Design and Implementation of 32-Bit RISC processor using Verilog HDL

[1]Tanvi Sawant, [2]Narayan Korgaonkar, [3]Vishwanath Sherlekar, [4]Guruprasad Bhise, [5]Prof.Bhavna Thakur
[1]B.E. Student, [2]B.E. Student, [3]B.E. Student, [4]B.E. Student, [5]proffesor of Electronics and Telecommunication Engineering
Department of Electronics and Telecommunication Engineering
Shivajirao S. Jondhale College of Engineering
Dombivli East, Thane, Maharashtra 421204, India.

Abstract : *RISC is a design technique used to reduce the amount of area required, complexity of instruction set, instruction cycle during the implementation of the design. This paper presents an 32-bit RISC processor design using Verilog Hardware Description Language (HDL) on FPGA board.The proposed processor is designed using Harvard architecture, having separate instruction and data memory. The salient feature of proposed processor is pipelining, used for improving performance, such that on every clock cycle one instruction will be executed.Another important feature is that instruction set contains only 16 instructions, which is very simple, easy to learn and compact. The proposed processor has 32-bit ALU, Eight 32-bit general-purpose registers. The pipelined controller is designed by using four units and they are fetch unit,decode unit, execute unit and internal register unit.The proposed processor is physically verified on Xilinx Spartan 3E Starter Board FPGA at 12 MHz clock frequency.*

Keywords – *RISC, FPGA, Verilog, Pipelining, Instruction, Opcode.*

I. INTRODUCTION

This project presents Design, Simulation and Implementation of a 32-bit RISC processor using Verilog Hardware Description Language (HDL) on spartan3E250 board.

Introduction to RISC:

RISC stands for Reduced Instructions Set Computer (RISC). It is a processor which uses simple instructions, which performs low level operations in a single clock cycle. It has a very high performance capacity and capable of executing instructions in a single microprocessor cycle. It has advantage of having pipelining therefore multiple instructions can execute in a single clock cycle. This results in high speed instruction execution. It also has Load/Store architecture where memory is accessed through particular instructions. It also has a simple Arithmetic Logic Unit (ALU) for basic operations with simple and uniform instruction set [1]. This work presents the design of reconfigurable 32 bit RISC processor.

Introduction to FPGA:

FPGA- Field Programmable Gate Array; is an integrated circuit designed and configured by a customer or designer after manufacturing hence called “Field Programmable”. The specification of configuration is normally done by using Hardware Description Language (HDL). It contains Logic Components which are programmable called Logic box and it has the ability to interconnect that cell or blocks to be wired together. It contains over 10000 logic cells. The individual cells are interconnecting by a matrix of wires and programmable switches [2]

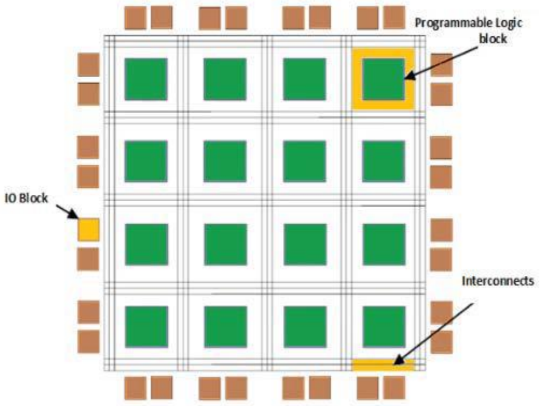


Fig 1.Basic FPGA

FPGA plays a major role for customizing the processor and reconfiguration of many electronic devices. The Hardware Description Languages (HDLs) increase the range of options available to FPGA designers by allowing designers to implement flexible intellectual Property (IP), often referred to as 2
IP cores. IP is the implementation of reusable components, which describe and implement hardware functionality
[3].
Field Programmable Gate Arrays (FPGA) is growing fast with cost reduction compare to ASIC design. This work is concerned with the design and implementation of a low cost 8bit Reduced Instruction Set Computer (RISC) processor on a FPGA. It provides the benefits of custom VLSI design while avoiding the initial cost, time delay and inherent risk of a conventional masked gate array [4]. They are customized by loading configuration data into the internal memory cell. RAM based FPGA's can be infinitely reprogrammed in-circuit in only a fraction of seconds. Design revisions even for fielded products can be implemented quickly and precisely [5].
Verilog HDL is a standard language which describes hardware of a digital system therefore it is called Hardware Description Language (HDL). It allows the user to design to be simulated earlier before implement to correct errors or experiments with different architectures [1].

