Delineation16 Bit RISC Processor using Verilog HDL

Dr. Hemlata Sinha, Anmol Chaura, Kavita Singh Thakur, Aditi Chaudhary, Harshita Bhagat
Electronics and Telecommunication, Shri Shankaracharya Instititute of Professional Management and Technology,Raipur, India

Abstract- The Reduced Instruction Set Computer or RISC is a processor delineation principle which implements a smaller and elementary set of instructions. A RISC processor executes approximately one instruction per clock cycle. RISC architectures are streamlined versions of traditional complex instruction set computers. In the proposed paper the behavioral delineation and functional characteristics of 16-bit RISC processor is presented, which utilizes minimum functional units without compromising in performance. The delineation is based on Harvard architecture having separate data memory and instruction memory. The control unit and data path modules are coded in verilog. These individual modules are designed and tested at each level of implementation and finally integrated in a top level module by appropriate mapping. The design entry and synthesis is done using Xilinx ISE 13.2 tool and simulation results are verified using ISim 13.2

Keywords- RISC,16-bit RISC,Processor, ALU,Verilog.

I. INTRODUCTION

The RISC architecture was developed with an eye to reducing complexity by using a simpler instruction set on processors that clock fewer cycles per second than the CISC architecture, hence making it faster.

RISC architecture is the groundwork for most computer terminals and UNIX based servers in use today, and is globally recognized as the computing architecture of the future .The RISC concept streamlines the instructions given to then active computers, making them much faster and more powerful. Based on pioneering work at IBM's Thomas J Watson research center starting 1975, a first prototype was completed in 1980, providing a lower cost method of conducting high performance calculations necessary for engineers and scientists. [1]

From the analysis of millions of instructions that were executed, we observed that for 90% of the time only about 10 instructions from the instruction repertoire were actually used. Then the obvious question was asked: "why not favor implementation of those selected instructions so that they execute in a short cycle, and emulate the reset of instructions". The following reasoning was used: "If the presence of a more complex set adds just one logic level to a 10 level basic machine cycle, the CPU has been slowed down by 10%. The frequency and performance improvement of the complex functions must first overcome this 10% degradation, and then justify the additional cost". [2]

Therefore RISC architecture starts with a small set of most frequently used instructions which enable fast execution of instructions in one cycle and thus upgrading the speed of the processor.[3][4]

II. ARCHITECTURE

The purpose of the project is the delineation of 16 bit RISC processor which has Harvard type architecture. The Verilog code consists of different modules which represents different parts of the CPU; Instruction Memory, Register Files, Data Memory, ALU unit, ALU control unit, Data path, and Control unit.

A program counter (PC) drives the instruction memory. It consists of two adders; one of the adders is used to add four to the program counter for the execution of instruction, as the instruction is of 4 bytes and the other adder for the computed branches. A multiplexer helps in loading the program counter for the next instruction or for a jump target. The processing of data is carried out using an Arithmetic logic unit (ALU) which performs various operations on the specified data. The destination of the processed data can be main memory or any one of the registers. The register file consists of two output ports to reduce the execution time of the instructions. Analogous to most CPUs, the multiplexers decide the entire control of the data path, that is, what gets sent where. [5][6]

