IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# FPGA BASED PROCESSOR IMPLEMENTATION

[1] Ghatam A S S P Pavankumar Sastri, [2] D. Aditya Sharma, [3] G S R N Aditya Sasidhar, [4] Mrs. V. Uma

[1,2,3] B.E IV year, [4]Assistant Professor

Department of Electronics and Communication Engineering

Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Kanchipuram, Tamilnadu, India

***Abstract:*** Computers are needed for various applications in our present life. As a result, Application specific soft processor cores for FPGA based embedded applications are being designed in which user can configure the processor as per requirement. These processors are gaining more importance in course of time. The architectural simplicity of RISC processors makes the suitable for high performance, low power applications. Hardware description languages (HDL's) are commonly used to construct hardware system. FPGA provides the re-configurable platform, so reuse of the design is a commonly used to improve the productivity. In this projct, the data processing instructions of RISC processor and the control unit are implemented using very high-speed integrated circuits, using the Verilog hardware description language and the design is verified by applying test bench on Xilinx's Spartan 6 based FPGA development board using software XILINX ISE.

***Index Terms*** **- Verilog HDL, Xilinx ISE, Xilinx Spartan 6.**

## 1. INTRODUCTION

Electronic devices and gadgets like televisions, computers, wireless speakers, smart watches etc are gaining importance, as they are helping us to lead life in a smarter way. These gadgets are having processors as heart of the system. Processors are required for different electronic applications as per the purpose. Generally, manufacturing different processors for different applications is difficult in practical world. In this project, design of a processor in which the core can be programmed by the user so that one common processor can be used for different applications. So, the trend is towards the designing of RISC Processors that are efficient for specific applications. Performance is the main criteria for designing such processors. The proposed RISC processor designed here is an effort towards an efficient processor suitable for small applications. Depending on the instruction set used the architecture of the processor is classified into RISC architecture and CISC architecture.

### RISC Architecture

Reduced Instruction Set Computer (RISC) is a type of microprocessor architecture that utilizes a small, highly-optimized set of instructions, rather than a more specialized set of instructions often found in other types of architectures. It is a Software centric design with simple and standardized instructions and these instructions take only one cycle time. In RISC Pipelining is easy and "Load" and "Store" are independent instructions. It only needs more amount of RAM. So, RISC architecture is Preferred. We used Verilog HDL to define the basic components of a processor and when this is paired with an FPGA board, an ASIC Processor is the result. Changing very few instructions the processor can be easily used for different types of applications. Proposed RISC processor has four stages in the pipeline and in general each stage takes one clock pulse to complete its operation. It has 16-bit address lines, 16-bit data lines, one 16-bit input port and 16-bit output ports. All the registers are 16-bit. The block diagram is shown in figure 1.

## 2. INSTRUCTION SET ARCHITECTURE (ISA):

As a true 16-bit RISC processor, all the instructions are 16-bit in length and use the 3-operand notation method (2-source operands and 1-destination operand). The design has 8-internal general-purpose registers like an array all are 16-bit in length. The General instruction format designed RISC processor is given below in the Tabular Column 1.

The actual Instruction format may vary from instruction to instruction. As each instruction has different format, they follow different data path, which are combined together to form final data path.

