INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

VHDL Design and Simulation of a 32 Bit MIPS RISC Processor

Prerna Kumar
Department of Electronics and Telecommunication Engineering
G.H. Raisoni Institute of Engineering and Technology for Women, Nagpur, India
Radhika Bajaj
Department of Electronics and Telecommunication Engineering
G.H. Raisoni Institute of Engineering and Technology for Women, Nagpur, India
Kasturi Ramesh Iyer
Department of Electronics and Telecommunication Engineering
G.H. Raisoni Institute of Engineering and Technology for Women, Nagpur, India
Dinesh R. Rotake
Department of Electronics and Telecommunication Engineering
G.H. Raisoni Institute of Engineering and Technology for Women, Nagpur, India

Abstract:
Today's smart phones, tablets, cameras, routers and play stations make use of MIPS RISC technology due to several key advantages. Due to the presence of the reduced instruction set, the processing time of the processor is reduced. It has improved flexibility and adaptability, is faster and inexpensive. So, here we are designing a microprocessor without interlocked pipelining stages i.e. MIPS RISC Processor using Xilinx software. For that firstly we are designing the Read only memory, i.e. ROM and then the random access and then we have designed the instruction fetch unit.

Key words*: MIPS, RISC, VHDL, Xilinx, RTL Schematic, Technology Schematic, ROM, RAM, Fetch unit*

1. Introduction

The concept of RISC was popularised in the 1980s, through two projects in Stanford University and University of California, Berkeley. Reduced instruction set computing (RISC) is a CPU design strategy based on the insight that simplified instructions can provide higher performance. It has load/store architecture. MIPS implementation is widely used in embedded systems. MIPS RISC is an abbreviation for microprocessor without interlocked pipelining stages reduced instruction set computer. It is designed in Xilinx as to reduce the instruction set in the programmable memory. As a result the processor will contain the necessary logics for the implementation.

2. Various Units of the Processor

2.1. Read only Memory (ROM)

ROM stands for Read only memory. Data stored in ROM cannot be modified. ROM is a hardwired memory which can only be read from. The different types of ROM are:

- PROM
- EPROM
- EEPROM
- EAROM

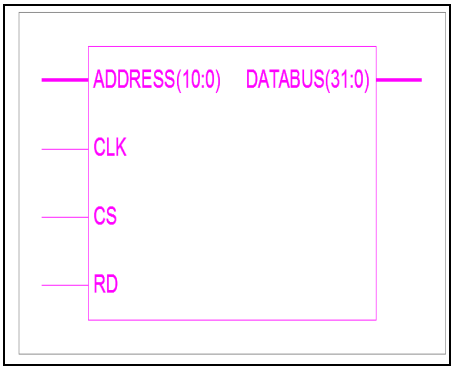


Figure 1: Block diagram of ROM

CLK	CS	RD	OUTPUT
0	0	0	ZZ
0	0	1	ZZ
0	1	0	ZZ
0	1	1	ZZ
1	0	0	ZZ
1	0	1	ZZ
1	1	0	XX
1	1	1	RD

Figure 2: Truth Table

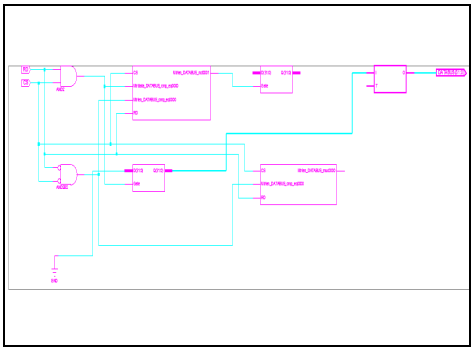


Figure 3: RTL Schematic

ROM reads data from the specified address when the clock is provided CLK is enabled and when the ROM chip is selected i.e. CS is enabled and when read is enabled.

