

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 10, October 2017

## RISC(16 Bit)Processor Design using Verilog in Modelsim

Jaina Patel<sup>1</sup>

U.G. Student, Department of ECE, L.D. College of Engineering, Ahmedabad, Gujarat, India<sup>1</sup>

**ABSTRACT:** The Reduced Instruction Set Computer or the RISC, is a microprocessor CPU design philosophy that provides complexity of instructions implemented at the hardware level since it is easier to lend flexibility and smartness in software than the hardware. As a result, a RISC design has smaller and simpler set of instructions such that almost all can get executed in a single clock cycle. The most widely used RISC processors are ARM, IBM's PowerPC, MIPS, RISC-V, Adapteva Epiphany, DEC Alpha, PA-RISC, and SPARC. The CISC processors which are complex instruction set computers and sometimes called "Complete" instruction set computers since they contain every kind of instruction. How many of these instructions are used and how often is another matter. The limited number of instructions in RISC processor increases the code and so needs more memory but that is not a problem because memory is cheap.

**KEYWORDS:** RISC (16 bit), Veriolog and Modelsim.

### I. INTRODUCTION

An intervening characteristic of CISC machines is its approach to the instruction set architecture that emphasizes doing more with each instruction. As a result, CISC machines have a varied range of addressing modes. The capabilities of CISC allows programmer to perform more operations in the same program size. During those days, more importance was given to the program and data storage since cost of memory was high. As a result, an attempt was made to narrow the semantic gap, that is, the gap that prevailed between the instruction sets and the high level language constructs, with complicated instructions and addressing modes to increase performance.<sup>[5]</sup> A number of systems, going back to the 1960s, have been credited as the first RISC architecture, partly based on their use of load/store approach.<sup>[5]</sup> The term RISC was coined by David Patterson of the Berkeley RISC project, although somewhat similar concepts had appeared before.

The question that now arises is why RISC design? Since the earliest day of computing, various attempts were made to increase the execution rates of the instruction by overlapping the execution of more than one instruction. The most common ways of overlapping are pre-fetching, pipelining and superscalar operation.

1) Pre-fetching: It is a process of fetching next instruction or instructions into an event queue before the current instruction is completed.<sup>[4]</sup> The earliest 16-bit microprocessor, the Intel 8086/8, pre-fetches into a non-board queue up to six bytes following the byte currently being executed thereby making them immediately available for decoding and execution, without latency.

2) Pipelining: Before the current instruction has executed completely, starting or issuing another instruction is called Pipelining. The current generation of machines carries this to a substantial extent.<sup>[4]</sup> The PowerPC 601 has 20 separate pipeline stages in which various portions of various instructions are executing simultaneously.

3) Superscalar operation: Issuing more than one instruction simultaneously is called Superscalar operation.<sup>[4]</sup> The PPC 601 has independent integer, floating-point and branch units, each of which can be executing an instruction simultaneously.

CISC machine designers merged all the above mentioned overlapping methods in their designs however with those instructions that were long and complex, it was difficult to make efficient use of these new speed-up techniques.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 10, October 2017

Additionally, complex instructions and addressing modes hold down clock speed compared to simple instructions. RISC machines were so designed that they competently exploit the caching, pre-fetching, pipelining and superscalar methods that were invented in the days of CISC machines.

## II. RISC ARCHITECTURE

All RISC processors, use a load-store architecture. This means it has two instruction types for transferring data in and out of the processor: load instructions copy data from memory to registers in the core, and conversely the store instructions copy data from registers to memory. There are no data processing instructions that directly manipulate data in memory. Thus, data processing is carried out solely in registers. The basic architecture of a RISC processor is as given below

