RESEARCH ARTICLE OPEN ACCESS

DESIGN AND ANALYSIS OF 16-BIT RISC PROCESSOR USING LOW POWER PIPELINE

P. Brundavani[1], Dr. D. Vishnu Vardhan[2], P.Mahesh[3], R.Suresh[4]
1(Asst.professor Department of ECE, AITS, Rajampeta
Email: brundavenky@yahoo.com)
2(Asst.proffessor Department of ECE,JNTUACE,Ananthapuram
Email:vishnushreyan@yahoo.com)
3(Department of ECE, AITS, Rajampeta
Email: maheshraina110@gmail.com)
4(Department of ECE, AITS, Rajampeta
Email :gupthasuresh123@gmail.com)

Abstract:

RISC is a design philosophy to reduce the complexity of instruction set that in turn reduces the amount of space, cycle time, and cost, power consumption taken into account during the implementation of the design. The existing system is an Arithmetic and Logical Unit which performs only Arithmetic operations such as ADD,SUB, MUL, DIV and logical operations such as AND, OR, NAND, NOR, NOT, XOR, XNOR operations and consumes more power .The execution delay is also high. To overcome the above mentioned problems, 16-bit RISC processor is proposed. 16-bit low power RISC processor consists of the blocks mainly ALU, Universal shift register and Barrel Shifter. This system consumes less power and the execution delay is also less. The processor is designed using XILINX ISE Design suit 14.3 & the total power estimation is done by the X Power analyzer.

Key words – ALU, RISC Processor, Low Power

INTRODUCTION

Basically an ALU is a combinational digital electronic circuit that performs arithmetic and bitwise logical operations and produces output of the performed operation. ALU also exchanges additional information with a status register. ALU can be designed to perform complex operations resulting higher complex circuit, large size, cost and power consumption .consequently ALU's are limited to simple operations that can be executed at high speed and the external processor is responsible for performing complex functions by arranging as a sequence of simpler ALU operation. When the design of a controller become more complex in CISC processor and the performance was also not up to the operations, an alternative that was found in RISC processor

FEATURES

- Uniform instruction encoding.
- A homogeneous register set, allowing any register to be used in any context and simplifying compiler design.
- Few data types supported in hard ware.

HISTORY

The concept of ALU is proposed in 1945 by a mathematician john von Neumann in a report on the foundations for a new computer called EDVAC.All serial computers and many early computers had a simple ALU that operated on one data bit at a time one of the earliest computer to have multiple discrete single bit ALU's was the 1948 Whirlwind I, which used 16 such math units to perform operation on 16-bit words first ALU-IC was implementedin1967, {Fairchild introduced}, With 8-bit ALU. In 1970's. ALU's for 4-bit &8-bit operations were appeared. Today modern alu's have wide word widths and hierarchical enhancements

such as shifters, binary multipliers that allow them to perform operations on a single clock cycle.

RISC

A RISC processor CPU design that favors a smaller and simpler set of instructions that all take same amount of execution time.A RISC architecture has a very few instructions, where processor asses data from memory probably not other than load and store operations since memory access across kills speed of the processor. Also the complexity of the controller design has been overcome by having operand and opcode bits are fixed in instruction register. At the end of pipelining is added to improve the speed with basic three stages as fetch, decode and execute.

Depending on the demand of performance, cost, power and programmability, processor can be 8-bit/16bit/32-bit. 8-bit processors have low cost and less power for simple application. Whereas a 32-bit processor have high programmability, higher performance and high power consumption on the other hand 16-bit processor have high performance and power than 8-bit processor and low power consumption than 32-bit processor. Hence 16-bit fully functional single cycle processor is applicable for real tasks and is presented in this paper.

RISC ARCHITECTURE

In RISC architecture, the instruction set of processor is simplified to reduce the execution time. It uses small and highly optimized set of instructions which are generally register to register operations.

The speed of the execution is increased by using smaller number of instructions .This uses pipeline technique for execution of any instruction. The figure-1 shows the architecture of RISC processor, which uses separate instruction and data caches and their access paths also different. There is one instruction per machine cycle in RISC processor.