2.2. Random Access Memory (RAM)

RAM stands for random access memory. It is a form of computer data storages. Random access device allows stored data to be accessed directly in any random manner. RAM is of two types:

- DRAM
- SRAM

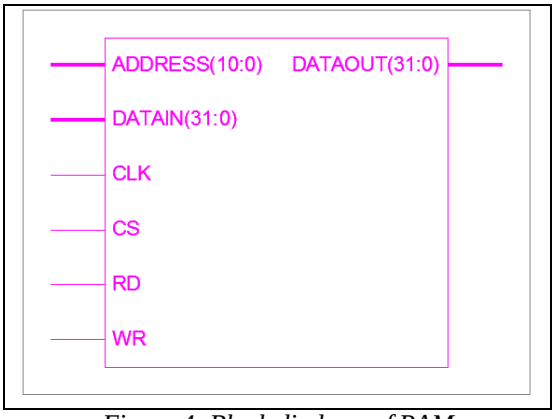


Figure 4: Block diadram of RAM

CLK	CS	RD	WR	OUTPUT
0	0	0	0	ZZ
0	0	0	1	ZZ
0	0	1	0	ZZ
0	0	1	1	ZZ
0	1	0	0	ZZ
0	1	0	1	ZZ
0	1	1	0	ZZ
0	1	1	1	ZZ
1	0	0	0	ZZ
1	0	0	1	ZZ
1	0	1	0	ZZ
1	0	1	1	ZZ
1	1	0	0	XX
1	1	0	1	WR
1	1	1	0	RD
1	1	1	1	XX

Figure 5: Truth Table

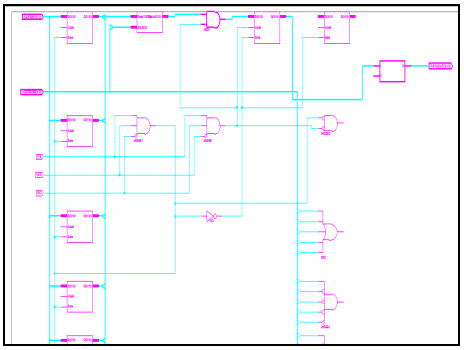


Figure 6: RTL Schematic of RAM

RAM performs read operation when chip select is enabled and clock is enabled and read pin is enabled.RAM performs write operation when chip select, clock and write pin are enabled. When read and write both are enabled no operation is performed.

2.3. Instruction Fetch Unit

Fetch unit is a part of the CPU. Instruction fetch unit fetches the Op-code as well as the operand. It reads data from the memory and stores data into the memory. Fetch unit is connected to the memory and the ALU. It performs load and store operations

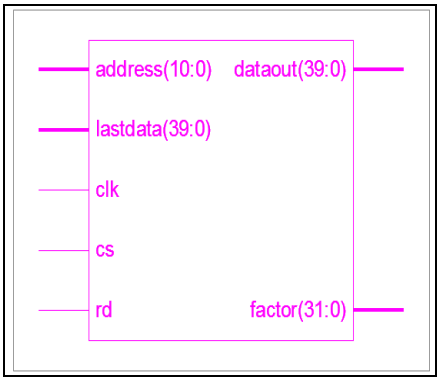


Figure 7: Block diagram of Fetch unit

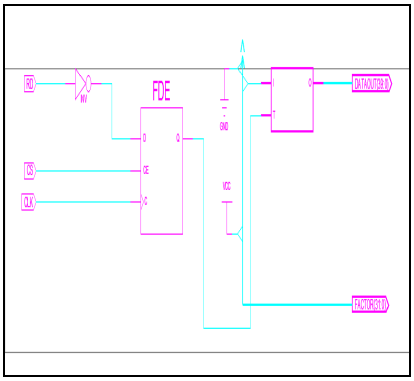


Figure 8: RTL Schematic of Fetch Unit

The last data and data out pin in the fetch unit consist of 40 bits wherein the first 8 bits are Op-code and next 32 bits are operand. Factor is a 31 bit entity that is disabled when the fetch cycle is non-functional and is enabled when the fetch cycle is functional. Fetch unit is functional when chip select is enabled and when clock is enabled. The full disk encryption unit encrypts data when data is being written into memory and decrypts data when data is being read from the memory.