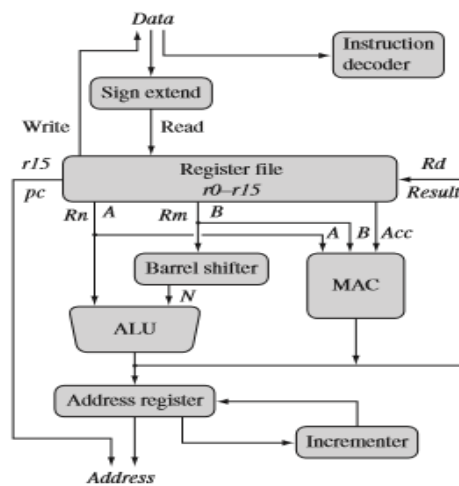


Fig.a RISC Processor

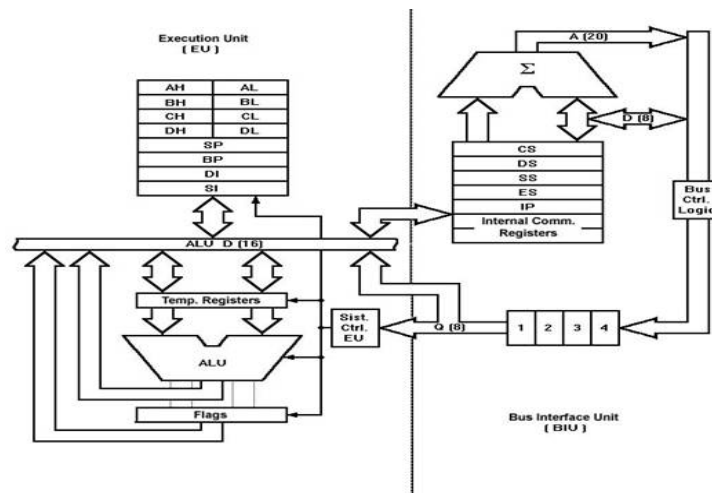


Fig.b Architecture of a 16 bit RISC processor

## International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 10, October 2017

Each of these modules namely Program Counter, Instruction Register and Register File, ALU and Operand Registers, Control Unit Design have their own micro-architecture

### III. RISC: TOP LEVEL DESCRIPTION

Implementation of an 8/16-bit RISC microprocessor is possible based on a simplified version of the MIPS architecture. The processor that I have designed has 16-bit instruction words and 16 general purpose registers. Every instruction is completed in four cycles. An external clock is used as the timing mechanism for the control and data-path units. Along with this 16 external address lines are also available.

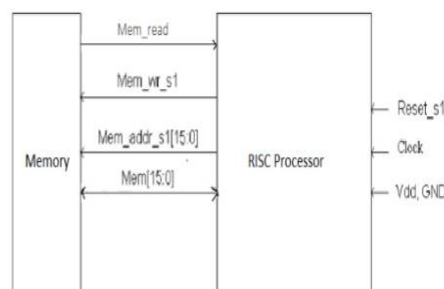


Fig.cBlock Diagram of the external interface of the chip

- The input signals to the processor are: Vdd= Power, GND= Ground, Clock= Clock, Reset = reset Mem\_read= Read from Memory.
- The output signals are- Mem\_wr\_s1= Chip write Data to memory. Mem\_addr\_s1[15:0]= 16 bits address line to address the main memory.
- Input/Output signals- Mem[15:0]= 16 bit address line to address the main memory.

### IV. APPROACH

To implement the 16 bit RISC processor I have used Verilog Language and Modelsim Application.

- Verilog – In electronic circuits and systems, to describe a textual format we use Verilog, which is a Hardware Description Language. Applied to electronic design, Verilog is intended to be used for verification through simulation, for timing analysis, for test analysis (testability analysis and fault grading) and for logic synthesis. It is also used in the verification of analog and mixed signal circuit. A Verilog design contains of a hierarchy of modules.
- Modelsim- ModelSim is a multi-language HDL simulation environment by Mentor Graphics,<sup>[1]</sup> for simulation of hardware description languages such as VHDL, Verilog and SystemC, and includes a built-in C debugger.

