IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

A Review On Comparative Analysis Of 16-Bit And 32-Bit RISC Pipelined Processors By Verilog Design

Tejaswini D, Shrisha M R
MTech Student, Assistant Professor
Department of Electronics & Communication Engineering,
VLSI Design and Embedded Systems
BMSCE, Bengaluru, India

Abstract: This review paper presents a comparative analysis of 16-bit and 32-bit Reduced Instruction Set Computing (RISC) pipelined processors, with a focus on their design and implementation using Verilog Hardware Description Language (HDL). The study explores key architectural differences, performance metrics, and resource utilization associated with both processor types. By delving into the pipeline stages, instruction sets, and data paths, the review highlights the trade-offs between the two architectures in terms of complexity, speed, power consumption, and scalability. Detailed simulations and synthesis results will be analyzed to provide a comprehensive understanding of the advantages and limitations of each processor size. The findings aim to guide designers in selecting the appropriate processor configuration for specific applications, balancing the need for efficiency, cost, and performance. The paper concludes with insights into future trends in RISC processor design and potential areas for further research in the context of evolving computational demands and technological advancements.

Index Terms **– RISC architecture, Pipeline stages, Verilog HDL, power consumption.**

I. INTRODUCTION

In the era of evolving landscape of digital design, the Reduced Instruction Set Computing (RISC) architecture stands out for its streamlined instruction sets and emphasis on performance efficiency. RISC processors are fundamental to a wide range of applications, from embedded systems to high-performance computing. This review paper aims to provide a comprehensive comparative analysis of 16-bit and 32-bit RISC pipelined processors, focusing on their design and implementation using Verilog Hardware Description Language (HDL).

The decision between utilizing 16-bit and 32-bit architectures is crucial and involves understanding the inherent trade-offs. A 16-bit processor typically offers advantages in terms of reduced complexity, lower power consumption, and cost-effectiveness, making it ideal for resource-constrained environments and applications requiring moderate computational power. Conversely, a 32-bit processor delivers superior computational capabilities, larger addressable memory space, and enhanced performance, essential for advanced embedded systems, real-time processing, and scientific computations.

A novel aspect of this review is the integration of floating-point operator's capabilities within the comparative framework. Floating-point operations are increasingly vital in applications that demand high precision and dynamic range, such as digital signal processing, graphics, and scientific simulations. By incorporating floating-point units (FPUs) in both 16-bit and 32-bit RISC processors, we can assess their impact on performance, resource utilization, and power efficiency. This paper delves into the architectural intricacies of 16-bit and 32-bit RISC pipelined processors, with a particular emphasis on their floating-point processing capabilities. We explore key design elements, including pipeline stages, instruction sets, data

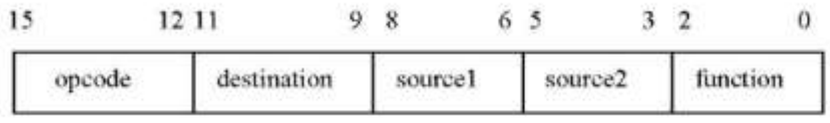
paths, and control mechanisms, to understand how these factors influence overall processor performance and resource utilization. The use of Verilog HDL as a design and simulation tool enables precise modelling and verification, providing valuable insights into the operational characteristics of these processors. Our analysis is grounded in detailed simulations and synthesis results, highlighting the strengths and weaknesses of each processor type. By examining metrics such as speed, power consumption, scalability, and the efficiency of floating-point operations, we aim to offer a balanced perspective that can inform the selection of appropriate processor configurations for specific applications.
The paper is structured as follows: We begin with an overview of RISC architecture, pipelining concepts, and the importance of floating-point operators. This is followed by a detailed comparison of 16-bit and 32-bit processors in terms of their design and performance metrics, including floating-point capabilities. Subsequent sections discuss the implications of our findings for practical applications and future trends in processor design. Through this comparative analysis, we seek to contribute to the ongoing discourse on processor design optimization, providing a valuable reference for researchers and engineers in the field of digital design.

