

Design and Implementation of 32 Bit RISC Processor Using XILINX

Asst. Prof. B. Ashok, Yadlapalli Bhargavi, Velagapudi Sirisha, Musunuri Murali Gopala Krishna

Department of ECE,
Chalapathi Institute of Engineering and Technology,
Lam, Andhra Pradesh, India.

Abstract- These RISC or Reduced Instruction Set Computer is a design philosophy that has become a mainstream in Scientific and engineering applications. The main objective of this paper is to design and implement of 32 – bit RISC (Reduced Instruction Set Computer) processor using XILINX VIRTEX4 Tool for embedded and portable applications. The design will help to improve the speed of processor, and to give the higher performance of the processor. The most important feature of the RISC processor is that this processor is very simple and support load/store architecture. The important components of this processor include the Arithmetic Logic Unit, Shifter, Rotator and Control unit. The module functionality and performance issues like area, power dissipation and propagation delay are analyzed using Virtex4 XILINX tool.

Keywords- RISC (Reduced Instruction Set Computer) Processor, Arithmetic Logic Unit, Shifter, Rotator and Control unit.

I. 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. Verilog HDL has evolved as a standard hardware description language. A hardware descriptive language is a language used to describe a digital system.

HDL's allows the design to be simulated earlier in the design cycle in order to correct errors or experiment with different architectures. Designs described in HDL are technology independent, easy to design and debug, and are usually more readable than schematics, particularly for large circuits.

More recently Verilog is used as an input for synthesis programs which will generate a gate-level description for the circuit. The simulator which is used for the language is Xilinx ISE and Modelsim. Verilog is capable of describing simple behaviour. Machine cycle instructions allow the processor to handle several instructions at the same time. The processor can work at a high clock frequency and thus yields higher speed. This paper is about design of a simple RISC processor and synthesizing it. The

RISC architecture follows single-cycle instruction execution.

II. LITERATURE SURVEY

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. The design is modeled and simulated using VHDL and then implemented on FPGA successfully. The maximum frequency of operation on the Xilinx's Spartan-II FPGA is 26-MHz[3].

Soumya Murthy, Usha Verma, "FPGA based Implementation of Power Optimization of 32 Bit RISC Core using DLX Architecture," By using fetch, decode, ALU, comparator, GPR memory, execute, pipelined RISC processor core is developed using DLX architecture. Using low power technique i.e. verilog HDL modification a lower version of the processor is designed to reduce power consumption of the core. Lower version of the processor is designed to reduce power consumption of the core. The overall optimization achieved from HDL technique is 13.33%[4].

Mohit N. Topiwala, N. Saraswathi, "Implementation of a 32-bit MIPS Based RISC Processor using Cadence," In this paper, design of 32-bit MIPS based RISC processor is implemented successfully with pipeline functionalities. Every instruction is executed in one clock cycle with 5-stage pipe lining.

This design shows the implementation of MIPS based CPU capable of handling various R-type, J-type and I-type of instruction and each of these categories has a different format.

These instructions are verified successfully through testbench. Designing Forwarding unit and hazard detection unit to overcome the data dependencies was critical task and it was implemented successfully. The design is implemented using VerilogHDL and synthesized using Cadence RTL compiler using

typical libraries of TSMC 0.18 urn technology. Design of MIPS processor is optimized both in timing and area. Also complete ASIC flow till RTL to GDS II have done using Cadence SoC Encounter, and analyzed the complete physical design flow[5].

III. PROPOSED SYSTEM

The below figure (1) shows the architecture of proposed system. The RISC processor architecture consists of Arithmetic Logic Unit (ALU), Control Unit (CU), Barrel Shifter, Booth's Multiplier, Register File and Accumulator. RISC processor is designed with load/store (Von Neumann) architecture, meaning that all operations are performed on operands held in the processor registers and the main memory can only be accessed through the load and store instructions.

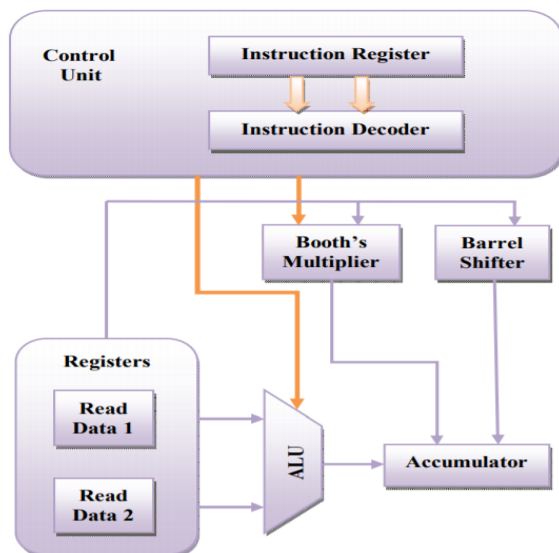


Fig 1. Proposed System.

