[1]Anu Samanta

[2] Subhadip Goswami

[3]Kumaresh Pal

[4]Madhu Sudan das

A Comprehensive Survey and Comparison on Pipelined RISC System Architectures

Journal of Electrical Systems

Abstract: In the modern world, processors are crucial when multiple components are housed on a single chip. Here, technology is developing quickly, and as a result, a variety of risks related to processors could arise and cause degradations in power, area, and speed. Thus, the evolution of these MIPS-RISC CPUs has improved performance. The establishment of the needed characteristics and quantities for MIPS (Micro Processor without Interlocked Pipeline Stages) is the primary topic of this study. Verilog HDL is utilized for the CPU design, Xilinx is used for synthesizing, and several simulators are used for simulation. A comparison of the outcomes for several parameters is made, which aids the intern in choosing the appropriate challenge.

Keywords: MIPS, RISC, VHDL, Xilinx, Synthesizing, Simulation.

I. INTRODUCTION

The central processing unit (CPU) is a key component of computers and is responsible for many of its characteristics, including speed, storage capacity, and other capabilities. Reduced instruction set computers, or RISCs, are microprocessors that carry out short, comparable commands in roughly the same amount of time. The goal is to decrease instruction complexity, which does away with costs, cycle times, and operational power. Despite being existed since the 1970s, 16-bit RISC has not lived up to expectations and has presented a great deal of technological challenges. This was the driving force for the creation of pipelining and the 32- and 64-bit RISC CPUs. New features and designs in processor design have led to new resolutions in RISC (Reduced Instruction Set Computer) architectures. Previously, the performance of compilers was simplified and improved by using CISC (Complex Instruction Set Computer) processors, but with several drawbacks including small and slow storage [1], a wide range of addressing modes, and longer execution times. Subsequently, MIPS RISC processors became popular, offering single-cycle execution, register-register operations, load/store functions, comprehensible addressing modes, and a fixed instruction set length. Additionally, this layout made the pipeline construction fairly straightforward, admitting to the concurrent execution of instructions.

With a wide range of potential applications, vision chips combine a vision processor and an imager onto a single chip to enable real-time intelligent image processing and capture. A programmable, adaptable and a dual-issue micro architecture-based vision processor is suggested by Qian Luo et al. [35] in this brief. A non-maximum suppression (NMS) block, a flexible memory access network, and a reconfigurable vector unit make up the processor. It is capable of effectively implementing both conventional computer vision (CV) methods and deep neural network (DNN) algorithms. With variable vector width, the vector unit does parallel operations with single-instruction multiple-vector (SIMV). Under varying vector widths, the flexible memory access network allows numerous vector operations in an adaptable manner. The vector unit's four-MAC processing element (PE) is intended to boost processing capacity and data reuse rate. The detection networks' object location processing can be accelerated by the NMS block. The 28nm technique is used to create the chip. The maximum clock frequency, peak energy efficiency and peak performance are 600MHz, 2.03TOPS/W, and 1.2TOPS respectively, according to the experimental results. On the ImageNet dataset, the Mobilenet-V1 processing achieves an accuracy of 87.15% (top-5) and a throughput of 404 fps under 256x 224 image size.

The symmetric overloaded reduced instruction set computer architecture (SORA) and its open instruction set architecture described by Jungwirth P. et al [36]. An article outlining the symmetric overloaded minimal instruction set computer architecture (SOMA) was provided by A&M last year. The SOMA design concentrated on an overloaded instruction set, represented in BNF notation as R1 = R2 + [R3 | const], where integer and register-to-register instructions are symmetric and overloaded. There is an integer class instruction for every register instruction. For branch instructions, the SOMA architecture uses an if{} statement, akin to one in C. The goal of the SORA paper is to develop a reduced instruction set computer (RISC) architecture using the concepts from the SOMA paper with some enhancements. K. Booth created assembly language back in 1947. RISC-V, ARM, IBM 360, Intel 4004, 6502, 8085, x86, 68k, MIPS, etc. all make use of an assembly language that is text-

[1] Anu Samanta, Asst. Prof. ECE Dept. Brainware University
[2] Subhadip Goswami, Research Scholar, NIT Arunachal Pradesh
[3] Kumaresh Pal, Research scholar, NIT Jamshedpur
[4] Madhu Sudan Das, Senior Research Fellow, ISM Dhanbad

based and usually has four fields. Assembly language hasn't changed in 76 years. It is necessary to reinvent assembly language for the twenty-first century. These days, assembly languages having high level language features are referred to as high level assembly or meta-assembly. Circa 1964, the IBM 360 assembly language included if statements and loops similar to those found in contemporary high-level languages. Around 1975, the assembly language of the Xerox Meta-Symbol Sigma 5-9 Computers included high level language characteristics. Basic operations are shared by all RISC architectures. Therefore, in future the SORA C-like assembly statements to develop a proto-assembly language that works with several RISC architecture families can be used.

Kulshreshtha A. et al [37] examined the independent instruction sets and behavioral models of 16- and 32-bit RISC processors. The Harvard architecture-based 16-bit RISC CPU is a non-pipelined CPU with independent data and instruction memory. The pipelined 32-bit RISC processor takes its implementation techniques from the MIPS architecture. General Purpose Registers (GPRs) and Flag registers (Carry, Zero, etc.) are among the processors. Due of the higher power consumption and lengthy execution delays associated with arithmetic and logical operations, the model under discussion will simulate the enhanced multiplier method and attempt to optimize the data flow. The purpose of the paper is to compare the models according to their instruction sets and performance metrics, such speedup, power dissipation, etc. After the different models were created and run, they were combined with power analysis using XILINX ISE Design suit 14.7 to create a top-level module.