II. LITERATURE SURVEY ON RISC PROCESSORS

Rajeshwari Bhat, [1] et al. proposal for a Processor and 32-bit MIPS Processor: A Review, RISC (Reduced Instruction Set Computing) is an instruction set architecture characterized by a smaller instruction set compared to CISC (Complex Instruction Set Computing). This reduction in instruction set complexity simplifies the implementation of various instructions, thereby lowering dynamic power consumption, reducing cycles per instruction (CPI), and ultimately decreasing costs. The 32-bit MIPS processor utilizes a five-stage pipeline, which significantly decreases CPI and enables the processor to function at higher frequencies. The pipeline stages also facilitate the implementation of more complex instructions. This paper provides a comprehensive review of studies comparing and analyzing 16-bit and 32-bit microprocessors. These models, implemented using Verilog HDL and analyzed with tools such as Xilinx Vivado, ISE, and Modelsim, as well as Cadence tools in some designs, are examined based on several parameters. These parameters include dynamic power consumption, combinational delay, operating frequency, and CPI.

Mrs. Rupali S. Balpande [2] have proposed an MIPS instruction format, instruction data path decoder module function and design theory based on RISC CPU instruction set. Furthermore, they propose a design of instruction fetch (IF) module of 32-bit CPU based on RISC CPU instruction set. Through analysis of function and theory of RISC CPU instruction decoder module, they also propose a design of instruction decoder (ID) module of 32-bit CPU by pipeline theory. The instruction decoder includes register file, write back data to register file, sign bit extends, relativity check, and it is simulated on QuartusII successfully. They have also designed an instruction set, dataflow and pipeline design of RISC CPU based on MIPS. In this research, they have adopted top-down design method and use VHDL to describe system. At first, they design the system from the top, and do in-depth design gradually. The structure and hierarchical of design is very clear. It is easy to edit and debug. Design of instruction fetch (IF) stage simulates, integrate and routes on Quartus II 4.3. Their result indicates IF stage completes prospective function.

Pravin S. Mane [3] describes a 16-bit RISC processor design using VHDL. They use a hierarchical approach so that basic units can be modelled using behavioural programming. Then, these basic units are combined using structural programming. They use four stages (viz. instruction fetch stage, instruction decode stage, execution stage and memory/lO, write back stage) pipelining to improve the overall CPI (Clock Cycles per Instruction). They also use hardwired control approach to design the control unit as against micro programmed control approach in conventional CISC processor. They have shown the general instruction format for 16-bit RISC Processor.



This processor has one input port, one output port and six hardware vectored interrupts along with 16- bit address bus and 16-bit data bus. Structural hazards are dealt with the implementation of pre-fetch unit, data hazards are dealt with forwarding and control hazards are dealt with flushing and stalling. They have implemented the design on FPGA for verification purpose. They achieved the maximum throughput of execution as one instruction per clock pulse provided that there are no stalls by using pipelining approach. This is possible due to hardwired approach for design of control unit and fixed length instruction format. To resolve data hazards, result forwarding is efficient than stalling as it remove the penalty of time in handling such conflicts. They use prefetch unit for handling the structural hazards while flushing is used to handle control hazards. The prefetch buffer is like a small cache storing tag along with instruction fetched. It does not work on FIFO principle. This design is modelled and simulated using VHDL and then implemented on FPGA successfully. The maximum frequency of operation they have obtained on the Xilinx's Spartan-Il FPGA is 26- MHz.

Animesh Kulshreshtha [4] et al. introduces the approach which aims to minimize instruction complexity, thereby reducing cost, cycle time, and operating power. Despite the introduction of 16-bit RISC processors in the 1970s, they have encountered significant technical challenges, leading to the development of more advanced 32-bit and 64-bit RISC processors and the adoption of pipelining techniques. In this review, studies on the behavioural models of 16-bit and 32-bit RISC processors and their respective instruction sets. The 16-bit RISC processor, based on a non-pipelined Harvard architecture, features separate data and instruction memory. In contrast, the 32-bit RISC processor utilizes pipelining techniques inspired by the MIPS architecture. Both processors include General Purpose Registers (GPRs) and flag registers (e.g., Carry, Zero). The reviewed models simulate an optimized multiplier algorithm and aim to enhance the data path, given that arithmetic and logical operations are power-intensive and have high execution delays. This paper compares these models based on their instruction sets and performance metrics such as speedup and power dissipation. The individual models have been designed, simulated, and integrated into a top-level module using the Xilinx ISE Design Suite 14.7, including power analysis.

Agnesh Savaliya [5] found the way to design and simulate a single-precision floating-point Arithmetic Logic Unit (ALU), which functions as part of a math coprocessor. Floating-point representation offers the advantage of supporting a much wider range of values compared to fixed-point and integer representations. The primary arithmetic operations performed by this unit include addition, subtraction, multiplication, and division. In this floating-point unit, inputs are provided in IEEE 754 format, representing 32-bit single-precision floating-point values. This arithmetic unit is primarily utilized in math coprocessors, commonly known as Digital Signal Processors (DSPs). DSPs require high-precision values for signal processing, and because these calculations are iterative, they need to be performed as quickly as possible. Standard processors are unable to meet these requirements, leading to the adoption of floating-point representation, which enables fast and accurate computations.

