



ANALYSIS OF 16 BIT RISC PROCESSOR USING LOW POWER PIPELINING

Shweta S. Chimurkar

M.Tech Student
Department Of Electronics Engineering
Priyadarshini College of Engineering
Nagpur, India
Chimurkarshweta25@gmail.com

Prof. P. J. Suryawanshi

Assistant Professor
Department Of Electronics Engineering
Priyadarshini College of Engineering
Nagpur, India
pradnyajs@rediff.com

Abstract— The aim of the paper is to design a 16-bit RISC processor. It is having four stages pipelining which is designed using VHDL. RISC processors have a unique feature called pipelining. 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 is designed for these processors. Multiplier is also designed using ADD instruction. Proposed instructions are simulated using Xilinx ISE 13.2.

A 16-bit low power pipelined RISC processor is proposed by us in this project, the RISC processor consists of the block mainly ALU, program counter, ram, instruction memory, 2:1 mux. This project presents the design and implementation of a low power pipelined 16-bit RISC Processor. In this project, low power technique is proposed in front end process. Low power consumption helps to reduce the heat dissipation, lengthen battery life and increase device reliability.

Keywords— RISC, VHDL, Pipelining, Low Power, Xilinx

I. INTRODUCTION

The processors are characterized by nature of their instruction set architecture. There are basically two ways of designing instruction sets CISC and RISC. RISC stands for reduced instruction set computer. It has faster and simpler set of instructions. The main feature of RISC is a pipeline. It has simple addressing modes and fixed length of instructions. RISC processor reduces cycles per instruction at the cost of a number of instructions per program. Reduced instructions in RISC require less number of transistors. RISC is used in portable devices such as Apple iPod due to its power efficiency.

The RISC architecture boost the computer speed also used in control algorithms. Using of RISC Processor the time required to execute each instruction can be shortened

and the number of cycles reduces. This paper depicts a RISC building design in which 2 cycle operation is gotten utilizing a pipelined outline. The pipelined architecture is used which minimizes the latency and increases the speed and in the new innovation the 4 stage pipelining which works on the positive edge and as well as in the negative edge minimizes the latency and increases the speed and also reduce the stalling in instruction. The whole architecture of RISC processor work on the 2 cycle. the fixed size of instruction allows the given instruction to be easily piped. RISC processor has a flexible architecture. The clock gating technique is used to minimize the PoWritIn the present research work, the outline of 16 bit RISC processor is introduced; implemented for high efficiency and low power. The architecture supports 33 instructions. The instruction cycle consist of 4 stage pipelining and perform fetch, decode, execute, write back operation simultaneously. The control unit Generate signals from the given instructions. The architecture supports arithmetic, logical, shifting and rotation operations.



RISC (Reduced instruction set computing) :

The RISC Processor have reduced number of Instructions, fixed instruction length, more general purpose registers, load-store architecture and simplified addressing modes which makes individual instructions execute faster,



achieve a net gain in performance and an overall simpler design with less silicon consumption as compared to CISC. 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 following features typically found in RISC based systems.

1) Pre-fetching: The process of fetching next instruction or instructions into an event queue before the current instruction is complete is called pre-fetching.

2) Pipelining: Pipelining allows issuing an instruction prior to the completion of the currently executing one.

3) Superscalar operation: Superscalar operation refers to a processor that can issue more than one instruction simultaneously.

VHDL:

VHDL stands for very high-speed integrated circuit hardware description language. This is one of the programming languages used to model a digital system by dataflow, behavioral and structural style of modeling. This language was first introduced in 1981 for the department of Defense (DoD) under the VHSIC program. In 1983 IBM, Texas instruments and Intermetrics started to develop this language. In 1985 VHDL 7.2 version was released.

II. METHODOLOGY

The block consists of four stage pipelining: Instruction fetch, Instruction decode, Execute, Memory Read/Write Back. Its comprises of low power unit. The block of the pipelined RISC processor is shown in the above Figure 1. Arithmetic operations either take two registers as operands or take one register and a sign extended immediate value as an operand. Some arithmetic instructions are ADD, SUB and MUL. The result is stored in a third register. Logical operations such as AND OR, XOR do not usually differentiate between 32-bit and 64-bit. Other logical instructions are NAND, NOR, NOT etc. Load/Store Instructions usually take a register (base register) as an operand and a 16-bit immediate value. The sum of the two will create the effective address. A second register acts as a source in the case of a load operation. In the case of a store operation the second register contains the data to be stored. Examples are: LW, SW etc. Branches and Jumps instruction: Conditional branches are transfer of control. A branch causes an immediate value to be added to the current program counter. Some common branch instructions are BZ (Branch Zero), BRZ (Branch Register Zero), JMP (Jump Instruction), JMPZ (Jump when zero) etc.