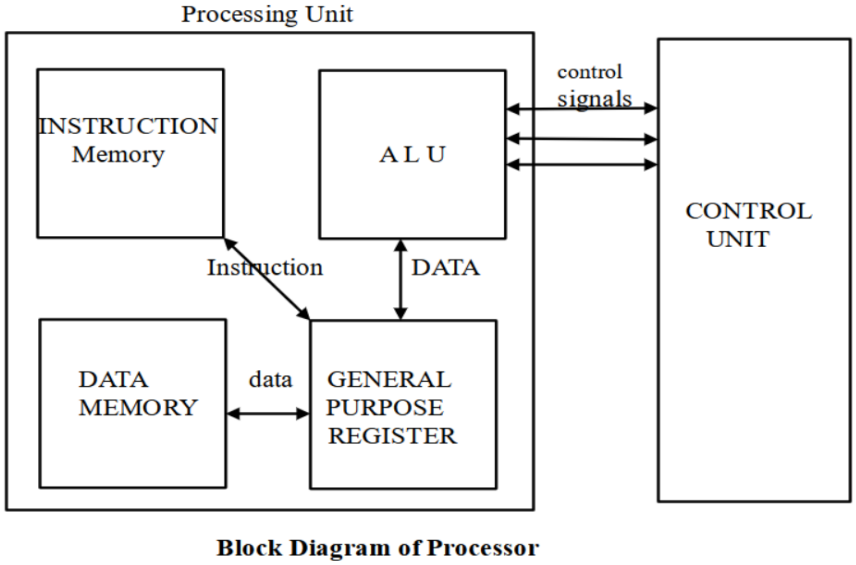



Fig 1 Block diagram of Processor

Table.1 Instruction Set

| Operand | Instruction Operand | General Instruction Format |
|---|---|---|
| 0000 | Load word | OP 4 DEST1 3 SRC1 3 OFFSET 6 |
| 0001 | Store word | OP 4 DEST1 3 SRC1 3 OFFSET 6 |
| 0002 | Add | OP 4 DEST1 3 SRC1 3 SRC2 3 OFFSET 3 |
| 0003 | Sub | |
| 0004 | Invert(1's Complement) | |
| 0005 | Logical shift Left | |
| 0006 | Logical Shift Right | |
| 0007 | Bitwise AND | |
| 0008 | Bitwise OR | |
| 0009 | Less than | |
| 0010 | Hamming distance | |
| 0011 | Branch on equal | OP 4 DEST1 3 SRC1 3 OFFSET 6 |
| 0012 | Branch on not equal | |
| 0013 | Jump | OP 4 OFFSET 6 |

## 3. Processor Architecture:

Proposed RISC Processor has been divided into two parts.
(i) Processing unit.
(ii) Control unit.

### 3.1 Processing Unit:

Data path unit refers to the collection of various units in the CPU like instruction memory, Registers, ALU control, ALU, Data Memory which are interconnected to make one combinational circuit. All the components are then connected to Control Unit.

### ALU:

Arithmetic and Logic unit is a block where Mathematical and logical operations are performed. It is connected to Alu Control unit through which it gets input data and opcode are received. The ALU we designed is very simple. Its functions include basic Arithmetic operations, Shifting and logical operations.

**General Purpose Register Unit:**

In Proposed design eight 16-bit registers are designed which are arranged in an Array. These are used to store the values and Instructions on those the operation is performing. In the design Registers are connected to data memory, ALU.

**Data Memory:**

Data memory otherwise called RAM (Random access Memory) is the main memory of the CPU. It is both Readable and writable. The Main operation of the RAM is to read or write data into which is accessed CPU Randomly. In Proposed system RAM has 16-bit input and output with a clock.

**3.2 Control Unit:**

Control Unit is a component of the processor which controls the Data Memory, Arithmetic and logic unit and input and output devices how to respond to the instructions that have been sent to the processor. It directs the operation of the other units by providing time and control signals. Hardwired control units are implemented using combinational logic units containing finite number of gates that can generate specific results based on instructions given.