Chandran Venkatesan [6] addressed the issue of enhancing system reliability and speed, the architecture of a MIPS (Microprocessor without Interlocked Pipeline Stages) based RISC (Reduced Instruction Set Computer) microprocessor employs a Harvard data path structure designed to execute a small set of instructions at high speed. This project explores the design and implementation of a low-power processor utilizing a 4-stage pipelining approach. The pipelining stages include fetch, decode, execute, and memory read/write operations. To achieve low power consumption, the project employs a clock gating technique, which eliminates unnecessary clock usage when a module is not in use. The main objective is to design a 4-stage pipelined RISC processor, progressing from RTL (Register Transfer Level) to GDSII (Physical Design). The processor is coded in Verilog HDL and implemented using the Cadence Encounter Compiler tool. The project also includes calculations of area, power, delay, and clock gating effectiveness using the Cadence RTL Compiler with slow and fast libraries of 45nm technology.

Sangeeta Palekar [7] has propose a high-speed MIPS-based 32-bit RISC processor equipped with a single-precision floating point unit specifically for DSP applications. The focus of the design is on enhancing the performance of the floating-point arithmetic unit, thereby improving the overall performance of the RISC processor. The proposed processor is capable of executing a wide range of instructions, including arithmetic, logical, floating point, data transfer, memory, shifting, and rotating operations. Digital signal processing (DSP) applications have grown significantly since the invention of technology in fields like space exploration, medical research, and numerous commercial industries. Many high-speed embedded and DSP applications rely on RISC processors. Because it can handle a wide dynamic range of values. In this study, we particularly design a 32-bit RISC processor with a single-precision floating point unit for DSP applications, based on a fast MIPS architecture. The design is centered on optimizing the floating-point arithmetic unit's performance, which will raise the RISC processor's total performance. Many other types of instructions, such as arithmetic, logical, floating point, and data transmission, can be carried out by the suggested processor.

Mamun Bin Ibne Reaz [8] found the work which described about VHDL-based design technique for a single clock cycle MIPS RISC processor with the goal of streamlining the hardware realization, simulation, description, and verification procedures. The RISC processor has 32-bit general-purpose registers with 32-bit memory words and uses fixed-length 32-bit instructions based on three formats: R-format, I-format, and J-format. A control unit oversees operations across the five stages of the MIPS processor, which are write back, data memory, instruction fetch, instruction decode, and execution.
For every module in the design, VHDL is used, and its concurrency features are utilized to manage the inherent parallelism of digital hardware. All phases are combined into a single, coherent system by the top-level module. During the design phase, inputs, outputs, the main block, and other modules are specified. Then, the program is run.

Mr. Sagar P. Ritpurkar [9] has introduces about the MIPS instruction format, instruction data flow, decoder module function, and design theory based on the instruction set of RISC CPUs will all be examined in this study. Here, the pipeline design process—which includes write-back, data memory (MEM), execution (EXE), instruction fetch (IF), and instruction decoder (ID)—will be used. (WB) modules of a 32-bit CPU that use the instruction set of a RISC CPU. The fetch instruction, latch, address arithmetic, check validity of instruction, and synchronous control modules are the primary functions of the IF module. The register file, write back data to register file, sign bit extension, and relativity check are all included in the instruction decoder. Lastly, the Xilinx ISE simulator will be used for synthesis, and VHDL will be used for coding.

Iro Pantazi [10] found out the pipeline approach, Pipelining is an implementation approach that uses the parallelism between the operations required to execute an instruction to its advantage by overlapping numerous instructions in execution. These days, the primary implementation method for creating quick CPUs is pipelining. But the majority of the time, Data dependencies exist that cause issues during execution and must be resolved. In this work, we applied pipelining to the MIPS architecture and examined how our system managed data dependencies.

Vinayak Patil [11] et al. proposed that in many fields of science and engineering, floating point operations play a crucial role. The floating-point coprocessor, or FPU, carries out operations such as multiplying, accumulating, dividing, square rooting, addition, subtraction, and comparison. The instruction sets of ARM, MIPS, RISC-V, and other processors include floating point operations. The FPU may be implemented in software or as a component of hardware. The architectural investigation for a floating-point coprocessor with RISC-V floating point instructions is described in this work. The RISC-V floating point instructions used in the construction of the Floating unit make it completely compatible with the IEEE 754-2008 standard. Both single and double precision floating point data operands can be handled by the floating-point coprocessor in an in-order commit and out-of-order execution scenario.

