DIO: 30.0485/ijearst.06.01.3479 https://ijearst.co.in/

ENGINEERING IN ADVANCED RESEARCH SCIENCE AND TECHNOLOGY

ISSN 2386-7876
Vol.01, Issue.03
Mar-2026
Pages: 739-750

DESIGN AND ANALYSIS OF 16-BIT RISC PROCESSOR

[1]M MADHAVI, [2]PADE ARUNA, [3]ITHADI SWAPNA, [4]ANDRA AKSHAYA, [5]MEDAPI ANJANI LAKSHMI

[1]Guide, Dept of ECE, ABR College of Engineering and Technology, Kanigiri, A.P.
[2,3,4,5]B. Tech, Dept of ECE, ABR College of Engineering and Technology, Kanigiri, A.P.

ABSTRACT: This project describes a 16-bit RISC microprocessor core that has been designed for portable applications. To implement these instructions the design incorporates various design blocks like Control Unit (CU), Arithmetic and Logic Unit (ALU), Accumulator, Program Counter (PC), Instruction Register (IR), Memory and additional logic. The RISC is designed to enhance processor performance by keeping the following goals in mind. The RISC uses simple constructs and has small instruction set. It is basically designed in order to achieve faster executions. 16 bit RISC processor is implemented with following specifications. Ripple carry adder is used to implement addition operation, separate General subtraction block is used to produce subtraction output of operands. Serial multiplier is used for multiplication purpose. And some basic gates are design for implementation of logical operations.

Keywords: RISC, Load/Store architecture, Von Neumann architecture.

Introduction: Reduced Instruction Set Computers (RISCs) are now use for all type of computational tasks. In the area of scientific computing, RISC workstations are being increasingly used for compute task such as DSP, DIP etc. RISC concepts help to achieve given levels of performance at significantly lower cost than other systems. Pipelined RISC improves speed and cost effectiveness over the ease of hardware description language programming and conservation of memory and RISC based designs will continue to grow in speed and ability. The main features of RISC processor are the instruction set can be hardwired to speed instruction execution. In the present work, the design of a 4-bit data width Reduced Instruction Set Computer (RISC) processor is presented. It has a complete instruction set, program and data memories, general purpose registers and a simple Arithmetical Logical Unit (ALU) for basic operations. In this design, most instructions are of uniform length and similar structure, arithmetic operations are restricted to CPU registers and only separate load and store instructions access memory. The architecture supports 8 instructions to support Arithmetic, Logical, Shifting, and load - store operations. When the controller design become more complex in CISC and the performance was also not up to expectations, people started looking on some other alternatives. It had been found that when a processor talks to the memory the speed gets killed. So the one improvement on CPI was to

keep the instruction set very simple. Simple in not the way it works but the way it looks. That‟s why there are very few instructions in any typical RISC architecture where processor asks data from memory probably not other than Load and Store. At the end the pipelining added a new dimension in the speed just with the help of some additional registers, which increases throughput by reducing CPI. Hence the instruction can be executed effectively in one clock cycle. A common misunderstanding of the phrase "Reduced Instruction Set Computer" is the mistaken idea that instructions are simply eliminated, resulting in a smaller set of instructions. In fact, over the years, RISC instruction sets have grown in size, and today many of them have a larger set of instructions than many CISC CPUs. The term "Reduced" in that phrase was intended to describe the fact that the amount of work any single instruction accomplishes is reduced at most a single data memory cycle compared to the "complex instructions" of CISC CPUs that may require number of data memory cycles in order to execute a single instruction. Most microprocessors in today‟s market are based on either RISC or CISC architectures. Research has shown that RISC architecture greatly boosts computer speed by using simplified machine instructions for frequently used functions This RISC processor is designed using pipelined architecture; through this we can improve the speed of the operation. In this we are using 5-stage pipelining. The 5 stages are Fetch, Decode, Execute, Memory and Write Back. The reason for wide use is that they are small; therefore, they do not take up much die area and are cost effective to fabricate. This processor will follow the RISC architecture because it supports a predefined set of instructions. In this all the instructions have same length. RISC processors, first developed in the eighties. Embedded processors demand instruction set suitable for the specific applications, various and fast interrupt handling, consumption and so on. For this properties RISC is more efficient in comparison with micro programmed than CISC. New society is highly dependent upon technological things such as gadgets.

