"Design and Simulation of IEEE 754 Based Single Precision MIPS RISC Processor Using VHDL"

Prof. Vishal D. Jaiswal[1], Ms. Ashwini R. Golghate[2]
Department of Electronics and Telecommunication
Datta Meghe Institute of Engineering Technology and Research
***Corresponding Author's email id**: ashwinigolghate@gmail.com*

Abstract

This paper describes study of Floating Point 32-Bit MIPS RISC Processor Using VHDL (Very High Speed Integrated Circuit Hardware Description Language). In this IEEE 754 Single Precision Format is use for the floating point concept. This processor has fixed-length of 32-Bit instructions based on four different format R-format, I-format, J-format and I/O format and eight 32-Bit general-purpose registers. The instruction format used in this is based on RISC (Reduced Instruction Set Computer) Processor but the design process is simple which involves instruction fetch (IF), instruction decoder (ID), execution (EXE), data memory (MEM) and write back (WB) modules of 32-Bit CPU. The control unit controls the operations performed in these module. All the modules in this design are coded in VHDL language, as it is very useful in digital hardware. A more than 45 instructions are designed in initial development step of the processor. The instruction set consists of Arithmetic, Logical, Immediate, Jump, Load, store, I/O and HALT type of instruction. Finally, Simulation of the design is done in XILINX 14.5i ISE Simulator.

***Keywords:** VHDL, RISC, MIPS, Instruction Format, ISA*

INTRODUCTION

RISC architecture is a reverberation to the developing technology and the accumulation of philosophy from the CISC designs. CISC processors were designed to simplify compilers and to improve performance under constraints such as small and slower memories. The

Page 10-22 © MANTECH PUBLICATIONS 2018. All Rights Reserved

outstanding appearance of Reduced Instruction Set Computer (RISC) are they accommodate simple yet efficacious set of instruction that can execute in single clock cycle , Register-to- Register Operations, load and store operations are used to access memory, Simple Addressing Modes, Large Number of Registers, uses the Harvard architecture, pipeline is easy due to the fixed length instruction.

In this paper, the simulation of a 32 bit MIPS RISC processor based on floating point concept will be done by using VHDL. MIPS are the abbreviation for Million Instructions Per Second but we use here Microprocessor Interlocked Pipelined Stage. This 32-bit processor design using VHDL (Very High Speed Integrated Circuit Hardware Description Language) mainly consists of eight 32-bit General Purpose Registers, 32-bit Flag Register, A Control Unit, an Arithmetic Logic Unit (ALU), Decoder and Execution Unit and Memory Unit.

This processor has fixed-length of 32-bit instructions based on four different format R-format, I-format, J-format and I/O format and eight 32-bit general-purpose registers. This processor used floating point IEEE 754 Standard format for ALU operation

Microprocessors are widely used in the embedded sector based on general purpose application and special purpose application. Microprocessors are used in instruments to make it intelligent using structural coding as well as behavioral coding.

TECHNIQUE OF PROJECT

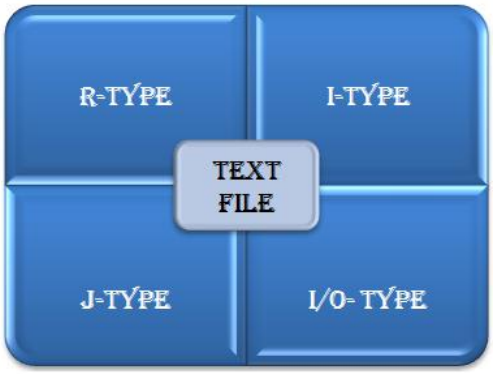

Fig 1.1 Overview of MIPS RISC processor

- Figure 1.1 shows the overview of MIPS RISC processor.

- In which the number of text file are design to write the data so that there is no need to go into software and change data.

- This text files method the simplest method to operate Xilinx software.

