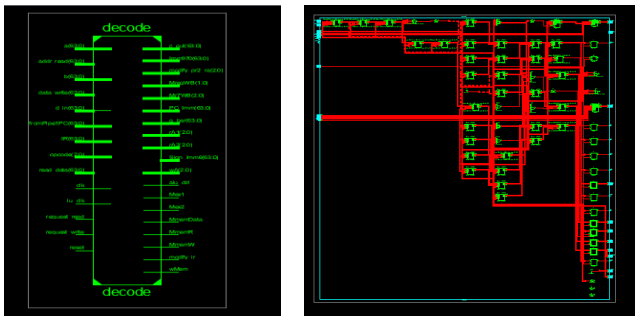

Fig. 7 RTL Schematics of Decode Stage

The Decode stage combined with Register Bank not only performs the decoding operation, for opcode, but also prepares the effective address calculations for different addressing modes for the next stage.

TABLE 2. POWER CONSUMPTION OF DECODE STAGE

Frequency (MHz)	Time ns	Delay ns	Total Power mW
250	3.26	0.924	181
500	4.48	1.12	362
750	5.47	1.58	513
1000	5.59	2.23	744

C. Execute(EXE) stage:

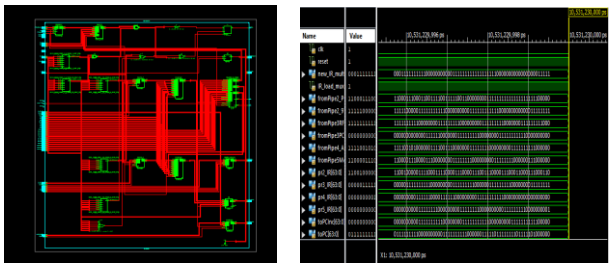

Fig. 8 RTL schematic of Executive Stage

In this stage apart from Arithmetic and logic operations shift/ rotate and multiplication/division operations are also being performed with shifter and multiplier devices.

TABLE 3. POWER CONSUMPTION OF EXECUTION STAGE

Frequency (MHz)	Time ns	Delay ns	Total Power mW
250	6.53	1.776	348
500	8.96	2.00	702
750	9.78	2.19	1008
1000	10.44	2.74	1382

D. Memory –Access (MEM) stage:

At this stage, load/store instructions are used for data storage and retrieval from the data memory. Here, for fast transactions, data cache is used. The results of the ALU are stored and the required data to perform ALU operation are supplied from this memory.

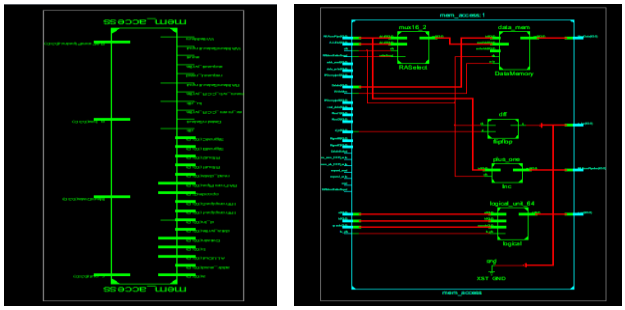

Fig. 9 RTL Schematic of Memory Access stage

TABLE 4. POWER CONSUMPTION OF MEMORY _ACCESS STAGE

Frequency (MHz)	Time ns	Delay ns	Total Power mW
250	5.41	1.148	225
500	7.43	1.67	563
750	8.27	2.28	741
1000	8.98	2.83	1098

E. 'Write_Back (WB) stage:

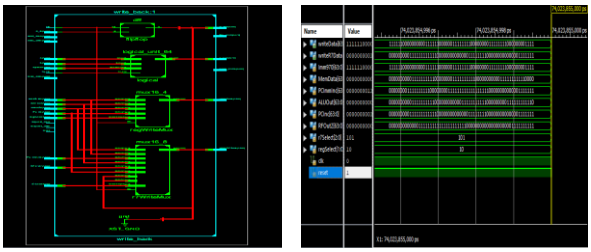

Fig. 10 Output of Write-back stage

This is the last stage of instruction pipeline where the results are written back to the register bank by using the feedback path. This is done because the register bank is associated with the decode stage, where the data is transferred easily to the integer unit.

TABLE 5. POWER CONSUMPTION OF WRITE_BACK STAGE

Frequency (MHz)	Time ns	Delay ns	Total Power mW
250	3.26	0.878	172
500	4.52	1.37	353
750	5.23	1.91	512
1000	5.89	2.22	803

VI. SOFTWARE TOOL

The Xilinx vivado ISE suit is used as a software tool for designing this five stage pipelining of an instruction. Verilog HDL coding is used to obtain the data of Power, Delay, operating frequency, etc. Here, the total operating frequency for an instruction to be executed takes 394.532 MHz with the path delay 5.74 ns. The total time taken to complete the task is approximately 22.13 ns.