Prabhakaran S. et al [38] created a 32-bit pipelined processor based on the open-source RV32I Version 2.0 RISC-V ISA that has several clock domains. A processor known as a RISC uses less hardware than a CISC in order to minimize the complexity of the instruction set and accelerate the execution time per instruction. In addition, they built this processor with five levels of pipelining, which allows instructions to be executed in parallel. The required block diagrams help to provide a thorough description of each procedure. To guarantee that variable delays, such as clock skew and metastability, are avoided within the stage pipeline registers, multiple clock domains using two clock sources are employed. This processor was designed and synthesized in Verilog HDL using Quartus Prime. This design was validated using ModelSim, and every instruction has been carefully examined. Additionally, the processor is used to calculate device utilization on the "ALTERA Cyclone 10 LP" board.

Avinash N J et al [39] created a 32-bit RISC-V processor with five stages that is flexible and pipelined using the Verilog system and its dynamic thermal management technique. The MIPS instruction set architecture (ISA) serves as the foundation for this design. The pipeline's stages are memory access, write back, execute, retrieve, and decode instructions. In contrast to a CISC, a RISC computer executes instructions in a single clock cycle and uses a defined instruction format based on opcode for all instructions. Because of this, RISC is better suited for low-power and simpler workloads. According to the literature, the suggested design achieves low power consumption and offers notable increases in simulation speeds thanks to the flexible pipeline technique. This paper's main goal is to simulate and synthesize the design using fundamental ALU operations and produce the required results.

Usually, internal and external events are used to drive embedded systems, which are implemented by (static priority) interrupts. permitting for interrupt nesting, or permitting a higher priority interrupt to override the execution of an interrupt handler that is presently executing, can enhance response time. Per Lindgren et al [40] examined interrupt nesting for the RISC-V architecture and suggest ENEST, a stacking strategy that minimizes interference and blocking while having a predictable overhead. The ESP32-C3 single core MCU from today is used to validate the claims made for the proposed mechanism. Further static scheduling analysis of ENEST-based applications is made possible by the quantification of blocking and interference that our experimental results yield.

Soundari D.V. et al [41] designed the RISC-V 32b processor's base. Numerous other commands, such as post-upgrade and save instructions, hardware loops and extra ALU instructions are supported by it. RISC processors have many applications based on their performance and power consumption. arranged to use design and solution approaches to create low power RISC CPUs. The right clock method can also be used to improve the pace. Decoding of memory, rewriting, and execution. For the middle stage, a single-sided clock signal is employed. In a list of 64-bit floating point numbers, single associated (PS), a new data type added to MIPS V, holds two 32-bit single-point floating point numbers. For arithmetic operations the existing floating-point statement versions have been included, appropriateness and restrictions to work with this sort of data utilizing the SIMD approach. There are now additional download, transfer, and PS data transfer instructions available. To bypass the floating-point SIMD with the available typefaces, this is the first command.

The Coarse-Grained Reconfigurable Architecture, or CGRA, is a popular accelerator that allows flexibility to fulfill a variety of application requirements while also reducing energy consumption and improving performance. Even with the aforementioned benefits, CGRAs often have a large number of processing components, which means area overhead that may prevent it from being integrated into systems with hard area constraints, including embedded systems and mobile devices. In order to address that, Francisco Carlos Silva Junior et al [42] assessed a CGRA known as ATHENA (A Thin reconfigurable Architecture) for systems with hard area constraints. The thinness notion is based on a CGRA that requires a significant reduction in processing elements compared to CGRAs identified in existing literature. ATHENA is dynamically mapped and connected to a superscalar processor. To assess the many area, energy, and performance tradeoffs that ATHENA and the superscalar processor can provide, a design space exploration is conducted. The results demonstrate that ATHENA was able to save up to 32% of energy and achieve speeds of up to 2.43x even with fewer processing elements. ATHENA offers competitive performance and is up to 4 times smaller than comparable dynamically mapped CGRAs of the state of the art.

The growth of Internet of Things applications has spurred the creation of embedded SoCs that are energy-efficient and capable of utilizing scarce energy resources. SoCs are made up mostly of very simple general-purpose processor cores, and one of the main challenges in SoC design is to maximize power consumption, performance, and size. Thus, the power, performance, and area of several 32-bit RISC-V cores with various microarchitectures were objectively examined by Kadomoto J et al [43] here. Every processor underwent simulation assessments with several pipeline topologies, both with and without a multiplier and divider. In addition to the predicted power consumption and area based on logic synthesis and place-and-route using several CMOS process technologies, the benchmark execution performance of the processors in a register transfer level (RTL) design has been shown. They offered a succinct set of recommendations for choosing microarchitectures for energy-efficient embedded SoCs in light of the findings.

Functional verification is a crucial phase in the VLSI design cycle that is required to find errors in the hardware description. Because digital designs are now larger and more complicated, functional verification is now essential. According to M.S.S.D.Amruth et al [44] surveys, the design phase only needs thirty percent of the project's time, whereas verification can take up to seventy percent. Therefore, in order to prevent delays in time to market, it is imperative to design an effective verification framework. The creation of a simulation-based verification platform for the RISC-V processor is the main objective of this research study. The paper outlines the architecture of a five-stage pipelined MIPS processor with 32-bit instruction capability that uses UVM. The design has various components, namely Data Memory, Registers, Instruction Memory, and ALU. Furthermore, an automated hazard detection and forwarding unit has been put in place to identify potential data hazards during verification.