1. Control Unit:

The control unit of the RISC processor examines the instruction opcode bits and decodes the instruction to generate nine control signals to be used in the additional modules. The RegDst control signal determines which register is written to the register file. The Jump control signal selects the jump address to be sent to the PC. The Branch control signal is used to select the branch address to be sent to the PC. The MemRead control signal is asserted during a load instruction when the data memory is read to load a register with its memory contents.

The MemtoReg control signal determines if the ALU result or the data memory output is written to the

register file. The ALUOp control signals determine the function the ALU performs. (e.g. and, or, add, sbu, slt) The MemWrite control signal is asserted when during a store instruction when a registers value is stored in the data memory. The ALUSrc control signal determines if the ALU second operand comes from the register file or the sign extend. The RegWrite control signal is asserted when the register file needs to be written

2. Arithmetic Logic Unit (ALU):

The arithmetic/logic unit (ALU) executes all arithmetic and logical operations. The arithmetic/logic unit can perform arithmetic operations or mathematical calculations like addition, and subtraction. As its name implies, the arithmetic/logic unit also performs logical operations include Boolean comparisons, such as AND, OR, XOR, NAND, NOR and NOT operations.

3. Barrel Shifter:

The design consists of a total of eight 8x1 multiplexers. The output of one multiplexer is connected as input to the next multiplexer in such a way that the input data gets shifted in each multiplexer thus performing the rotation operation. Depending on the select lines the number of rotation varies. With select lines low there is no output. If select line c0 is high 1-bit rotation takes place, if c1 is high 3-bit rotation.

4. Booth's Multiplier:

The Multiplier is implemented using the modified Booth algorithm. The two main advantages of this algorithm are speed and the ability to do signed multiplication (using two's complement) without any extra conversions. The multiplier is implemented using the modified booth algorithm. The multiplier is scanned sequentially from right to left. In this case, however, three adjacent bits are examined during each step of the procedure. According to the value of the three bits, the multiplicand is added to or subtracted from the accumulated partial product and the later is then shifted.

5. General Purpose Register:

The eight bit input data is stored in this register. This register acts as a source register. The register file has two read and one write input ports, meaning that during one clock cycle, the processor must be able to read two independent data values and write a separate value into the register file. The register file

was implemented in VHDL by declaring it as a one-dimensional array of 32 elements or registers each 8-bit wide. It consists of eight D – flip flops and eight AND gates.

IV. RESULTS

Table 1. Delay, Total Power and Area Calculation.

Topology	Delay (ns)	Slices Utilized (Area)	AT	Power Dissipation $10^{-3}W$	Total Power (W)
Control Unit	0.905	15	13.575	0.159	0.165
ALU	4.495	60	269.7	0.159	0.168
Barrel Shifter	2.905	262	761.11	0.159	0.164
Booth's Multiplier	4.495	60	269.7	0.159	0.166
Register File	5.443	74	402.782	0.159	0.166
Total	18.243	471	8592.453		0.829

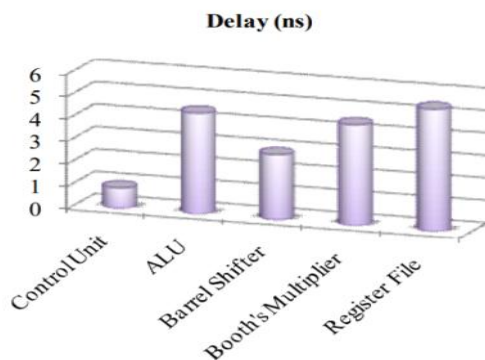


Fig 2. Delay Estimation.

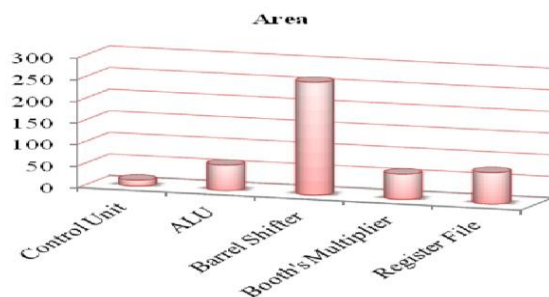


Fig 3. Area Estimation.

V. CONCLUSION

A 32-bit RISC processor with 16 instruction set has been designed. Every instruction is executed in one clock cycles with 5-stage pipelining.

The design is verified through exhaustive simulations. The processor achieves higher performance, lower area and lower power dissipation.

This processor can be used as a systolic core to perform mathematical computations like solving polynomial and differential equations. Apart from this it can be used in portable gaming kits.

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 High-performance 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 (IJEIERD), 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.