Design and Performance Analysis of 8-bit RISC Processor using Xilinx Tool

R. Uma
Research Scholar, Department of Computer Science, Pondicherry University, Pondicherry.

ABSTRACT

RISC or Reduced Instruction Set Computer is a design philosophy that has become a mainstream in Scientific and engineering applications. Increasing performance and gate capacity of recent FPGA devices permits complex logic systems to be implemented on a single programmable device. So the main objective of this paper is to design and implement an 8-bit Reduced Instruction Set (RISC) processor using XILINX Spartan 3E tool. The enhanced feature of Spartan-3E deliberately reduces the cost per logic cell designed. The most important feature of the RISC processor is that this processor is very simple and support load/store architecture. The important components of this processor include the Arithmetic Logic Unit, Shifter, Rotator and Control unit. The module functionality and performance issues like area, power dissipation and propagation delay are analyzed at 90 nm process technology using SPARTAN 3E XCS500E XILINX tool

***Keywords* –** RISC, Load/store architecture, Pipeline, Uniform bit-stream.

INTRODUCTION

Nowadays, computers are indispensable tools for most of everyday activities. With the rapid development of the silicon technology and the decreasing cost of the integrated circuit, RISC processor is increasing widely used in every field. RISC is an extension of the architecture principles of the Reduced Instruction Set Computer (RISC). The simple design provides exceptional performance and is ideal for use in a broad family of cost-effective, compatible systems. Some typical applications include: commercial data processing, computation-intensive scientific and engineering applications, and real-time control.

The main features of RISC processor are the instruction set can be hardwired to speed instruction execution. No microcode is needed for single cycle execution. All instructions are one word (fixed bit) in length. This simplifies the instruction fetch mechanism since the location of instruction boundaries is not a function of the instruction type. The processor has small number of addressing modes. Only load and store instructions access memory. There are no computational instructions that access memory; load/store instructions operate between memory and a register. Control hardware is simplified and the machine cycle time is minimized. The instructions were designed to be easily divisible into parts. This and the fixed size of the instructions allow the instructions to be easily piped. RISC provides a flexible and expandable architecture that maximizes performance from any given semiconductor technology. RISC includes extensions to RISC concepts that help achieve given levels of performance at significantly lower cost than other systems. With FPGA based core design any set of task can be configured. It is also capable of employing any control algorithm. It sustains any system level change without costly hardware replacement and thus the design process is very fast and cost effective.

In the present work, the design of an 8-bit data width Reduced Instruction Set Computer (RISC) processor is presented; it was developed with simplicity and implementation efficiency in mind. It has a complete instruction set, program and data memories, general purpose registers and a simple Arithmetical Logical Unit (ALU) for basic operations. In this design, most instructions are of uniform length and similar structure, arithmetic operations are restricted to CPU registers and only separate *load* and *store* instructions access memory. The Instruction cycle consists of three stages namely fetch, decode and execute. After every instruction fetch, Control Unit generate signals for the selected Instruction. The architecture supports 16 instructions to support Arithmetic, Logical, Shifting and Rotational operations.

The remainder of this paper is organized as follows. Section II explains the architecture detail of 8-bit RISC processor. Section III presents the design module of ALU, Control unit and general purpose registers. Section IV presents the simulation results implemented in advanced 90nm process technology. Section V discusses summary with the implementation of the RISC design topology. The final section presents the conclusion.

I. ARCHITECTURE OF 8-BIT RISC PROCESSOR

The architecture of an 8-bit RISC processor is shown in Figure (1). This architecture consists of arithmetic logic unit, control unit, shifter and rotator. The processor is designed with load/store (Von Neumann) architecture .One shared memory for instructions (program) and data with one data bus and one address bus between processor and memory. Instruction and data are fetched in sequential order so that the latency incurred between the machine cycles can be reduced. Three stages of pipelining have been incorporated in the design which increases the speed of operation. The pipelining stages are fetch, decode and execute. In fetch, the instruction and the necessary data are drawn from the memory. Whereas in decode, the instruction and data that are drawn from the memory are separated activating the components and the data path so as to execute And finally in execution, the instruction is performed, the data is manipulated and the result is stored.