Since the implementation and operation of "algorithms" are at the core of the three courses we concentrate on (Digital System Design, Computer Architecture/Organization, and Embedded System Design), these subjects are focused by Jamieson P. et al [45]. From this foundation, we have developed instruments to produce code samples and tests, as well as methods for virtualizing labs and practical exercises. Specifically, they have developed tools for Python that let teachers customize code and problems, generate these codes and issues (as text files or embedded in word documents), and send these documents to students via email. In order to mitigate some of the difficulties associated with live and proctored exams, this gives students the ability to design problems and code samples that are distinct from those of their peers and can be evaluated individually.

According to Ali, Z et al [46] research, the RISC beat the CISC because it was more energy-efficient, had superior thermal characteristics, and was more power-efficient. That may be the cause of Apple's migration from CISC to RISC architecture, and they will now concentrate on building their ecosystem around it. In the Internet of Things, ARM can lower the total cost of the device by increasing the battery timing of these devices and removing the requirement for a cooling fan. Using ARM chips instead of x86 chipsets can cut the energy usage of data centers and servers by 3x to 4x, a step towards green computing. The survey also showed that, while designing or selecting an architecture in the current era, performance and power consumption rank among the most crucial considerations. People are looking for laptops and mobile devices with longer-lasting batteries and enough processing power to run demanding apps. Demands can be met by ARM design for a very long time.

The design of the RISC processor and its implementation from RTL to GDSII are discussed by Aradhya, H. et al [47]. Verilog (RTL file) and the test bench associated with that Verilog file are used to verify the Harvard structure of the RISC processor. The tool Cadence NC Launch is used to simulate code. Gate level net list was obtained from the Cadence Genus tool and later checked along with the Verilog file, sdc, and.lib files. The front-end portion of the design will continue until this point in the gate level net list creation process. Verified gate level net list files and sdc constraints files are the Genus tool's outputs. Results of pre-layout simulations are performed for timing, power, and area. Cadence Innovus was the tool used for floor planning, power planning, and routing in backend design. Post layout simulations for power, time, and area are also carried out here. The 180nm technology cadence tool is used for all of these procedures. Using the Cadence Innovus tool, the Harvard Structure RISC Processor is successfully realized physically. Once pre-clock tree synthesis, post-clock tree synthesis, and post-routing of the circuit are completed, optimal timing 9.236ps, power 0.53155682W, and area 170677.7584μm2 are attained.

The bridge between hardware and software, an instruction set is a collection of commands that a central processing unit (CPU) uses to compute and manage a computer system. CISC and RISC are the two common instruction sets. In order to increase the effectiveness of executing instructions, pipeline technology is frequently employed in instruction set processor design. The differences between CISC and RISC pipeline implementation are discussed by He, Y. et al [48] along with an overview of basic pipelining, two advanced pipelining techniques—superscalar and super pipelining—and multiple pipelining techniques using CISC and RISC architecture processors, such as ARM, RISC-V, Longarch, and X86.

II. MIPS ARCHITECTURE:

MIPS stands for Microprocessor without Interlocked Pipelined Stages, a RISC processor developed in the 1980s by MIPS Computer Systems Inc. In the embedded sector, MIPS is now the performance leader. These fundamental concepts can be found in everyday items like play stations, Windows CE, digital cameras, and more. But, because MIPS is based on instructions with set lengths and simplified architecture, by the 1990s every RISC processor design was created using this architecture. However, the load and store instructions, which are accessed from memory independently, are the sole limitations. [2]. MIPS is mostly used for its pipelining technique, which uses parallelism to execute instructions in phases while overlapping the execution of several instructions.

These days, all processors use this crucial implementation strategy. The pipelining structure of MIPS processors consists of five stages: write back (WB), data memory (MEM), execution (EXE), instruction fetch (IF), and instruction decode (ID). Each pipeline stage is in charge of completing instructions on its own and registers are used to connect them.

The architecture of the 32-bit MIPS-based RISC CPU is shown in Fig. 1 below. It consists of the Control unit (CU), register file, Datapath, code and data memory, and Datapath.

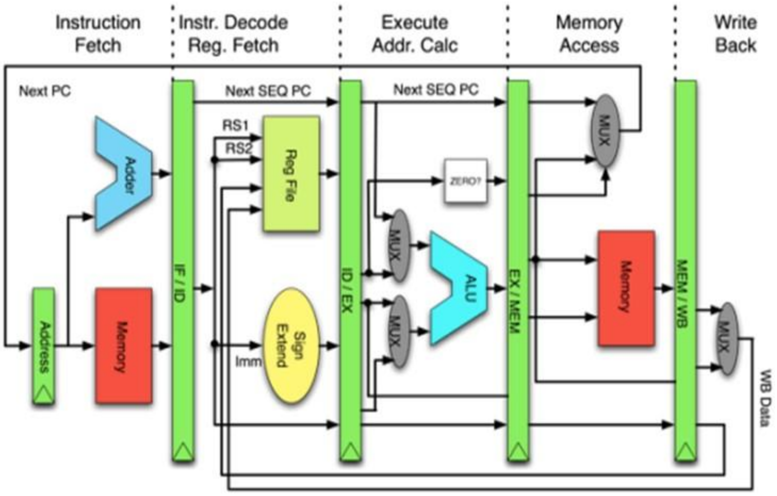

Fig.1: 32-bit MIPS processor

A. INSTRUCTION MEMORY

The instructions that the CPU executes are stored in the instruction memory.

