# Design approach for 5 Stage Pipelined RISC Processor

**[1]Mr.Venkanna Mood, Research Scholar, Dept .of ECE, UCE, Osmania University,Telangana India**

**Working at St.Martin's Engineering College , Dhulapally Hyderabad-500100**

**[2]Dr. Rameshwar Rao, Rtd. Professor, Dept.of ECE, UCE, Osmania University, Telangana, India.**

**[3] Dr Santosh Kumar Patra , Professor& Principal, Dept .of CSE, St.Martin's Engineering College, Telangana,India.-500100**

**Corresponding Author: venkatmood03@gmail.com**

Abstract:
The proposed research work is the design of a 32 bit RISC (Reduced Instruction Set Computer) processor. The design helps to improve the speed of processor, and to give the higher performance of the processor. It has 5 stages of pipeline viz. instruction fetch, instruction decode, instruction execute, memory access and write back all in one clock cycle. The control unit controls the operations performed in these stages. All the modules in the design are coded in VHDL.

*Keywords* –**RISC, SISC, Pipeline Processor, VHDL**

## I. INTRODUCTION:

Perhaps the most significant characteristic of von- Neumann computer architecture is the use of a single program counter (PC) to control the flow of executing programs. Program instructions are executed in the same order as they appear in the main memory. Branching to subroutines or other programs is allowed. However, a return to the calling routine is usually made available.
Generally we can call a computer a von-Neumann machine if it satisfies the following requirements:

1. It has three basic units:
   a) A CPU: Central Processing Unit
   b) A Main memory
   c) An I/O unit
2. Its programs are stored in the main memory. A program can manipulate its data which can reside in the main memoryas well.
3. It executes its programs sequentially andone instruction is executed at any given time.

## II.PROBLEM STATEMENT

There are several drawbacks & deficient effects of earlier design of Microprocessors, earlier designs was proposed in which finite state machines has been taken for controller but low power issues are not considered for encoding and cyclic controlling. If the encoding is grey, then it will consume less power, other issues of previous designs are there low speed and there controlling mechanism for instruction scheduling. In today's era of high speed systems and ubiquitous computing, the need for real-time embedded systems is always on the rise. These embedded systems must operate within stringent requirements that are often at the intersection of the conflict between speed and area. Increasing complexity of signal, image and control processing in embedded real-time applications requires very high computational power. This power can be achieved by high performance programmable components like RISC or CISC processors etc and non-programmable specific chips like ASICs or FPGA based hardware. Naturally, to enhance the speed of such systems we need to design algorithms that can computed with low running time complexity. Another way of increasing the speed of the system is to directly design high speed VLSI chips for these embedded systems. However, the present design was concerned more with demonstrating the power of multi-stage pipelining for fast instruction execution and also to demonstrate that FPGA based processors can be equally useful for real-time operating systems rather than to design just a singleprocessor.

CISC processors have gained the common marketplace over the years. They support various addressing modes and various data types. The instruction length varies from instruction to instruction. They frequently access data in external memory. They are generally implemented using micro-programmed control. There is little semantic gap between instructions of CISC processor and statements in higher-level languages. Although they may be saving memory space, design is complicated and as instructions is variable in length; special hardware for boundary marking of instruction is required. Study over the years proved that simple instructions are used 80% of the time and many complex instructions can be replaced by group of simple instructions [3].

RISC processor operates on very few data types and does the simple operations. It supports very few addressing modes and is mostly register based. Most of the instructions operate on data present in internal registers. Only LOAD and STORE instructions access data in external memory. Also the instruction length is fixed and hence decodingis easier.

## III.PROPOSED METHOD

**Instruction Fetch Unit**: The first stage in the pipeline is the Instruction Fetch. Instructions are fetched from the memory and the Instruction Pointer (IP) is updated. The function of the instruction fetch unit is to obtain an instruction from the instruction memory using the current value of the PC and increment the PC value for the next instruction as shown in Figure 3. This stage is where a program counter will pull the next instruction from the correct location in program memory. In addition the program counter will updated with either the next instruction location sequentially, or the instruction location as determined by a branch.

