DESIGN OF A DYNAMICALLY RECONFIGURABLE RISC CPU UTILISING THE MIPS INSTRUCTION SET

Sashi Bhusan Panda[a], **Sreelekha Panda**[b], **Sasmita Kumari Behera**Department of Electronics & Communication Engineering, Raajdhani Engineering ,Bhubaneswar, Odisha

Abstract: *The world of today encourages multipurpose in all things. This work develops a multifunctional RISC processor using the MIPS instruction set architecture. Dynamic reconfiguration refers to the processor's capacity to modify its internal Instruction Decode and Execute stage while the system is in operation in order to accommodate new functions. This project outlines a principle for exploiting the MIPS instruction set to increase performance in the context of microprocessor unit applications.*

Keywords - *RISC, MIPS, dynamically reconfigurable.*

I. INTRODUCTION

Microprocessor without Interlocked Pipelining Stages, or MIPS, is an acronym. Initially, Sony, Nintendo, and NEC worked on its development. It was created to solve the issues with the typical design, which is that using the same instruction set for all applications makes the instruction set busy and the system slow. For easier implementation, the MIPS processor core gave various formats to different instructions.

Additional benefits of RISC CPUs include fewer instructions, higher performance, and simpler structures that are simpler to implement. In embedded systems, the RISC CPU is widely used. There is a load store load instruction on RISC processors.

This research focuses on the performance-enhancing integration of the MIPS instruction set into a RISC processor.

II. LITERATURE SURVEY

This section provides a quick overview of some RISC static processor design works.

Make sure a RISC processor's power consumption is reduced in the plan shown in [1]. Here, they used the MIPS architecture to develop a RISC processor. To cut back on power, they used the multi Vt method and clock gating. To eliminate any potential hazards, they have also included a hazard detecting unit. In the paper [2], an FPGA-designed RISC processor was created. They used VHDL to define the system using a top-down design approach. They examined the RISC CPU instruction set's design philosophy as well as the MIPS instruction format, instruction data flow, and decoder module functionality.

Stall in the MIPS architecture has been minimised in section [3]. In pipeline architecture, stall frequently occurs, resulting in longer clock cycles. Pre-fetching units were included to reduce stall in this situation.

A 16-bit non-pipelined RISC processor with a 24 instruction set is proposed in another work [6] and is utilised for signal processing applications.

III.PROPOSED WORK

A. MIPS PROCESSOR ARCHITECTURE

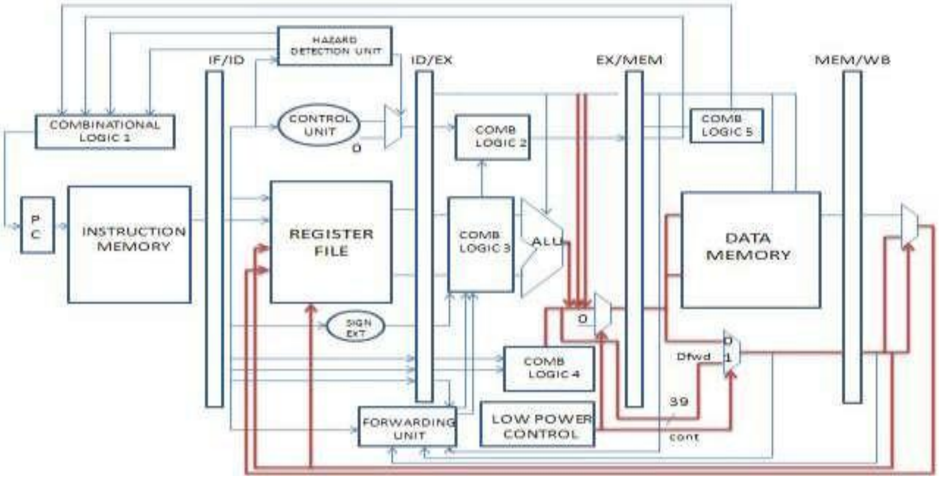
Fix-length straightforward decoded instructions, memory access restricted to load and store instruction format, and a sizable general purpose register file are all features of the MIPS RISC microprocessor's architecture.