B. Data Memory: The data memory is accessed by load and store instructions. The data memory has a width of 32 by 256 bytes.

C. Register File: There are 32-bit general purpose registers in the register file.

D. Datapath: The pipelined structure of the datapath consists of five stages. The instructions are executed in five stages: fetch from memory, decode, execute, access memory, and write back stage. Pipeline registers are used to transfer the outcome of the first stage to the subsequent stages and are positioned in between each stage.

Instruction Set:

Eliminating hardware interlocks that arise between CPU pipelining stages is the primary goal of the MIPS design. Similar to other RISC processors, MIPS offers simple instructions and a load/store architecture that makes use of general-purpose registers [4]. Other instructions can only access the values stored in registers. Generally, there are three sorts of MIPS instructions:

1. Memory Reference Instructions
2. Arithmetic-Logic Instructions
3. Branch Instructions

Additionally, the instructions come in three different formats:

1. R-Type instruction
2. I-Type instruction
3. J-Type instruction

The instruction set for MIPS is shown in figure 2.

R (Register) Format:

Opcode (6)	Rs (5)	Rt (5)	Rd (5)	Shamt (5)	Funct (6)

Most arithmetic and logic instructions (except ‘immediate’)

I (Immediate) Format:

Opcode (6)	Rs (5)	Rt (5)	16-bit Immediate value (16)

Data Transfer, Immediate, and Cond. Branch instructions

J (Jump) Format:

Opcode (6)	26-bit word address (26)

Unconditional Jump instructions

Fig.2: MIPS Instruction Format

The MIPS instruction forms are shown in Fig. 2. The instructions have a length of 32 bits. The type of instruction format is selected using the 6-bit Opcode field.

III. LITERATURE SURVEY

In order to identify the delay caused by longer pathways, Kumar D. et al. [5] proposed a five-stage MIPS pipelining method that makes use of several processor technologies. The primary idea is to reduce the critical path by improving hardware efficiency within the circuit, which boosts performance. To do this, pipelines are arranged so that high frequency clocks must arrive at pipelining. The resulting architecture was then synthesized with the intention of targeting different processing device technologies, including Virtex4, Virtex5, and Virtex 6. Thus, the primary crucial delay path—which was noted in the execution unit—is retrieved through the use of this synthesizing method. The execution unit's maximum frequency is increased from 51MHz to 152MHz at 45nm to 40nm technology, while the minimum longest path delay is 6.57ns at 40nm technology and the maximum longest path is 41.405ns at 90nm technology. At every level of pipelining, the clock frequency is high at Vertex-6. The comparison of results for other technologies only indicates that the CPU operates at 178 MHz frequency, or 49.7% more efficient than those other technologies, at 40nm technology.

V. Prasanth and others [6] presented a 28nm technology, five step pipelined MIPS 32 bit CPU. Here, there is less switching between stages as a result of stage registers that cause flip-flops to be toggled in a design. This study describes a procedure for enhancing the speed and performance of a Harvard architecture processor, which leads to a decrease in device power consumption. Thus, a power-saving method is used to lessen this.

The most often used method for reducing dynamic power dissipation is clock gating. When it is not necessary for the flip-flops to update their state values, this technique temporarily turns off their clock signal. If there is valid data to be saved, the control signal activates the clock signals of the flip-flops. Thus, a novel method of exploiting the FPGA device's RTL clock gating is presented in this research.

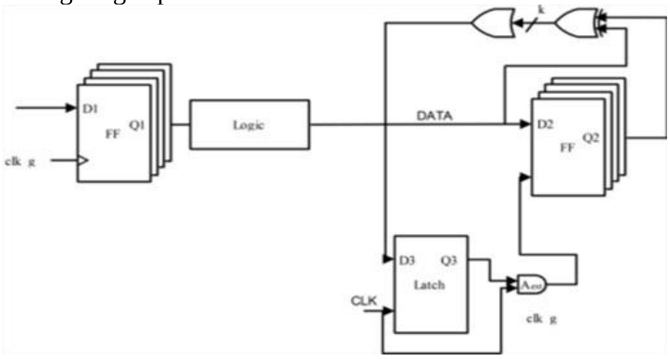

Fig.3: Clock Gating Technique

Therefore, 73% of dynamic power is decreased in inactive models by effective clock gating, since the dynamic power is 0.031 W with clock gating and 0.152 W without. The Kintex-7 platform board was utilized in this work to analyze design functioning under various conditions.

Omkar, et al. [8] presented the idea of using a single precision floating point multiplier at the ALU, which is located in the execution block of a 32-bit RISC processor, which is made up of blocks like the write back unit, instruction fetch, instruction decode, and execute block. This one precision demonstrates the increase in execution speed and accuracy. This study uses the power gating technique to reduce static power. When no activity is required, the power sources of the logic blocks in the design are turned off using the power gating technique, which makes use of a specialized power management unit.

