

Efficient Hardware Implementation of 32-Bit Single Cycle RISC Microprocessor

Meghana Shanthappa

Graphics Hardware Design Engineer, intel Corporation, USA
Corresponding Author: Meghana Shanthappa

Abstract : Any microprocessor is the heart of any general purpose computing systems which is a form of embedded system. The total efficiency of the system is mainly depends upon the efficiency of the main processing elements. In this paper we proposed efficient hardware architecture for 32-bit microprocessor. The proposed architecture is implemented on Digilent ATLYS (Spartan-6) board and the coding is done by VHDL language. For synthesis purpose we use Xilinx ISE 14.5 version. The proposed architecture is optimized with respect to various hardware parameters such as power, frequency etc. which is discussed in the paper briefly. The main reason for these optimizations is simpler architectures of internal components.

Keywords - RISC Architecture, Microprocessor, FPGA implementation, Parallel Processing, Instruction Set etc.

Date of Submission: 13-05-2019

Date of acceptance: 30-05-2019

I. Introduction

Due to revolution of electronics, now a day's operating system based embedded system gained more popularity. This is mainly due to the re-configurability features which are based on operating system through microcode. This will give the user to add or remove additional optional features. The heart of such system must contain either microcontroller or microprocessor as central processing system. Since the processing element acts as vital role of the overall performance of the respective embedded system, the architecture must be optimized in all sorts of hardware and software aspects.

Most of the general purpose computing system uses microprocessor as central processing unit. To implement any microprocessor various architectures are invented, among them RISC based architecture is most popular. With that architecture to support parallel task execution MIPS techniques [1] are used. In this paper we propose efficient hardware architecture for 32-bit microprocessor. The comparison result shows that the proposed architecture is better compared to existing architectures.

Cesar et al., [2] presented Adiabatic computing based on MIPS microprocessor. To overcome the power dissipation, the authors used Adiabatic computing process. To show the power dissipation comparison the authors built the MIPS microprocessor using both CMOS technique and Adiabatic technique. Both structures were designed using backend custom design tools. The comparison result shows effective power reduction in the case of Adiabatic processor. Buse and Berna [3] presented 32-bit MIPS microprocessor based on fault tolerant technique. The main advantage of this technique is that fault tolerant capability. The proposed architecture is implemented on Cadence tool using 90 nm CMOS technology. Ahmed Eissa et al., [4] proposed the optimized architecture of instruction set of a microprocessor based on Secure Hash Algorithm. For implementing this approach the authors used two techniques namely Native Datapath and Coprocessor Based design. Both approaches were implemented on Virtex-6 FPGA. The comparison result shows the hardware area reduction in the case of both architectures. Jikku Jeemon [5] presented 8-bit pipelined microprocessor using Harvard architecture. For improving the performance, parallelism was implemented. In this case one instruction is executed in each clock cycle. The architecture was designed using verilog language and verified in Spartan-3E FPGA board. Ritpukar et al., [6] presented 32-bit RISC processor. The main aim of this paper was to understand the working of each sub-modules present in the processor architecture. Here each module is coded using VHDL language and synthesized using Xilinx ISE 13.1i software. Kui and Yue [7] designed an efficient 32-bit RISC processor instruction decoder module. This architecture was designed using VHDL language and implemented on Quartus-II FPGA.

Contributions: The main contributions of this paper are (i). The whole architecture is divided into suitable sub-blocks and the architecture of each sub-blocks are optimized. (ii). The Op-Code structure is made such way that the corresponding encoding and decoding circuits are simpler.

II. Proposed Architecture

The proposed architecture for RISC microprocessor is given in Fig.1. It consists of Adder, Shifter, Memory, ALU etc.