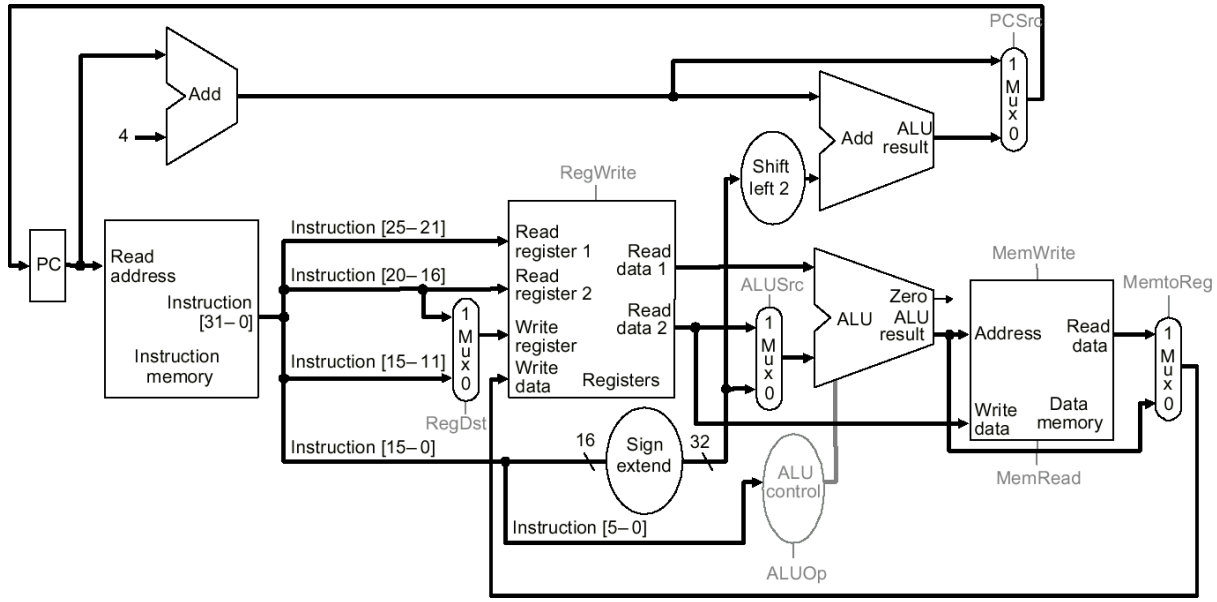

Fig.2:

A. Instruction Memory

The instruction memory consists of the instructions that are to be executed. It is the memory which is referenced during an instruction fetch. It is delineated as read only at user level. In the RISC processor the instructions are fetched from this memory at the maximum speed of memory.

B. Registers

These are temporary memory cells which are used to store data during the execution of the program and are attainable by users through instructions. All the operands that are to be operated on are stored in the register files. The result of this operation is also stored in register files. Once the operands are stored in registers there is no need for repeated loads and stores from or to memory. [7]

C. Data Memory

This memory consists of data that is to be operated on. Data memory is modified using program and hence is accessible by the user.

D. ALU

The ALU is delineated to execute 16 bit arithmetic operations such as ADD, SUB and logical operations such as AND, OR, INVERT, LSR (Logical shift right), LSL (Logical shift left). Two branching instructions BEQ (Branch on equal), BENQ (Branch on not equal) and a JUMP instruction. [8]

E. ALU Control

ALU Control				
ALUOp	Opcode	ALUcnt	ALU Operation	Instruction
10	xxxx	000	ADD	LW,SW
01	xxxx	001	SUB	BEQ,BNE
00	0010	000	ADD	D-type: ADD
00	0011	001	SUB	D-type: SUB
00	0100	010	INVERT	D-type: INVERT
00	0101	011	LSL	D-type: LSL
00	0110	100	LSR	D-type: LSR
00	0111	101	AND	D-type: AND
00	1000	110	OR	D-type: OR
00	1001	111	SLT	D-type: SLT

Table 2.4.1

F. Control Unit

This unit controls the CPU and directs its operations. It applies the required control signals to the ALU, register and ports. It commands the program counter (PC) to fetch the next instruction.

G. Control Signals

Control signals									
Instruction	Register Dst	ALUSrc	Memory to Register	Register Write	Memory Read	Memory Write	Branch	ALUOp	Jump
Data-processing	1	0	0	1	0	0	0	00	0
Load Word	0	1	1	1	1	0	0	10	0
Store Word	0	1	0	0	0	1	0	10	0
BEQ,BNE (Branch)	0	0	0	0	0	0	1	01	0
Jump	0	0	0	0	0	0	0	00	1

Table 2.5.1

III. INSTRUCTION FORMAT

An instruction consists of an opcode; operational code which decides which operation is to be performed and the source and destination of the operands.[7]