Kaur Harpreet, et al. [9] suggested a MIPS processor with a five-stage pipelining architecture that lowers dynamic power consumption and boosts the processing speed of instructions per second. By including NOP instruction, the risks associated with pipelining are also removed. Since there is no practical work for this NOP command, power is squandered. Therefore, the dual write port register file in this CPU is employed to facilitate dual right-back operations, which lessened the requirement for NOP in pipelining and further lowered power consumption. Hazard detection units in pipelines prevent pointless transitions, thereby reducing stalls.

Indira, P., et al. [10] presented a MIPS 6 stage pipelined 64-bit RISC processor for a structure free of hazards by combining the capabilities of the branch and jump prediction unit, hazard unit, forward unit, and pre-fetch unit. The use of dual edge type triggering flip-flop designed registers reduces power consumption in the pipeline. These flip-flops are low power consuming for enhancing the system performance further we use low power unit to control wastage of power. Other devices used in this design include cache, balancing pipeline stages, etc., which result in low power consumption and high speed performance. Due to its reputation for delay reliability, DDR4 SDRAM is utilized with the most modern controller versions.

It is demonstrated that in comparison to other counterparts, our suggested design uses the least Total Power (3.6 µW) and achieves the highest frequency (255.88 MHz) with a significant area utilization; of the 63400 LUTs, only 515 are used, demonstrating that less area is used for design. All things considered, the various actions are done to maximize improving the system's performance. The MATLAB tool is only used for graphical help when plotting graphs; Xilinx14.7 is utilized for simulation.

A six-stage pipeline architecture for a 32-bit RISC processor was proposed by P. Indira et al. [12] in order to achieve a low power and high-performance structure. Configurable logic blocks, input/output blocks, RAM blocks with digital clock managers, and these blocks enable routing by linking various sources. These functional blocks are the main components used by processors. The author conducted a comparison between the suggested and traditional technology. Here, 90nm CPU technology is employed, with the primary goals of the article being to minimize power consumption, maximize execution speed, and minimize area utilization. For additional power reduction, deeper pipelining and power gating are employed. Appropriate steps are taken to lessen the risks. The graphical depiction is done using a MATLAB program and the Spartan 3E series simulator. The suggested model's overall power usage is 0.129W, which is significantly less than the conventional model's 11.180ns, and it has a 22% faster speed than the GPPM model. Different frequencies are employed to quantify the total, leakage, and dynamic powers.

A three-stage pipeline architecture for a 32-bit MIPS RISC processor was presented by P. Indira et al. [13] in order to address design dangers and enhance performance factors like speed and power consumption. The well-designed DDR3 SDRAM-powered Virtex-7 series of processors guarantees high-speed performance while employing 28nm processor technology. The author employs Xilinx to demonstrate the effectiveness of her suggested technique by comparing various metrics with graphs.14.3 uses MATLAB for graphical simulation and Verilog HDL for modelling.

IV. Review Results

Numerous studies that have been studied up to this point show that a variety of suggested strategies are applied to MIPS RISC processors, which may have pipeline topologies that vary from 32 to 64 bits. These designs employ a variety of processing technologies, including 28nm, 40nm, 45nm, 60nm, and 90nm, along with distinct modelling and synthesis methodologies through the usage of devices like Spartan-3, spatran-5, Virtex-4, Virtex-5, Virtex-6, Virtex-6LP, and Vitex-7.

Let us now examine the outcomes of these various strategies with respect to the development of MIPS, by contrasting all parameter results from various suggestions, as indicated in table 1 below.

Reference No.	Delay(ns)	Frequency (MHz)	Total Power(W)
[15]	14.348	179.092	0.829
[14]	1.05	100.145	0.0004
[13]	1.143	420.028	0.023
[12]	11.180	285.583	0.129
[11]	-	401.881	0.363
[10]	1	255.88	3.60µ
[9]	-	193.8	0.340
[8]	0.562	-	0.210
[7]	-	-	0.177
[6]	-	-	0.031
[5]	6.57	178	-

Table.1 Comparison of parameters

Table 1 compares various papers, denoted as "Reference No.," and displays power consumption, delay, and frequency in columns 4, 2, and 3, accordingly. Table 1 shows a power reduction from 0.829W to 0.031W. A maximum dynamic power reduction of 90% was attained for the Recent MIPS CPU.

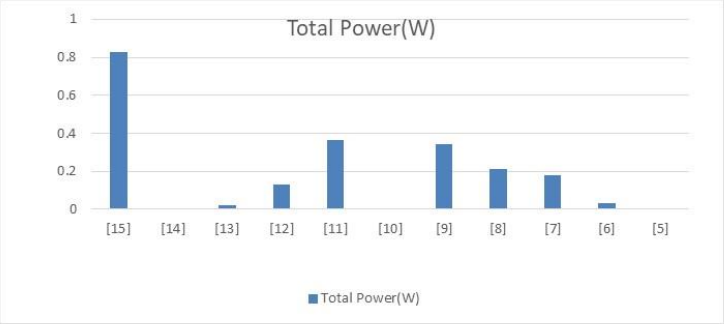

Fig.4: Power Consumption of different Processors

Figure 4 displays a comparison graph of power usage across CPUs that have and do not have clock gating mechanisms. In comparison to other CPUs, the 32 bit MIPS processor with clock gating approach and low power edge triggered flipflop uses 0.031 W less power.