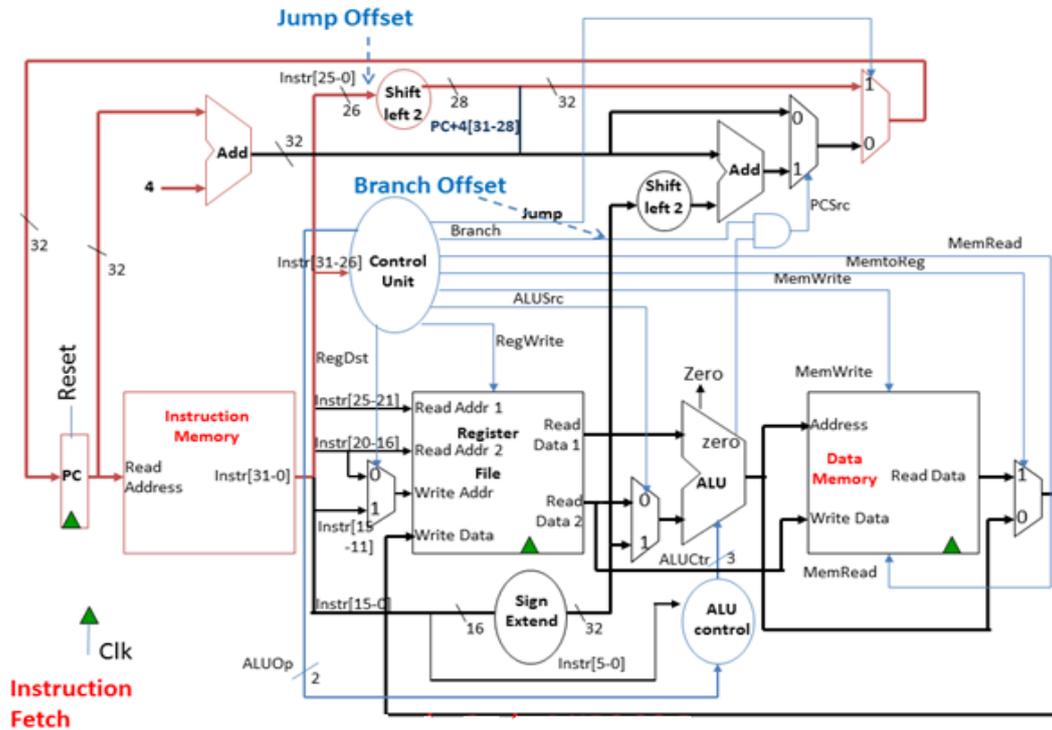


Fig.1: Proposed 32-bit RISC Microprocessor Architecture

2.1. Arithmetic Logic Unit (ALU): The operating management unit is implemented by ALU to perform all the logical operations relating to registers, like adding, subtracting, ANDing, ORing etc. At this point the design had all the practicality necessary for ALU. However, we split the ALU in three elements as Sign extender, ALU management unit and ALU.

2.1.1. Main ALU: The main ALU is used to perform all arithmetic and logic related operations. This module is integrated one of the MUX to reduce hardware complexity. The MUX is present in between register files “Read Data 2” output and sign extender output in Fig.1 is used for this purpose.

2.1.2. Sign Extender: The sign extender is required to implement the actual CPU for performing signed operation along with unsigned operation. Here we used 32 bits Sign Extender Unit for our design.

2.2. ALU Control Unit: We implemented ALU control unit for providing proper command defined by the operation. Basically the task of the ALU control unit is to specify what type of arithmetic operation the main ALU has to carry out is given in Table 1.

Table 1: Functions of ALU Control Unit

Op-Code	ALUOp	Operation	instr	ALU function	ALUCtr
lw	00	Load word	XXXXXX	Add	010
sw	00	Store word	XXXXXX	Add	010
beq	01	Branch if equal	XXXXXX	subtract	110
R_type	10	Add	100000	Add	010
		Subtract	100010	Subtract	110
		AND	100100	AND	000
		OR	100101	OR	001
		SLT	101010	Set on less than	111

2.3. Control Unit: One of the fundamental modules for a hardware is that the operation management in terms of Control unit. This unit is different from ‘ALU Control’ unit and this ‘Control Unit’ is that the first block is used

to control the operation of ALU only and this block is used to control the whole architecture. The main functions of this unit is lw, beq, bne, j, add, sub, and, or, slt, addi etc shown in Fig.1.

2.4. Instruction Memory: It is used to store the instructions in terms of some binary numbers. This is mainly a single ROM module. The operation of this module as follows

1. The address is only an input from PC.
2. The data width of this module is 32 bits.
3. With simple operation we are accessing sequentially the array address by a step of 4. Instead of skipping 3 array elements on every iteration.