II. ANALYSIS AND DESIGN

Flow chart of the process:
The following flowchart shows the steps for the design of our proposed system:

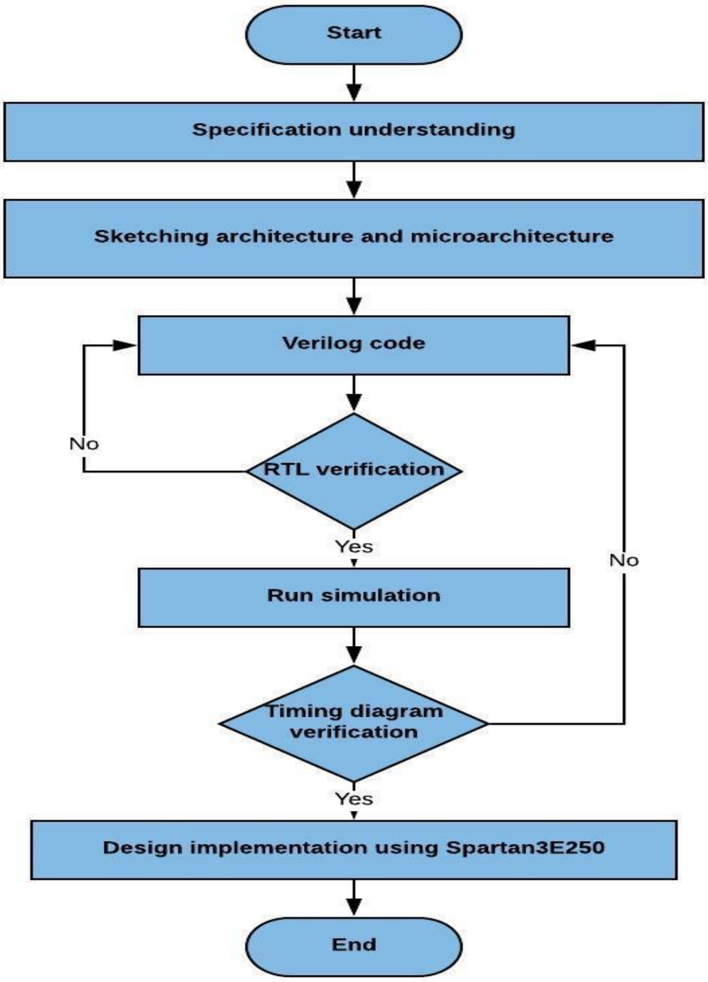

Start
Specification understanding
Sketching architecture and microarchitecture
Verilog code
No
RTL verification
Yes
No
Run simulation
Timing diagram verification
Yes
Design implementation using Spartan3E250
End

Fig 2. Flowchart of the process

- It was decided to implement 16 instructions in RISC processor. For that 4-bit unique opcode was required. Also for the store unit (source and destination) it was decided to use 8 registers each of 32-bit. So to uniquely define these 8 registers, a 3-bit opcode is required.

- Specification understanding includes the determination of input pins and output pins of proposed RTL for particular unit.
- Sketching the architecture and micro architecture include proposed architecture and micro-architecture of 32-bit RISC Processor.
- RTL Design includes RTL schematic of particular unit depending upon the RTL code.
- Analysis of RTL design, timing diagram and simulation is the part of Functional verification and synthesis.
- After verifying all above steps implementation of code on spartan3E250 development board will be done. It includes final verification of program.

Proposed Architecture:

The pipelined processor consists of instruction memory, data memory and pipelined processor. The architecture is evolved from the design specifications. Every Processor has the processing unit, internal register storage, and instruction memory with instruction cache.