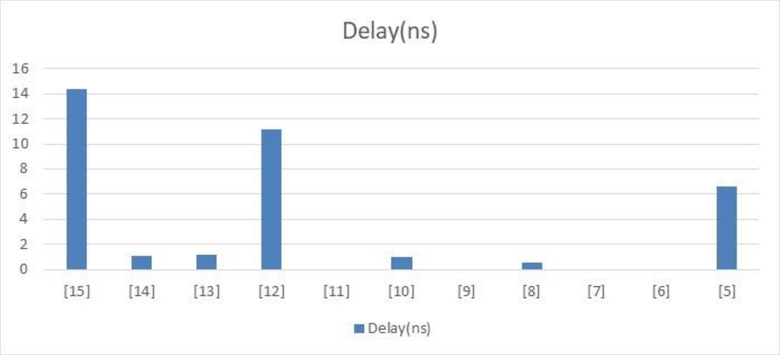

Fig.5: Delay of Various Processors

Figure 5 displays the different processors' delays. In comparison to other designs, the 32-bit MIPS RISC processor with parallel architecture in the execution block exhibits a lower delay of 0.562 ns.

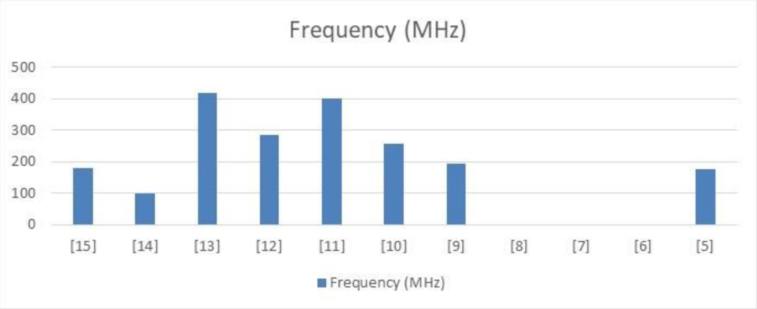

Fig.6: Frequency of Various Processors

Figure 6 displays the different processors' frequency. In comparison to other designs, the three-stage pipeline architecture for a 32-bit MIPS RISC processor in the execution block exhibits a highest frequency of 420.028 MHz.

V. CONCLUSION

A thorough analysis of the performance of MIPS RISC processors is presented in this work. Since the 1981 proposal for MIPS processors, numerous studies on the architecture have been conducted. The use of clock gating and pipelining techniques in MIPS RISC processor architectures has been demonstrated in this work to result in considerable performance gains.

REFERENCES

[1] K Mohit N. Popiwala, Saraswathi, “Implementation of a 32- bit MIPS based RISC processor using cadence”, Proc.of IEEE International Conference on Advanced Communication control and Computing Technologies (ICACCCT),2014,pp.979-983.

[2] Rupali Balpande, Rashmi Keote, “Design of FPGA based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor”, Proc.of International Conference on Communication Systems and Network Technologies, 2011 IEEE ,ISBN:978-0- 7695-4437-3/11,pp.409-413.

[3] Nan Wang, Ranjith Kumar and Rajath M. Basavaraj, “Implementation and Performance Evaluation of Pipelining Mechanism in 32-bit MIPS Architecture”, proc. of 7th International Workshop on Computer Science and Engineering (WCSE),2017doi: 10.18178/wcse.2017.06.069 WCSE 2017, ISBN: 978-981-11-3671-9,pp.405-409.

[4] Sagar P. Ritpurkar1, Mangesh N. Thakare, Girish D. Korde, “ Review on 32- bit MIPS RISC Processor using VHDL”, IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)(ICEAT),2014, e- ISSN: 2278-1676, p-ISSN: 2320-3331 ,pp. 46-50.

[5] Dr.Dilip Kumar and Kiratpal , “Design of High performance MIPS-32 Pipeline Processor”,Conference Paper · April 2012 DOI: 10.13140/RG.2.1.4883.6567.

[6] V.Prasanth, V.Sailaja, P.Sunitha, B.Vasantha Lakshmi, “ Design and implementation of low power 5 stage Pipelined 32 bits MIPS Processor using 28nm Technology”, International Journal of Innovative Technology and Exploring Engineering (IJITEE) March, 20, Volume-8, Issue-4S2,ISSN: 2278-3075, pp.502-507.

[7] G.Jhansi, “ Design of MIPS based 64-bit RISC Processor”, International Journal of Engineering and Advanced Technology (IJEAT), October 2017,Volume-7 ,Issue-1, ISSN: 2249 – 8958,pp.48-51.

[8] Omkar A. Shastri, “Floating Point Parallel Processing Multiplier Based RISC (MIPS) Processor”, International Journal of Advanced Research in Computer and Communication Engineering, July 2016, Vol. 5, Issue 7,ISO 3297:2007 Certified, ISSN: 2278 – 1021,pp.477-481.

[9] Harpreet Kaur, Jaspal Singh, “MIPS Processor With Reduced Dynamic Power” IJCST , Jan - March 2013, Vol. 4, Issue 1, ISSN : 0976-8491 (Online) | ISSN : 2229- 4333,pp.559-562.

[10] P. Indira, M. Kamaraju, “Design and Implementation of 6-Stage 64-bit MIPS Pipelined Architecture”, International Journal of Engineering and Advanced Technology (IJEAT) , August 2019,ISSN: 2249 – 8958, Volume-8, Issue-6S2,pp.790- 796.

