IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676, p-ISSN: 2320-3331
PP 46-50
www.iosrjournals.org

Review on 32-bit MIPS RISC Processor using VHDL

Mr. Sagar P. Ritpurkar[1], Prof. Mangesh N. Thakare[2], Prof. Girish D. Korde[3]

[1]*(Department of EXTC, B. D. College of Engineering Wardha, INDIA)*
[2]*(Department of EXTC, B. D. College of Engineering Wardha, INDIA)*
[3]*(Department of EXTC, B. D. College of Engineering Wardha, INDIA)*

ABSTRACT : *In this paper, we will analyze MIPS instruction format, instruction data path, decoder module function and design theory based on RISC CPU instruction set. Here, we will use pipeline design process which involves instruction fetch (IF), instruction decoder (ID), execution (EXE), data memory (MEM) and write back (WB) modules of 32-bit CPU based on RISC CPU instruction set. Function of IF module mainly includes fetch instruction, latch module, address arithmetic module, check validity of instruction module and synchronous control module. The instruction decoder includes register file, write back data to register file, sign bit extend and relativity check. Finally, the coding will be done in VHDL and synthesis will be done using Xilinx ISE simulator.*

Keywords *- Data Flow, Data Path, Instruction Set, MIPS, Pipeline.*

I. INTRODUCTION

MIPS Technologies has a strong customer licensee base in home electronics and portable media players; 75 percent of Blu-ray Disc players are running on MIPS Technologies processors. In the digital home, the company's processors are predominately found in digital TVs and set-top boxes. The Sony PlayStation Portable uses two processors based on the MIPS32 4K processor. Within the networking segment, licensees include Cavium Networks and Net logic Microsystems. Licensees using MIPS to build smart phones and tablets include Actions Semiconductor and Ingenic Semiconductor. Tablets based on MIPS include the Cruz tablets from Velocity Micro. TCL Corporation is using MIPS processors for the development of smart phones. Companies can also obtain an MIPS architectural license for designing their own CPU cores using the MIPS instruction set. MIPS Technologies is predominately used in conjunction with Android and Linux operating systems.

Building on the success of the award-winning Aptiv generation of CPU cores, Imagination is now giving a preview of the next generation of MIPS processors. Codenamed 'Warrior', the MIPS Series5 CPUs feature cores based on the MIPS32 and MIPS64 instruction set architectures, with a focus on superior performance efficiency across the high-end, mid-range and entry-level/microcontroller CPUs. Just as the 'Rogue' codename for our latest Power VR Series6 GPU family proved to be a favorite with our ecosystem, we thought we'd give our CPUs a similar naming convention: enter the MIPS Series5 'Warrior' CPU cores, a forthcoming line-up of new MIPS32 and MIPS64 CPU IP cores from Imagination, designed to address a wide range of markets, from mobile and networking to embedded and microcontroller applications.

RISC or Reduced Instruction Set Computer is a design philosophy that becomes mainstream in the last few years, as the quest for raw speed has dominated the highly competitive computer industry. 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. RISC design resulted in computers that execute instructions faster than other computers built of the same technology. In a RISC machine, the instruction set is based upon a load store approach. Only load and store instructions access memory. This is the key to single-cycle execution of instructions. Comparing to CISC, RISC CPU have more advantages, such as faster speed, simplified structure easier implementation. RISC CPU is extensive use in embedded system. Developing CPU with RISC structure is necessary choice [1].

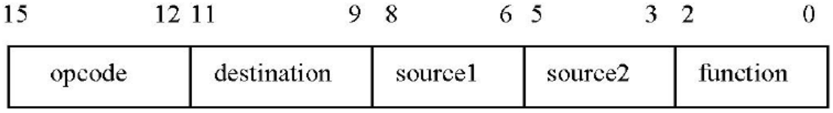
II. RELATED WORK

Mrs. Rupali S. Balpande [1] have proposed an MIPS instruction format, instruction data path decoder module function and design theory based on RISC CPU instruction set. Furthermore, they propose a design of instruction fetch (IF) module of 32-bit CPU based on RISC CPU instruction set. Through analysis of function and theory of RISC CPU instruction decoder module, they also propose a design of instruction decoder (ID) module of 32-bit CPU by pipeline theory. The instruction decoder includes register file, write back data to register file, sign bit extend, relativity check, and it is simulated on QuartusII successfully. They have also design an instruction set, dataflow and pipeline design of RISC CPU based on MIPS. In this research, they have adopt top-down design method and use VHDL to describe system. At first, they design the system from the top, and do in-depth design gradually. The structure and hierarchical of design is very clear. It is easy to edit and debug. Design of instruction fetch (IF) stage simulates, integrate and routes on Quartus II 4.3. Their result indicates IF stage completes prospective function.

