Design and Implementation of High Speed 32-Bit RISC Processor for Real Time Embedded Applications using VERILOG HDL

[1] Sreekaram Jagadeesh, [2] S Mohan Rao, [3] V G Hamsaveni
[1]Assistant Professor, [2] Assistant Professor, [3]Associate Professor
[1] SITAMS (AUTONOMOUS), [2]SVCE, [3] SITAMS (AUTONOMOUS), JNTUA University
[1]sreekaramjagadeesh@gmail.com
[2]mohannaidu.vkt@gmail.com
[3]vghamsaveni@gmail.com

Abstract

Reduced instruction set computer (RISC) processors is coming in compact size with highly sophisticated features, for real world examples. In this context this research paper projecting the concept of the Design and implementation of High speed 32-Bit Processor for Real –time Embedded Applications. In order to improve the speed and performance of the processor here we used pipelining technique. Pipelining architecture reduces the execution time for instructions with in a consolidated single machine cycle; this process is often called as fastest performance of the Machine. Design and Implementation of the processor work has been carried out using Xilinx tool, code developed in Verilog HDL, Results verified using FPGA Hardware. In Real time verification we developed a code vending machine, which can supply newspapers according to user options. Using this RISC Processor we achieved greater accuracy in delivering the requirement with the machine.

Keywords: RISC Processor, Verilog, FPGA, Xilinx software, Real-time embedded systems.

1. Introduction

Very Large Scale Integration (VLSI) is playing an important role in the semiconductor sector all over the world. Technology aspect day by day, there is a lot of changes and advancements in every sector. In this regards VLSI field, minimization of components size came to 9 nm and 3nm etc. VLSI industry starting from Designing, Layout, physical Processing and Fabrication process almost there tremendous changes came in every sector. Starting from basic components to processors design there are a lot of changes in fabricating processes. Coming to Processors design changing based on applications. According to user requirements, hardware processors are manufactured in the market. General Processors are like complex instruction set computers (CISC) and RISC, Digital Signal Processors, Application specific integrated circuits (ASIC) and embedded controllers. In order to increase the performance of the design, we need a high-speed processing unit. [1]. generally, embedded systems become more popular with its nature that is application specific. In Embedded real-time applications, it often needs high computational power with processing speed in order to achieve these specifications need the RISC processor mechanism. RISC reduces the processing time with great accuracy. In this RISC mechanism instructions will execute with single machine cycle. Here in this concept 32- Bit RISC instructions implemented with Pipelining process. Pipelining techniques greatly reduce the instruction execution time with machine cycles. [2-3].

For the Design and implementation, Verilog Hardware Description Language (HDL) is greatly flexible with syntax, it looks like C-language but not C. Xilinx software allows getting into hardware unit constraints. Here processor code developed and implemented on FPGA. There are some advantages of FPGA's over traditional processors, for implementing code over FPGA. This

is flexible, reusable and quicker etc. FPGA can be reconfigurable and reprogrammable. While developing the code we can modify the design according to system requirements. [3-4].

2. IMPLEMENTATION

2.1 RISC PROCESSOR DESIGN – ARCHITECTURE

RISC is a microprocessor which has been designed in such a way that is can operate at a higher speed which includes smaller instruction sets. The proof of concept of this research of processor is based on certain conditions. One main assumption is that the design must have few register sets for memory elements. In addition to Register Banks instruction register, ALU and memory register also integrated. FPGA's having internal memories those are very fast as registers and those registers are pre-emptive in multitasking functions. Usage of the small number of registers tends to faster switching. Another assumption is RISC based Processor is most flexible and suitable for real-time embedded applications. Moreover coming to RISC operation it simplifies the instructions into multiple instructions that can perform low level operations with high accuracy and speed. In addition to this concept RISC architecture reduces the complexity of the design aspects. Based on these analysis parameters we have chosen RISC Processor. [5-6]. RISC Processor design mainly consists of Instruction register (IR), Register Banks (RB), Arithmetic and Logical Unit (ALU) and Memory unit. The basic block diagram of the processor can be seen in Fig. 1 below. Functional aspects will discuss in detail.