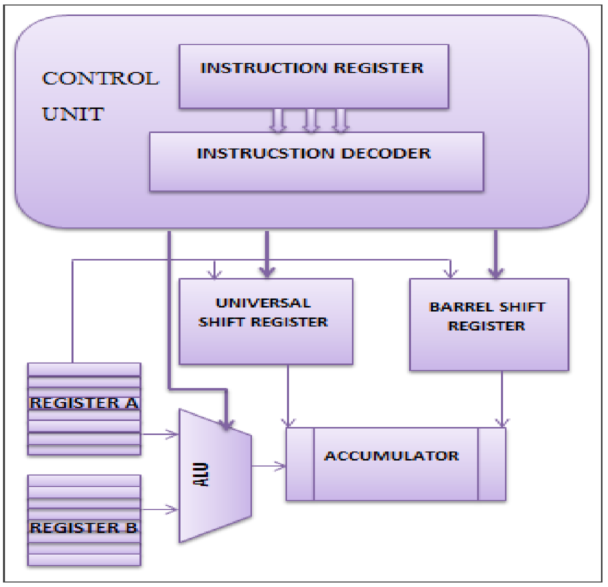


Figure 1. Architecture of 8-bit RISC processor

The control unit reads the opcode and instruction bits and then creates control signals as outputs that triggers the respective components and data path to perform the desired task. The control unit has two instruction decoders that decodes the instruction bits and the decoded output of the control unit is fed as control signal either into Arithmetic logic unit (ALU) or Universal shifter or Barrel shift rotator. The operands are received from register A and register B by the ALU. Depending on the control signal from the control unit the ALU performs either arithmetic or logic operations. After the execution of the instruction, the result is stored in the accumulator register. Input is taken from source register A and is either loaded or shifted in right or left direction based on the control lines activated by the control unit. The shifted data is saved in the destination register which is nothing but the accumulator register. Input data is given from source register A and rotated N number of times based on the opcode fed from the control unit. The rotated data is stored in the accumulator register.

II. MODULES DESIGN OF 8-BIT RISC PROCESSOR

This section presents the design of different modules line control unit, ALU, Universal shift register, barrel shifter register and general purpose registers.

A. Control unit

Control unit is designed using finite state machine as depicted in Figure (2).The state machine is designed to perform the logical, arithmetic, shifting and rotate operations. For example, if the instruction bit is 0010 the operation performed is NOR operation and the next consecutive opcode is 1001 then it remains in the same state or else it will have a transition to the next state depending on the opcode it receives. The overall operation is shown in Table (1).The control unit consists of two decoders in which the first decoder performs logical and arithmetic opcode generation and the second decoder performs shifting and rotating opcode generation. The top block of the 5 control unit is shown in Figure(3). The circuit is simulated in xilinx environment and its simulation results are shown in Figure (4) and Table (1).