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

   Minimum period: 3.224ns (Maximum Frequency: 394.532MHz)
   Minimum input arrival time before clock: 3.085ns
   Maximum output required time after clock: 2.201ns
   Maximum combinational path delay: 5.740ns
```

Fig. 11 Time and delay summary report

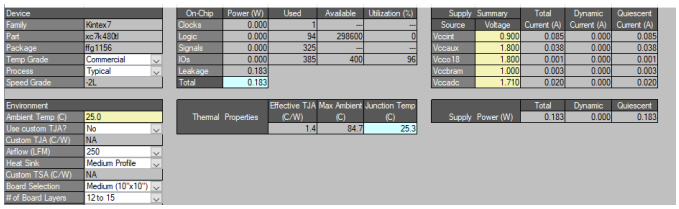

Fig 12. Voltage Summary Report

Figure 12 describes the kintex7 XC7K480T device with package FFG1156. Dynamic power, leakage power, thermal properties, on-chip utilization, and other voltage and current details are described.

```
Device utilization summary:
---------------------------

Selected Device : 7k480tlffg1156-2l


Slice Logic Utilization:
 Number of Slice Registers:              35  out of  597200     0%
 Number of Slice LUTs:                  328  out of  298600     0%
    Number used as Logic:               328  out of  298600     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:    143
   Number with an unused Flip Flop:     108  out of    143    75%
   Number with an unused LUT:            15  out of    143    10%
   Number of fully used LUT-FF pairs:    20  out of    143    13%
   Number of unique control sets:         4

IO Utilization:
 Number of IOs:                         901
 Number of bonded IOBs:                 385  out of    400    96%