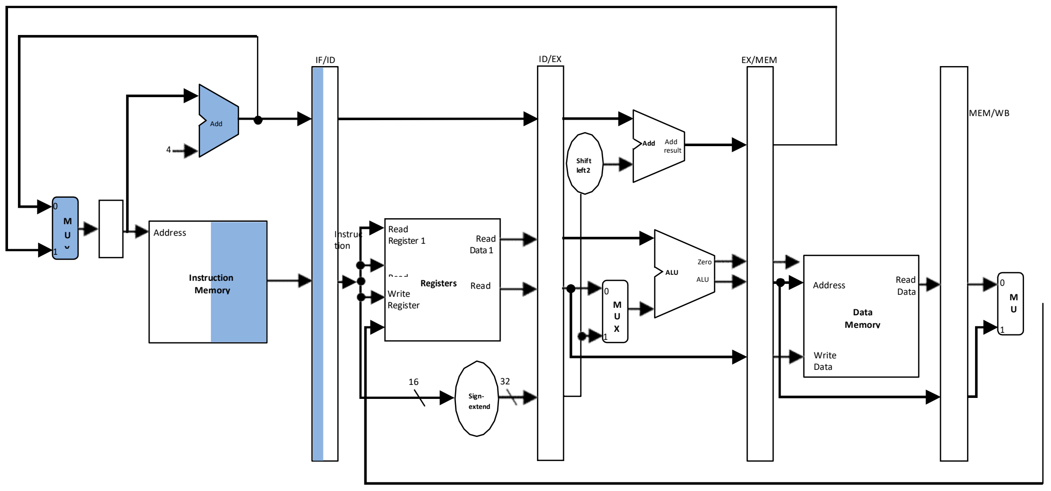

Figure 3: Instruction fetch unit in 5 stage pipelined RISC processor

The instruction fetch stage is also responsible for reading the instruction memory and sending the current instruction to the next stage in the pipeline, or a stall if a branch has been detected in order to avoid incorrect execution. The instruction fetch unit contains the following logic elements that are implemented in VHDL: 8-bit program counter (PC) register, an adder to increment the PC by four, the instruction memory, a multiplexor, and an AND gate used to select the value of the next PC. Program counter and instruction memory are the two important blocks of Instructions Fetch Unit.

**Instruction Decode Unit:** The Instruction Decode stage is the second stage in the pipeline. Branch targets will be calculated here and the Register File, the dual-port memory containing the register values, resides in this stage. The forwarding units, solving the data hazards in the pipeline, reside here. Their function is to detect if the register to be fetched in this stage is written to in a later stage. In that case the data is forward to this stage and the data hazard is solved. This stage is where the control unit determines what values the control lines must be set to depending on the instruction. In addition, hazard detection is implemented in this stage, and all necessary values are fetched from the register banks. The Decode Stage is the stage of the CPU's pipeline where the fetched instruction is decoded, and values are fetched from the register bank. It is responsible for mapping the different sections of the instruction into their proper representations (based on R or I type instructions).