The 32 bit RISC MIPS processor has 5 stages.

1. Instruction fetch (IF)
2. Instruction Decode (ID)
3. Execution (EXE)
4. Data memory (MEM)
5. Write back (WB)

The MIPS single cycle processor can fetch, decode, execute, read memory, and write back instructions all in a single clock cycle.

Figure 1 depicts the RISC processor's architecture, including all five stages. Processor performance is enhanced through pipelining.



B. *DYNAMIC PROCESSOR*

Fig 1: Block Diagram of RISC processor

4. If the processor is 16 bit, Load 16 bit LSB

Dynamic processors have the ability to configure themselves autonomously. The processor in this instance is constructed as an 8-bit, 16-bit, and 32-bit processor. We can alter our CPU to meet our needs.

Using selection lines, we may choose the appropriate processor. We have a 2-bit selection line available.

01--8 bit processor

10--16 bit processor 11—32bit processor

In real time projects it provides the flexibility to switch between the processors.

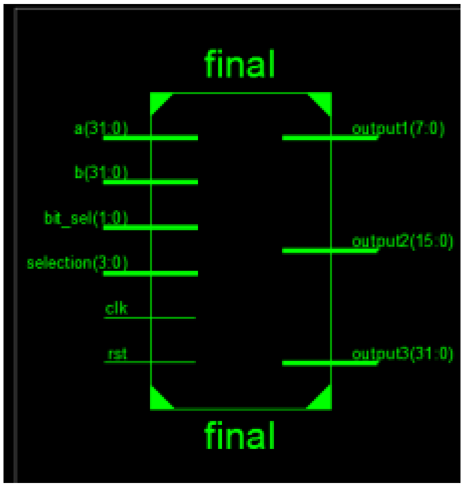


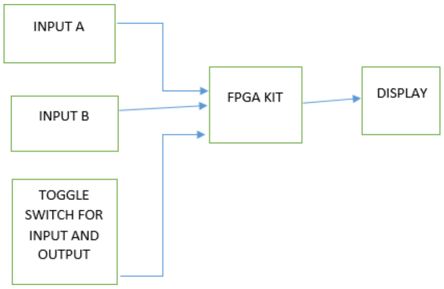
Fig 2: Schematic view of dynamic processor Algorithm:

1. Accept the inputs as 32 bit
2. Select the processor.
3. If the processor is 8 bit, Load 8 bit LSB of inputs.of inputs.
5. If the processor is 32 bit, load the complete value.
6. Select the instruction.

7. All the values are fetched, decoded and executed.
8. The result will be stored in the output register
9. According to the processor selection only the corresponding set of registers will be active.
10. The output will be displayed in corresponding 8 bit,16 bit,32 bit output lines.

C.FPGA IMPLEMENTATION

The processor is synthesized in Xilinx FPGA. Field programmable gate array.



RESULTS

The processor is created using VHDL and ModelSim for simulation. Results were checked using the synthesis tool Xilinx Spartan - 3E, and the work was then implanted in an FPGA.