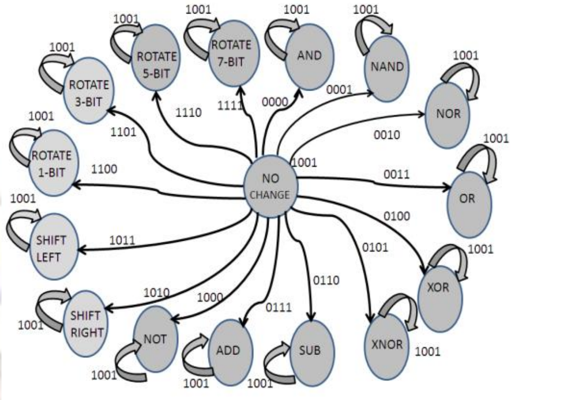


Figure 2. State Diagram of Controller

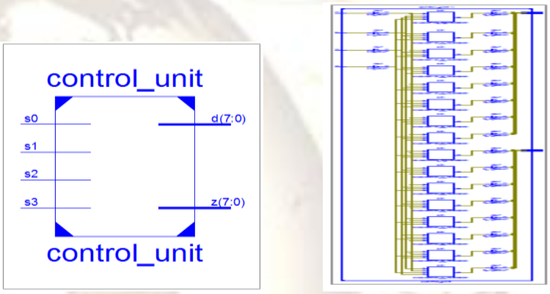


Figure 3. Top block of Controller

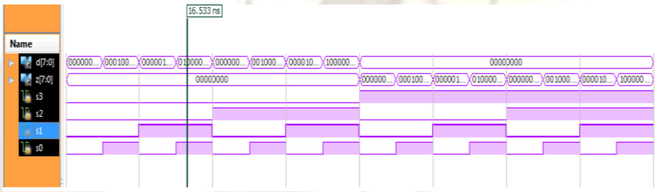


Figure 4. Timing diagram of control unit

Table1. Operation of control unit

B. Arithmetic Logic Unit

SELECT LINES	OUTPUT OF DECOCDER								FUNCTION PERFORMED
S3 S2 S1 S0	d 0	d 1	d 2	d 3	d 4	d 5	d 6	d 7	OPERATION
0 0 0 0	1	0	0	0	0	0	0	0	AND
0 0 0 1	0	1	0	0	0	0	0	0	NAND
0 0 1 0	0	0	1	0	0	0	0	0	NOR
0 1 1 0	0	0	0	1	0	0	0	0	OR
0 1 0 0	0	0	0	0	1	0	0	0	XOR
0 1 0 1	0	0	0	0	0	1	0	0	XNOR
0 1 1 0	0	0	0	0	0	0	1	0	SUB
0 1 1 1	0	0	0	0	0	0	0	1	ADD
S3 S2 S1 S0	Z 0	Z 1	Z 2	Z 3	Z 4	Z 5	Z 6	Z 7	
1 0 0 0	1	0	0	0	0	0	0	0	NOT
1 0 0 1	0	1	0	0	0	0	0	0	NO CHANGE
1 0 1 0	0	0	1	0	0	0	0	0	SHIFT-RIGHT
0 0 1 1	0	0	0	1	0	0	0	0	SHIFT-LEFT
1 1 0 0	0	0	0	0	1	0	0	0	ROTATE 1-BIT
1 1 0 1	0	0	0	0	0	1	0	0	ROTATE 3-BIT
1 1 1 0	0	0	0	0	0	0	1	0	ROTATE 5-BIT
1 1 1 1	0	0	0	0	0	0	0	1	ROTATE 7-BIT

The ALU design comprises of 2 units.One unit is meant for logic operation containing eight bit logic gates such as AND,NAND,OR,NOR,XOR,XNOR and the other unit is meant for arithmetic operations such as ADD and SUBTRACT. In arithmetic unit, based on the control input Cin the Add and Subtract operations take place. For Cin low, addition of the given input data is performed whereas for Cin high subtraction performed. The entire design of the ALU in RTL view is represented in Figure (5) and the internal submodule of arithmetic unit is shown in Figure (6) and the timing waveform for arithmetic unit simulated using Xilinx tool is shown in Figures (7)(7.1).

