ISSN 2320-9984 01 >
9 772320 998022

Volume 2 Issue 1 | March 2014 ISSN: 2320-9984 (Online)

International Journal of Modern Engineering & Management Research

Website: www.ijmemr.org

New Approach for Affine Combination of A New Architecture of RISC cum CISC Processor

Shaily Agrawal
Research Scholar
Department of Electronics and Communication Engg.
Takshshila Institute of Engineering and Technology,
Jabalpur M.P., [INDIA]
Email : juhi1896@gmail.com

Prof. Shadma Pragi
Professor & Guide,
Department of Electronics & Communication Engg,
Takshshila Institute of Engineering and Technology,
Jabalpur, (M.P.) [INDIA]

Abstract*—Microprocessor is a general purpose IC which follows the instructions given to it, and the instructions set for the microprocessor designed such a way that it can handle any type of computations. Different type of architectures are available in the market like CISC, RISC, ARM SHARC etc. all of them have their own different approaches to perform computations. This Paper work which is RISC cum CISC processor architecture's concept comes after the research and mulling over on all these architecture and proposed architecture is a new microprocessor architecture which will have CISC (Complex instruction Set Computing) type instruction set (large instruction set good for multiple applications) and RISC (Reduced Instruction Set Computing) features for executing every instruction in one cycle. This paper work aims to reduce the area requirements than the conventional RISC processor architecture and to support powerful CISC instruction set.*

Keywords:-*CACHE memory, CISC, FPGA, microcode, op-code, RISC, SHARC*

1. INTRODUCTION

From the architecture point of view, the microprocessor chips can be classified into two categories: Complex Instruction Set Computers (CISC) and Reduce Instruction Set Computers (RISC). In either case, the objective is to improve system performance. The debates between these two architectures made this research area very interesting, challenging, and sometimes confusing. CISC computers are based on a complex instruction set in which instructions are executed by microcode. Microcode allows developers to change hardware designs and still maintain backward compatibility with instructions for earlier computers by changing only the microcode, thus make a complex instruction set possible and flexible [2, 3] .The instructions set for the microprocessor designed such a way that it can handle any type of computations. [4] Although CISC designs allow a lot of hardware flexibility, the supporting of microcode slows microprocessor performance because of the number of operations that must be performed to execute each CISC instruction. A CISC instruction set typically includes many instructions with different sizes and execution cycles, which makes CISC instructions harder to pipeline. or RISC is a CPU design strategy based on the insight that simplified (as opposed to complex) instructions can provide higher performance if this simplicity enables much faster execution of each instruction [1].

2. CONVENTIONAL RISC & CISC

To illustrate the large contrast between the instruction encoding formats used by CISC and RISC processors, the instruction formats for the Intel x85 and Compaq Alpha processor

architectures [10] are shown in Figure 1. In the case of x85 there is a lot of sequential decoding that has to be accomplished (although modern x85 processors often pre decode x85 instructions[10] while loading them into the instruction cache, and store instruction hints and boundary information as 2 or 3 extra bits per instruction byte). For the Alpha (and virtually every other classic RISC design) the instruction length is fixed at 32-bits and the major fields appear in the same locations in all the formats. It is standard practice in RISC processors to fetch operand data from registers (or bypass paths) even as the instruction op-code field is decoded.

There is still considerable controversy about which architecture is better. For ex. RISC is cheaper and faster and therefore the architecture of the future. Sometimes it is note that by making the hardware simpler, RISC puts a great burden on the software. Software needs to become more complex. Software developers need to write more lines for the same tasks [3].