A. Memory access: Load

B. Memory access: Store

Opcode (4 bits)	Source Register(Rs) (3 bits)	Destination Register(Rd)(3 bits)	Offset (6 bits)

Opcode (4 bits)	Register1 (3 bits)	Register2 (3 bits)	Offset (6 bits)

C. Data Processing

Opcode (4 bits)	Register1 (3 bits)	Register2 (3 bits)	Destination Register(Rd)(3 bits)	Useless

D. Branch

Opcode (4 bits)	Register1 (3 bits)	Register2 (3 bits)	Offset (6 bits)

E. Jump

Opcode (4 bits)	Offset (12 bits)

IV. SIMULATION RESULT

Simulation refers to testing of delineation, its purpose and execution. A software simulator is a computer program which is used to verify functional correctness of a digital delineation that is delineated using a HDL (Hardware Descriptive Language) like Verilog. Simulation removes the time taking need for continual physical prototyping. [9][10]

A. Control Signals

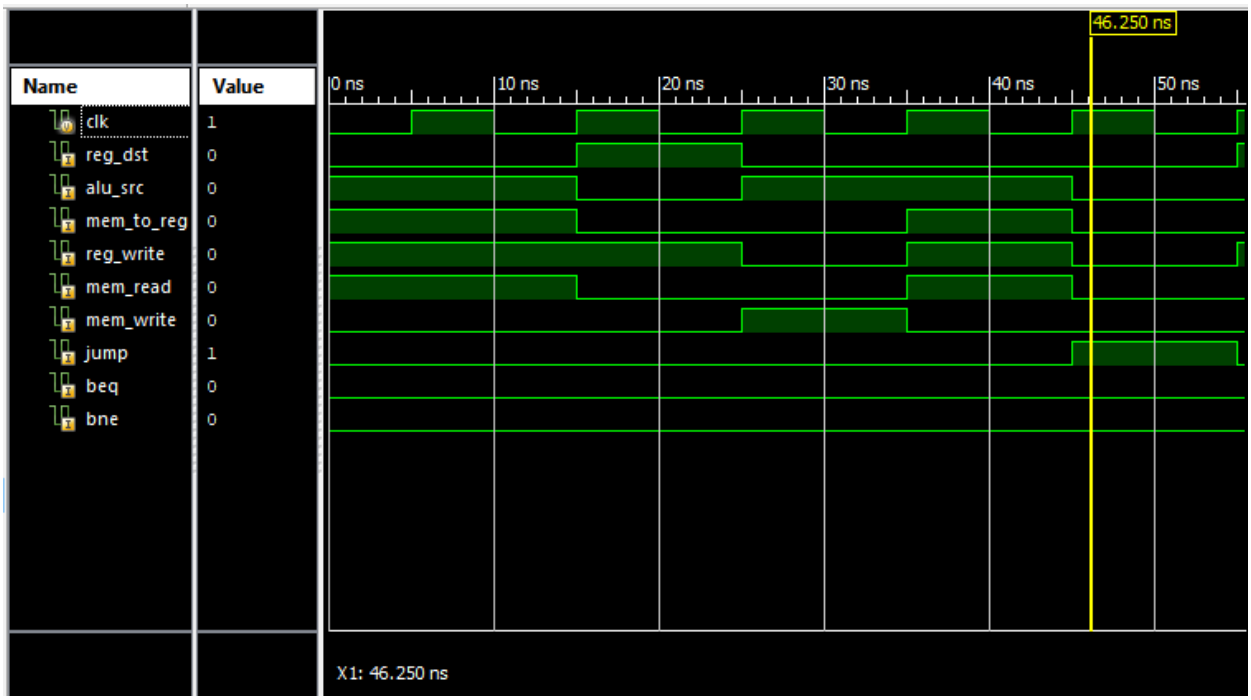

Fig.3:

Working RISC

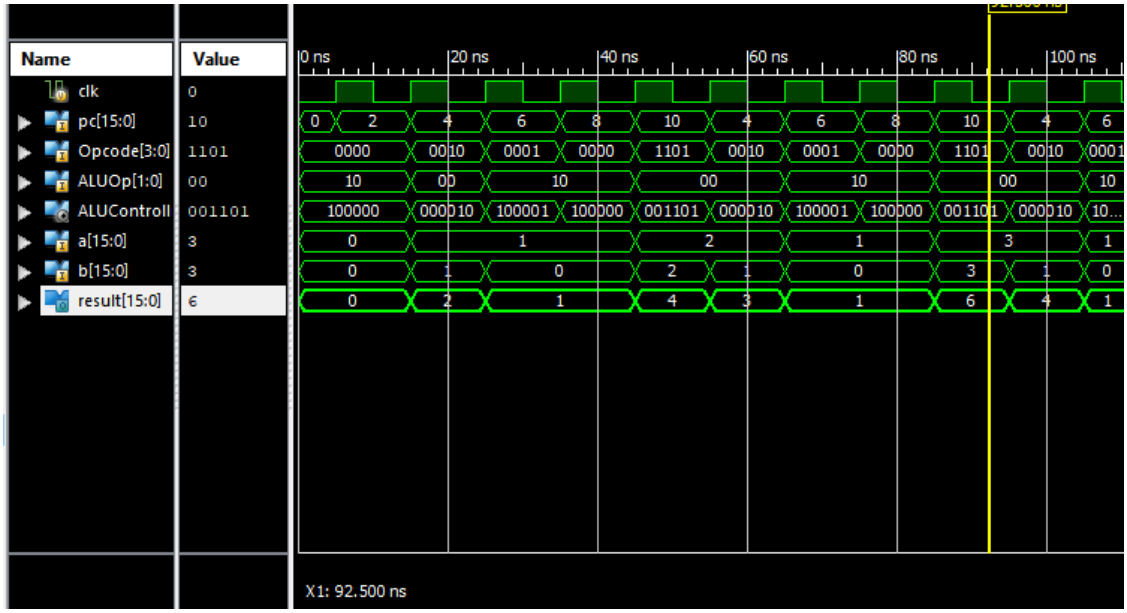

Fig.4:

V. CONCLUSION

A 16 bit RISC processor based on Harvard architecture has been designed.The processor is implemented using various functional blocks which are controlled sequentially by a control unit. It consists of three basic elements - Control unit, General Purpose Registers, Arithmetic and Logical unit (ALU). Xilinx ISE design suite 13.2 has been implemented for the design entry and synthesis. This delineation can be enhanced in multiple ways. A more refined delineation can be attained by adding more attributes to the present. Number of operations carried out by the processor can be increased. To

enhance the performance of the proposed design, pipelining can be added.

VI. REFERENCES

[1]. www.ibm.com/ibm/history/documents/pdf/rs6000.pdf "A Brief History of RICS, the IBM RS/6000 and the IBM eServer pSeries"

[2]. G. Radin, "The 801 Minicomputer", IBM T.J.Watson Research Center, Report RC 9125, November 11, 1981, also in SIGARCH Computer Architecture News 10, No.2, p.39-47, March 1982

[3]. www.alanclements.org/rischistory.html "Rise of the RISC Processor ".

[4]. Supraj Gaonkar, Anitha M "Delineationof 16 bit RISC processor" International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 7, July - 2013

[5]. www.hackaday.com/2017/05/05/learn-by-fixing-another-verilog-cpu/ Al Williams "Learn by fixing"

[6]. Prof. Vojin G. Oklobdzija, "Reduced Instruction Set Computers"

[7]. Gaonkar R. Microprocessor architecture, programming and applications with the 8085. PENRAM International Publishing Pvt Ltd; 2010.

[8]. Vishnuvardhan Rao, K & Angeline, Anita & Bhaaskaran, V s kanchana. (2015). Delineationof a 16 bit RISC processor. Indian Journal of Science and Technology. 8. 10.17485/ijst/2015/v8i20/78320.

[9]. www.testbench.in "Verilog for verification"

[10]. www.web.eecs.umich.edu "Logic Simulation (University of Michigan)".