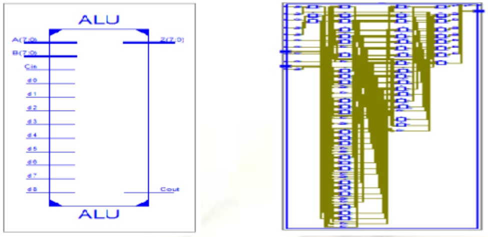


Figure 5. Top block of 8 bit arithmetic and logic unit

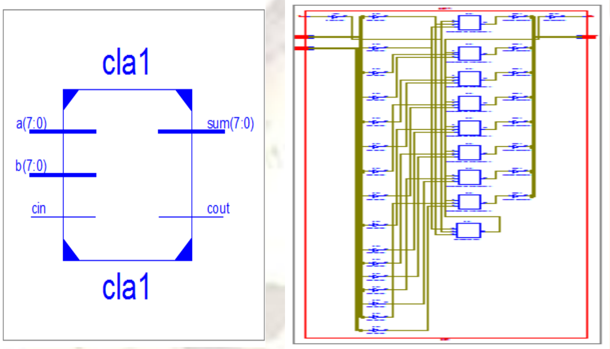


Figure 6. Carry look ahead adder/subtractor

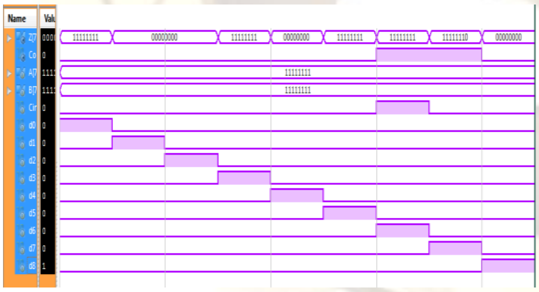


Figure 7. Simulated timing diagram of carry look ahead

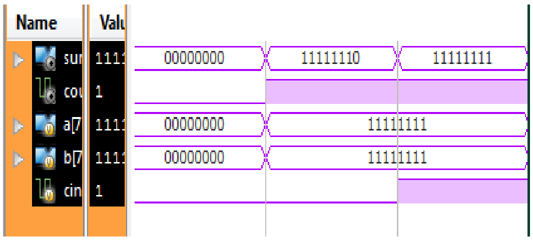


Figure 7.1. Simulated timing diagram of carry look ahead

C. *Universal Shift Register*

The Universal shift register is designed with features such as loading, right shift, left shift and no change. The design has eight 4x1 multiplexers and nine basic gates and is shown in the Figure (8). Loading the input is attained by applying eight bits of data as input with control lines S0 and S1 taken as low. Right shifting takes place for the given eight bit input data with control lines S0 high and S1 low and similarly the left shift takes place for the eight bit data as input provided the control lines S0 should be low and S1 should be high. The output remains low for the control lines S0 and S1 taken high. The entire operation is represented in Table (2) and Figure (9) shows the simulated result of the universal shift register.

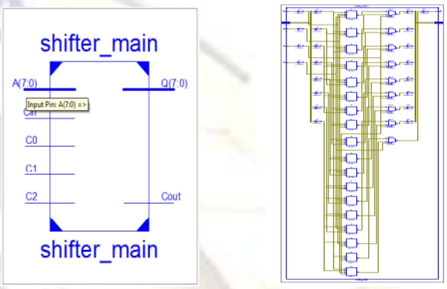


Figure 8. Top block of universal shift register

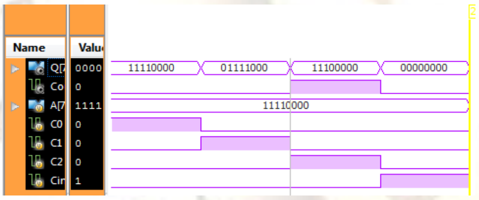


Figure 9. Simulated timing diagram of universal shift register

Table 2.Operation of the universal shift register

SELECT LINES			OPERATION PERFORMED		
INPUT			OUTPUT		
A A A A A A A A 7 6 5 4 3 2 1 0	Ci n	S1S0	Q Q Q Q Q Q Q Q 7 6 5 4 3 2 1 0	Co ut	
1 1 1 1 0 0 0 0	0	0 0	1 1 1 1 0 0 0 0	0	Load
1 1 1 1 0 0 0 0	1	0 0	1 1 1 1 0 0 0 0	0	Load
1 1 1 1 0 0 0 0	0	0 1	0 1 1 1 1 0 0 0	1	Right shift
1 1 1 1 0 0 0 0	1	0 1	0 1 1 1 1 1 1 1	1	Right shift
1 1 1 1 0 0 0 0	0	1 0	0 1 1 1 1 0 0 0	0	Left shift
1 1 1 1 0 0 0 0	1	1 0	1 1 1 1 1 0 0 0	0	Left shift
1 1 1 1 0 0 0 0	0	1 1	0 0 0 0 0 0 0 0	0	No change
1 1 1 1 0 0 0 0	1	1 1	0 0 0 0 0 0 0 0	0	No change

D. *Barrel Shift Rotator*

The design consists of a total of eight 8x1 multiplexers. The output of one multiplexer is connected as input to the next multiplexer in such a way that the input data gets shifted in each multiplexer thus performing the rotation operation. Depending on the select lines the number of rotation varies. With select lines low there is no output.If select line S0 is high 1-bit rotation takes place,if S1 is high 2-bit roation takes place and the roation continues untill all select lines are high. The rotation of the input data for different select lines is shown in Table (3) and Figure (11).The Figure (10) shows the top block of the barrel shift rotator.