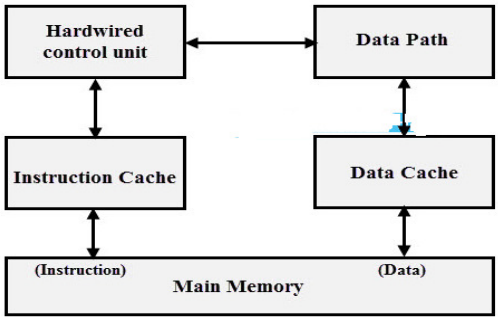

Figure-1: Architecture of RISC processor

The pipelining technique allows the processor to work on different steps of instruction execution like fetch, decode and execute instructions at the same time. Below is figure-2 showing execution of instructions in pipelining technique.
Generally, execution of second instruction is started, only after the completion of the first instruction. But in pipeline technique, each instruction is executed in number of stages simultaneously.
When the first stage of first instruction is completed, next instruction is enters into the first stage. These processes continuous until all the instructions are executed.

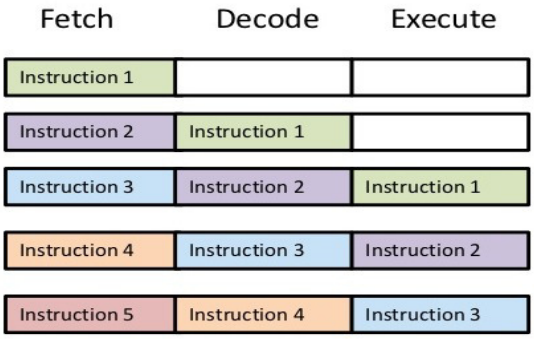

Figure-2: Pipelining Process

PROPOSED 16-BIT RISC PROCESSOR

The architecture of 16 bit RISC processor has been shown in the figure-3. It comprises of Control unit, general purpose register, ALU, Barrel shifter, universal shift register and accumulator. The control unit consists of two registers i.e. instruction register and instruction decoder. Instruction and

data are fetched sequentially in order to reduce the latency in the machine cycle. Pipeline structure has been incorporated that further utilizes three execution cycle fetch, decode and execute. This pipeline structure helps in enhancing the speed of operation. In fetch cycle, instruction and relevant data are inferred from the memory while in decode cycle, instruction and data drawn from the memory are bifurcated to activate Component and data path for execution and in the execution cycle instruction is executed, data is manipulated and result is stored in the accumulator.

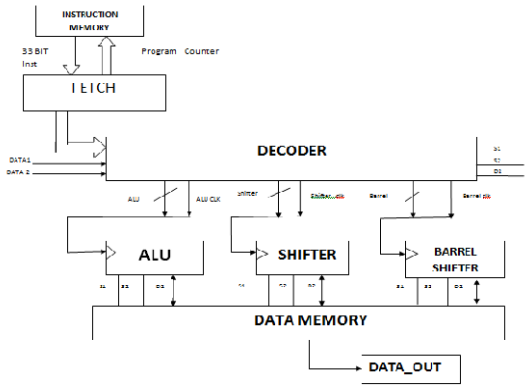

Figure-3: Architecture of 16-Bit RISC processor

The control units accept the Opcode and generate the signals that trigger the components and data path to work accordingly and perform the desired function. The control unit has two instruction decoders. These two decoders decode the instruction bits and direct the signal to either into ALU, universal shift register or barrel shift rotator. The operands are received from register A or register B. Upon receiving the operands from registers and the decoded instruction bits arithmetic and logical unit perform arithmetic and logical functions. Universal shift register and barrel shift rotator receives the input from register A and depending upon the decoded information perform the desired operation of either shifting or rotation and the result is stored in the accumulator register.

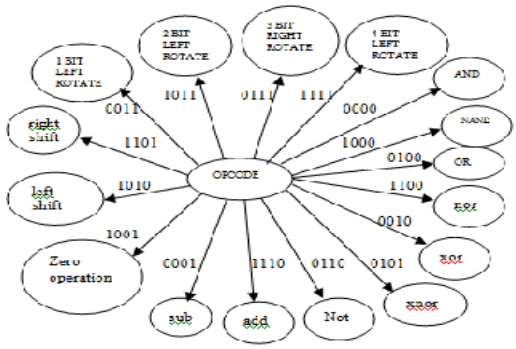

Figure-4: State Diagram

INSTRUCTION DECODER

Instruction decoder is an FSM, which has 4 states. if rst signal is asserted the instruction decoder will enter into initial state. After this for every positive edge of clock the state will change from into to fetch and then to execute and then to load.