2.4. Arithmetic Logic Unit (ALU)

In digital electronics, an ALU is a digital circuit that performs arithmetic and logical operations. The ALU is a fundamental block of central processing unit (CPU) of a computer, and even the simplest microprocessors contain one for purposes such as maintaining timers. Here, the ALU can perform 24 operations, having 8-bit op-code. Only the last 5 bits are changed, corresponding to the bits assigned to each operation.

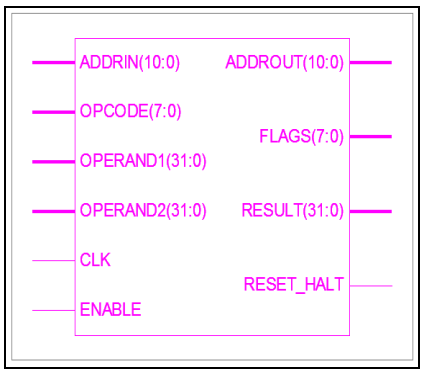


Figure 9: Block diagram of ALU

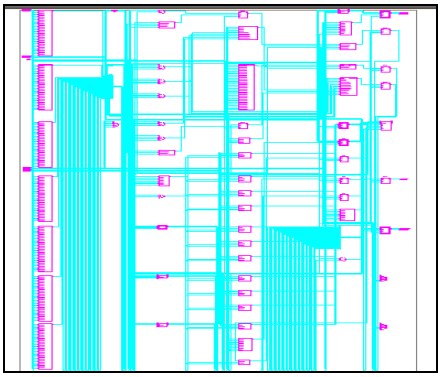


Figure 10: RTL Schematic of ALU

2.5. Central processing Unit (CPU)

It is the hardware within a computer that carries out instructions of a computer program by performing the basic arithmetic, logical, and input/output operations of the system. Here different devices like ROM memory, RAM memory, fetch unit and ALU combine to form an integrated CPU. The figure below shows a high level diagram of CPU.

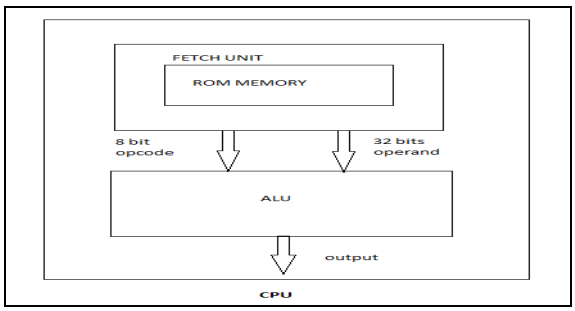


Figure 11: Central Processing Unit (CPU)

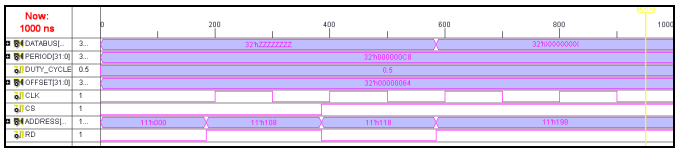
OP-CODES	OPERATION
00000000	RESET
00000001	ANDING
00000010	ORING
00000011	XORING
00000100	NOT(OPERAND1)
00000101	NOT (OPERAND2)
00000110	SHIFT OPERAND1 RIGHT ARITHMETICALLY OPERAND2 TIMES
00000111	SHIFT OPERAND1 RIGHT LOGICALLY OPERAND2 TIMES
00001000	SHIFT OPERAND1 LEFT ARITHMETICALLY OPERAND2 TIMES
00001001	SHIFT OPERAND1 LEFT LOGICALLY OPERAND2 TIMES
00001010	ADDITION
00001011	SUBSTRACTION
00001100	STORE OPERAND1 INTO RAM ADDRESS GIVEN BY ADDRIN
00001101	LOAD RAM TO OUTPUT, RAM ADDRESS IS GIVEN BY ADDRIN
00001110	JUMP TO A PARTICULAR ADDRESS, CALCULATED BY ADDING OPERAND1 AND OPERAND2
00001111	STORE OPERAND1 TO OUTPUT
00010000	NO OPERATION
00010001	INCREMENT OPERAND 1
00010010	DECREMENT OPERAND 1
00010011	2'S COMPLEMENT
00010100	SET CARRY
00010101	RESET CARRY
00010110	COMPLEMENT CARRY
00010111	ROTATE OPERAND1 RIGHT THROUGH CARRY
00011000	ROTATE OPERAND1 LEFT THROUGH CARRY
00011001	ROTATE OPERAND1 RIGHT WITHOUT CARRY
00011010	ROTATE OPERAND1 LEFT WITHOUT CARRY
00011011	COMPARE THE TWO OPERANDS, AND REFLECT RESULT IN THE FLAG REGISTER
00011100	RESET THE PROCESSOR
00011101	HALT THE PROCESSOR