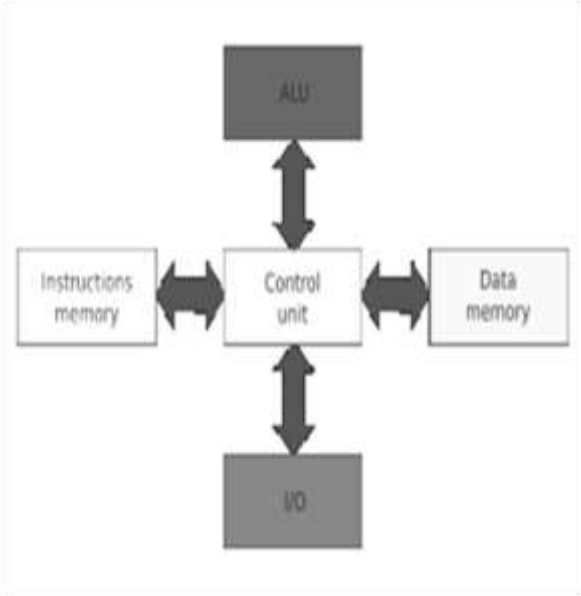

Figure 1: Conventional CISC Design

3. PROPOSED ARCHITECTURE

This proposed Design is of eight bit Processor using Verilog HDL, the designed module will be synthesized using Xilinx ISE 9.1i Web pack, and the verification will be done on ISE simulator, and then for validation be the design module will be implemented on Xilinx FPGA (Field programmable Gate Array) Spartan3E XC3500-6PQ208.[11]

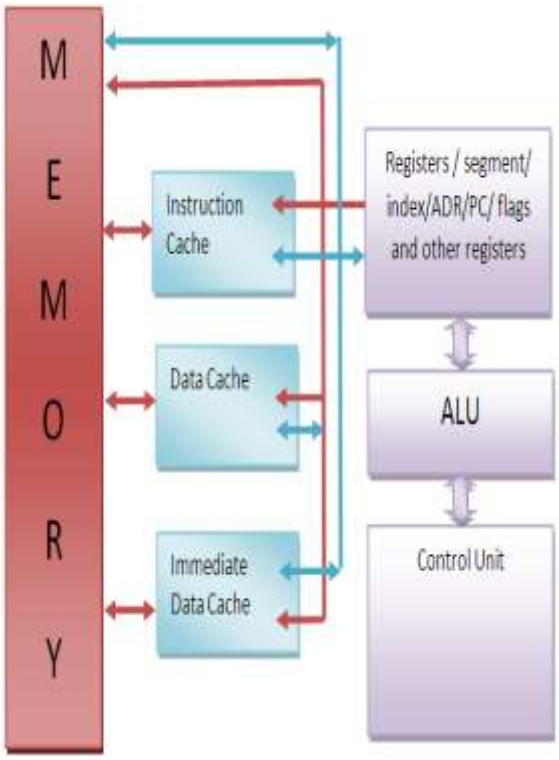

Figure 2: Proposed Architecture

The Proposed model for this paper comes after mulling over all the architectures like RISC, CISC, Automatic RISC Machine (ARM), Super Harvard architecture (SHARC) and paper work is proposing a new microprocessor architecture which will have CISC type instruction set (large instruction set good for multiple applications) and RISC type feature for executing every instruction in one cycle. To achieve these we just exclude few instructions (total five) of CISC (i.e. x85). Though still it has covered all small scale and medium scale applications with this proposed architecture on cost of nothing and all sophisticated scale application on cost of extra few nanoseconds.

The proposed design is shown in Figure 2 which comprises of the mixture of both RISC type and CISC type architecture. It employs the double Harvard architecture which further subdivides the program memory into

instruction memory, immediate value and immediate address.

A processor generally interfaces with main memory for data. Proposed design have four isolated memory blocks to store different kind of data, instruction cache for op-code data cache for frequent data and immediate data cache for program immediate operands & main memory is for storing rest of the data, proposed work has separated interfacing line for all three cache it makes interfacing and execution of every instruction in one clock cycle only like RISC though instruction can have size more than one byte like CISC.[8]

The Proposed design is a new application specific RISC processor architecture to overcome the performance bottleneck of traditional RISC processor.