## IV.SIMULATION AND RESULTS

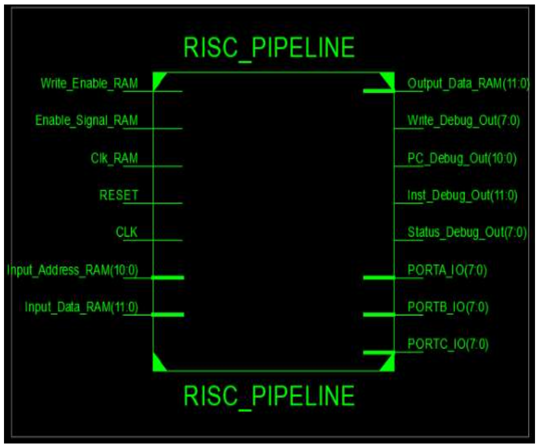

Fig4. RTL View of RISC Pipeline

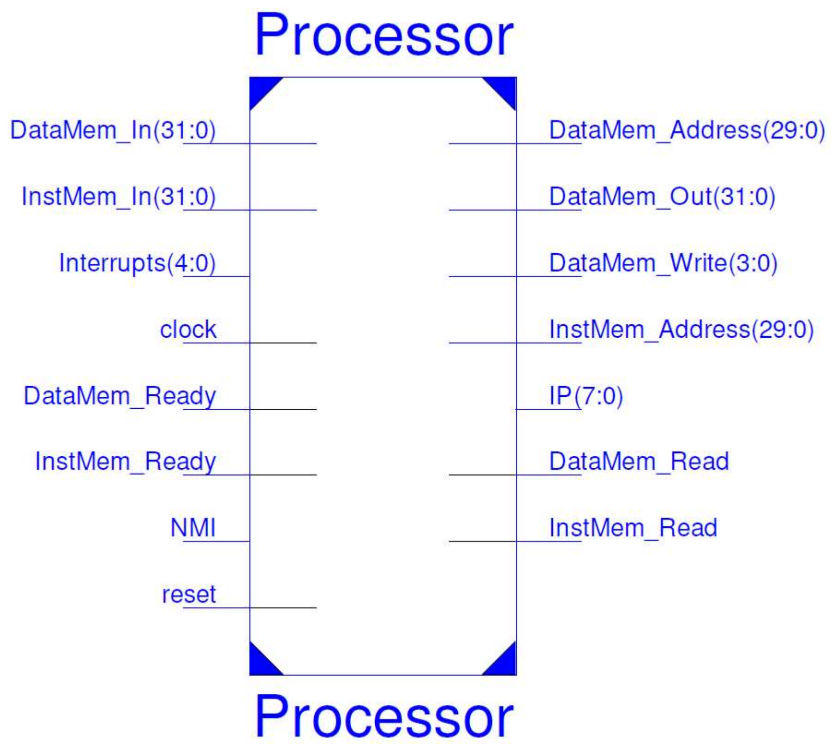

Figure 5: Pin diagram for proposed 32 bit RISC processor

Figure 4 shows the pin diagram of proposed 5 stages pipelined architecture of RISC. Synthesis has been carried out on Vertex-6 board using Xilinx 14.5 ISE. RTL View is a register transfer level graphical representation of the design. This representation (.ngr file produced by Xilinx Synthesis Technology (XST)) is generated by the synthesis tool at earlier stages of a synthesis process when the technology mapping is not yet completed. The goal of this view is to be as close as possible to the original HDL code. In the RTL view, the design is represented in terms of macro blocks, such as adders, multipliers, and registers. Standard combinatorial logic is mapped onto logic gates, such as AND, NAND, and OR

*A.* Device Utilization Summary **Selected Device : 6slx100tfgg900-3Slice Logic Utilization:**

Number of Slice Registers: 1273 out of 1265761%
Number of Slice LUTs: 2886 out of 63288 4%
Number used as Logic: 2886 out of 63288 4%

**Slice Logic Distribution:**

Number of LUT Flip Flop pairs used: 3011
Number with an unused Flip Flop: 1738 out of3011 57%
Number with an unused LUT: 125 out of 3011 4%
Number of fully used LUT-FF pairs: 1148 out of3011 38%
Number of unique control sets: 4

**IO Utilization:**

Number of IOs: 180

**Specific Feature Utilization:**

Number of BUFG/BUFGCTRL/BUFHCEs: 1 outof 16 6%
Number of DSP48A1s: 3 out of 180 1%

**Timing Summary:**

Speed Grade: -3
Minimum period: 14.222ns (Maximum Frequency:70.312MHz)
Minimum input arrival time before clock: 3.924ns Maximum output required time after clock: 4.521ns

Table 1: Comparison with previous work

| Device Utilization Parameter | Previouswork [10] | Proposed Approach |
|---|---|---|
| Number of Slice Registers | 8493 | 1273 |
| Number of LUTFlip Flop pairs Used | 7476 | 3011 |

.