Table 3. Operations of Barrel rotator

INPUT OF ROTATOR		OUTPUT OF ROTATOR	FUNCTION PERFORMED
A A A A A A A A 7 6 5 4 3 2 1 0	S2 S1 S0	Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0	
1 1 1 1 0 0 0 0	0 0 0	0 0 0 0 0 0 0 0	Zero
1 1 1 1 0 0 0 0	0 0 1	1 1 1 0 0 0 0 1	1 Bit Rotate
1 1 1 1 0 0 0 0	0 1 0	1 1 0 0 0 0 1 1	2 Bit Rotate
1 1 1 1 0 0 0 0	0 1 1	1 0 0 0 0 1 1 1	3 Bit Rotate
1 1 1 1 0 0 0 0	1 0 0	0 0 0 0 1 1 1 1	4 Bit Rotate
1 1 1 1 0 0 0 0	1 0 1	0 0 0 1 1 1 1 0	5 Bit Rotate
1 1 1 1 0 0 0 0	1 1 0	0 0 1 1 1 1 0 0	6 Bit Rotate
1 1 1 1 0 0 0 0	1 1 1	0 1 1 1 1 0 0 0	7 Bit Rotate

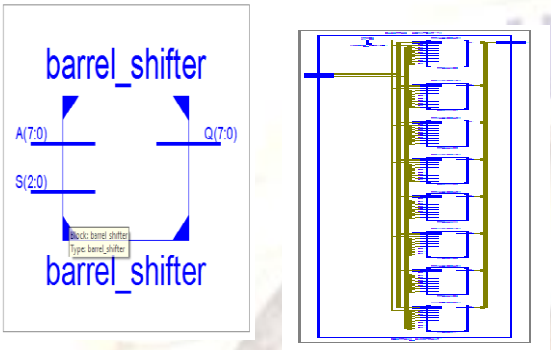


Figure 10.Top Block Of Barrel Shift Rotator

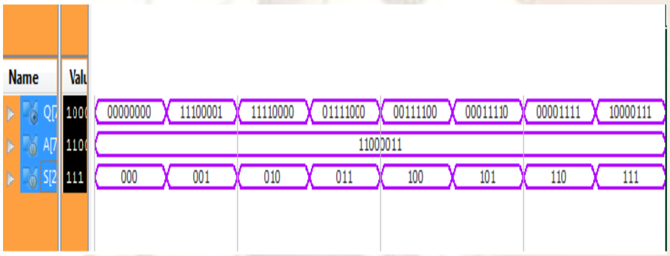


Figure 11. Timing Diagram of Barrel shift rotator

E. *General Purpose Register*

The eight bit input data is stored in this register. This register acts as a source register. It consists of eight D – flip flops and eight AND gates. The gate level view of the register is given by Figure (12). Initially the RESET is set high to clear the register. Taking RESET as low and CLOCK as low or high and READ as high the data is stored in the register.The conditions for which the data is stored in the register is clearly shown in Table (4) and simulated timing waveform in Figure (13).

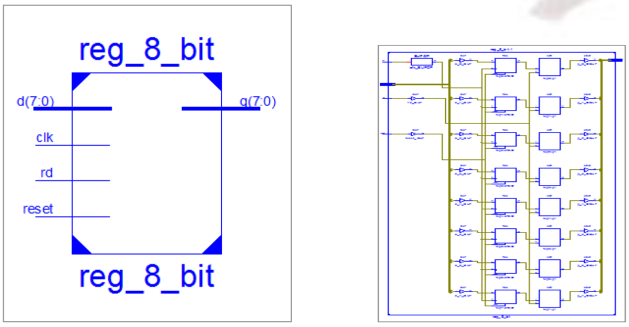


Figure 12.Top block of general purpose register

Table 4.Operations of general purpose register

INPUT				OUTPUT
CLK	RESET	RD	Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0	Q7 Q6 Q5 Q4 Q3 Q2 Q1 Q0
0	1	0	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0
1	0	1	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 1
0	0	1	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 1