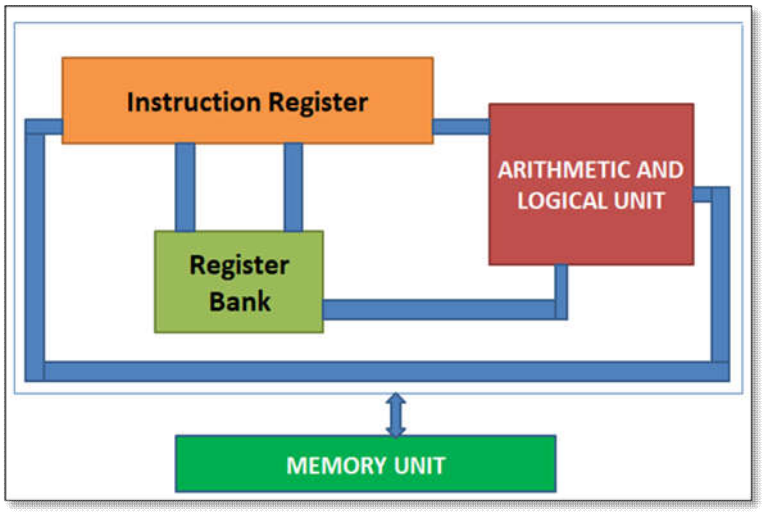

Fig.No.1. Functional Block Diagram of RISC Processor

2.2. INSTRUCTION REGISTER

Instruction register unit consists of FETCH and DECODE sub blocks, here it receives the opcode and generates the necessary signal that activates the remaining components and selects the data path to work according to the instruction. The IR unit has two separate Decoders. These decoders decode the instruction bits and send the signal to either ALU or to the Register Banks. The operands are received from the input registers, based on receiving input instruction, ALU performs the functions with decodes instruction bits. [7].

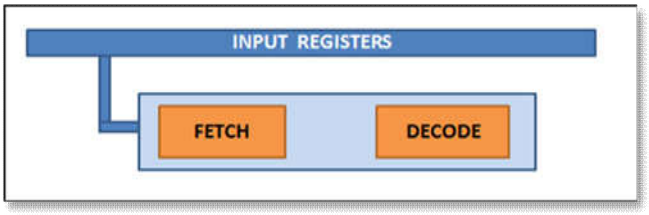

Fig.No.2. INSTRUCTION REGISTER

2.2. ARITHMETIC AND LOGICAL UNIT

Arithmetic and Logical Unit (ALU) is a multi-functional combinational digital circuit. It performs both arithmetic and logic functions in bitwise or binary number wise. The proposed architecture design of ALU supports to 32-bit Arithmetic and Logical operations. ALU will receive instructions from the IR unit and execute the desired operation. [8]

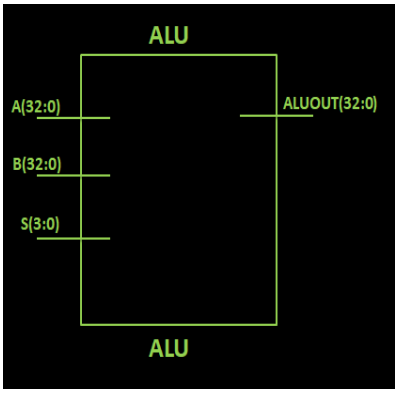

Fig.No.3. ALU IMPLEMENTED BLOCK IN XILINX

For example, if the input instruction is 0011, and input register sends the input bits to the functional register, the decode bits will be will be 32- bit data and after receiving the input instructional bits from the decoder, ALU executes the "NOR" operation. ALU functions mentioned in table no.1.