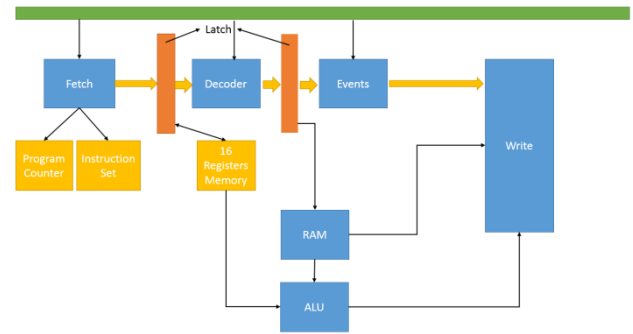


Fig.1. Block Diagram of RISC

Detail of logical blocks

Control unit-

The control unit comprises in 2 parts of operations. In our proposed architecture the control unit work as all the coordination needed between the entire components of the processor it creates the desired control signals. It is responsible for generating the signals that decide to use.

Fetch-

Instruction memory- in the fetch 33 bit of instruction is drawn from instruction memory,

in the instruction memory it contains the instruction that are executed by the processor.

Program counter-

The program counter contains the address of the instruction that will be fetched from the instruction memory during the next clock cycle. The pc is incremented by one during each given clock cycle. The fetch unit is work in the positive edge of the cycle.

Decode-

In the decoder consist of 2 addressing mode as input immediate addressing mode or a register addressing mode the three operation ALU ,universal shifter ,shifter rotator are performed. The two decoder are exist in the control unit the first decoder perform the arithmetic and logical operation And second decoder performs rotating and shifting operations. The decoder operation is performed in negative cycle.

Execute Unit –

The execution stage is where actual computation takes place. All the operations that are perform by the given control signals are executed. The ALU operation, shifter operation and shifter rotator operation are performed exactly given control signal. This operation is performed in the positive clock edge of the cycle.

Write back-



In this stage, their result is written into the register file by both single and two cycle instructions.

III. INSTRUCTION SET

The instructions can be classified into the following five functional categories: data transfer operations, arithmetic operations, logical operations, branching operations and control operations.

Data Transfer Operations

This group of instructions copies data from a location called source to another location called destination without modifying the contents of the source. MVI, LOAD and STORE instructions come under this category.

Arithmetic Operations

These instructions perform arithmetic operations such as addition (ADD), subtraction (SUB), multiplication (MUL), increment (INC) and decrement (DEC).

Logical Operations

These instructions perform various logical operations such as AND, OR, XOR, NOT, SHL, SHR.

Branching Operations

The instruction JUMP alters the sequence of program execution Control Operations The instruction HLT does not produce any result but stops the program execution.

IV. PERFORMANCE PARAMETER

2:1 MUX:

In electronics, a multiplexer (or mux) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A multiplexer of inputs has select lines, which are used to select which input line to send to the output.

Instruction Memory:

The Instruction Memory contains the instructions that are executed by the processor. The input to this unit is a 9-bit address from the program counter and the output is 9-bit instruction word.

Program Counter:

The program counter (PC) contains the address of the instruction that will be fetched from the Instruction memory during the next clock cycle. Normally, the PC is incremented by one during each clock cycle unless a branch instruction is executed. When a branch instruction is encountered, the PC is incremented by the amount indicated by the branch offset.

Arithmetic Logic Unit (ALU):

The ALU is responsible for all arithmetic and logic operations that take place within the processor. These operations can have one operand or two, with these values coming from either the register file or from the immediate value from the instruction directly. All operations are done according to the control signal coming from ALU control unit.