The pipelined processor is designed to perform the operations on 32 bit binary inputs and has maximum 16 instructions. So it needs 4 bit instruction code called as Opcode. The design has 8 internal registers of 32 bit each and to address these registers it requires 3 bit address. The processor operates on 32 bit data input and generates the 32 bit or 64 bit result depending on the operation performed.

The top level architecture is shown in the Figure 3 and it consists of pipelined processor, instruction memory and data memory. The major emphasis of this paper is to describe the micro-architecture of pipelined processor and to implement the pipelined processor on FPGA.

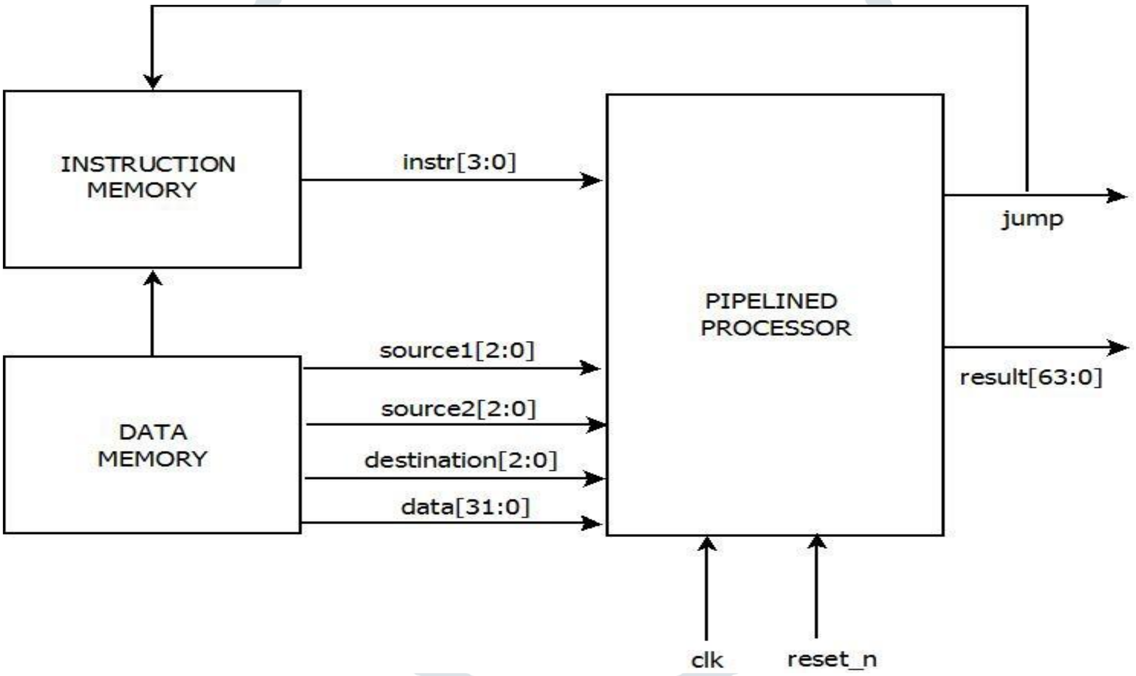


Fig 3. Proposed architecture of Pipelined Processor

Proposed Microarchitecture:

The Micro-architecture for pipelined processor is derived from the architecture. The micro-architecture is detail low level abstraction of architecture. For the pipelined processor shown in Figure 3.2, the micro-architecture is derived to implement the required instructions with four stage pipeline. Since we have decided to implement 4 stages pipelining, the fetch, decode, execute and Internal Register unit were designed individually and interfaced by using Verilog code and defined as top module. The Micro-architecture sub blocks are: fetch, decode, execute and Internal Register unit and are shown in Figure 4.

Figure 4 describes the micro-architecture for the pipelined Processor [Note: It is assumed that all the units shown in the Figure 4 has synchronous clock signal clk and reset]. As shown in the architecture the instruction passes through fetch, decode, internal register unit and execute units.