Mamun Bin Ibne Reaz [2] Presents a design methodology of a single clock cycle MIPS RISC Processor using VHDL to ease the description, verification, simulation and hardware realization. The RISC processor has fixed-length of 32-bit instructions based on three different format R-format, I-format and J-format, and 32-bit general-purpose registers with memory word of 32-bit. The MIPS processor is separated into five stages: instruction fetch, instruction decode, execution, data memory and write back. The control unit controls the operations performed in these stages. All the modules in the design they have coded in VHDL, as it is very useful tool with its concept of concurrency to cope with the parallelism of digital hardware. The top level module connects all the stages into a higher level. After detecting the particular approaches for input, output, main block and different modules, the VHDL descriptions are run through a VHDL simulator, followed by the timing analysis for the validation, functionality and performance of the designated design that demonstrate the effectiveness of the design. By simulating with various test vectors, they have successfully designed, implemented and tested a single clock cycle 32-bit MIPS RISC processor using VHDL. They use 8-bit datapath with the intention of hardware realization onto Altera FLEX 10K FPGA chip. Currently, they are conducting further research that considers further reductions in the hardware complexity in terms of synthesis and then download the code into Altera FLEXIOK: EPFlOKlOLC84 FPGA chip on LC84 package for hardware realization.

Rohit Sharma [4] presents the design and implementation of a 64-bit reduced instruction set (RISC) processor with built-in-self test (BIST) features. A built-in self-test (BIST) or built-in test (BIT) is a mechanism that permits a machine to test itself. Key features of the design including its architecture, data path, and instruction set are presented. The design is implemented using VHDL and verified on Xilinx ISE simulator. The processor is designed keeping in mind specific applications. The proposed design may find applications where automation and control is required. Future applications may include its use in vending machines, ATMs, mobile phones, and portable gaming kits. This 64 bit RISC processor works on two clock cycles. ‘xclk’ is the external clock which is always equal to one. ‘clk’ is the clock which triggers the inputs and gives us the desired output. The instruction set, architecture, and data path are analyzed. The processor is designed considering MICA architecture and is implemented using VHDL. The architecture of the ALU consists of two parts, the Operation Architecture, which does the actual operation of the ALU, and the Testing Architecture, which comes into play only during testing. The Operation Architectures consists of a five units, 4-bit Carry Look Ahead adder (CLA), and a 4-bit AND, OR, XOR and INVERTER gates. There is a MUX which uses the select pins to select one of the results from the above five units. The data path and the architecture for the proposed design are designed. Instruction set is design for 64-Bit RISC Processor. A 64-bit RISC processor with 33 instruction set has been designed. Every instruction is executed in two clock cycles. The design is verified through exhaustive simulations. Some of the applications are presented. In the near future more applications like the ATMs, mobile phones and portable gaming kits can be implemented.

Pravin S. Mane [5] describes a 16-bit RISC processor design using VHDL. They use a hierarchical approach so that basic units can be modeled using behavioral programming. Then, these basic units are combined using structural programming. They use four stage (viz. instruction fetch stage, instruction decode stage, execution stage and memory/lO, write back stage) pipelining to improve the overall CPI (Clock Cycles per Instruction). They also use hardwired control approach to design the control unit as against micro programmed control approach in conventional CISC processor. They have shown the general instruction format for 16-bit RISC Processor.



15 12	11 9	8 6	5 3	2 0
opcode	destination	source1	source2	function

FIGURE 1: General Instruction Format [5]

This processor has one input port, one output port and six hardware vectored interrupts along with 16-bit address bus and 16-bit data bus. Structural hazards are dealt with the implementation of pre-fetch unit, data hazards are dealt with forwarding and control hazards are dealt with flushing and stalling. They have implemented the design on FPGA for verification purpose. They achieved the maximum throughput of execution as one instruction per clock pulse provided that there are no stalls by using pipelining approach. This is possible due to hardwired approach for design of control unit and fixed length instruction format. To resolve data hazards, result forwarding is efficient than stalling as it remove the penalty of time in handling such conflicts. They use prefetch unit for handling the structural hazards while flushing is used to handle control hazards. The prefetch buffer is like a small cache storing tag along with instruction fetched. It does not work on FIFO principle. This design is modeled and simulated using VHDL and then implemented on FPGA successfully. The maximum frequency of operation they have obtained on the Xilinx's Spartan-Il FPGA is 26-MHz.