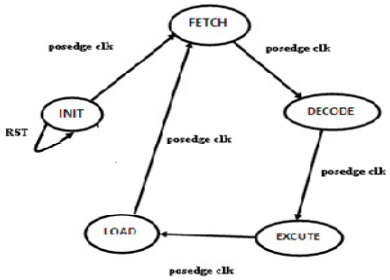

Figure-5: ***State machine for RISC processor***

MEMORY

- Memory of 16 locations depth and 16 bit width
- Receives address from operand1 which is give to the instructions decoder
- If r/w is high it will assign the data to DATA_OUT.
- The memory block will get activated only when CHIP_SE is high.

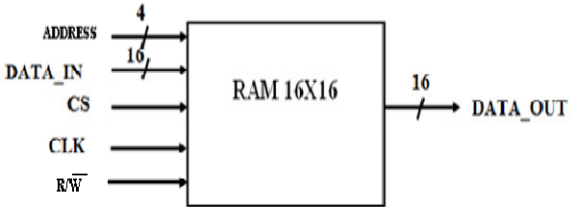

Figure-6 Memory block

ALU

Arithmetic and logical unit is a digital circuit that performs arithmetic and logical operations. The proposed design performs seven logical functions and two arithmetic functions.

The logical operations to be executed are AND, NAND, OR, NOR, XOR, XNOR and NOT while two logical operations are performed Addition and Subtraction. ALU will receive instruction bits from control unit and will execute the desired operation. For example, if input to control unit is 0000, the decoded bits will be 10000000 and after receiving the instruction bits from the decoder AND operation is performed by ALU according to the operands from register A and register B. The top block is shown in the figure7.

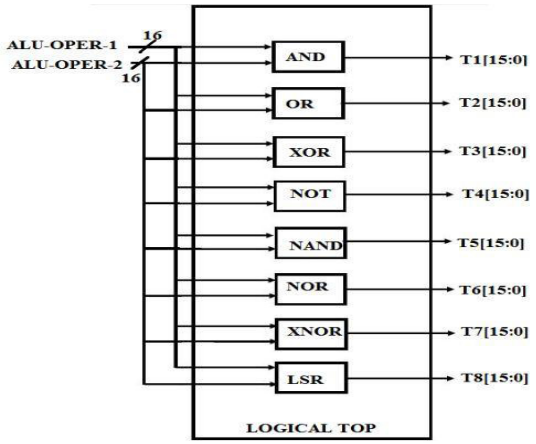

Figure-7: ***ALU Sub block (Logical Operations)***

BARREL SHIFTER

Barrel shifter is shown in the figure. It is a digital circuit that shifts the number of bits by specified times. It will receive the decoded instruction bits from the second instruction decoder inside the control unit and performs the desired operation depending upon operand from register A and select lines.

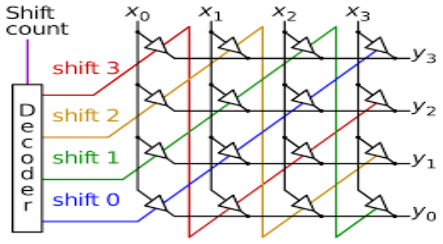

Figure-8: Barrel Shifter Block Diagram

Select lines			Decoder output								Function performed
S0	S1	S2	Zo0	Zo1	Zo2	Zo3	Zo4	Zo5	Zo6	Zo7	Operations
0	0	0	0	1	0	0	0	0	0	0	Zero opration
0	0	1	0	0	0	0	1	0	0	0	1 bit left rotate
0	1	0	0	0	0	0	0	1	0	0	2 bit left rotate
0	1	1	0	0	0	0	0	0	1	0	3 bit right rotate
1	0	0	0	0	0	0	0	0	0	1	4 bit left rotate

Figure-9: ***Operation of Barrel shifter***

UNIVERSAL SHIFT REGISTER

The architecture is shown in the figure. This architecture performs four main functions as follows loading the value, left shift and right shift and no change. If s4 and s5 both are low while z is equal to 01000000 the value is loaded. If s4 is low and s5 is high with decoded output z as 00100000 left shift operation is performed.