Literature Survey: Uma [1] presented a new design for 8-bit microprocessors. The presented architecture is implemented on Spartan-3E FPGA board. To code architecture in FPGA the author use standard HDL language. To check the functionality and generate the bit-stream to the corresponding FPGA chip Xilinx ISE tool is used. In this case the author implemented most of the essential instruction set effectively using structural modeling. This allows the author to reduce the hardware requirement. The performance of the architecture is improved due to this reason. Also the implementation cost of the architecture reduces drastically due to this reason. Moreover to increase throughput of the architecture the author use some degrees of pipelining. ShahlaGul et al. [2] Made a comparison between RISC and CISC microprocessor architectures in this paper in a wide way. Also they have showed that the types of operations are performed by both RISC and CISC architectures. Like CISC architecture divides the total algorithm into a various simplifier instructions which executes step by step. This simplifies the program having smaller architecture but for larger or complex architecture this method is difficult. In such case RISC architecture is helpful. Because the RISC architecture embedded many

instruction set. Also in the design prospective the design of RISC architecture is harder than the CISC architecture. This is mainly due to the complexity of the controller which is used to cascade instruction. Mrudul S. Ghaturle et al In 2016 Sarika U. Kadam, S.D. Mali, designed "Design of RISC Processor using VHDL". The proposed 16-bit RISC processor is designed using a parallel programming language called VHDL. It is simulated and synthesized using Xilinx ISE 13.1i. Pipelining is used to make processor faster. In Pipelining instruction cycle is divided into parts so that more than one instruction can be operated in parallel. Number of instructions are designed for this processors. Multiplier is also designed using ADD instruction. All instructions are simulated successfully. Simulation results show that the proposed processor is working correctly. The proposed processor has a delay of 4.744 ns and operating frequency of 210.775 MHz. When the proposed work compared with previous processors, it can be seen that proposed processor has less delay[1]. Swati Joshi, Sandhya Shinde, Amruta Nikam,"32- bit pipeline Risc Processor in VHDL using Booth Algorithm",. The aim of paper is to design instruction fetch unit and ALU which are part of RISC processor architecture. Instruction fetch is designed to read the instructions present in memory. ALU is in the execution stage of pipelining which performs all computations i.e. arithmetic and logical operations. Xilinx 8.1i is used to simulate the design using VHDL language. This paper proposes ALU which performs operations such as addition, subtraction, AND, OR, NOT, XOR etc. successfully. ALU provides correct results according to Opcodes and operands provided. ALU designed in this paper is used in execution stage of pipelined processor. Instruction fetch unit works correctly when provided with address it fetches correct instruction from memory. It is used to read instruction from memory which is the first step of pipelined processor. The designed fetch unit and ALU are used in pipelined RISC processor[2]. Vishwas V.Balpande, Vijendra P.Meshram, Ishan A. Patilm, Sukeshini N.Tamgadem, Prashant Wanjari, "Design and Implementation of RISC processor on FPGA", In proposed paper 16-bit RISC processor is designed using VHDL programming. Four stage (viz. instruction fetch stage, instruction decode stage, execution stage and memory/IO - write back stage) pipelining is used to improve the overall CPI (Clock Cycles per Instruction). Hardwired control approach is used to design the control unit as against microprogrammed control approach in conventional CISC processor. Structural hazards are dealt with the implementation of prefetch unit, data hazards are dealt with forwarding and control hazards are dealt with flushing and stalling.

PROPOSED METHOD:

16bit RISC using array multiplier Ripple carry adder:

The proposed RISC processor has a simple 16-bit instruction set format, it is non-pipelined processor shown in figure (1). The processor has a load and store architecture, in which the operations will be performed on registers, and not on memory locations. It supports the typical von-Neumann architecture with one common memory bus for both data and instructions. A prior work is to design the processor with 15 instructions as a first step in the development of the processor. The instruction set consists of

Logical, jump, read, write and HALT type of instructions. The processor performs in three phases viz; fetch, decode and execute. In the first phase, the instruction and the necessary data is obtained from the memory. While in decode phase, the data and the instruction drawn from the memory gets separated and activates the components and data path to be executed. Finally, in execution phase the instruction is executed, the data is computed and the result is stored.