III. PROPOSED WORK

The proposed work is likely to achieve;

(a) The instruction set which will include the instruction formats of R-type, I-type and J-type.

	(31 to 26)	(25 to 21)	(20 to16)	(15 to11)	(10 to 6)	(5 to 0)
R-Format	Opcode	Rs	Rt	Rd	shift	Function

FIGURE 2: REGISTER FORMAT (R-TYPE)

	(31 to 26)	(25 to 21)	(20 to 16)	(15 to 0)
I- Format	Opcode	Rs	Rt	Address/Immediate value

FIGURE 3: IMMEDIATE FORMAT (I-TYPE)

	(31 to 26)	(25 to 0)
J- Format	Opcode	Branch target address

FIGURE 4: BRANC H TYPE FORMAT (J-TYPE)

(b) Datapaths will include the R-Format, RI-Format, Load Word and Memory Word Data Path.

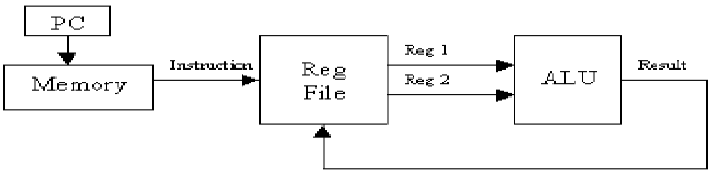


FIGURE 5: R-FORMAT DATAPATH

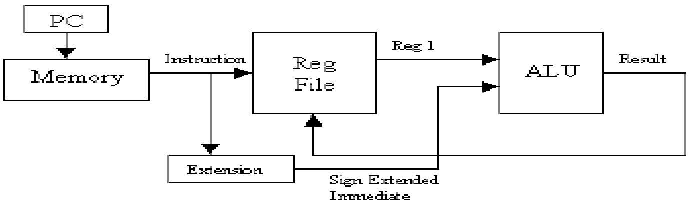


FIGURE 6: RI-FORMAT DATAPATH

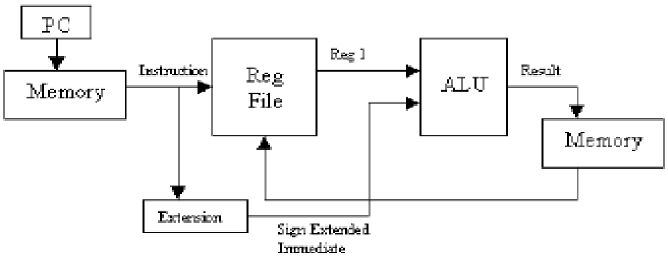


FIGURE 7: LOAD WORD DATAPATH

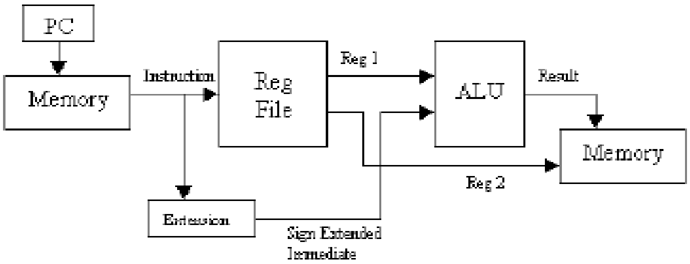


FIGURE 8: MEMORY WORD DATAPATH

(c) Pipeline will include instruction fetch (IF), instruction decoder (ID), execution (EXE), memory/ IO (MEM) and write-back (WB).