SELECT LINE	FUNCTION
0000	A and B
0001	A nand B
0010	A or B
0011	A nor B
0100	A xor B
0101	A xnor B
0110	not A
0111	A + B
1000	A – B

Table .No.1. ALU Functions

2.3 REGISTER BANKS

The register banks are temporary memory units within the architecture to support the functionality of the core processor. These are arranged in part one by one in overlapping each in the way. There is a different structure for each kind of register bank for each processor mode. The Banked registers will give accurate switching context for dealing with processor computational executions and privileged operations.[9 -10]

.2.3.1. Register Bank selection:
Generally, register banks store the temporary data, which can be accessed at the time of the instruction execution. Here 8-Bit Register bank designed with D- Flip-flops. According to the clock and reset signal, and Write/Read signal, the input signal goes to execution and stores the data in the bank.

clock	Reset	Wr/rd	Input
1	1	X	XXXX
1	0	1	1010
1	0	0	1010

Table .No.2.General Purpose Register Functions

3. PIPELINED ARCHITECTURE DESIGN FOR HIGH-SPEED APPLICATION

General Pipeline structure for RISC Processors contains 5 stages and are summarized in this section.

3.1 Instruction Fetch (IF): In order to execute the instructions, instruction cache had a delay of one cycle, during the instruction fetching stage, it would be ready on the coming clock cycle. Normally 32- bit instruction fetched from the instruction cache.

3.1.2 The Program Counter (PC): This is the internal register, performs the operation and holds the address of the current executing instruction. It sends to the PC predictor, then it sends to the Program Counter (PC) to the instruction cache to read the status of the current executing instruction.

3.2 Instruction Decode(ID): Normally in RISC machines, instructions fetched from the instruction cache and the fetched instruction bits will be shifted down to the pipeline, so that the combinational logic in each of a pipeline stage produce the required control signals for the execution, the decoding section decodes the data path from the instruction bits.

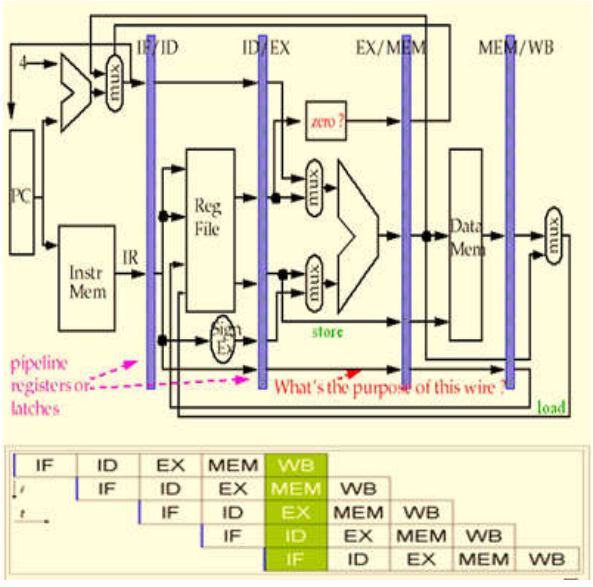

Fig.No.4. PIPELINE STRUCTURE OF THE RISC PROCESSOR

3.3. Instruction Execute (IE): This stage is responsible for computations. Generally in this stage consists the ALU and Shift registers like barrel shifter etc. to perform the operations?

3.4 Memory Access (MA): Here, if the data memory required to be accessed, it allows to the other units to access the memory. During the executing stage, single cycle latency instructions
Results forwarded to the next stage from the unit. It stores instructions to execute the functionalities.

3.5 WRITEBACK (WB): In this stage, both the single and two cycle instructions can write their respective results into the register file.

Pipelining decreases execution time but it will increase cycle time. The output is increased since a single instruction ideally finishes every clock. However, it generally increases the latency of each instruction. [11-13].

4. RESULTS

In real time embedded applications scenario, we have implemented the vending machine design for the supply of requirements with high speed accessing. 32 – Bit RISC processor designed and implemented on FPGA, and tested with Vending machine code unit. Fig no 5 and 6 shows that the implementation block of RISC Processor and Internal structure of RISC Processor Implementation.