Page 10-22 © MANTECH PUBLICATIONS 2018. All Rights Reserved

BLOCK DIAGRAM

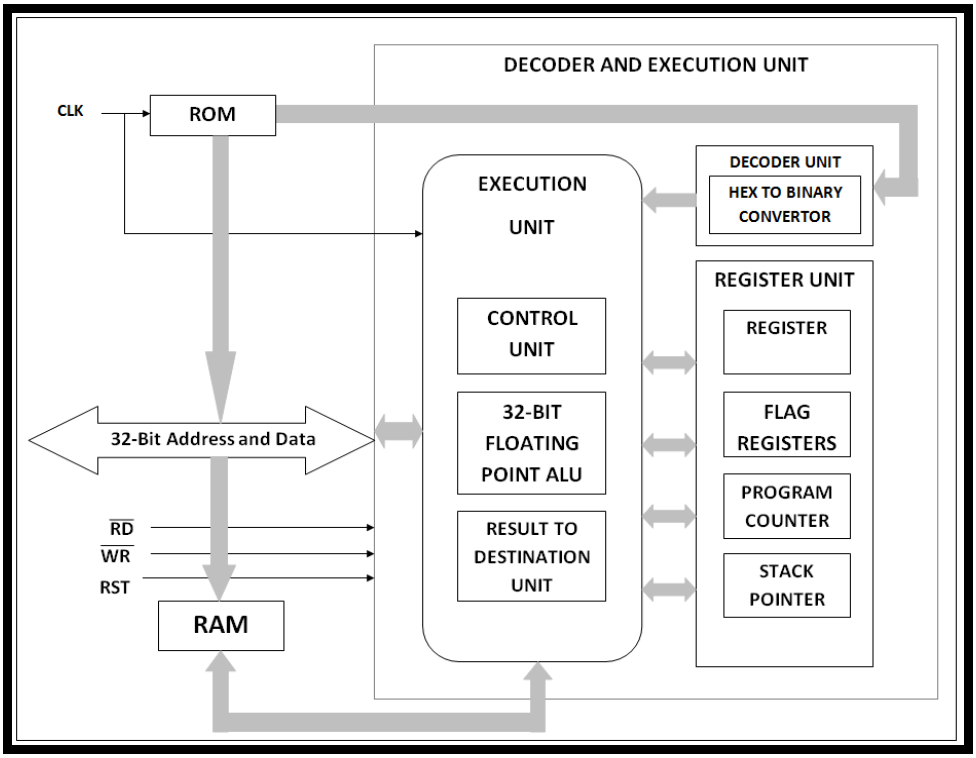

Fig 3.1 Block Diagram of Processor

This 32-Bit processor consists of following main blocks:

1) 32-Bit ALU
2) Decoder and Execution Unit
3) Register unit
4) Control Unit
5) Memory unit

1. 32-Bit Floating Point ALU:

In floating point 32-Bit processor, ALU is of 32-Bit which can perform different arithmetic and logical operation on maximum 32-Bit floating number at a time. The different arithmetic operation is addition, subtraction, multiplication, division, increment, decrement etc. The different logical operation is ANDing, ORing, EX-ORing, complement etc. While

ALU perform any operation either arithmetic or logical the 32-Bit result will be store into Result to Destination Unit.

2. Decoder and Execution Unit:

Decoder performs the function of instruction register and instruction decoder. Instruction register is a 32-Bit register which is used to store 32-Bit opcode. During execution of program processor will fetch the opcode from program memory and store it in instruction

Page 10-22 © MANTECH PUBLICATIONS 2018. All Rights Reserved

register. From instruction register, this opcode will be given as an input to instruction decoder. Depending upon the input to instruction decoder, one of the decoder output become active and corresponding instructional circuit available at that active output.

3. Control Unit:

The function of control unit to generate the control signal at that activeoutput and the signals are given to all those blocks which are involved in that operation to complete the operation. The control unit communicates with both the ALU and memory.

4. Register Unit:

Register unit consist of four main registers. They are as follows:

Register's:

It is a 32-Bit register which is used to store 32-Bit data or 32-Bit result temporary while performing any arithmetic or logical operation.