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:                1  out of     32     3%
```

The device utilization summary describes the circuit element distribution, where the total slices utilized are 328, i.e., minor utilization of total (298600) slices.

A 3D graphic description narrates the relationship of various parameters in pipelining design.

The auxiliary supply voltage used between 0-2 V and frequency 0-40MHz. Here, the load is a capacitive type and it varies from 0-8 pf.

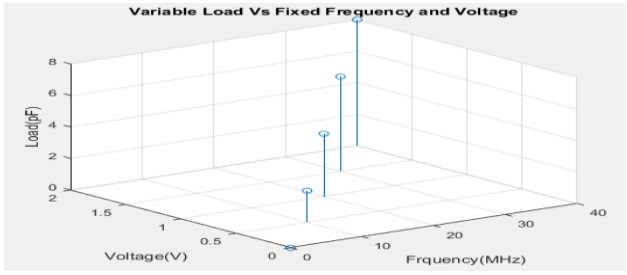

Fig. 13 Variable load vs. fixed frequency and voltage.

VII. COMPARITIVE ANALYSIS.

TABLE 6 .POWER COMPARISON OF VARIOUS PIPELINE MODELS

Parameters [12]	Process Technology	LUTs	Frequency (MHz)	Power (W)
Spartan3: XC3S1500L-4FG676	90nm	417	98.09	0.144
Virtex5: XC5VFX30T – 3FF665	65 nm	300	321.048	0.777
Virtex6: XC6VLX75T – 3FF784	40 nm	307	401.881	1.440
Virtex6Low Power: XC6VLX75TL–IL-FF784	40 nm	307	335.233	0.920
Our Model (Kintex 7)	28 nm	328	394.532	1.125

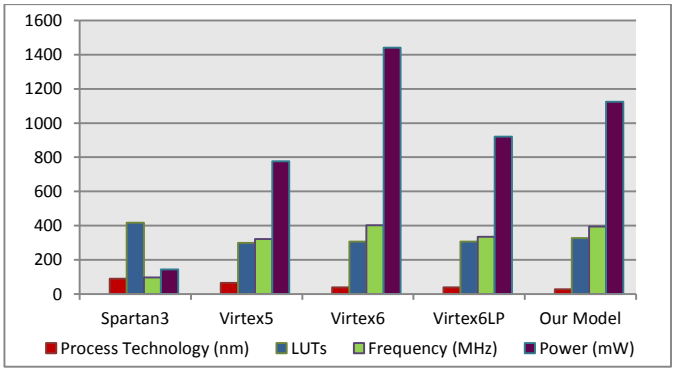

Fig. 14 Performance Comparison of various process cores

In the comparative study, process technology, LUTs, Frequencies, and Power are compared with existing pipeline processor cores. Our model got the total power of 1.125 watts, when compared to the nearest counterpart; it is getting reduced by 28%.

TABLE 7. FREQUENCY COMPARISON OF VARIOUS PIPELINE MODELS

Parameters	Our Model	Single core processor [13]	Low Power MIPS [14]	MIPS Core [15]	Tiny CPU [16]
Max. Frequency	394.532 MHz	277.90 MHz	205.7 MHz	95.5 MHz	89 MHz
LUT	328	1168	1890	2340	336

The other parameters like speed and number of slices utilized are also compared with existing pipeline models. The LUT utilization is 2.4% less and frequency better enhanced by 42% when compared to the nearest counterpart.

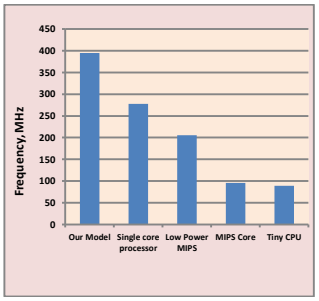

Fig. 15 Frequency comparison

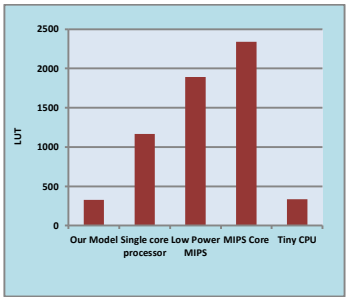

Fig. 16 LUTs comparison

VIII. CONCLUSION

In this research, 5-stage 64-bit MIPS RISC processor-based pipeline model is implemented and the desired parameters are measured. Kintex7 family XC7K480T device processor is utilized for this purpose. The features are described and parameters are related through a 3D graph.

The Xilinx vivado platform with verilog HDL coding is used to implement the results. The main hazards and exceptions of pipeline models are described with remedies.

In a comparison study mainly key features like the power and speed of the pipeline models are compared. The superiority of our model with the optimal balance of these parameters are maintained and proved.

REFERENCES

[1] Divya.G, Bhagya lakshmi, P and L. Srinivas, "Design and implementation of High Speed Pipelined DDR SDRAM memory Controller," International Journal of Engineering Research and Applications. National Conference on developments, advances & Trends in Engineering Sciences (NCDATES), January, 2015.

[2] F. Rashidah, Olanrewaju, E. Fawwaz, Fajingbesu, S.B. Junaid, Farhat Anwar and Bisma Rasool Pampori, "Design and Implementation of a 5-stage pipelining architecture simulator for RISC-16 instruction set," Indian Journal of Science and Technology, vol. 10, no. 3, January, 2017.

[3] Raja Krishnamoorthy and S. Sarvanan, "A novel flip-flop based error free, area efficient and low power pipeline architecture for finite impulse recursive system," Cluster Computing, Springer, February, 2018.

[4] Saranya Krishnamurthy, Ramani Kannan, Emran Azwan Yahya, Kishore Bingi, "Design of FIR Filter using Novel Pipelined Bypass Multiplier," Third International Symposium on Robotics and Manufacturing Automation (ROMA), IEEE, 2017.

[5] M. Zulkifli, Y.P.P. Yudhanto, N.A. Soetharyo and T. Adiono, "Reduced Stall MIPS Architecture using Pre-Fetching Accelerator," International Conference on Electrical Engineering and Informatics. Malaysia, IEEE, August, 2009.

[6] Macha Ashok Kumar and T. Krishna Moorthy, "Design and Analysis of 64-Bit MIPS Processor," International Journal of Electronics, Electrical and Computational System (IJEECS), vol. 7, no. 4, April, 2018

[7] Mangalwedhe, S. Roopa Kulkarni, Y. Kulkarni, "Low Power Implementation of 32-Bit RISC Processor with Pipelining

Proceeding of the Second International Conference on Microelectronics, Computing & Communication Systems, pp 307-320, November, 2017

[8] Husainali S. Bhimani, Hitesh N. Patel and Abhishek A. 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) – ISSN : 2249-0868, May, 2016.

[9] M.R. Rakesh, "RISC Processor Design in VLSI Technology Using the Pieline Technique," International Journal of Innovative Research in Electrical, Electronics, Instrumentation and control Engineering, vol. 2, no. 4, April, 2014.

[10] M. Indu and M. Arun Kumar, "Design of Low Power Pipelined RISC Processor," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 2, no. 8, August, 2013.

[11] Charu Sharma and Gurpreet Singh Saini, "Design and Analysis of High performance RISC Processor using Hyperpipelining Technique," International Journal of Advances in Scientific Research and Engineering, vol.3, no. 5, June, 2017.

[12] Narender Kumar and Munish Rattan, "Implementation of embedded RISC Processor with Dynamic Power Management for Low Power embedded system on SOC," Proceedings of 2015 RAECS UIET conference, IEEE, December, 2015.

[13] Nishant Kumar and Ekta aggrawal, "General purpose Six-Stage Pipelined Processor," International Journal of Scientific & Engineering Research, vol 4, no. 9, September, 2013.

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

[15] P. Gautham, R. Parthasarathy and Karthi Balasubramanian, "Low Power Pipelined MIPS Processor Design," 2009 ISIC proceedings, 2009.

[16] 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, 2008