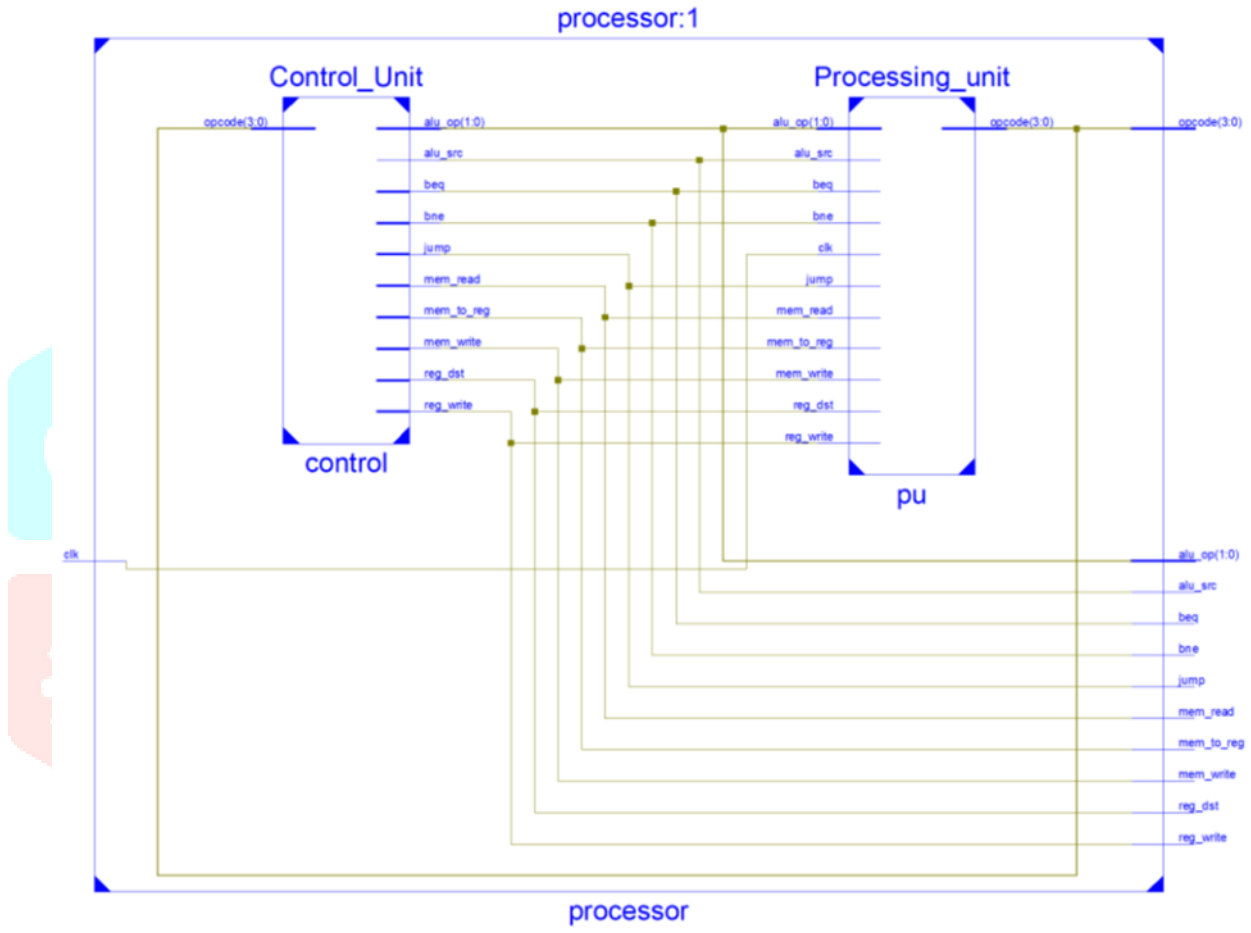



Fig 2 Inner Architecture of Processor

**4. RESULTS AND IMPLEMENTATION**

**4.1 Implementation On FPGA:**

FPGA (Field Programmable Gate Array) is a device used for verification of design. It works like a raw IC in which user can implement his design of it and can verify the design. It consists of LUTs (Look Up Tables), CLBs (configurable logic blocks), Logic cells, and Input Output Blocks. The designed processor has been implemented on Xilinx Spartan-6 FPGA board for verification purpose.