Jadapalli Sreedha [12] introduce a novel design to further enhance program runtime effectiveness and reduce power consumption in low-cost embedded and Internet of Things (IoT) devices, ARM and Intel developed Bit Manipulation Instructions (BMIs). Therefore, focused on the design of the open-source RISC-V (RV32I)-based "bitRISC" processor, a fully synthesizable 32-bit CPU. The "bitRISC" processor incorporates up to 16 arithmetic and logical operations within its ISA, aiming to simplify the design while maintaining high performance. The processor is a single-cycle design developed using Verilog HDL, following a streamlined design approach. It was prototyped using Cadence tools, and power and area reports were generated and analyzed to ensure optimal performance and efficiency.

Mr. S. P. Ritpurkar [13] investigates the use of VHDL (Very High-Speed Integrated Circuits Hardware Description Language) in the design and implementation of a Reduced Instruction Set Computer (RISC) CPU architecture based on Microprocessor Interlock Pipeline Stages (MIPS). VHDL is widely used as a reliable solution for high-volatility applications and for the emulation of Application Specific Integrated Circuits (ASICs). Furthermore, these systems frequently require Field Programmable Gate Arrays (FPGAs), which are well-known for their rapid time-to-market and reprogram ability. The instruction set, architectural layout, and timing diagrams of the RISC CPU architecture are all thoroughly analyzed in this study. Using a Float to Fixed number converter module, this study's conversion of floating-point numbers to fixed numbers is a key component. John Bunda [14] noticed that in any stored-program computer system, information frequently transfers between memory and the instruction processor, with machine instructions constituting a significant portion of this traffic. Since transfer bandwidth is a limited resource, inefficient encoding of instruction information (low code density) can lead to substantial hardware and performance costs. This review paper compares the performance of two instruction encodings for the same instruction processing core, starting with a parameterized baseline RISC design. One encoding is a variant of DLX, a typical 32-bit RISC instruction set, while the other is a 16-bit format that sacrifices some expressive power but retains essential RISC features. Utilizing optimizing compilers and software simulation, the study measures code density and path length for a suite of benchmark programs, relating performance differences to specific instruction set features. Additionally, the paper evaluates time-to-completion performance by varying memory latency and instruction cache size parameters. The results demonstrate that the 16-bit format offers significant cost-performance advantages over the 32-bit format under typical memory system performance constraints. This comparative analysis provides valuable insights into the trade-offs and efficiencies of different instruction encodings in RISC architectures, contributing to the optimization of processor design for enhanced performance and resource utilization.

III. CONCLUSION AND FUTURE WORK

This review paper provides an in-depth survey of various RISC Processor architectures, internal data pathways and fewer transistors, 16-bit computers are more expensive but can only be used in a growing number of niche applications. Notwithstanding their increased cost and power consumption, 32-bit processors offer noticeably improved performance, economy, and operational speed particularly in demanding applications like floating-point arithmetic. 32-bit processors use more dynamic power because of its higher operating frequency, which allows for more operations to be performed in a cycle, resulting in quicker processing rates and less combinational delays. 32-bit CPUs' pipelined architecture optimizes the instruction cycle, which further improves speed.

Future research should focus on further optimizing 32-bit processor designs to balance performance and power consumption. Investigating advanced power-saving techniques, such as dynamic voltage and frequency scaling (DVFS) and clock gating, could reduce power consumption without sacrificing performance. Additionally, exploring hybrid architectures that combine the benefits of 16-bit and 32-bit processors could offer a versatile solution for a wider range of applications. Another promising direction is the development of adaptive processors that can dynamically switch between 16-bit and 32-bit modes based on the computational demands of the application, thereby optimizing power usage and processing efficiency. Finally, further research into the integration of specialized co-processors for specific tasks, such as DSP and AI acceleration, can enhance the overall capabilities of the processor while maintaining energy efficiency.

REFERENCES

[1] R. Bhat, D. Pandey, F. Ahmad and P. Gupta, "Analysis and Optimization of 16-bit RISC Processor and 32-bit MIPS Processor: A Review," 2023 3rd International Conference on Intelligent Technologies (CONIT), Hubli, India, 2023, pp. 1-6, doi: 10.1109/CONIT59222.2023.10205898

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

[3] Pravin S. Mane, Indra Gupta, M. K. Vasantha, Implementation of RISC Processor on FPGA, 1-4244-0726-5/06, 2006 IEEE.