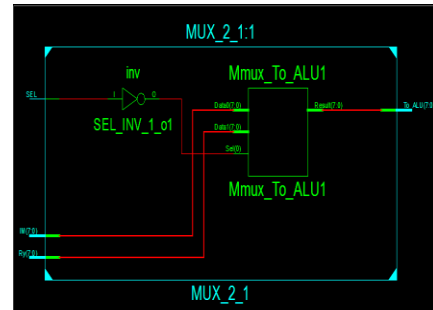


Fig.2. RTL Schematic of 2:1 MUX

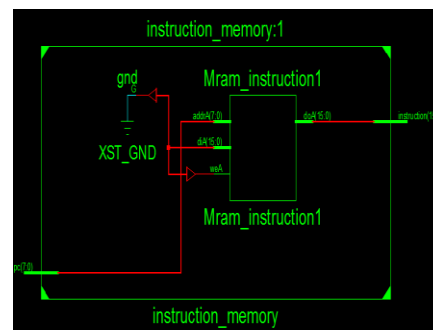


Fig.3. RTL Schematic of Instruction Memory

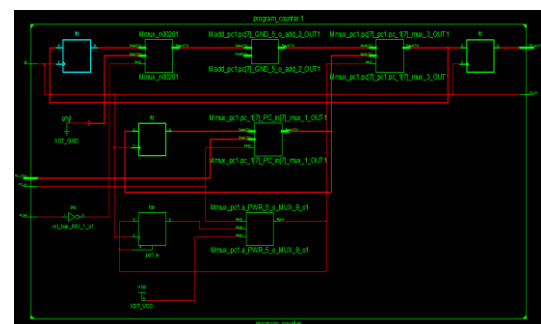


Fig.4. RTL Schematic of Program Counter

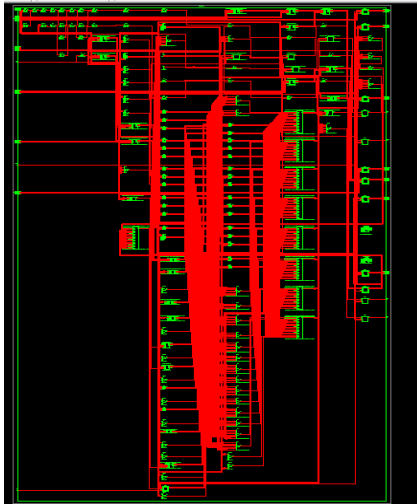


Fig.5. RTL schematic view of ALU

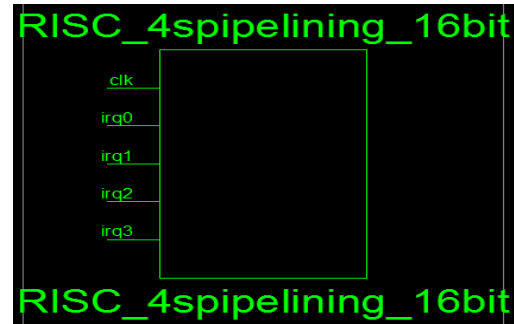


Fig.6. Technology Schematic of RISC Pipelining

Features of proposed architecture

- It is 4- stage low power pipelining processor.
- It provides hazard detection unit to determine when stall must be added.
- Each functional unit can only be used once per instruction.
- Each functional unit has to be used at the same stage as all other instructions.
- Pipeline Control: that is each stage control signal depends only on the instruction that is currently in that stage.
- It provides halt support.
- It is capable of providing high performance.
- Pipelining would not flush when branch instruction occurs as it is implemented using dynamic branch prediction.
- Take cares of order-of-execution

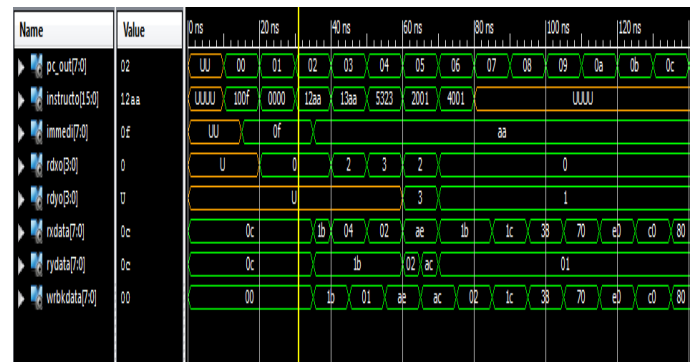


Fig.7. Simulation Waveform of Risc Pipelining

V. SIMULATION AND RESULTS

Modelsim is used for simulation and results have been verified. The complete project has been developed on RISC, VHDL, Pipelining, and Low Power. Xilinx 13.2 for results. The 16 bit RISC Processor using 4 stage pipeline that reduces the latency and increases the speed. A low power design technique was employed to reduce the power consumption .This method drastically reduces the dynamic power & the quiescent power is reduced total power consumption which is very less than the conventional & also final latency comes out cycle.