Figure 12

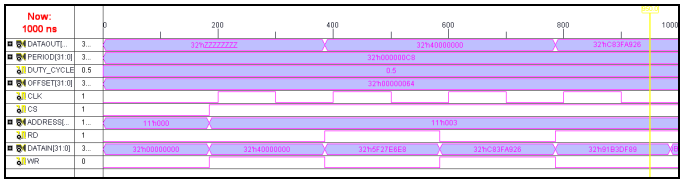
3. Software Tool

Xilinx ISE is a software tool produced by Xilinx for synthesis and analysis of HDL design, enabling the developer to synthesis their designs, perform timing analysis, examine RTL diagrams, Simulate a design's reaction to different stimuli, and configure the target device with the programmer.

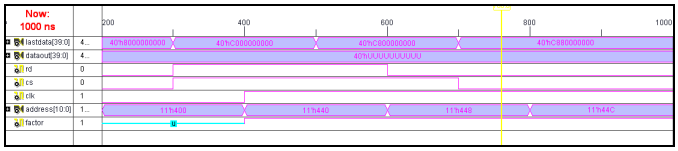
4. Simulation



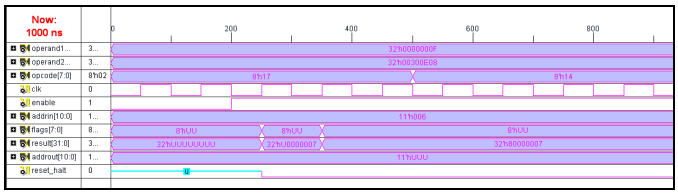
Output Waveform of ROM



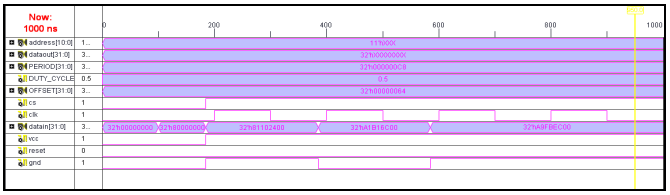
Output Waveform of RAM



Output Waveform of Instruction Fetch Unit



Output Waveform of ALU



Output Waveform of CPU

5. References

1. VHDL Implementation of a MIPS RISC Processor by Anjana R. & Krunal Gandhi, Ph.D Scholar (ECE) , Lecturer (ECE) Laxmi Institute of Technology, Gujarat, India, August, 2012.
2. FPGA Implementation of an 8-bit Simple Processor, E. Ayeh, K. Agbedanu, Y. Morita, O. Adamo, and P. Guturu, Department of Electrical Engineering, University of North Texas, Denton, TX. 76207, USA, July, 2008.
3. Fundamentals of Digital Logic with VHDL design by Stephen Brown and Zvonko Vranesic.
4. VHDL, A design oriented approach by S.S. Limaye.
5. A FPGA Implementation of a MIPS RISC Processor for Computer Architecture education, by Victor P. Rubio, B.S., December, 2006