The concept of this proposed design is to design an area and speed optimized MASIP (Multiple Applications Specific Intrusions Processor), an ideal processor core for all small scale applications[5, 6]. Proposed processor will be a RISC architecture with CISC instruction set and multi level cache memory along with separate op-code, operands and temporary data memories. Proposed work is a combination of Super Harvard & RISC and multiple caches to achieve a better execution of CISC instruction.

4. PROBLEM & REMEDY

Problem with existing general purpose systems are that it required a huge set of hardware and very complex design which support small and big every application. Proposed work is suitable with all instructions of 8085. Problem with embedded system is that it performs only specific work and hardware need to select as per the application. ASIC is best performer if specific work and computation requires for any task but it requires lots of manufacturing time in case of Full custom when go for semicustom design again it requires selection specific FPGA. To overcome the above problems thesis work is going to implement a unique processor architecture which will be powerful enough as general purpose processor at most of applications and better than embedded processors and controllers. Since paper work has applied the pipelining architecture so this will lead to a limitation, and which causes the JUMP instruction to fail. In order to remove this we will fetch and decode the JUMP instruction at the same cycle which help us in utilizing the machine cycle efficiently.

5. RESULT & SIMULATION

The Proposed core architecture has been designed and simulated with the help of Xilinx ISE software. The result is as shown below.

Table 1: Logic Utilization

Logic Utilization	Used
No. of slice registers	324
No. of slice LUT's	1271
No. of fully used bit slices	214
No. of bonded IOB's	88
No. of block RAM/FIFO	1

In proposed design 324 no. of slice registers, 1271 LUT's, 214 no. of bit slices, *88* bounded IOB 's and 1 RAM are used.

Figure 3 explains the simulation results of overall design. By this simulation result it can be seen that the proposed design uses 324 no. of slices. And max delay is 8.7 ns. So the area is reduced and this proposed design is fast.

Table 2: Comparative Analysis

Para - meter	Base [1]	Base [2]	Base [3]	Proposed design	Remarks
No. of slices	428	448	--	324	25% less than [1] 28% less than [2]
Max. Delay (ns)	--	10.38	11.1	8.7	16% less than[2] 22% less than[3]

This table shows that number of slices are used by base[1] and base [2] is 428 and 448 respectively. But in proposed design this is 324. So the area will be reduced in proposed architecture. And maximum delay by base [2] and base[3] is 10.38ns and 11.1ns respectively. But in proposed design max delay is 8.7ns. so this is faster than previous results.