Flag Register:

In 32 bit processor, flag register is of 32 bit which consists of 32 flip-flops. Flag register is used to indicate the status of result obtained in Result and Destination Unit.

Program Counter:

Program counter is a 32 bit register, which is used to store 32 bit address of that memory location from where program opcode is to be fetched. After every opcode read operation, PC will be auto increment by one. So that next time next opcode will be read or fetch. Thus PC can also be defined as 32 bit register which hold 32 bit address of next memory location from where next opcode is to be fetched or read.

Stack Pointer:

Stack pointer is of 32 bit which is used to store 32 bit address of stack top memory location in internal RAM or data memory. During PUSH operation SP will be auto increment by one.

Memory unit:

The memory unit is classified as ROM and RAM. The ROM memory is expandable memory and size of ROM memory 4 KB. As the instructions which can be performed in RAM is up to 8 bit thus size of RAM can be defined as 2^8=256 bytes.

Page 10-22 © MANTECH PUBLICATIONS 2018. All Rights Reserved

MIPS INSTRUCTION SET

Table 4.1 MIPS Instruction Set

INSTRUCTION	INSTRUCTION DETAIL
ADD D,A,B	D <=A+B
SUB D,A,B	D<=A-B
MUL D,A,B	D <=A*B
DIV D,A,B	D <=A/B
AND D,A,B	D <=A AND B
OR D,A,B	D <=A OR B
NAND D,A,B	D <=A NAND B
NOR D,A,B	D <=A NOR B
XOR D,A,B	D <=A XOR B
XNOR D,A,B	D <=A XNOR B
NOT D,A	D <=A NOT B
RRA D,A	D <=A(Rotate Right Arithmetic)
RLA D,A	D <=A(Rotate Left Arithmetic)
RRL D,A	D <=A(Rotate Right Logically)
RLL D,A	D <=A(Rotate Right Logically)
ADDI D,A ,IMMEDIATE	D <=A + Immediate data(16 bit)
SUBI D,A, IMMEDIATE	D <=A - Immediate data(16 bit)
MULI D,A, IMMEDIATE	D(16 bit) <=A * Immediate data(16 bit)
DIVI D,A ,IMMEDIATE	D <=A / Immediate data(16 bit)
ANDI D,A, IMMEDIATE	D <=A AND Immediate data(16 bit)

Page 10-22 © MANTECH PUBLICATIONS 2018. All Rights Reserved

There are more than 60 instructions were design in this processor some are given above.

The selection of type of instruction on basis of opcode bits are as follows:

Table 4.2 Instruction type selection bit

SR NO.	OPCODE		TYPE OF INSTRUCTION
	BIT 31	BIT 30	
1	0	0	Register Type
2	0	1	Immediate Type
3	1	0	Jump/Branch type
4	1	1	Input/output type

The new MIPS instruction format is as follows:

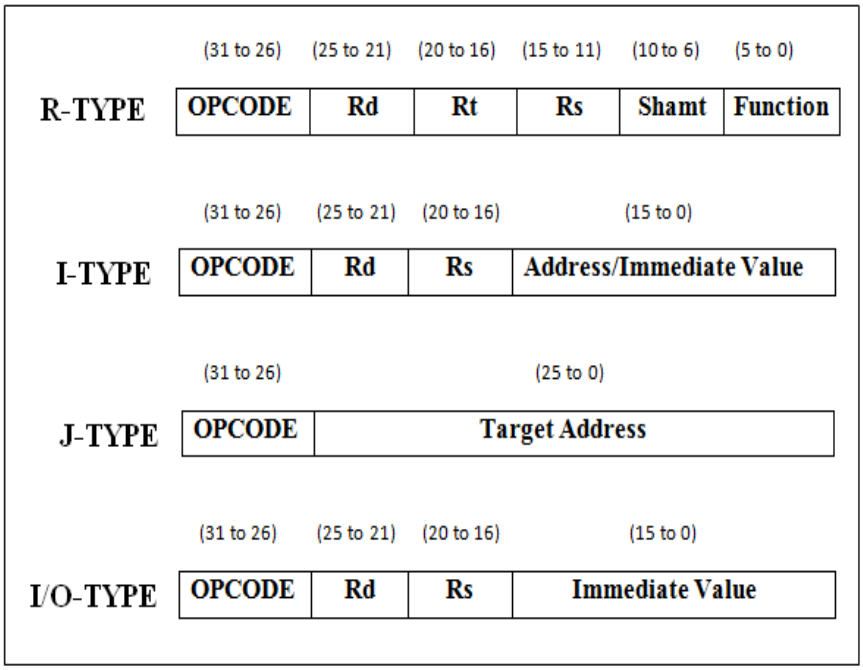

Fig 4.1 MIPS Instruction format of proposed work

Page 10-22 © MANTECH PUBLICATIONS 2018. All Rights Reserved

EXPERIMENTAL RESULTS

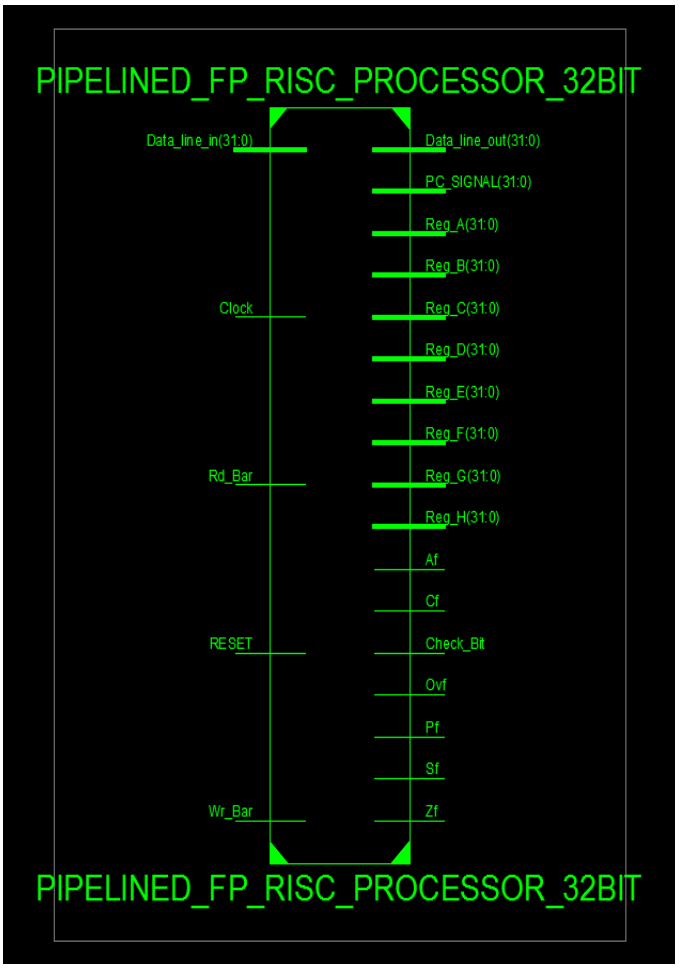

Fig 5.1 MIPS RISC Processor

The figure 5.1 shows the experimental result of MIPS RISC Processor. In which various signals like control signals, read signals, write signals etc. are designed.

SIMULATION RESULT USING MIPS INSTRUCTION FORMAT

• R-TYPE FORMAT

Addition:-

The simulation result show the operation of instruction ADD D, A, B. In this instruction the data from register A and B are added and the result is stored in register D.