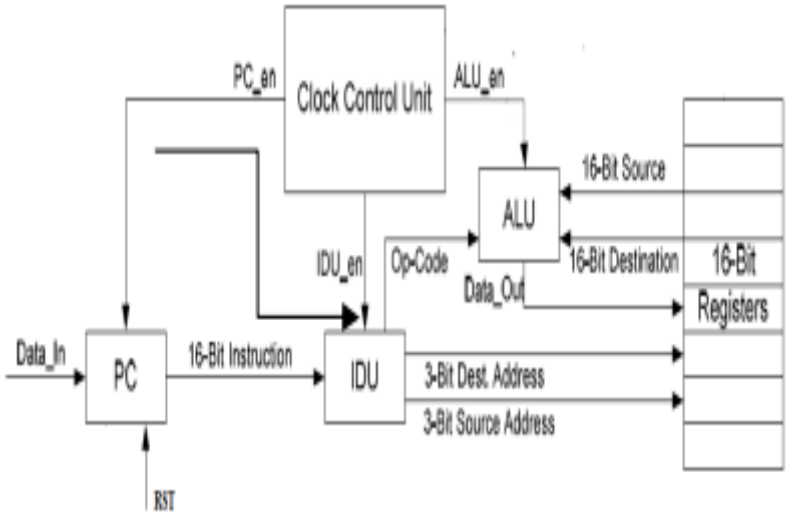


Fig1: Block Diagram of RISC processor.

The architecture of the proposed RISC CPU is a uniform 16-bit instruction format, single cycle processor. It has a load/store architecture, where the operations will only be performed on registers, and not on memory locations. It follows the classical vonNeumann architecture with just one common memory bus for both instructions and data The instruction set consists of Load, store and HALT type of instructions. The Halt instruction acts as a border line between the instruction and data memory.Each of the register is of 16-bits width capacity.

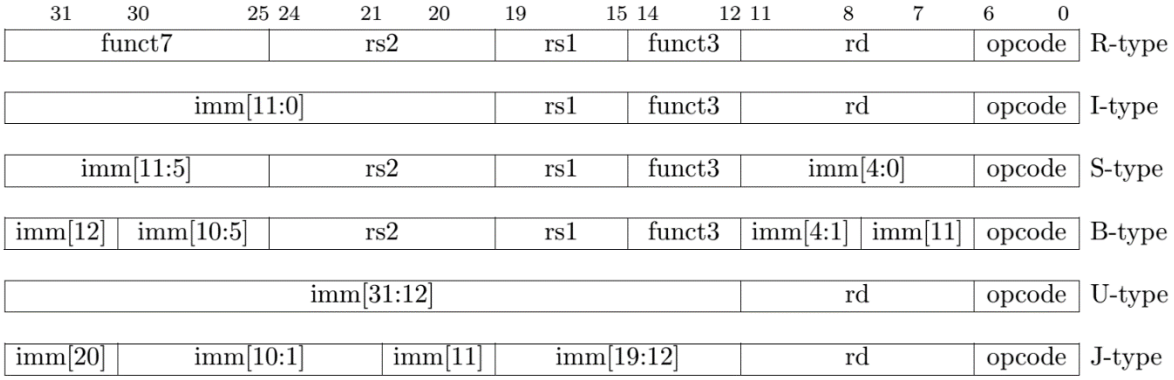
Program Counter: The Program Counter (PC) is a 16-bit latch that holds the memory address of location, from which the next machine language instruction will be fetched by the processor. The proposed PC is the largest sub-block and second to the control unit in complexity.