[11] Sneha Mangalwedhe, Roopa Kulkarni, S.Y.Kulkarni, “Low Power implementation of 32-bit RISC Processor with pipelining”, Proc.of Second International Conference on Microelectronics, Computing & Communication Systems (MCCS), November 2017,springer . pp 307-320.

[12] P. Indira, M. Kamaraju, Ved Vyas Dwivedi, “DESIGN AND ANALYSIS OF A 32-BIT PIPELINED MIPS RISC PROCESSOR”, International Journal of VLSI design & Communication Systems (VLSICS), October 2019, Vol 10, Issue 5, ISBN: 10.5121/vlsic.2019.10501, pp.1-18.

[13] P. Indira, M. Kamaraju, Ved Vyas Dwivedi, “ Verilog Implementation of a MIPS RISC 32-bit Pipelined Processor Architecture”, IOSR Journal of Electronics and Communication Engineering (IOSR- JECE) e-ISSN: 2278-2834,p- ISSN: 2278- 8735.Volume 14, Issue 1, Ver. I (Jan.-Feb. 2019), pp. 31-40.

[14] B. Kiran Kumar, Anilkumar B, “Functional Verification and the Physical Design of MIPS Architecture”, International Journal of Recent Technology and Engineering (IJRTE), July 2019, Volume-8 Issue-2,ISSN: 2277-3878, pp.3151-3154.

[15] Galani Tina,R.D.Daruwala,“Performance Improvement of MIPS Architecture by Adding New Features”, International Journal of Advanced Research in Computer Science and Software Engineering, February 2013 , Volume 3, Issue 2, ISSN: 2277 128X,pp.423-430.

[16] P.V.S.R .Bharadwaja, K. Ravi Teja], M.Naresh babu,K.Neelima ," Advanced Low Power RISC Processor Design using MIPS Instruction Set",Proc.of IEEE ,2015, ISBN:978-1-4788-7225-8/15/$31.00 ©2015 IEEE, pp.1252-1258.

[17] Priyanka Trivedi, Rajan Prasad Tripathi," Design & Analysis of 16 bit RISC Processor Using low Power Pipelining",Proc.of International Conference on Computing, Communication and Automation (ICCCA) ,2015, ISBN:978-1- 4799-8890-7/15/$31.00 ©2015 IEEE, pp.1294-1297.

[18] Pejman Lotfi-Kamran. Ali-Asghar Salehpour. Amir-Mohammad Rahmani. Ali Afzali-Kusha, and Zainalabedin Navabi. "Dynamic Power Reduction of Stalls in Pipelined Architecture Processors",International Journal Of Design, Analysis And Tools For CircuitsAnd Systems, June 2011. Vol. I, 8 Issue-I,pp.9- 14.

[19] Sharda P. Katke, G.P. Jain,"Design and Implementation of 5 Stages Pipelined Architecture in 32 Bit RISC Processor", IJETAE, April 2012,Volume 2. Issue 4, pp. 340-346.

[20] Mrs. Rupali Balpande, Mrs. Rashmi Keote, "Design of FPGA based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor",Proc.of International Conference on Communication Systems and Network Technologies, 2011,ISBN: 978-0- 7695-4437-3/11 $26.00 © 2011 IEEE,pp.409-413.

[21] Kui YI, Yue-Hua DING, "32-bit RISC CPU Based on MIPS Instruction Fetch Module Design" Proc.of International Joint Conference on Artificial Intelligence, 2009, ISBN: 78-0-7695-3615-6/09, IEEE,PP.754- 760.

[22] Mrs. Rupali S. Balpande, Mrs. Rashmi S. Keote, "Design of FPGA based Instruction Fetch & Decode Module of 32- bit RISC (MIPS) Processor", Proc. of International Conference on Communication Systems and Network Technologies, 2011 IEEE , ISBN:978-0-7695-4437-3/11,pp.409- 413

[23] Rohit Sharma, Vivek Kumar Sehgal, Nitin Nitin1, Pranav Bhasker, Ishita Verma, "Design and Implementation of a 64-bit RISC Processor using VHDL", Proc.of 11th International Conference on Computer Modelling and Simulation, UKSim 2009 IEEE, ISBN:978-0-7695-3593-7/09,pp.568- 573.

[24] I.Dalal, A.Ganesh, Aishwarya.D “An 8 bit Power-Efficient MIPS Processor “, Advanced VLSI Design Sping ,2014.

[25] Zulkifli, Yudhanto, Soetharyo " Reduced Stall MIPS Architecture using Pre-fetching Accelerator "International Conference on Electrical Engineering and Informatics (ICEEI),2009 ISBN: 978-1- 4244-4913-2/09/$25.00 IEEE,Pp. 611-616.

[26] Indu, Arun Kumar, “Design of Low Power Pipelined RISC Processor”, International Journal of Advanced Research in Electrical & electronics & instrumentation Engineering, August 2013,vol.2, Issue 3,pp.3747-375 .

[27] Samiappa Sakthikumaran ,S. Salivahanan,V.S,kanchan.Bhaaskaran, “16 bit RISC Processor Design For Convolution Application”,Proc.of IEEE - International.Conference on Recent Trends in Information technology, June 2011, ISBN:978-1-4577-0590-8/11/$26.00, pp.394- 397.

[28] B. Ramkumar and Harish M Kittur “Low-Power and Area-Efficient Carry Select Adder” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2012, vol. 20, Issue 2, pp. 371-75.