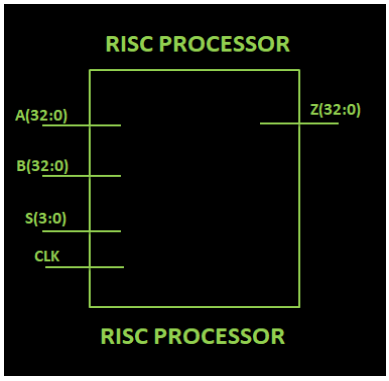

Fig.No.5. RISC PROCESSOR IMPLEMENTATION

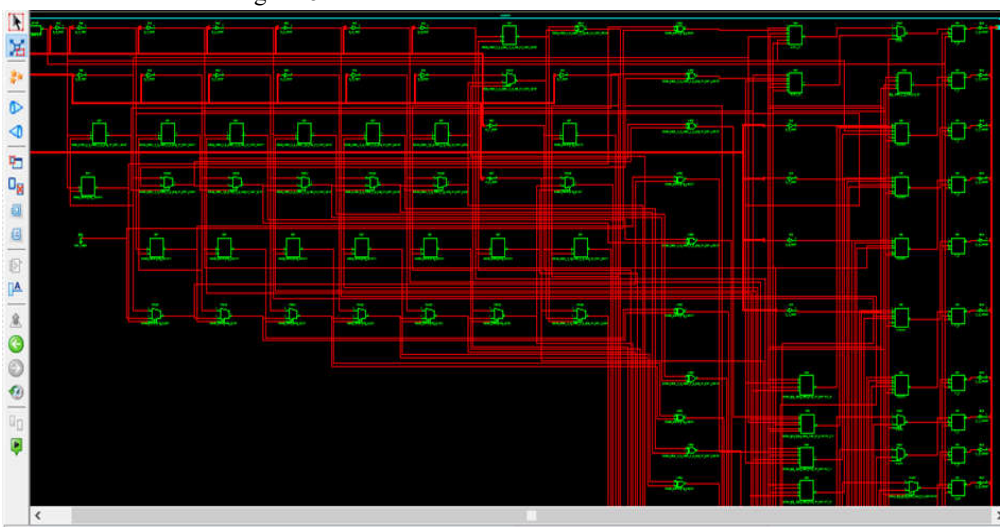

Fig.No.6. RISC PROCESSOR INTERNAL STRUCTURE

4.1 Real Time – Embedded Application:

Vending Machine: Newspaper serving device – Automatic Newspaper serving device can able to process the requirements based on user selections. Flowchart diagram represents the functionality of the machine.