Arithmetic and Logic unit: The arithmetic and logic unit (ALU) performs arithmetic and logic operations. It also performs the bit operations such as rotate and shift by a defined number of bit positions. The proposed ALU contains three sub-modules, viz. arithmetic, logic and shift modules. The arithmetic unit involves the execution of addition operations and generates Sign flag and Zero flag as per the result shown in the process. In order to reduce the complexity of the adder circuits used in the arithmetic unit of the RISC CPU, a very fast and low power carry select adder circuit has been introduced. The ALU also consists of a modified Wallace tree multiplier, which uses compressor circuits to achieve low power and improved speed of operation. The multiplier is designed to execute

in a single cycle. Hence, it satisfies the requirement of the RISC design, to execute single cycle instructions.

INTRUCTION FORMAT:

The design of RISC-V instruction sets is modular. Rather than take the approach of a large and complex monolith, a modular design enables flexible implementations that suit specific applications. RISC-V defines base user-level integer instruction sets. Additional capability to these are specified as optional extensions, thus giving implementations flexibility to pick and choose what they want for their applications. The specifications of the base ISA has been frozen since 2014. Some of the extensions are also frozen while many others are being defined



- RISC-V comprises of a base user-level 32-bit integer instruction set. Called **RV32I**, it includes 47 instructions, which can be grouped into six types:

- R-type: register-register
- I-type: short immediates and loads
- S-type: stores
- B-type: conditional branches, a variation of S-type
- U-type: long immediates
- J-type: unconditional jumps, a variation of U-type

RV32I has x0 register hardwired to constant 0, plus x1-x31 general purpose registers. All registers are 32 bits wide but in RV64I they become 64 bits wide. RV32I is a **load-store architecture**. This means that only load and store instructions access memory; arithmetic operations use only the registers. User space is 32-bit byte addressable and little endian.

Correspondingly, **RV64I** is for 64-bit address space and **RV128I** is for 128-bit address space. The need for RV128I is debatable and its specification is evolving. We also have **RV32E** for embedded systems. RV32E has only 16 32-bit registers and makes the counters of RV32I optional.

Base	Version	Frozen?
RV32I	2.0	Y
RV32E	1.9	N
RV64I	2.0	Y
RV128I	1.7	N
Extension	Version	Frozen?
M	2.0	Y
A	2.0	Y
F	2.0	Y
D	2.0	Y
Q	2.0	Y
L	0.0	N
C	2.0	Y
B	0.0	N
J	0.0	N
T	0.0	N
P	0.1	N
V	0.2	N
N	1.1	N

Versions of ISA base and extensions. Source: Waterman and Asanović 2017, pg. i.

RISC-V defines a number of extensions, all of which are optional. Some of them are frozen and these are noted below:

- M: Integer multiplication and division.
- A: Atomic.
- F: Single-precision floating point compliant with IEEE 754-2008.
- D: Double-precision floating point compliant with IEEE 754-2008.
- Q: Quad-precision floating point compliant with IEEE 754-2008.

- C: Compressed instructions (16-bit instructions) to yield about 25-30% reduced code size. "RVC" refers to compressed instruction set. Among the evolving or future extensions are L (decimal float), B (bit manipulation), J (dynamically translated languages), T (transactional memory), P (packed SIMD), V (vector operations), N (user-level interrupts), and H (hypervisor support). When multiple extensions are supported, that ISA variant can be described by concatenating the letters; such as, RV64IMAFD. To represent the standard general purpose ISA, "G" is defined as a short form for "IMAFD". RV32I uses one-eighth of the encoding space. This means there's plenty of room for custom extensions.

ALU Design: Mainly ALU design have LHI, LLI ,Xor, left shift, right shift, adder ,multiplier and halt. ALU by optimizing the design of arithmetic circuits.

RESULTS:

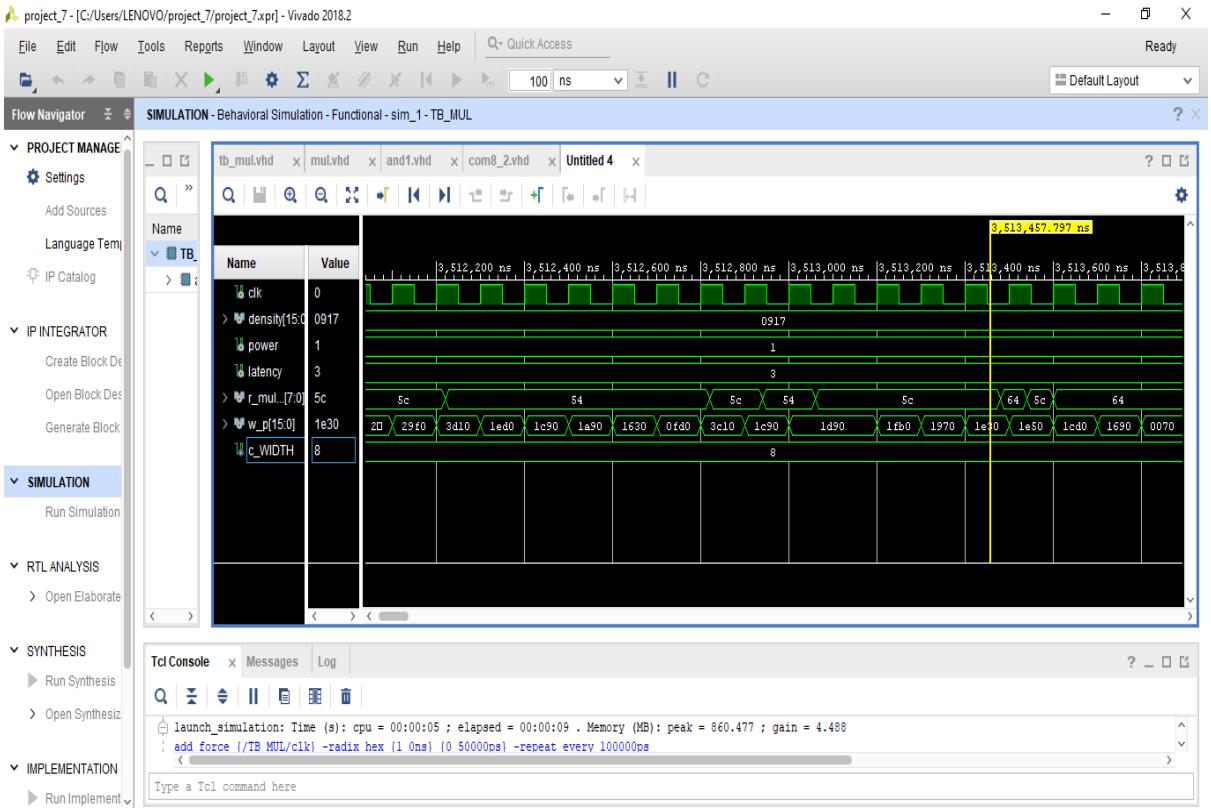


Fig: a Proposed Simulation result

ADVANTAGES:

- RISC(Reduced instruction set computing)architecture has a set of instructions, so high-level language compilers can produce more efficient code
- It allows freedom of using the space on microprocessors because of its simplicity.
- Many RISC processors use the registers for passing arguments and holding the local variables.
- RISC functions use only a few parameters, and the RISC processors cannot use the call instructions, and therefore, use a fixed-length instruction that is easy to pipeline.

- The speed of the operation can be maximized and the execution time can be minimized. A very less number of instructional formats, a few numbers of instructions, and a few addressing modes are needed.

APPLICATIONS: RISC is used in high-end applications like video processing, telecommunications, and image processing. Laser printer raster image processing. ARM, in partnership with Apple, developed a low-power design and then specialized in that market, which at the time was a niche. With the rise in mobile computing, especially after the introduction of the iPhone, ARM became the most widely used high-end CPU design in the market.

CONCLUSION: The design of a single cycle 32bit RISC processor has been presented. A Low latency adder and multiplier structures have been employed in the RISC architecture. The processor has been designed for executing based on the user requirements.

FUTURE ENHANCEMENT Finally, we can expect more and more suppliers to create complex RISC-V cores for high-performance computing in the future. Floating point complex arithmetic units for double precision floating architectures will be expecting in future.

REFERENCES:

[1] David A. Patterson, John L. Hennessy, "Computer Organization and Design-The Hardware/ Software Interface" Second Edition (1998) Morgan Kaufmann Publisher, Inc.