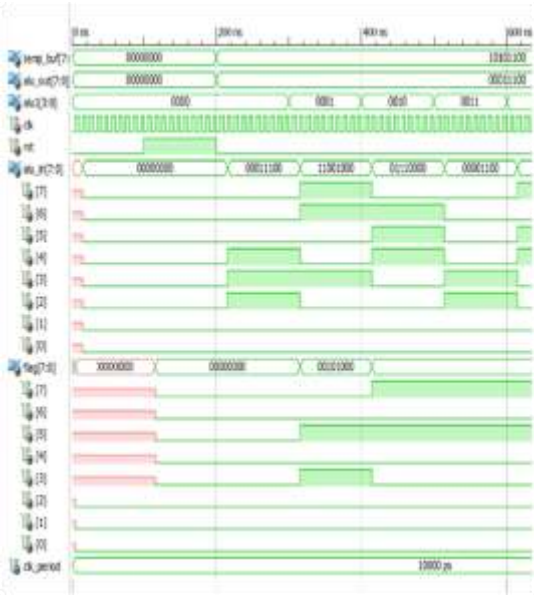

Figure 3: The simulation results of overall design

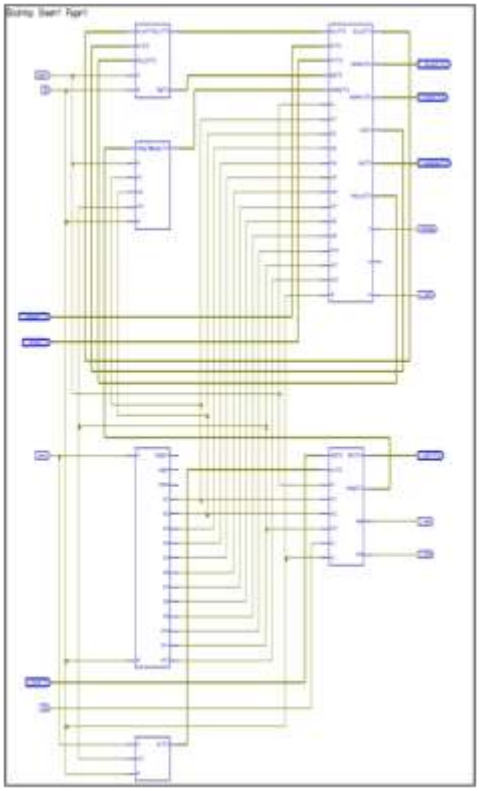

Figure 4: RTL schematic of proposed Processor design

6. CONCLUSION

A new trend of CISC and RISC architectures is addressed. This paper gives a new trend of fast processor design with less no. of splices used. Some of previous works was highlighted, and a new technology is presented. For the best performance and scalability processor, the following are important factors are been introduced in proposed microprocessor architecture: (a) fast cache-to-cache communication, (2) large L2 or shared capacity, (3) fast L2 access delay, and (4) fair resource (cache) sharing. Results observed are better than previous work.

REFERENCES :

[1] Yanfen Chen*,Wuchen Wu,Ligang Hou,Jie Hu, "Design and Implementation of 8-bit RISC MCU," *IEEE* 978-1-4244-6736-5©2010 IEEE.

[2] Tomas Balderas-Contreras, Rene Cumplido *, Claudia Feregrino-Uribe "On the design and implementation of a RISC processor extension for the KASUMI encryption algorithm. Sciencedirect, Computer and electrical engineering 34 (2008)

[3] D. Seal, ARM Architecture Reference Manual. 2nd Edition, Addison-Wesley 2001.

[4] M. K. Jain, M. Balakrishnan, and A. Kumar, "ASIP Design Methodologies: Survey and Issues," Proceeding of Fourteenth International Conference on VLSIDesign, pp. 76-8 1, Jan. 2001.

[5] K. Wakabayashi, and T. Okamoto, "C -Based SoC Design Flow and EDA Tools: An ASIC and System Vendor Perspective," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 19, No. 12, pp. 1507-1522, Dec. 2000.

[6] M. Gschwind, "Instruction set

selection for ASIP design," Proceeding of the Seventh of International Workshop on Hardware/ Software Codesign, pp. 7-1 1, May 1999.

[7] I. H. Jeng, F. Lai, and Y. D. Tseng, "FACE: Fine-tuned Architecture Codesign Environment for ASIP Development," Design Automation for Embedded Systems, Kluwer Academic Publishers, Vol. 4, No. 4, pp. 329-25 1, Oct. 1999.

[8] C. C. Lee, and S. P. Cheng, "Application-Specific RISC Architecture Enhancements: Circular Buffering of Registers for Efficient Filtering," Proceedings of 2004 International SOC Design Conference, Seoul Korea, Oct. 2004.

[9] V. G. Oklobdzija, "An Algorithmic and Novel Design Of a Leading Zero Detector Circuit: Comparison with Logic Synthesis," IEEE Transactions on Very Large Scale Integration Systems, Vol. 2, No. 1, pp. 124-128, Mar. 1994

[10] Single bit N channel Microprocessor 8085A/8085A-2 data sheet, Intel, APN-1242B, 1977

[11] http://www.xilinx.com/support.html