2.5. Instruction Fetch: To implemented actual instruction fetch circuit we divided the task in multiple subtasks of different single circuit elements as Adder, Binary multiplexer, Left shifter by 2 and Program Counter. The Adder and Binary multiplexer are implemented by normal method. The left shifter is customized in this architecture. At first we thought of making an n-bit shifter, where our component could be reused for different amounts of shift. The PC is implemented separately to make the circuit design easier. The proposed PC passes the input value on output at every rising edge of the input clock.

2.6. Data Memory and Register Files: The data memory and register files are mainly used to store intermediate calculation values at various stages respectively. The operations of those memories are controlled by 'Control Unit' block. To implement both memories we used write first RAM architecture.

III. Fpga Implementation

The proposed architecture is implemented on Xilinx ATLYS Board which having Spartan-6 XC6SLX45-3CSG324 chip. The proposed architecture is coded using VHDL language and Xilinx ISE 14.5 is used for synthesis purpose. The RTL schematic of the proposed microprocessor is given in Fig.2.

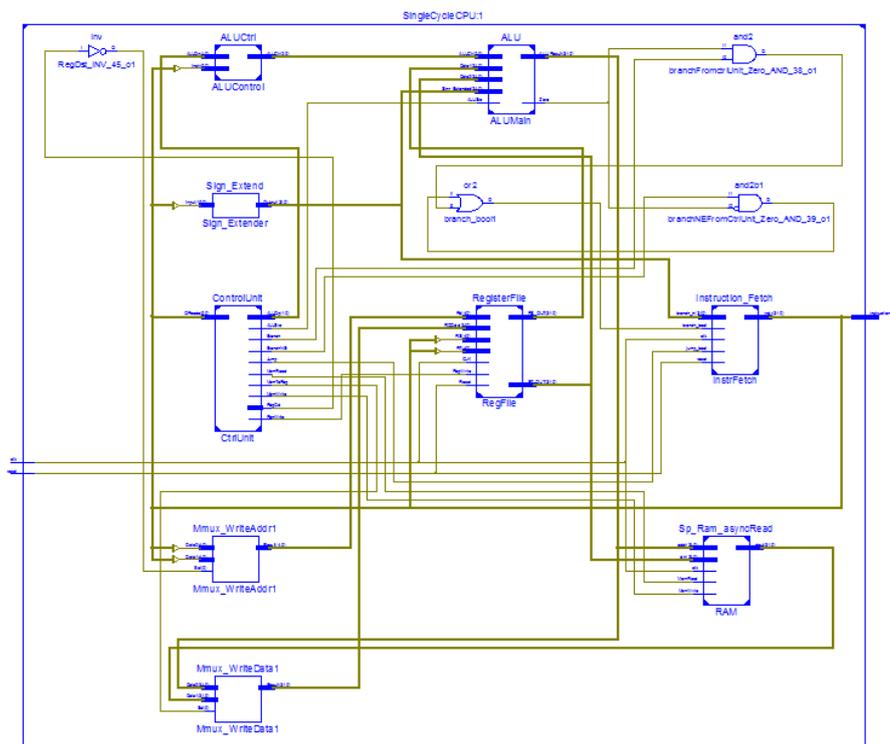


Fig.2: RTL Schematic of Proposed 32-bit RISC Microprocessor

The hardware utilizations and maximum operating frequency of the architecture is given in Table 2. The proposed architecture uses 552 slice registers, 676 slice LUTs, 128 bytes of memories, 2 BUFG/BUFGCTRLs. The minimum period of the architecture is 12.280 ns and the maximum frequency is 81.435 MHz. The total power requirement of the architecture is 0.037 W.