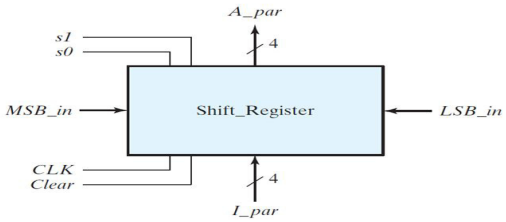

Figure-10: Universal Shift register block diagram

Select lines		Decoder output								Functions performed
s4	s5	Zo0	Zo1	Zo2	Zo3	Zo4	Zo5	Zo6	Zo7	Operations
0	0	0	1	0	0	0	0	0	0	Load
0	1	0	0	1	0	0	0	0	0	Left shift
1	0	0	0	0	1	0	0	0	0	Right shift
1	1	0	0	0	0	1	0	0	0	Zero operation

Figure-11: Operation of Universal shift register

RESULTS AND DISCUSSION

SIMULATION RESULTS

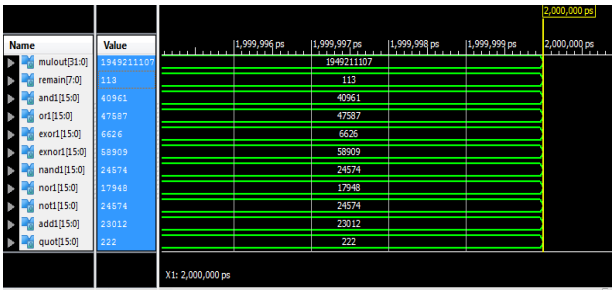

Figure-12 : Unsigned decimal representation of existing system

The figure-12 represents the unsigned decimal representation of the existing system. In this the inputs are unsigned decimals. Based on the inputs alu1, array1 and deno different alu operations such as xor,xnor,multiplication,addition ,division and some other logical operations are performed. The input deno is used as a divisor.

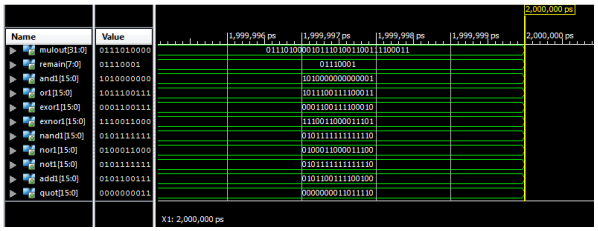

Figure-13 : Binary representation of existing system

The figure-13 represents the binary representation of the existing system. Based on the inputs alu1, array1 and deno different alu operations such as xor, xnor, multiplication, addition ,division and some other logical operations are performed. The input deno is used as a divisor.

RISC PROCESSOR SIMULATION RESULTS

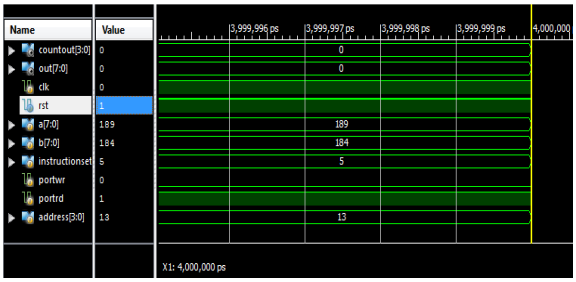

Figure-14: Unsigned decimal representation of proposed system

The figure-14 represents the unsigned decimal representation of the 16-bit RISC processor. Based on the clk and rst values operations are performed. The outputs portwr,portrd represents memory operations.