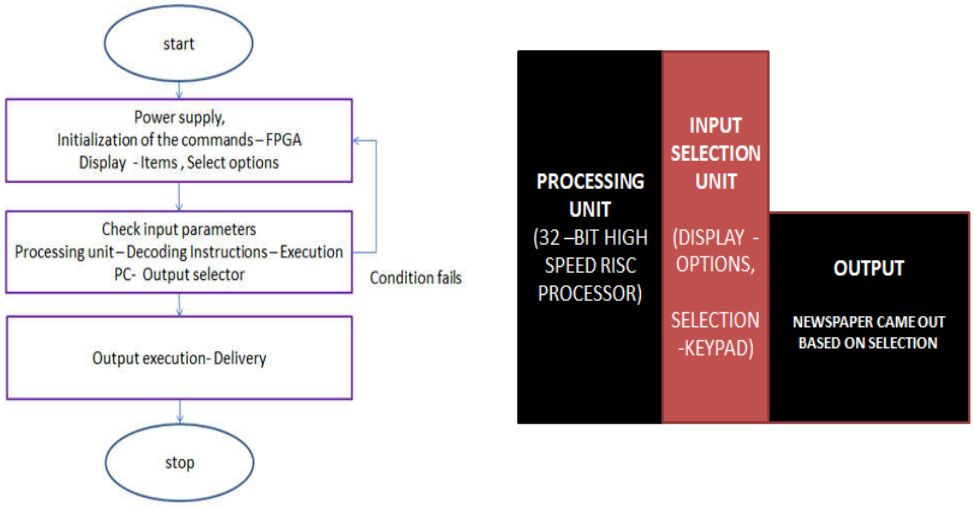

Fig.No.7. FLOW CHART AND VENDING MACHINE DESIGN WITH RISC PROCESSOR

5. CONCLUSION

In this research paper, we discussed and showed the design and implementation of a high-speed 32- Bit RISC Processor for real-time embedded applications. This RISC Processor is suitable for real-world embedded applications; the processor performance remains as the state of the art. Here RISC reduces the instruction count compare other Processors, it affects the performance of the embedded application. In this research we implemented the vending machine code using standard FPGA, the output remains as the significant achieved speed signifies the performance of the design. The design of the systems like real-time embedded, communications and signal processing and intelligent control systems, these RISC processors are much suitable.

REFERENCES

1. Tannu Chhabra, Md Tauheed Khan, VLSI Design of a 16-bit Pipelined RISC Processor, available at http://www.IJCSE.ORG
2. L. Kaouane et al, A Methodology to Implement Real-time Applications on Reconfigurable Circuits, available at http://wwwrocq. inria.fr/syndex
3. P. Kohlig et al, FPGA Implementation of high performance FIR Filters, In Proc. International Symposium on Circuits and Systems, 1997.
4. M. Shand, Flexible Image Acquisition using reconfigurable hardware In Proc. of the IEEE Workshop on Field Programmable Custom
5. Computing Machines, April 1995.
6. J. Villasenor, Video Communication using rapidly reconfigurable hardware, IEEE Transactions on Circuits and Systems for Video
7. Technology, Vol. 5, No. 12, pp. 565 – 567, Dec. 1995.

8. V.Prasanth and K.V.Rajeshkumar , FPGA Based 64-Bit Low Power RISC Processor Using Verilog HDL, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering.
9. Supraj Gaonkar, Anitha M. Design of 16-bit RISC Processor, International Journal of Engineering Research & Technology (IJERT)Vol. 2 Issue 7, July – 2013
10. K. Vishnuvardhan Rao*, A. Anita Angeline and V. S. Kanchana Bhaaskaran Design of a 16 Bit RISC Processor, Indian Journal of Science and Technology, Vol 8(20), IPL0221, August 2015
11. Jaina Patel RISC(16 Bit)Processor Design using Verilog in Modelsim International Journal of Innovative Research in Science, Engineering and Technology, www.ijirset.com
12. Xilinx Corporation. "Xilinx breaks one million gate barrier with delivery of new vertex series" October 1998.
13. Xilinx Corporation. Vertex Data Sheet 2000.

Author Profile –

Mr. S. Jagadeesh, currently working as Assistant Professor at SITAMS (Autonomous), JNTU_A University, Anantapuram. He received his master's degree in VLSI Design from JNTU_A University in the year 2015. He received his Bachelor's degree in ECE from JNTU_A University in the year 2011. He has published various national and international journals. His research interests are VLSI systems and image analytics, and image enhancement.

Mr. S. Mohan Rao, currently working as Assistant Professor at SVCE, JNTU_A University, Anantapuram. He received his master's degree in VLSI System Design from JNTU_H University in the year 2013. He received his Bachelor's degree in ECE from JNTU University in the year 2009. He has published various national and international journals. His research interests are image analytics, and image enhancement and VLSI systems.

Mrs. V G Hamsaveni, currently working as Associate Professor at SITAMS (Autonomous), JNTU_A University, Anantapuram. She received her master's degree in Energy Management from SV University in the year 2005. She received her Bachelor's degree in EIE from JNTU University in the year 2002. She has published various national and international journals. Her research interests are Nano technology and image analytics, and image enhancement.