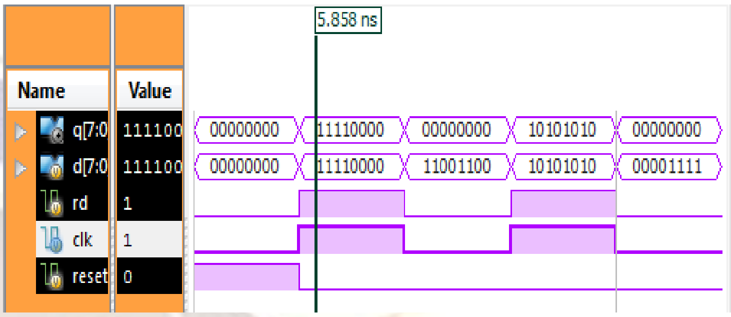


Figure 13. Timing diagram of general purpose register

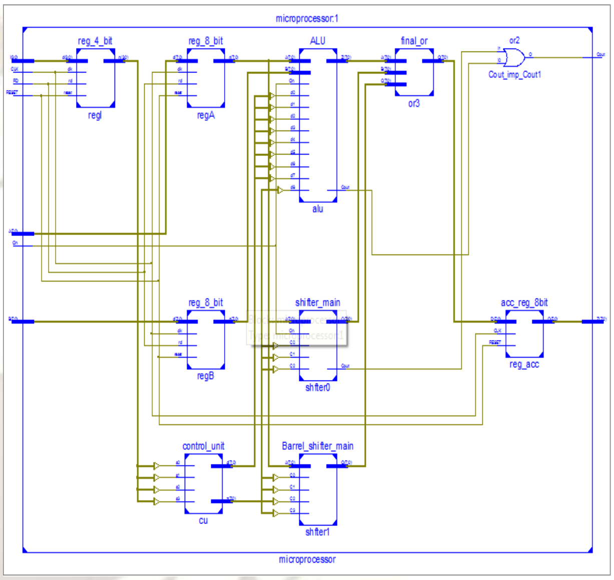


Figure 14 Top Block of 8-bit Processor

F. *The Instruction set format rule*

The instruction set of the RISC processor has been designed following several rules:

- All instructions are executed in just one clock cycle. Doing so, processor is simpler, smaller, faster and easier to understand.
- The instruction code is received at the beginning of each cycle, all operations are executed during the clock period, and results are stored at the end of it.
- ALU operations take two operands from registers and store the result in one of them.
- External read and write operations are synchronous.

III. RESULT

The performance of the RISC processor has been evaluated in this research work by using advanced XILINX SPARTAN 3E XCS500E technology. The design meets the need of high performance logic solution for high volume, very low cost, consumer-oriented applications. It employs a multi-voltage, multi-standard SelectIO™ interface pins with a voltage range of 3.3V,2.5V,1.8V,1.5V and 1.2V at a 622+ Mb/s data transfer rate. It is operated at a maximum frequency range of 5MHz to 300MHz.

The overall design of 8-bit processor is shown in Figure 14. The processor has two eight-bit input signals A7 - A0 and B7 - B0 taken externally and loaded into registers A and B respectively. Memory Interface Signal is a signal READ (RD).This signal indicates that the selected memory location is to be read and data is to be put on the data bus.

The synchronization of various operation are done using CLK signal. The processor is designed with two control signals RD and RESET. If reset is high then the processor will not perform any operation it will stay in idle state. If the reset is low and RD is high then the data is loaded into the data bus and its corresponding values are loaded into the general purpose registers A and B. Depending on the opcode provided by the control unit the particular operation is performed as stated in Table(1).

This 8- bit RISC processor works on one clock cycles. clk is the external clock which is always equal to one which triggers the inputs and gives us the desired output. RD triggers the state of the registers through which data is passed into the internal registers A and B. I0 to I3 specifies the opcode to enable the operation. For example if the opcode value is “0111” then the operation performed will be addition.

IV. SUMMARY

This section presents the performance of the processor in terms of its power consumption and delay that are obtained using the Xilinx tool. Table 5 presents the maximum power dissipation, area occupied and time taken by each module to operate the processor.

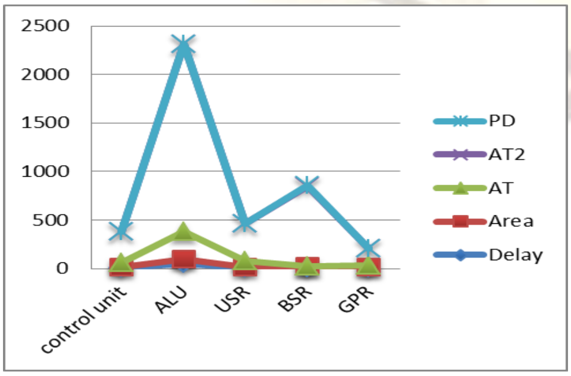
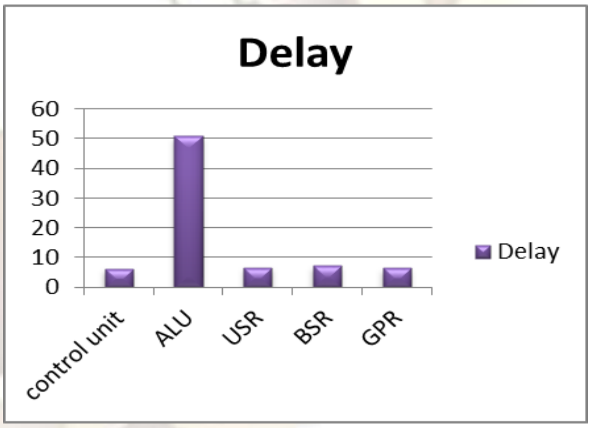