Table 2: Hardware Utilizations of the Proposed Architectures

Parameters	Utilizations
FPGA Board	Spartan-6 (XC6SLX45-3CGG324)
Slice Registers	552
Slice LUTs	676
Memory	128
LUT-FF Pairs	72
BUFG/BUFGCTRLs	2
Minimum Period (ns)	12.280
Maximum Frequency (MHz)	81.435
Power (W)	0.037

IV. Comparisons with Existing Techniques

The comparison of existing techniques with proposed technique is given in Table 3. The architecture presented by Galani et al., [8] was based on normal technique. This architecture was implemented on Virtex-4 FPGA board. The total power required by this architecture is 0.829 Watts. The minimum period is 18.243 ns and the maximum frequency is 54.815 MHz. Similarly the architecture presented by Neha and Pradeep [9] uses five stages of pipeline to implement parallel architecture. This will increase the area and power requirements. This architecture was implemented on Spartan-6 FPGA board. The maximum operating frequency of the architecture is 70.313 MHz and minimum period is 14.222 ns. Whereas the proposed architecture required only 0.037 Watts power and maximum frequency and minimum period are 81.435 MHz and 12.280 ns.

Table 3: Hardware Comparisons of the Proposed Architectures with Existing Architectures

Parameters	Galani Tina et al., [8]	Neha and Pradeep [9]	Proposed
FPGA	Virtex-4	Spartan-6	Spartan-6
Power (W)	0.829	----	0.037
Minimum Period (ns)	18.243	14.222	12.280
Maximum Frequency (MHz)	54.815	70.313	81.435

V. Conclusion And Future Works

In this paper we proposed efficient 32-bit microprocessor architecture. The total architecture can be able to perform any operation just in single cycle whereas most of the existing architectures required more than one clock cycle. Also due to the simpler architecture, the overall area and power requirements of the proposed processor is less than existing and the overall operating frequency is higher than existing. The proposed architecture is designed using Xilinx ISE 14.5 tool. For implementation purpose we use Digilent ATLYS FPGA board and on-chip verification is done using Chipscope Software through IP cores provided by Xilinx.

REFERENCES

- [1]. https://en.wikipedia.org/wiki/MIPS_architecture.
- [2]. Cesar O. Campos-Aguillon, Rene Celis-Cordova, Ismo K. Hanninen, Craig S. Lent, Alexi O. Orlov and G. Regory L. Snider, "A Mini MIPS Microprocessor for Adabatic Computing", IEEE International Conference on Rebooting Computing, pp. 1-7, 2016, USA.
- [3]. Buse Uataoglu and Berna Ors Yalcin, "Reliability Analysis of MIPS-32 Microprocessor Register File Designed with Different Fault Tolerant Technique", IEEE International Conference on Signal Processing and Communication Application, pp. 1-6, 2016, Turkey.
- [4]. Ahmed S Eissa, Mahmoud A Elmohr, Mostafa A Saleh, Khaled E Ahmed and Mohammed M Farag, "SHA-3 Instruction Set Extension for a 32-bit RISC Processor Architecture", 27th IEEE International Conference on Application Specific System, Architecture and Processors, pp. 233-234, 2016, UK.
- [5]. Jikku Jeemon, "Pipelined 8-bit RISC Processor Design using verilog HDL on FPGA", IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, pp. 2023-2027, 2016, India.
- [6]. S. P. Ritpurkar, M. N. Thakare and G. D. Korde, "Synthesis and Simulation of a 32-bit MIPS RISC Processor using VHDL", IEEE International Conference on Advances in Engineering and Technology Research, pp. 1-6, 2014, India.
- [7]. Kui Yi and Yue-Hua Ding, "32-bit RISC CPU Based on MIPS Instruction Decoder Module", 2nd IEEE Pacific-Asia Conference on Web Mining and Web Based Applications, pp. 1-5, 2009, China.
- [8]. Galani Tina, Riya Saini and R. D. Daruwala, "Design and Implementation of 32-bit RISC processor using Xilinx", International Journal of Emerging Trends in Electrical and Electronics, pp. 18-24, Vol. 5, Issue. 1, 2013.
- [9]. Neha Dwivedi and Pradeep Chhawcharia, "Design and Implementation of 32-bit RISC Processor with Five Stage Pipeline", Proceeding of International Conference on Recent Innovations in Engineering and Technology, pp. 84-88, 2017, India.

Meghana Shanthappa" Efficient Hardware Implementation of 32-Bit Single Cycle RISC Microprocessor" IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) , vol. 9, no. 2, 2019, pp. 44-47.