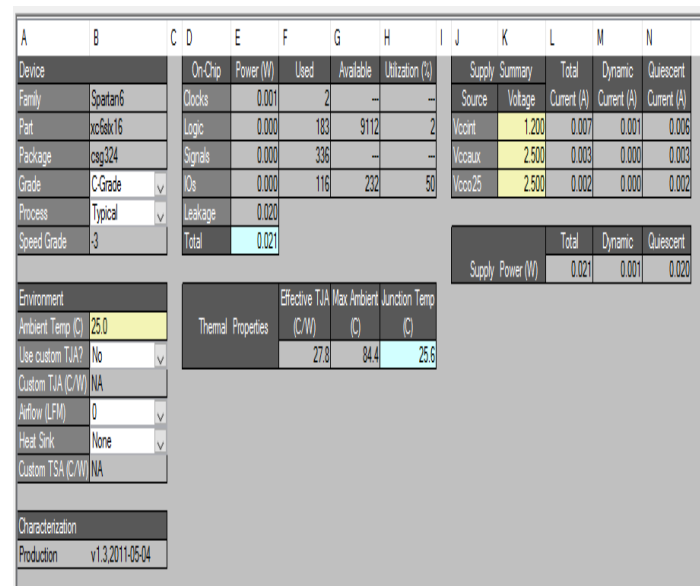


Fig.8. Power Report

VI. CONCLUSION

The design of 4-stage 16-bit low power RISC processor performing arithmetic, logical, memory and branch instructions is presented in this project. By increasing the number of instructions and make a pipelined design with less clock cycles per instruction and more improvement can be added in the future work.



VII. REFERENCES

- [1] Neenu Joseph, Sabarinath.S, Sankarapandiammala K, "FPGA based Implementation of High Performance Architectural level Low Power 32-bit RISC Core" International Conference on Advances in Recent Technologies in Communication and Computing, Annual IEEE pp.53-57, 2009
- [2] Shofiquil Islam, Debanjan Chattopadhyay, Manoja Kumar Das, V Neelima; Rahul Sarkar, "Design of High Speed Pipelined Execution Unit of 32-bit RISC Processor" India Conference, Annual IEEE ,pp. 1 - 5, 2006.
- [3] Samiappa Sakthikumar et al., "A Very Fast and Low Power Carry Select Adder Circuits", 3rd International Conference on Electronics Computer Technology -ICECT 2011.
- [4] Samiappa Sakthikumar et al., "A Novel Low Power and High Speed Wallace Tree Multiplier for RISC Processor", 3rd International Conference on Electronics Computer Technology – ICECT 2011.
- [5] Adamec, F., Fryza, T., "Design and Optimization of ColdFire CPU Arithmetic and Logic Unit", Proceedings of 16th International Conference on mixed design of integrated circuits & Systems, pp. 699 – 702, 2009.
- [6] Sarika U. Kadam, S. D. Mali, "Design of Risc Processor using VHDL", 2016 International Journal of Research Granthaalaya, Vol.4 (Iss.6): June, 2016, DOI:10.5281/zenodo.56647.
- [7] Vishwas V. Balpande, Vijendra P. Meshram, Ishan A. Patil, Sukeshini N. Tamgadem, Prashant Wanjari, "Design and Implementation of RISC processor on FPGA," Indian Journal of Advanced Research in computer science and software Engineering, ISSN:2277 128x Vol 9(8), Volume 5, Issue 3, March 2015, pp.11611165.
- [8] Soumya Murthy, Usha Verma, "FPGA based Implementation of Power Optimization of 32 Bit RISC Core using DLX Architecture," 2015 International Conference on Computing Communication Control and Automation, DOI 10.1109/ICCUBE.A.2015.191.
- [9] Harpreet Kaur, Nitika Gulati, "Pipelined MIPS With Improved Datapath", IJERA, Vol. 3, Issue 1, January -February 2013, pp.762-765.
- [10] J. Poornima, G.V.Ganesh, M. Jyothi, M. Shanti and A.Jhansi Rani, "Design and implementation of pipelined 32-bit Advanced RISC processor for various D.S.P Applications", Proceedings of International Journal of Computer Science and Information Technology, ISSN: 3208-3213, Vol-3(1), June-2012.
- [11] R. Uma, "Design and Performance Analysis of 8-bit RISC processor using Xilinx Tool", International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, Vol-2, Mar-Apr-2012.
- [12] Pejman Lotfi-Kamran. Ali-Asghar Salehpour. AmirMohammad Rahmani. Ali Afzali-Kusha, and Zainalabedin Navabi. "Dynamic Power Reduction of Stalls in Pipelined Architecture Processors", International Journal Of Design, Analysis And Tools For Circuits And Systems. Vol. I, No. I, June 2011.
- [13] Samir Palnitkar, "Verilog HDL: A Guide to Digital Design and Synthesis", Prentice Hall, 2nd Edition, 2003.