## VI. CONCLUSION

The 5 stage pipelined architecture of 32 bit RISC Processor (MIPS) has been designed using VHDL. The synthesis is performed on Xilinx ISE 14.5 using the device 6slx100tfgg900-3and simulations are done with Model-Sim simulator. The simulation result shows that the processor works perfectly. A Reduced Instruction Set computer is a microprocessor that had been designed to perform a small set of instructions, with the aim of increasing the overall speed of the processor. The RISC architecture follows the philosophy that one instruction executes in one clock cycle.
The future scope of this work includes:

- Future work will be added by increasing the number of instructions and to add more pipelined stages in the design to improve the performance of the design and to increase the speed of the processor.
- Hazard detection and management
- Interrupt facility can be added to thisprocessor.

## V.REFERENCE

[1] Pranjali S. Kelgaonkar, ShilpaKodgire, “Pipelined 32bit RISC MIPS Processor on Spartan-6 FPGA”, International Journal of Science, Engineering and Technology Research (IJSETR), ISSN: 2278-7798, Vol. 5, Issue 8, August 2016.

[2] P. Ajith Kumar, M. Vijaya Lakshmi, “Design of a Pipelined 32 Bit MIPS Processor with Floating Point Unit”, International Journal of Innovative Research in Science, Engineering and Technology, ISSN: 2319- 8753, Vol. 5, Issue 7, July 2016.

[3] S. P. Ritpurkar, M. N. Thakare, G. D. Korde, “Design and Simulation of 32-Bit RISC Architecture based on MIPS using VHDL”, IEEE, International Conference on Advanced Computing and Communication Systems (ICACCS -2015), Coimbatore, INDIA, January 2015.

[4] Topiwala, Mohit N., and N. Saraswathi, "Implementation of a 32-bit MIPS Based RISC Processor using Cadence", IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), pp. 979-983, May 2014.

[5] Sharda P. Katke, G.P. Jain, "Design and Implementation of 5 Stages Pipelined Architecture in
32 Bit RISC Processor", International Journal of Emerging Technology and Advanced Engineering, ISSN: 2250-2459, Vol. 2, Issue 4, April 2012.

[6] Samiappa Sakthikumaran, S. Salivahanan, V. S. Kanchana Bhaaskaran, "16-Bit RISC Processor Design for Convolution Application" IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011.

[7] Robert S. Plachno, VP of Audio, "A True Single Cycle RISC Processor without Pipelining", ESS Design White Paper – RISC Embedded Controller.

[8] V. B. Saambhavi and V. S. Kanchana Bhaaskaran, "A 16-BIT RISC MICROPROCESSOR USING DCPAL CIRCUITS", International Journal of Advanced Engineering Technology (IJAET), Vol. 2, no. 1, pp.1-9, 2011.

[9] Yasuhiro Takahashi, Member, IACSIT, Toshikazu Sekine, and Michio Yokoyama, "Design of a 16-bit Non-pipelined RISC CPU in a Two Phase Drive Adiabatic Dynamic CMOS Logic", International Journal of Computer and Electrical Engineering, Vol. 1, No. 1, April 2009.

[10] Shashidhar, R., A. M. Mahalingaswamy, and M. Roopa. "Design and Implementation of Low Power Pipelined 32-bit High Performance RISC Core." Journal of Electronics and Communication Systems 1, no. 1 (2016).

[11] Mood Venkanna, Rameshwar Rao and P. Chandra SekharAn Efficient Design of ASIP Using Pipelining Architecture, International Conference on Intelligent Computing and Applications, Advances in Intelligent Systems and Computing 846, https://doi.org/10.1007/978-981-13-2182-5_12.

[12] Mood Venkanna, Rameshwar Rao and P. Chandra Sekhar,Static Worst-Case Execution Time Optimizationusing dpso for asip Architecture,Ingeniería Solidariadoi: https://doi.org/10.16925/.v14i0.2230

[13] Mood Venkanna, Rameshwar Rao and P. Chandra Sekhar,Review of Design Space Exploration in ASIP Applicationsusing PSO ,International Journal of Electronics, Electrical and Computational System,ISSN 2348-117X,Volume 6, Issue 11,November 2017