[2] Xiao Li, Longwei Ji, Bo Shen, Wenhong Li, Qianling Zhang, "VLSI implementation of a Highperformance 32-bit RISC Microprocessor", Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on, Volume 2, 2002, pp. 1458 – 1461.

[3] Kusumlata Pisda, Deependra Pandey, "Realization & Study of High Performance MIPS RISC Processor Design Using VHDL", International Journal of Emerging trends in Engineering and Development, Volume 7, Issue 2, November 2012, pp. 134 – 139, ISSN: 2249 – 6149.

[4] Kirat Pal Singh, Shivani Parmar, "VHDL Implementation of a MIPS – 32 bit Pipeline Processor", International Journal of Applied Engineering Research, Volume 7, Issue 11, ISSN: 0973 – 4562.

[5] Samiappa Sakthikumaran, S.Salivahanan and V.S.Kaanchana Bhaaskaran,"16-Bit RISC Processor Design For Convolution Application",IEEE International Conference on Recent Trends In Information Technology, June 2011, pp.394-397.

[6] Rupali S. Balpande and Rashmi S. Keote, "Design of FPGA based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor, International Conference on Communication Systems and Network Technologies pp. 409 – 413.

[7] R. Uma, "Design and Performance Analysis of 8 – bit RISC Processor using Xilinx Tool", International Journal of Engineering Research and Applications, Volume 2, Issue 2, March – April 2012, pp. 053 – 058, ISSN: 2248 – 9622.

[8] V.N.Sireesha and D.Hari Hara Santosh, "FPGA Implementation of a MIPS RISC Processor", International Journal of Computer Technology and Applications, Volume 3, Issue 3, pp. 1251 – 1253, ISSN: 2229 – 6093.

[9] M.Yugandhar and N.Suresh babu, "VLSI Design of Reduced Instruction set Computer Processor Core Using VHDL", International Journal of Electronics, Communication & Instrumentation Engineering Research and Development (IJECIERD), Volume 2, Issue 3, pp. 42 – 47, ISSN: 2249 – 684X.

[10]Kui YI, Yue-Hua DING, "32-bit RISC CPU Based on MIPS Instruction Fetch Module Design", 2009 International Joint Conference on Artificial Intelligence, 978-0-7695-3615-6/09, 2009 IEEE.

[11] Seung Pyo Jung, Jingzhe Xu, Donghoon Lee,Ju Sung Park,Kangjoo, Kim, Koon-shik Cho" Design & Verification of 16 Bit RISC Processor "International SoC Design Conference 2008.

[12] R. Uma."Design and Performance Analysis of 8-bit RISC Processor using Xilinx Tool,"International Journal of Engineering Research and Applications(IJERA) Vol2,Issue 2,Mar-Apr 2012.

[13] Mr. Prashant Bhirange, "Hardware Implementation High Speed RISC Processor for Convolution ," International Conference on Recent Trends in Engineering Science and Technology(ICRTEST 2017)

[14] Vivek Dubey,Prof Ravi Mohan,"Architecture for RISC Processor with CISC insturction" International Journal of Engineering Research & Management Technology May-2014 Volume 1,Issue 3.

[15] Priyanka Trivedi, Rajan Prasad Tripathi , "Design & Analysis of 16 bit RISC Processor Using low Power Pipelining", International Conference on Computing, Communication and Automation (ICCCA2015).

[16] Akshatha S Patil,B G Shivaleelavathi.,"Design and Implementation of Pipelined 8-Bit RISC Processor using Verilog HDL on FPGA"International Research Journal of Engineering and Technology (IRJET),Volume: 04 Issue: 06 — June -2017

[17] Vishwas V. Balpande*, Abhishek B. Pande, Meeta J. Walke, Bhavna D. Choudhari, Kiran R. Bagade, "Design and Implementation of 16-Bit Processor on FPGA ",International Journal of Advanced Research in Computer Science and Software Engineering,Volume 5, Issue 1, January 2015.

[18] Jari Nurmi,Processor Design,System-on-Chip Computing for ASICs and FPGAs,Springer [9] Samir Palnitkar, Verilog HDL: A Guide to Digital Design and Synthesis.