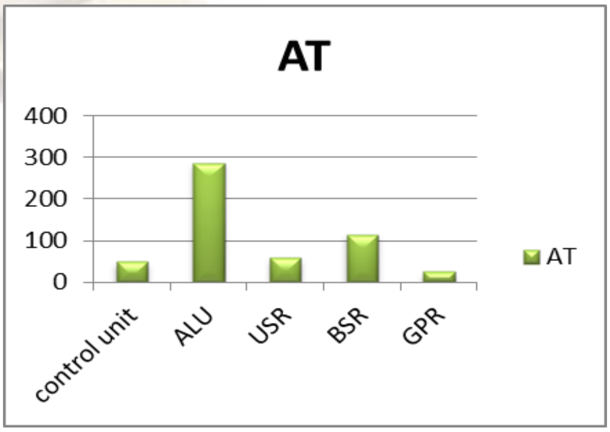
Figure 15. Variation of power disipation, delay and area of the RISC processor

Tabel 5 Delay, Power consumption and area calculation

TOPOLOGY	SUB BLOCKS	Delay(ns)	Slices Utilized (Area)	AT	AT^2	Power dissipation 10^{-9} W	PD
Control Unit	*Decoder*	6.275	8	50.2	315.0	0.081	0.508
ALU	*AND*	5.753	4	23.012	132.3	0.081	0.463
	NAND	5.753	4	23.012	132.3	0.081	0.463
	NOR	5.753	4	23.012	132.3	0.081	0.463
	XOR	5.753	4	23.012	132.3	0.081	0.463
	CLA	7.732	5	38.66	298.9	0.081	0.626
	Inverter	6.034	4	24.136	144.6	0.081	0.488
	AND	6.546	4	26.184	171.4	0.081	0.530
	OR	7.508	14	105.11	789.1	0.081	0.608
Universal Shift Reg	*MUX*	6.582	9	59.238	389.9	0.081	0.533
Barrel Shift Reg	*MUX*	7.198	16	115.16	828.9	0.081	0.583
GPR	*D-FF*	6.546	4	26.184	171.4	0.081	0.530
TOTAL		77.43	80	536.92	3638		6.258

It is observed that the overall delay of the circuit is 77.4 ns. The maximum area is occupied by the ALU unit and barrel shifter. The overall power dissipation of this processor is observed to be 6.258 W. The power dissipation can even be reduced if the circuit is designed with any adiabatic logic. Figure (15) and (16) shows the variation of area, power dissipation and delay of the whole RISC processor and for each sub modules respectively. From the graph it is seen that the maximum power dissipation and chip area is exhibited by the ALU unit.