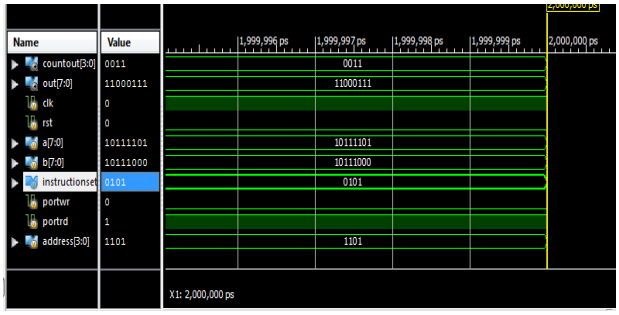

Figure-15:Binary representation of proposed system

The figure-15 represents the binary representation of the 16-bit RISC processor. The given inputs and outputs are in binary format. Based on the clk and rst values operations are performed. The outputs portwr,portrd represents memory operations.

RTL SCHEMATIC

The RTL Schematic of the proposed system RISC Processor with various inputs and outputs shown in the figure-16.

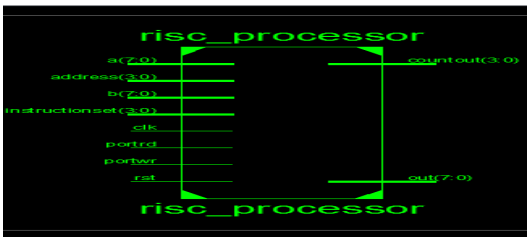

Figure-16: RTL Schematic of RISC processor

TECHNOLOGY SCHEMATIC

The existing system Technology schematic is as shown in the below figure with the large number of LUTs.

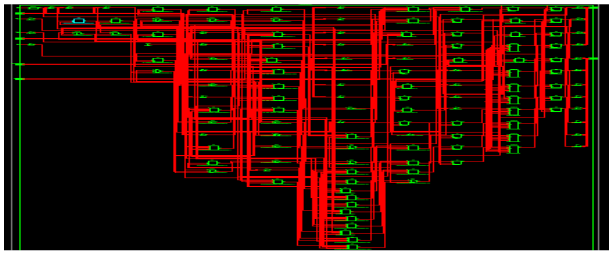

Figure-17: Technology schematic of proposed system

Table-1: Design summary of proposed system

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	283	4656	6%
Number of 4 input LUTs	517	9312	5%
Number of bonded IOBs	196	232	84%

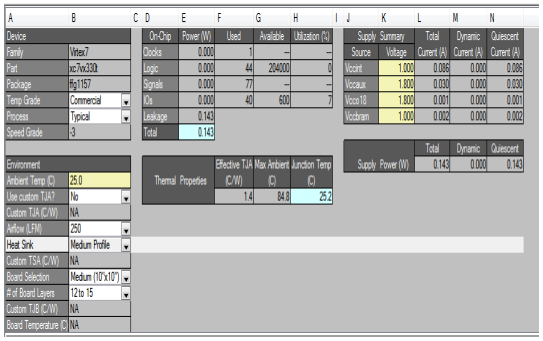

Figure-18: Total power estimated in X power analyser

The power and the temperature is determined by using the X power analyser during synthesis. The above figure-18 represents the power estimation in 16-bit RISC processor.

Table-2: Power Report

Parameters	**Existing System**	**Proposed System**
Power	1.293W	0.143W
Delay	72.331ns	15.221ns
Temperature	53.5	25.2

From the comparison table we can say that the proposed system requires very less power and also also delay is less when compared with the existing system.

CONCLUSION

The 16 bit RISC Processor using low power pipeline is designed and simulated using Xilinx 14.3 version software. The power is estimated using X power analyser. The proposed system consumes power about 0.143W which is less when compared with the existing system power 1.293W.Proposed 16 bit RISC processor consumes less power at lower delays instructions are executed. Hence the speed of operation is more in comparing with existing system.

FUTURE SCOPE

This work is limited to integer operation using pipeline process. In future operations for fraction can be pipeline process also it can be extended to ad super scalar architecture to perform instruction parallelism.

REFERENCES

1. "Supercomputer "K computer" Takes First Place in World". Fujitsu. Retrieved 20 June 2011.

2. Fisher, Joseph A.; Faraboschi, Paolo; Young, Cliff (2005). Embedded Computing: A VLIW Approach toArchitecture,CompilersandTools.p. 55. ISBN 155 86076.