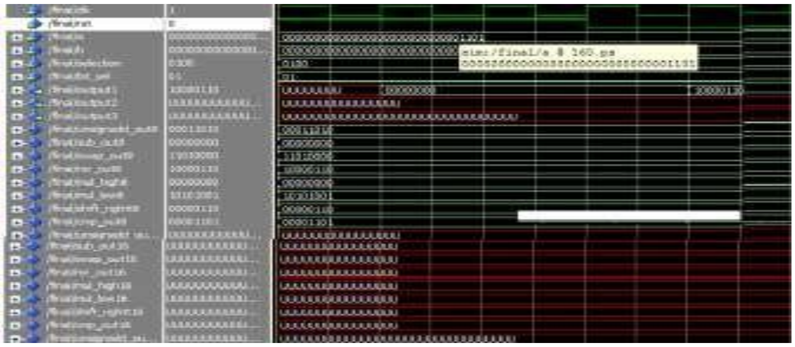


Fig 3: 8 bit RISC processor

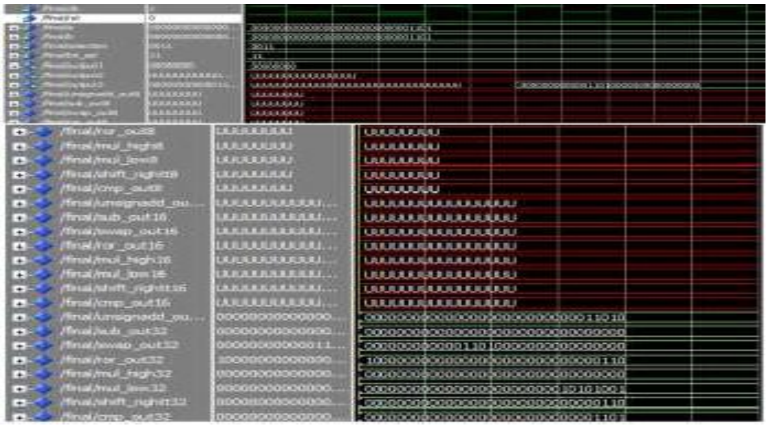


Fig43: 16 bit RISC processor

Fig 5: 32 bit RISC processor

CONCLUSIONS

Better performance is intended for the system under consideration. Verilog HDL was used to successfully design the CPU, and ModelSim 10.4 was used to simulate it before it was synthesised on a Xilinx Spartan-3E for FPGA implementation. While the CPU is functioning, the system can reconfigure itself. More instructions can be added for future work.

REFERENCES

[1] **P.V.S.R.Bharadwaja**, **K. Ravi Teja, M.Naresh babu**, **K.Neelima** (2015,May 20-21) Advanced Low Power RISC Processor Design using MIPS Instruction, *IEEE International Conference On Recent Trends In Electronics and Communication systems*.

[2] Gautham P, Parthasarathy R, Karthi Balasubramanian (2014) Low-Power Pipelined MIPS Processor Design .

[3] **Mrs. Rupali S. Balpande. Mrs.Rashmi S. Keote.** (2014,May) Design of FPGA based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor, *2011 International Conference on Communication Systems and Network Technologies*

[4] **Panjali S Kalgoankar,Prof.Shilpa.Kodgire** Design of 32 bit MIPS RISC Processor based on SoC, International journal of latest trends in Engineering and technology(IJLTET)

[5] **Zulkifli.M, Yudhanto, Y.P, Soetharyo NAdionoT** Reduced Stall MIPS Architecture using Pre-Fetching Accelerator**.**, *2009 International Conference on Electrical Engineering and Informatics5-7 August 2009, Selangor,*
Malaysia

[6] R.Uma,"Design and Performance Analysis of 8-bit RISC Processor using Xilinx Tool"2012 *International Journal of Engineering Research and Applications (IJERA)*

[7] Anand Nandakumar Shardul, "16-Bit RISC Processor Design for Convolution Application", -2013 *International Journal of Advancements in Research & Technology*.

[8] Ciletti, Michael D. "Advanced Digital Design with the Verilog HDL."Upper Saddle River, NJ: Pearson Education Inc. 2003

[9] Sagar Bhavsar , Akhil Rao , Abhishek Sen , Rohan Joshi ," A 16-bit MIPS Based Instruction Set Architecture for RISC Processor" *April 2013 International Journal of Scientific and Research Publications*.

[10] V. B. Saambhavi and V. S. Kanchana Bhaaskarana," 16-bit RISC microprocessor using dcpal circuits*"* *International Journal of Advanced Engineering Technology*

[11] .Dalal, A.Ganesh, Aishwarya.D "An 8 bit Power-Efficient MIPS Processor " Advanced VLSI Design Sping 2014