**4.2 Simulation Results:**

After modeling the design through Verilog HDL, the code is Simulated using Xilinx ISE Simulator. The input test bench waveform as a source for applying the input values and the output from simulator is used to verify the design. Here we are verifying the code by observing the control signals generated by the ISIM simulator for the test bench given. In the design the result is acquired in a specific file named "34323_334.o" as specified in the program. Given below is the example output waveform acquired for the test bench and output acquired through the file.

34323_334.o - Notepad

File Edit Format View Help

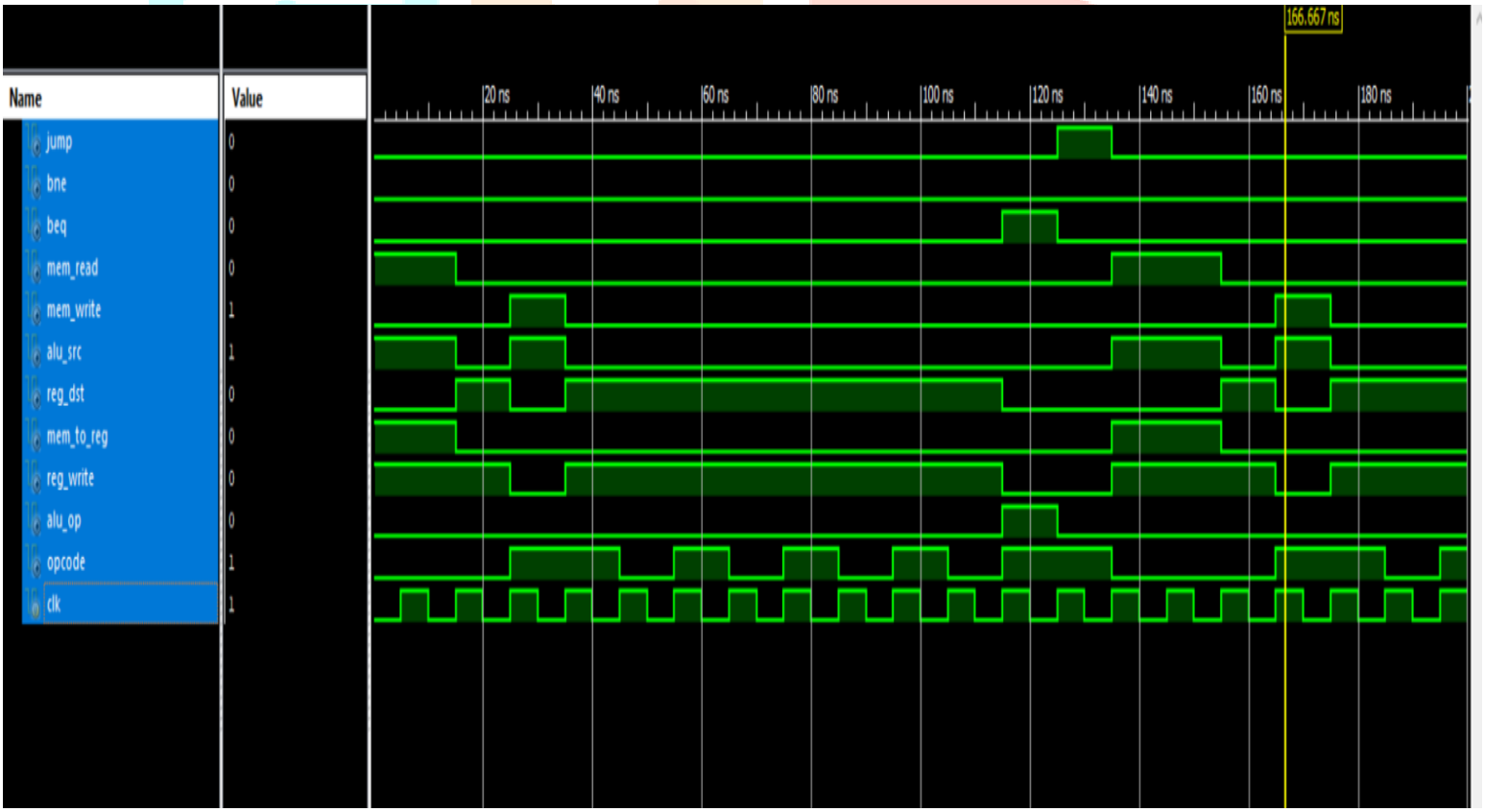
```
time =                    0
        memory[0] = 0000000000000001
        memory[1] = 0000000000000010
        memory[2] = 0000000000000001
        memory[3] = 0000000000000010
        memory[4] = 0000000000000001
        memory[5] = 0000000000000010
        memory[6] = 0000000000000001
        memory[7] = 0000000000000010

time =                   35
        memory[0] = 0000000000000001
        memory[1] = 0000000000000010
        memory[2] = 0000000000000011
        memory[3] = 0000000000000010
        memory[4] = 0000000000000001
        memory[5] = 0000000000000010
        memory[6] = 0000000000000001
        memory[7] = 0000000000000010

time =                  175
        memory[0] = 0000000000000001
        memory[1] = 0000000000000010
        memory[2] = 0000000000000011
        memory[3] = 0000000000000101
        memory[4] = 0000000000000001
        memory[5] = 0000000000000010
        memory[6] = 0000000000000001
        memory[7] = 0000000000000010
```