[29] Hai Li, Swarup Bhunia, Yiran Chen, Kaushik Roy, “DCG:Deterministic Clock - Gating for Low - Power Microprocessor Design”, IEEE Trans. On VLSI Systems, March 2004,vol. 12, Issue 3, pp.245-254.

[30] Indu, Arun Kumar, “Design of Low Power Pipelined RISC Processor”, International Journal of Advanced Research in Electrical & electronics & instrumentation Engineering Engineering (IJSRD), August 2013,vol.2, Issue 4, ISSN (Print) : 2320 – 3765 ,ISSN (Online): 2278 – 8875 pp.3747-3756.

[31] David A Patterson, John L Hennessy "Computer Organization and Design - The Hardware- Software Interface" 3rd Edition.

[32] Preethi Sudha Gollamudi, M. Kamaraju “Design Of High Performance IEEE- 754 Single Precision (32 bit) Floating Point Adder Using VHDL”, International Journal of Engineering Research & Technology (IJERT), July – 2013,Vol. 2 Issue 7, ISSN: 2278-0181 ,pp.2264-2275.

[33] Jinde Vijay Kumar, Boya Nagaraju, Chinthakunta Swapna and Thogata Ramanjappa, “Design and Development of FPGA Based Low Power Pipelined 64-Bit RISC Processor with Double Precision Floating Point Unit”, International Conference on Communication and Signal Processing, April 3-5, 2014, ISBN: 978-1- 4799-3358-7114/$3l.00,pp.1054-1058.

[34] Li Li and Ken Choi “SeSCG: Selective Sequential Clock Gating for Ultra - low- Power Multimedia Mobile Processor Design,” IEEE EIT Conference, May 2010.

[35] Luo, Q., Yao, C., Ning, K., Zheng, X., Zhao, M., Cheng, L., ... & Liu, L. (2022). A programmable and flexible vision processor. IEEE Transactions on Circuits and Systems II: Express Briefs, 69(9), 3884-3888.

[36] Jungwirth, P., Scott, A., Kondamadugula, S., & Xiao, Z. (2024, March). Alabama A&M Symmetric Overloaded Reduced Instruction Set Architecture (SORA). In SoutheastCon 2024 (pp. 1268-1273). IEEE.

[37] Kulshreshtha, A., Moudgil, A., Chaurasia, A., & Bhushan, B. (2021, March). Analysis of 16-bit and 32-bit RISC processors. In 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1, pp. 1318-1324). IEEE.

[38] Prabhakaran, S., Mathan, N., & Vedanarayanan, V. (2022, March). Design and Analysis of a Multi Clocked Pipelined Processor Based on RISC-V. In 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT) (pp. 1-5). IEEE.

[39] Avinash, N. J., Mishra, I., Ghorpade, T. C., Lokesh, M., Nishanth, H., & Kumar, M. (2023, January). Design and Implementation of 32-Bit MIPS RISC Processor with Flexible 5-Stage Pipelining and Dynamic Thermal Control. In 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE) (pp. 846-849). IEEE.

[40] Lindgren, P., Dzialo, P., Lunnikivi, H., & Ericsson, J. (2023, December). ENEST-Efficient Interrupt Nesting for RISC-V based CPUs. In 2023 IEEE 2nd Industrial Electronics Society Annual On-Line Conference (ONCON) (pp. 1-7). IEEE.

[41] Soundari, D. V., Ganesh, M. S., Raman, I., & Karthick, R. (2021). Enhancing network-on-chip performance by 32-bit RISC processor based on power and area efficiency. Materials Today: Proceedings, 45, 2713-2720.

[42] Junior, F. C. S., Silva, I. S., & Jacobi, R. P. (2021, August). Evaluating the performance, energy and area tradeoffs of ATHENA in superscalar processors. In 2021 34th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI) (pp. 1-6). IEEE.

[43] Kadomoto, J., Irie, H., & Sakai, S. (2022, December). Evaluation of Different Microarchitectures for Energy-Efficient RISC-V Cores. In 2022 IEEE 15th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC) (pp. 78-84). IEEE.

[44] Amruth, M. S. S. D., Swamy, B. P. R., Satyanarayana, C. M. V., Kumar, B. R., Aarthi, V. P. M. B., & Ramesh, M. (2023, June). Five Stage Pipelined MIPS Processor Verification Scoreboard Module using UVM. In 2023 International Conference in Advances in Power, Signal, and Information Technology (APSIT) (pp. 593-596). IEEE.

[45] Jamieson, P., Ferreira, R., & Nacif, J. A. (2021, October). Personalizing online computer engineering resources and labs for digital, embedded, and computer system courses. In 2021 IEEE Frontiers in Education Conference (FIE) (pp. 1-5). IEEE.

[46] Ali, Z., Tanveer, T., Aziz, S., Usman, M., & Azam, A. (2022, October). Reassessing the performance of arm vs x86 with recent technological shift of apple. In 2022 International Conference on IT and Industrial Technologies (ICIT) (pp. 01-06). IEEE.

[47] Aradhya, H. R., Kanase, G., & Vinayakgouda, Y. (2021, July). RTL to GDSII of Harvard structure RISC processor. In 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT) (pp. 1-4). IEEE.

[48] He, Y., & Chen, X. (2023, April). Survey and Comparison of Pipeline of Some RISC and CISC System Architectures. In 2023 8th International Conference on Computer and Communication Systems (ICCCS) (pp. 785-790). IEEE.