# International Journal of Innovative Research in Science, Engineering and Technology

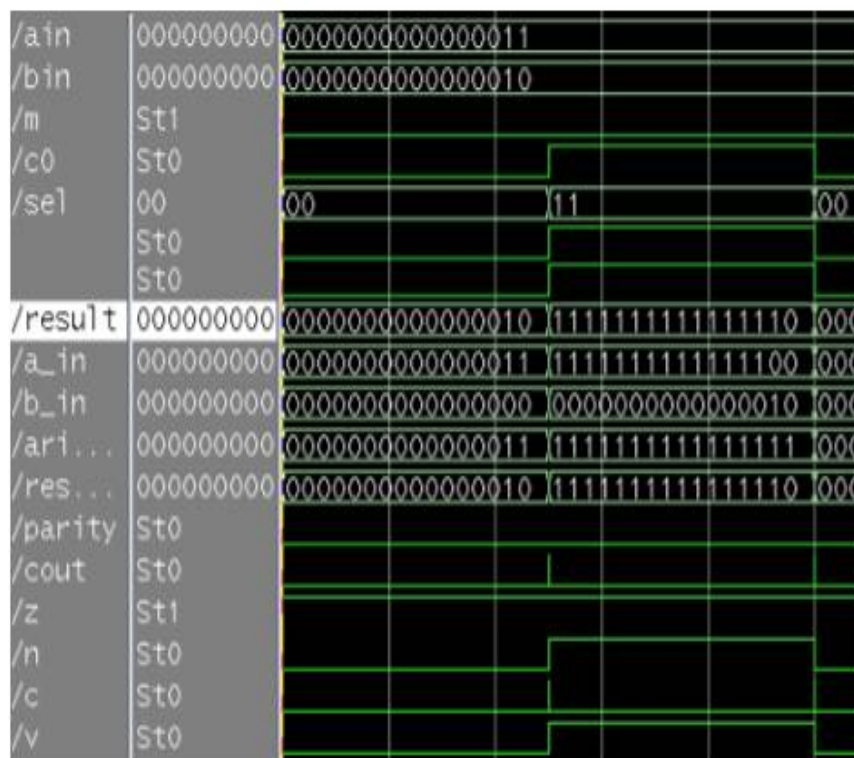
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 10, October 2017

## V.RESULT

The Final Modelsim results are as below



## VI.CONCLUSION

In this work, a 16 bit RISC processor is implemented with high performance and throughput being the main objective. This processor comprises of 4 main elements- Control unit, General Purpose Registers, Arithmetic and logical unit (ALU), shift registers. Control unit follows 3 stage instruction cycle- fetch, decode and execute. Every instruction has been executed in one clock cycle using these 3 pipelining stage. The control unit has two registers namely instruction register and instruction decoder. Instruction and data are fetched chronologically in order to reduce the inactivity in the machine cycle. Here verification has been achieved by Modelsim simulation. Thus this processor can be used for mathematical calculation in manageable calculators as well as for gaming tools.

## REFERENCES

- [1] Arif Ahmed, Arnab Das, Iqbalur Rahman Rokon, Md. Abu Obaidah, "RISC (16-bits) Processor Design Using Verilog & FPGA," 8th International Conference on Computer Science and Information Technology (ICCSIT'2016) Jan. 12-13, 2016 Dubai (UAE).
- [2] Ramandeep Kaur, Anuj. "8 Bit RISC Processor Using Verilog HDL", Int. Journal of Engineering Research and Applications ISSN: 2248-9622, Vol. 4, Issue 3 (Version 1), March 2014, pp.417-422.
- [3] AndrewN.Sloss, DominicSymes, ChrisWright, "ARM System Developer's Guide- Designing and Optimizing System Software."
- [4] A REPORT ON DESIGN OF 16 – BIT RISC PROCESSOR, under the guidance of Dr. Chandra Shekhar, Director, CEERI, Pilani.
- [5] [https://en.wikipedia.org/wiki/Reduced\\_instruction\\_set\\_computer#History\\_and\\_development](https://en.wikipedia.org/wiki/Reduced_instruction_set_computer#History_and_development).



ISSN(Online): 2319-8753  
ISSN (Print): 2347-6710

# International Journal of Innovative Research in Science, Engineering and Technology

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

Visit: [www.ijirset.com](http://www.ijirset.com)

**Vol. 6, Issue 10, October 2017**

- [6] Antonio Hernández Zavala, Oscar Camacho Nieto, Jorge A. Huerta Ruelas, Arodí R. CarvalloDomínguez, “Design of a General Purpose 8-bit RISC Processor for Computer Architecture Learning”, *Computación y Sistemas*, Vol. 19, No. 2, 2015, pp. 371–385 ISSN 1405-5546.
- [7] RISC Processors, [www.Williams.comp.ncat.edu](http://www.Williams.comp.ncat.edu), “Superscalar Architectures.”
- [8] Modelsimhelp Manual from Mentor Graphics.
- [9] B. Rajesh Kumar, Ravisaketh and Santha Kumar, 2014, “Implementation of A 16-bit RISC Processor for Convolution Application”, *Research India publications*, pp 441-446
- [10] CMOS Digital Integrated Circuits by Sung-Mo Kang and Yusuf Leblebici.
- [11] Yatin Trivedi and others, Verilog HDL, IC, 2000.
- [12] Berkeley, E.C. & Jensen, R.A. (1950). World’s Smallest Electric Brain. *Radio-Electronics*, Oct. 1950.