Page 10-22 © MANTECH PUBLICATIONS 2018. All Rights Reserved

EXAMPLE:- $X = 2345.125_{10}$

S	EXPONENT	MANTISSA
0	10001010	00100101001001000000000

$Y = .75_{10}$

S	EXPONENT	MANTISSA
0	01111110	10000000000000000000000

Result:-

S	EXPONENT	MANTISSA
0	10001010	00100101000011000000000

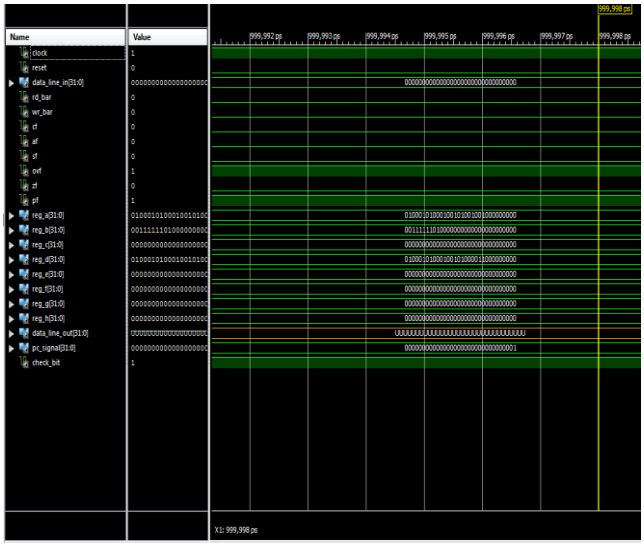

Fig 6.1 Simulation result of addition in R type format

This is only for floating point addition. The remaining arithmetic operation such as subtraction, multiplication etc. will perform by following IEEE 754 format.

• I-TYPE FORMAT

Considering instruction ANDI, A, IMMEDIATE. In this instruction the IMMEDIATE data directly given in instruction is ANDed with the data in register A and the result is stored in register D. The simulation result of this operation is as shown below-

EXAMPLE:-

OPCODE of ANDI		BINARY(lower 16 bit only)(upper 16bit are zero)
Upper 16 bit	Lower 16 bit	
5004	2229	0010 0010 0010 1001
REGISTER A		1001 0010 0000 0000
ANSWER		0000 0010 0000 0000

Page 10-22 © MANTECH PUBLICATIONS 2018. All Rights Reserved

Fig 6.2 Simulation result of addition in I type format

• J-TYPE FORMAT

Considering instruction JUMP TARGET. In this instruction the IMMEDIATE address is directly given in instruction and it will Jump to the target address. The simulation result of this operation is as shown below:

EXAMPLE:-

JUMP	**TARGET**	00000000000000000000000000000000
TARGET ADDRESS		00000000000000000000000000000000

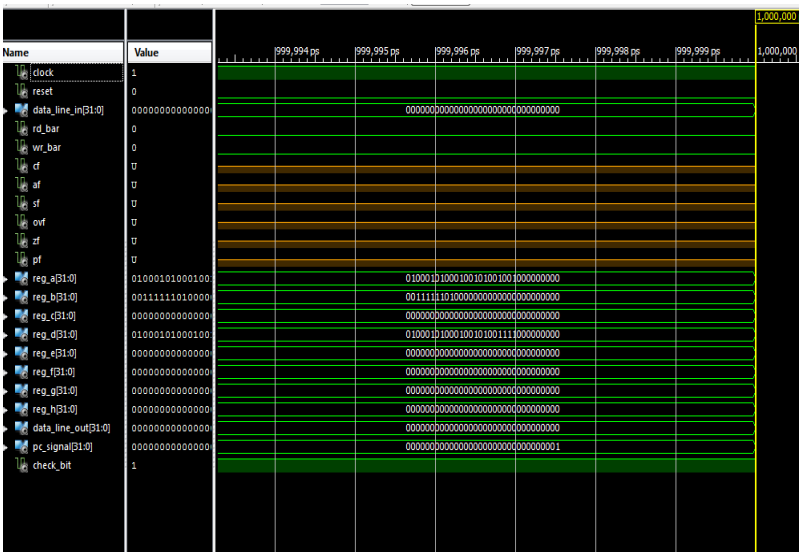

Fig 6.3 Simulation result of addition in J type format

Page 10-22 © MANTECH PUBLICATIONS 2018. All Rights Reserved

• I/O TYPE FORMAT

Considering instruction IN A. In this instruction the value from register A is read and then it is stored to the port using 'dout' signal. The simulation result of this operation is as shown below:

EXAMPLE:-

REG A	**A**	0100 0101 0001 0010 1001 0010 0000 0000
PORT	**dout**	0100 0101 0001 0010 1001 0010 0000 0000

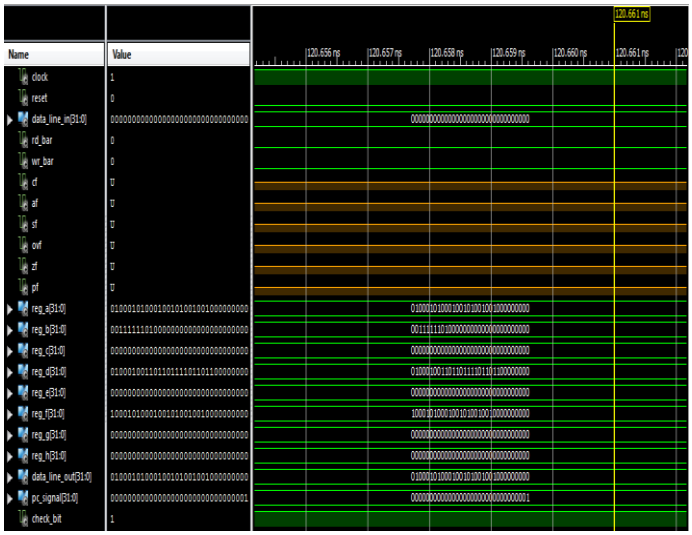

Fig 6.4 Simulation result of addition in I/O type format

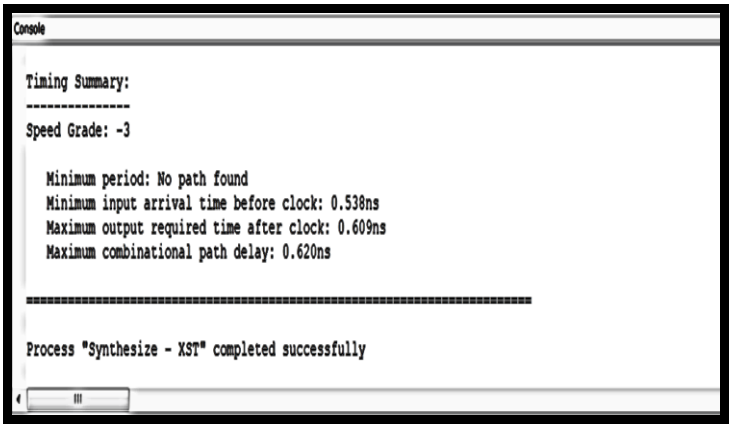

Fig 6.5 Final combinational path delay of MIPS processor

Fig 6.5 shows the final combinational path delay of this 32 – bit MIPS processor. This delay is common to all calculation were done in this processor. Hence this is very efficient processor

Page 10-22 © MANTECH PUBLICATIONS 2018. All Rights Reserved

COMPARION OF PARAMETER

PARAMETER	EXISTING WORK	PROPOSED WORK
Combinational Path Delay	20.105 ns	0.620 ns
Frequency	50 MHz	1.612 GHz
No. of Slices	464	1
No. of LUT's	805	0

CONCLUSION

In this proposed work 32-Bit MIPS RISC Processor has been design with less clock cycles per instruction having a delay of 0.620 ns and maximum operating frequency of 1.612 GHz. So, the processor can be called as a high performance processor. In this work Instruction format, 32-Bit ALU, Instructions Set, Flag register, eight MIPS General purpose Registers (A-H), Operation select, ROM and RAM are synthesized and simulated. More than 45 instructions are successfully designed. The design has been synthesized and simulated using Xilinx 14.5i ISE Simulator.