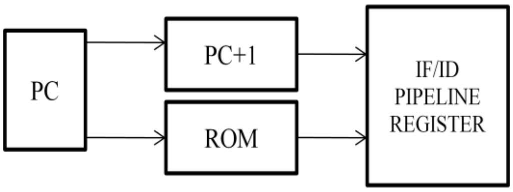


FIGURE 9: INSTRUCTION STAGE

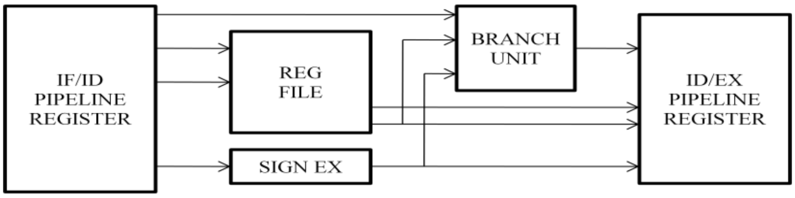


FIGURE 10: INSTRUCTION DECODER STAGE

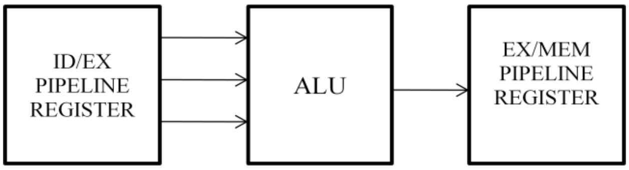


FIGURE 11: EXECUTION STAGE

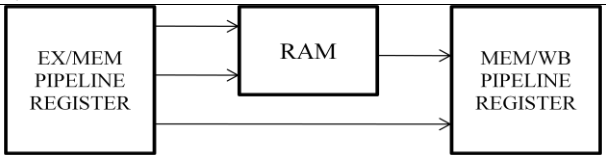


FIGURE 12: MEMORY AND I/O STAGE

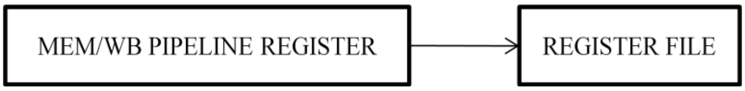


FIGURE 13: WRITE BACK STAGE

IV. CONCLUSION AND FUTURE SCOPE

The use of VHDL for modeling is especially appealing since it provides a formal description of the system and allows the use of specific description styles to cover the different abstraction levels (architectural, register transfer and logic level). The MIPS is one of the RISC Processor. This is proposed because of its area, less complicated, less power as well as faster speed. In particular, the MIPS RISC Processor uses reduced instruction set hence results to minimize human effort. Pipeline decomposition enhances throughput rate of instruction. There is a desire to enhance the processor's speed and simplify the hardware for reasons of cost. If we suppose to implement this processor using FPGA technology it is possible to upgrade the system with new features as per user requirements. Hence, a 32-bit MIPS RISC Processor with high speed will be probable outcome of this work and instructions will be design up to 30.

REFERENCES

[1] Mrs. Rupali S. Balpande, Mrs.Rashmi S. Keote, *Design of FPGA based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor*, 2011 International Conference on Communication Systems and Network Technologies, 978-0-7695-4437-3/11, 2011 IEEE.

[2] Mamun Bin Ibne Reaz, MEEE, Md. Shabiul Islam, MEEE, Mohd. S. Sulaiman, MEEE, *A Single Clock Cycle MIPS RISC Processor Design using VHDL*, ICSE2002 Proc. 2002, Penang, Malaysia, 0-7803-7578-S/02/S, 2002 IEEE.

[3] Kui YI, Yue-Hua DING, *32-bit RISC CPU Based on MIPS Instruction Fetch Module Design*, 2009 International Joint Conference on Artificial Intelligence, 978-0-7695-3615-6/09, 2009 IEEE.

[4] Rohit Sharma, Vivek Kumar Sehgal, Nitin Nitin1, Pranav Bhasker, Ishita Verma, *Design and Implementation of a 64-bit RISC Processor using VHDL*, UKSim 2009: 11th International Conference on Computer Modelling and Simulation, 978-0-7695-3593-7/09, 2009 IEEE.

[5] Pravin S. Mane, Indra Gupta, M. K. Vasantha, *Implementation of RISC Processor on FPGA*, 1-4244-0726-5/06, 2006 IEEE.

[6] Bai-ZhongYing, Computer Organization, Science Press, 2000.11.

[7] Wang-AiYing, Organization and Structure of Computer, Tsinghua University Press, 2006.

[8] Wang-Yuan Zhen, IBM-PC Macro Asm Program, Huazhong University of Science and Technology Press, 1996.9.

[9] MIPS Technologies, Inc. MIPS32™ Architecture For Programmers Volume II: The MIPS32™ Instruction Set June 9, 2003.

[10] Zheng-WeiMin, Tang-ZhiZhong, Computer System Structure (The second edition), Tsinghua University Press, 2006.