3. Numerical Linear Algebra on High-Performance Computers by Jack J. Dungaree, et al 1987 ISBN 0-89871-428-1.

4. Processor architecture: from dataflow to superscalar and beyond by Jurij Šilc, Borut Robič, Theo Ungerer 1999 ISBN 3-540-64798-8.

5. Readings in computer architecture by Mark Donald Hill, Norman Paul Jouppi, Gurindar Sohi 1999 ISBN 1-55860-539-8

6. Patterson, D. A.; Ditzel, D. R. (1980). "The case for the reduced instruction set computer". ACM SIGARCH Computer Architecture News. **8** (6): 25–33. CiteSeerX 10.1.1.68.9623.

7. RISC I: A Reduced Instruction Set VLSI Computer by David A. Patterson and Carlo H. Sequin, in the Proceedings of the 8th annual symposium on Computer Architecture, 1981.

8. Design and Implementation of RISC I by Carlo Sequin and David Patterson, in the Proceedings of the Advanced Course on VLSI Architecture, University of Bristol, July 1982 [1]

9. The MIPS-X RISC microprocessor by Paul Chow 1989 ISBN 0-7923-9045-8 pages xix-xx

10.Processor design: system-on-chip computing for ASICs and FPGAs by Jari Nurmi 2007 ISBN 1-4020-5529-3 pages 52-53

11.Waterman, Andrew; Lee, Yunsup; Patterson, David A.; Asanovi, Krste. "The RISC-V Instruction Set Manual, Volume I: Base User-Level ISA version 2 (Technical Report EECS-2014-54)". University of California, Berkeley. Retrieved 26 December 2014.

12. "RISC vs. CISC: the Post-RISC Era" by Jon "Hannibal" Stokes (Arstechnica)

13."Guide to RISC Processors for Programmers and Engineers": Chapter 3: "RISC Principles" by Sivarama P. Dandamudi, 2005, ISBN 978-0-387-21017-9. "the main goal was not to reduce the number of instructions, but the complexity".

Author's Profile

P.Brundavani, M.Tech, she received her Master of Technology degree from JNTUA, Currently working as Assistant Professor in ECE department of Annamacharaya Institute of Technology and Sciences,(Autonomous),Rajampeta, affiliated to JNTUA, Ananthapuram. A.P. India. She has published in International Journals and National Conferences. She has attended many workshops and Seminars. Her research areas are Low Power VLSI, Digital IC Design, Signal processing, and Image processing and communication systems. Currently she is pursuing her Ph.D. in the area of Low Power VLSI design from Bio Medical applications, from JNTUA, Ananthapuram. She is also an IEI member.

Dr.D.VishnuVardhan,Ph.D,received his Doctorate in 2015 from JNTUA, Ananthapuramu and M.Tech in 2005 from JNTUACE in Computers and Communication. He received Bachelor Degree in 2000 from RGM College of Engineering & Technoloy Nandayala, Affiliated to JNTUA, Ananthapuramu. He has teaching experience of 15 years. He published many International and many national Journals. His areas of Specialisation are Embedded Systems and Image Processing. Currently, he is working as an Assistant Professor, Department of ECE, JNTUACE, Ananthapuramu. He worked as Head of the Department, Placement Officer, Officer in charge examinations, Deputy Warden, Officer in charge of Hostels, Guest House in charge and Alumni Co-ordinator. Presently working as TEQIP Co-ordinator at JNTUACE ,Ananthapuramu. He Guided 28 M.Tech. Students and currently, he is guiding 04 Ph.D students and 10 M.tech. Students.

P.Mahesh, currently he is pursuing B.Tech ECE from Annamacharaya Institute of Technology and Sciences, Rajampeta. He attended many paper presentations. His General Area of Interest include VLSI, Digital IC Design.

R.Suresh, currently he is pursuing B.Tech ECE from Annamacharaya Institute of Technology and Sciences, Rajampeta. He also worked at Multiland Electronics Pvt, Ltd, Bangalore. Heattended many Project Expo's. His General Area of Interest include VLSI, Communication systems.