Fig 3 Simulation Output file

Fig 4 Processor Simulation Output



| Device Utilization Summary (estimated values) | | | [-] |
|---|---|---|---|
| **Logic Utilization** | **Used** | **Available** | **Utilization** |
| Number of Slice Registers | 19 | 54576 | 0% |
| Number of Slice LUTs | 356 | 27288 | 1% |
| Number of fully used LUT-FF pairs | 10 | 365 | 2% |
| Number of bonded IOBs | 16 | 296 | 5% |
| Number of BUFG/BUFGCTRLs | 1 | 16 | 6% |

Fig 5 Device Utilization Summary

## REFERENCES

**[1]** Implementation of RISC Processor on FPGA, Pravin S. Mane*, Indra Gupta**, M.K. Vasantha**, *Computer Science & Engineering Department, Mody Institute of Technology & Science, Lakshmangarh, **Electrical Engineering Department, Indian Institute of Technology Roorkee, Roorkee. (1-4244-0726-5/06/'2006IEEE).

**[2]** FPGA Prototyping of a RISC Processor Core for Embedded Applications, Michael G Schwind, Senior Member, IEEE, Valentina Salapura, and Dietmar Maurer IEEE TRANSACTIONS ON VERY LARGE-SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 9, NO. 2, APRIL 2001

**[3]** DESIGN OF FPGA BASED 8 BIT RISC PROCESSOR, S.V. KULKARNI*, A.I. NADAF**, P.P. SHAH*, M.K. BHANARKAR**, * Department of Electronics, Devchand College, Arjunnagar, MS-India, ** Department of Electronics, Shivaji University, Kolhapur, MS-India, Vol 07, Article 05675; June 2016

**[4]** Design and Implementation of 16-Bit RISC Processor on FPGA, Mr. Rajkumar D. Komati, Aditya Kolekar, Kunal Kasbekar, Ms. Avanti Godbole Dept. of ENTC, MIT College of Engineering, Kothrud, Pune IJCRT2006062

**[5]** Design of 16-bit RISC Processor, Supraj Gaonkar, Anitha M, Sir M Visvesvaraya Institute of Technology Bangalore, Karnataka, India (vol-1, issue-1, ISROSET-IJSRPAP-00068)

**[6]** Verilog HDL: A Guide to Digital Design and Synthesis, Second Edition by Samir Palnitkar

**[7]** Advanced Digital Design with the Verilog HDL by Michael Ciletti

**[8]** https://www.geeksforgeeks.org/computer-organization-and-architecture-tutorial