FUTURE SCOPE

1) In this microprocessor the number of instructions can be increase using the different instruction format.

2) The same concept can be used to design the processor with higher bits such as 64 bits, 128 bits etc.

3) If FFT concept is used in this processor then it is applicable in many areas with tremendous speed.

4) Making a GUI (Graphical User Interface) is possible using Xilinx Vivado 2013.14 Software and Visual Basics C# instead of text files.

REFERENCES

I. Mrs. Sangeetapalekar and Mr. Nitinnarkhede"32-bit RISC processor with floating point unit for DSP applications" IEEE International Conference On Recent Trends In Electronics Information Communication

Page 10-22 © MANTECH PUBLICATIONS 2018. All Rights Reserved

Technology, May 20-21, 2016, India.

II. Mr. AmitPandey"Simulating a Pipelined RISC Processor" Member, IEEE Department of Computer Science, Sunrise University, Alwar, India – 2016

III. AneeshRaveendran, VinayakBaramuPatil, David Selvakumar, Vivian Desalphine"A RISC-V Instruction Set Processor-Micro architecture Design and Analysis" Centre for Development of Advanced Computing (C-DAC) Bangalore, India – 2016

IV. Mrs. Rupali S. Balpande and Mrs. Rashmi S. Keote "Design of FPGA based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor" Department of Electronics Engineering, International Conference on Communication Systems and Network Technologies-2011

V. J. Bhaskar, ―VHDL Primer, Pearson Education, 3rd edition, 2000.

VI. S. S. Limaye ― VHDL: A Design Oriented Approach, January 13, 2009.

VII. V Udayashankara, M S Mallikarjunaswamy― 8051 Microcontroller.

VIII. Mr. Raj Prakash Singh, Mr. Ankit K Vashishtha and Mr. Krishna R "32 Bit Reconfigurable RISC Processor Design and Implementation for BETA ISA with Inbuilt Matrix Multiplier" Dept. of ECE Bangalore Institute of Technology Bangalore, India – IEEE 2016

IX. PriyankaTrivedi&Rajan Prasad Tripathi"Design& Analysis of 16 bit RISC Processor Using low Power Pipelining" Department of Electronics and Communication Galgotias University Gr.Noida, India – IEEE 2015

X. Mohit N. Topiwala and N. Saraswathi"Implementation of a 32-bit MIPS Based RISC Processor using Cadence" Department Of Electronics and Communication, SRM University,Chennai, India, IEEE International Conference on

Page 10-22 © MANTECH PUBLICATIONS 2018. All Rights Reserved

Advanced Communication Control and Computing Technologies (ICACCCT)-2014

XI. E. Ayeh, K. Agbedanu, Y. Morita, O. Adamo, and P. Guturu"FPGA Implementation of an 8-bit Simple Processor" Department of Electrical Engineering University of North Texas, Denton, TX. 76207 USA - 2008.

XII. MohamadHasman Bin Abdul Manas"8-BIT MICROPROCESSOR DESIGN USING VHDL" FakultiKejuruteraanElektronikdan KejuruteraanKomputerUniversitiTeknikal Malaysia Melaka - 2007.

XIII. Muhammad Ali Mazidi, Janice GillispieMazidi, Rolin D Mckinlay ― The 8051 Microcontroller and Embedded Systems , 2nd edition, 2008.

XIV. Vishwas V. Balpande, Abhishek B. Pande, Meeta J. Walke,Bhavna D. Choudhari, Kiran R. Bagade"Design and Implementation of 16 Bit Processor on FPGA " International Journal of Advanced Research in Computer Science and Software Engineering 5(1), January - 2015, pp. 704-708

XV. A K Ray, K M Bhurchandi ―Advanced Microprocessors And Peripherals―2nd Edition, Tata McGraw-Hill Education, 2006.

Page 10-22 © MANTECH PUBLICATIONS 2018. All Rights Reserved