[4] A. Kulshreshtha, A. Moudgil, A. Chaurasia and B. Bhushan, "Analysis of 16-Bit and 32-Bit RISC Processors," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 1318-1324, doi: 10.1109/ICACCS51430.2021.9441873.

[5] Yagnesh Savaliya, Jenish Rudani "Design and Simulation of 32-Bit Floating Point Arithmetic Logic Unit using VerilogHDL"2020 International Research Journal of Engineering and Technology (IRJET),

Nadiad, India.

[6] C. Venkatesan, M. T. Sulthana, M. G. Sumithra and M. Suriya, "Design of a 16-Bit Harvard Structure RISC Processor in Cadence 45nm Technology," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 173-178, doi: 10.1109/ICACCS.2019.8728479.

[7] S. Palekar and N. Narkhede, "32-Bit RISC processor with floating point unit for DSP applications," 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 2016, pp. 2062-2066, doi: 10.1109/RTEICT.2016.7808202.

[8] M. B. I. Reaz, M. S. Islam and M. S. Sulaiman, "A single clock cycle MIPS RISC processor design using VHDL," ICONIP '02. Proceedings of the 9th International Conference on Neural Information Processing. Computational Intelligence for the E-Age (IEEE Cat. No.02EX575), Penang, Malaysia, 2002, pp. 199-203, doi: 10.1109/SMELEC.2002.1217806.

[9] Mr. Sagar P. Ritpurkar, Prof. Mangesh N. Thakare, Prof. Girish D. Korde, “Review on 32-bit MIPS RISC Processor using VHDL,” 2014 IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE), Wardha, India.

[10] Pantazi-Mytarelli, "The history and use of pipelining computer architecture: MIPS pipelining implementation," 2013 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 2013, pp. 1-7, doi: 10.1109/LISAT.2013.6578243.

[11] V. Patil, A. Raveendran, P. M. Sobha, A. David Selvakumar and D. Vivian, "Out of order floating point coprocessor for RISC V ISA," 2015 19th International Symposium on VLSI Design and Test, Ahmedabad, India, 2015, pp. 1-7, doi: 10.1109/ISVDAT.2015.7208116.

[12] V. Jain, A. Sharma and E. A. Bezerra, "Implementation and Extension of Bit Manipulation Instruction on RISC-V Architecture using FPGA," 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), Gwalior, India, 2020, pp. 167-172, doi: 10.1109/CSNT48778.2020.9115759

[13] S. P. Ritpurkar, M. N. Thakare and G. D. Korde, "Design and simulation of 32-Bit RISC architecture based on MIPS using VHDL," 2015 International Conference on Advanced Computing and Communication Systems, Coimbatore, India, 2015, pp. 1-6, doi: 10.1109/ICACCS.2015.7324067.

[14] John Bunda, Don Fussell, W. C. Athas, Roy Jenevein,” 16-bit vs. 32-bit instructions for pipelined microprocessors,” 1993 ACM SIGARCH Computer Architecture News, doi: doi.org/10.1145/173682.165159

[15] J. -Y. Lai, C. -A. Chen, S. -L. Chen and C. -Y. Su, "Implement 32-bit RISC-V Architecture Processor using Verilog HDL," 2021 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Hualien City, Taiwan, 2021, pp. 1-2, doi: 10.1109/ISPACS51563.2021.9651130.

[16] Sivarama P. Dandamudi, "A Guide to RISC Processor For Programmers And Engineers" in, Springer

[17] Preetam Bhosle, Hari Krishna Moorthy, "FPGA Implementation of low power pipelined 32-bit RISC Processor", International Journal of Innovative Technology and Exploring Engineering (IJITEE), August 2012.

[18] M. N. Topiwala and N. Saraswathi, "Implementation of a 32-bit MIPS based RISC processor using Cadence," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, 2014, pp. 979-983.

[19] Patra, S., Kumar, S., Verma, S., Kumar, A. (2020). Design and Implementation of 32-bit MIPS-Based RISC Processor. In: Dutta, D., Kar, H., Kumar, C., Bhadauria, V. (eds) Advances in VLSI, Communication, and Signal Processing. Lecture Notes in Electrical Engineering, vol 587. Springer, Singapore.

[20] P. Trivedi and R. P. Tripathi, "Design & analysis of 16-bit RISC processor using low power pipelining," International Conference on Computing, Communication & Automation, 2015, pp. 1294-1297.