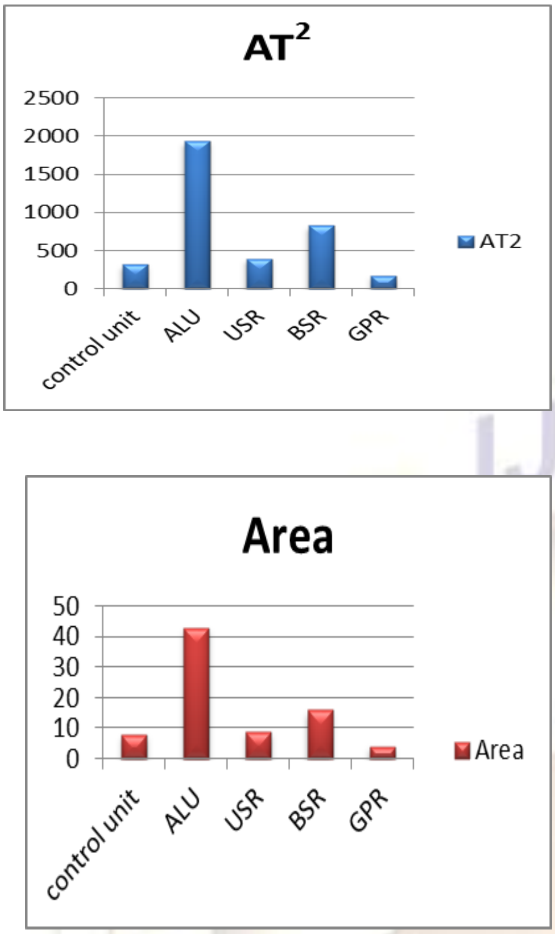


Figure 16.Variation of power dissipation, area and delay for eachs sub module

V. CONCLUSION

An 8-bit RISC processor with 16 instruction set has been designed. Every instruction is executed in one clock cycles with 3-stage pipelining. The design is verified through exhaustive simulations. The processor achieves higher performance, lower area and lower power dissipation. This processor can be used as a systolic core to perform mathematical computations like solving polynomial and differential equations. Apart from this it can be used in portable gaming kits.

REFERENCES

[1] Samiappa Sakthikumaran,S.Salivahanan and V.S.Kaanchana Bhaaskaran , June 2011, "16-Bit RISC Processor Design For Convolution Application",IEEE International Conference on Recent Trends In Information Technology, pp.394-397.

[2] Rohit Sharma, Vivek Kumar Sehgal, Nitin Nitin1, Pranav Bhasker, Ishita Verma , 2009, "Design And Implementation Of 64-Bit RISC Processor Using VHDL",UKSim : 11th International Conference on Computer Modeling And Simulation, pp. 568 – 573.

[3] Rupali S. Balpande and Rashmi S. Keote.2011, "Design of FPGA based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor, International Conference on Communication Systems and Network Technologies pp. 409 – 413

[4] Sivarama P.Dandamudi ,"A Guide To RISC Processor For Programmers And Engineers", Springer.

[5] Tom Wada, "Small Risc Processor (SPR) design specification v1.0", 12th Design Contest In OKINAWA, pp. 1-17

[6] Seung PyoJung, Jingzhe Xu, Donghoon Lee, Ju Sung Park, 2008, " Design And Verification Of 16 Bit RISC Processor" , International SOC Design Conference.

[7] Xiaoping Huang,Xiaoya Fan, Shengbing Zhang , 2008,"Design and Performance Analysis of One 32-bit Dual Issue RISC Processor for Embedded Application".

[8] R. N. Noyce and M. E. Hoff, "A History of Microprocessor Development at Intel," IEEE Micro, vol. no. 1, 1981, pp. 8-21 .

[9]. J.L. Hennessy, "VLSI Processor Architecture," IEEE Trans. Computers, vol. C-33, no. 12, Dec. 1984, pp. 1221-1246.

[10] John L. Hennessy, an d David A. Patterson, "Computer Architecture A Quantitative Approach" , 4th Edition ; 2006.

[11].Vincen t P. Heuring, and Ha rry F. Jordan, "Computer Systems Design and Architecture " , 2n d E dit i on, 2003.

[12].Wa yne Wolf, FPGA -Based System Design , Prentice Hall, 2005.

AUTHOR PROFILE

She is graduated B.E (EEE) from Bharathiyar University Coimbatore in the year 1998, Post graduated in M.E (VLSI Design) from Anna University Chennai in the year 2004. Currently she has been working as Assistant Professor in Electronics and Communication Engineering, Rajiv Gandhi College of Engineering and Technology, Puducherry. She has been teaching VLSI Design, Embedded Systems, Microprocessor and Microcontrollers for PG and UG students. She authored books on VLSI Design. She has published several papers on national conference and symposium. She is the guest faculty for Pondicherry University for M.Tech Electronics. she has been actively guiding PG and UG students in the area of VLSI, Embedded and image processing. She has received the best teacher award for the year 2006 and 2007. Her research interests are Analog VLSI Design, Low power VLSI Design, Testing of VLSI Circuits, Embedded systems and Image processing. She is a member of ISTE.