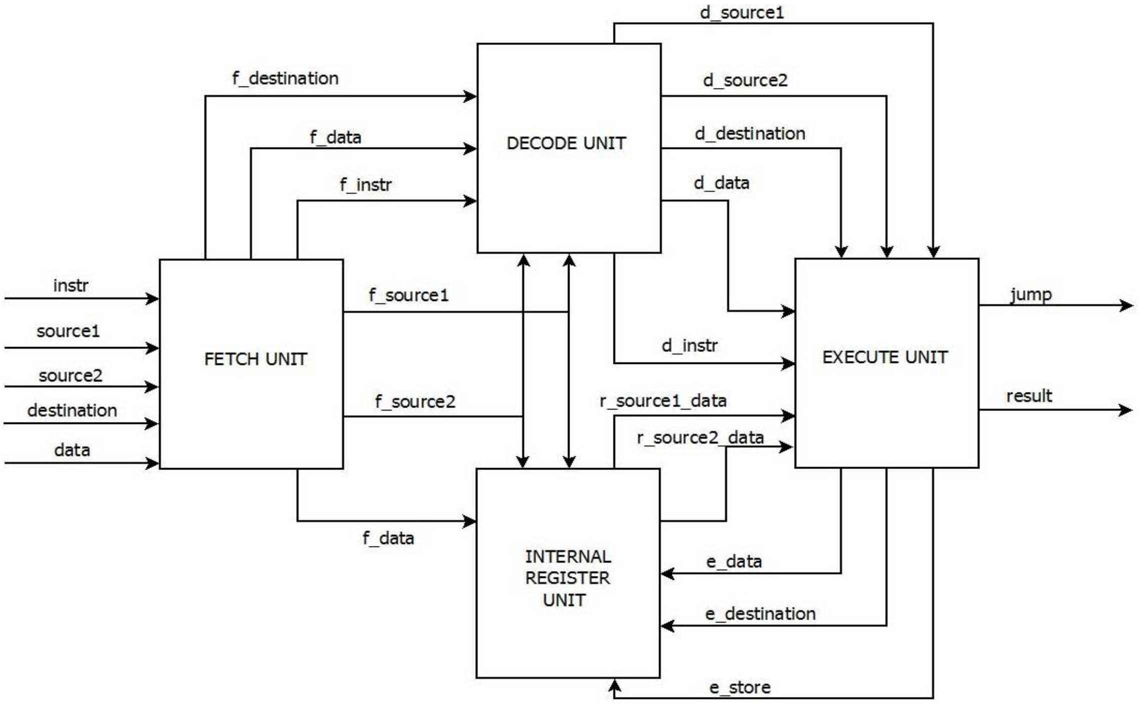


Fig 4. Proposed microarchitecture of Pipelined Processor

Proposed Instruction Set:

Table 1. List of Instructions

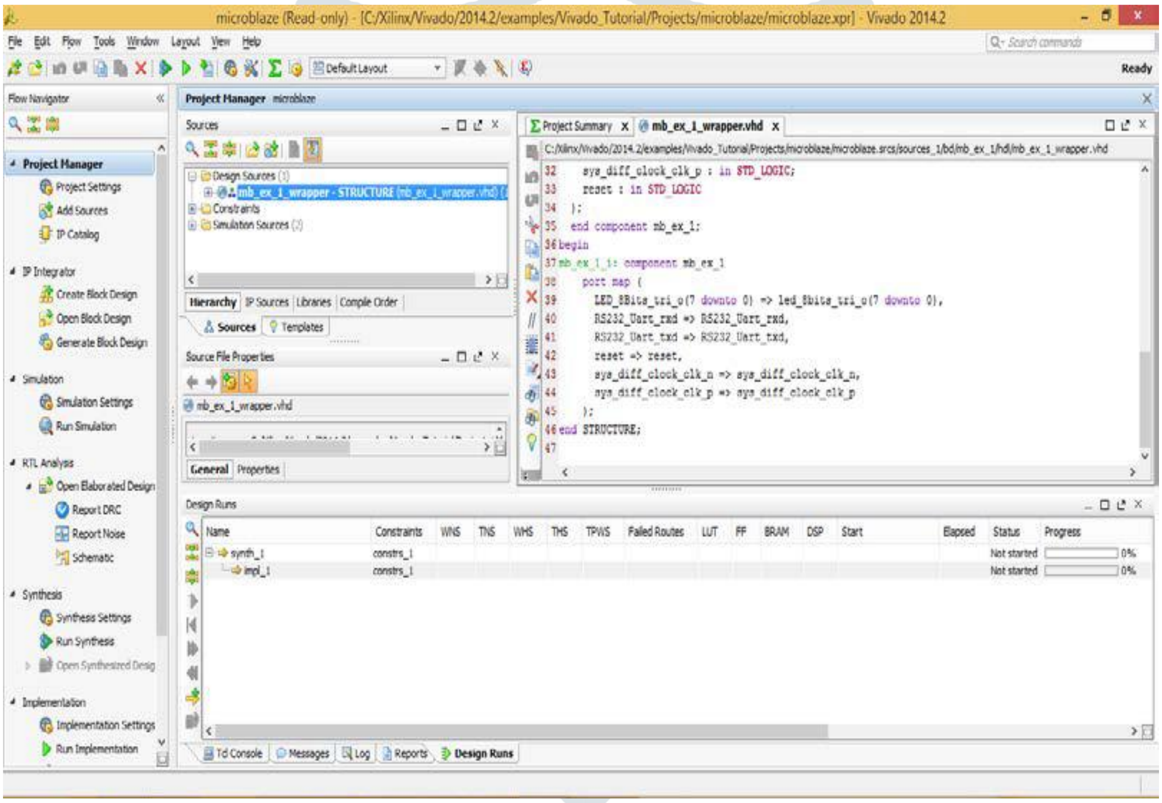
Instruction opcode (4bits)	**Instructions**
0000	Transfer/Read data in result
0001	Transfer/Read registers(ram) in result
0010	Move data of source1 into source2
0011	Add source1, source2 and store in destination
0100	Sub source1, source2 and store in destination
0101	Increment given data and store in destination
0110	Decrement given data and store in destination
0111	Multiply source1, source2 and store in destination
1000	Clear register
	Load data in destination
1010	Not source1, source2 and store in destination
1011	And source1, source2 and store in destination
1100	Or source1, source2 and store in destination
1101	Nand source1, source2 and store in destination

1110	Nor source1, source2 and store in destination
1111	Xor source1, source2 and store in destination

Software Requirement Specification:

Xilinx Vivado software is main part of our project used for writing an RTL code. Also, by using this software we can elaborate RTL design, synthesis and simulation. RTL schematic include verification of RTL schematic and input output pins of RTL diagram. Simulation includes study of timing diagram.

Vivado Design Suite is a software suite produced by Xilinx for synthesis and analysis of HDL designs, superseding Xilinx ISE with additional features for system on a chip development and high-level synthesis. Vivado represents a ground-up rewrite and re- thinking of the entire design flow (compared to ISE), and has been described by reviewers as "well-conceived, tightly integrated, blazing fast, scalable, maintainable, and intuitive"[6].



microblaze (Read-only) - [C:/Xilinx/Vivado/2014.2/examples/Vivado_Tutorial/Projects/microblaze/microblaze.xpr] - Vivado 2014.2

C:/Xilinx/Vivado/2014.2/examples/Vivado_Tutorial/Projects/microblaze/microblaze.srcs/sources_1/bd/mb_ex_1/hdl/mb_ex_1_wrapper.vhd

```
    sys_diff_clock_clk_p : in STD_LOGIC;
    reset : in STD_LOGIC
  );
  end component mb_ex_1;
begin
mb_ex_1_i: component mb_ex_1
    port map (
      LED_8Bits_tri_o(7 downto 0) => led_8bits_tri_o(7 downto 0),
      RS232_Uart_rxd => RS232_Uart_rxd,
      RS232_Uart_txd => RS232_Uart_txd,
      reset => reset,
      sys_diff_clock_clk_n => sys_diff_clock_clk_n,
      sys_diff_clock_clk_p => sys_diff_clock_clk_p
    );
end STRUCTURE;
```

Design Runs

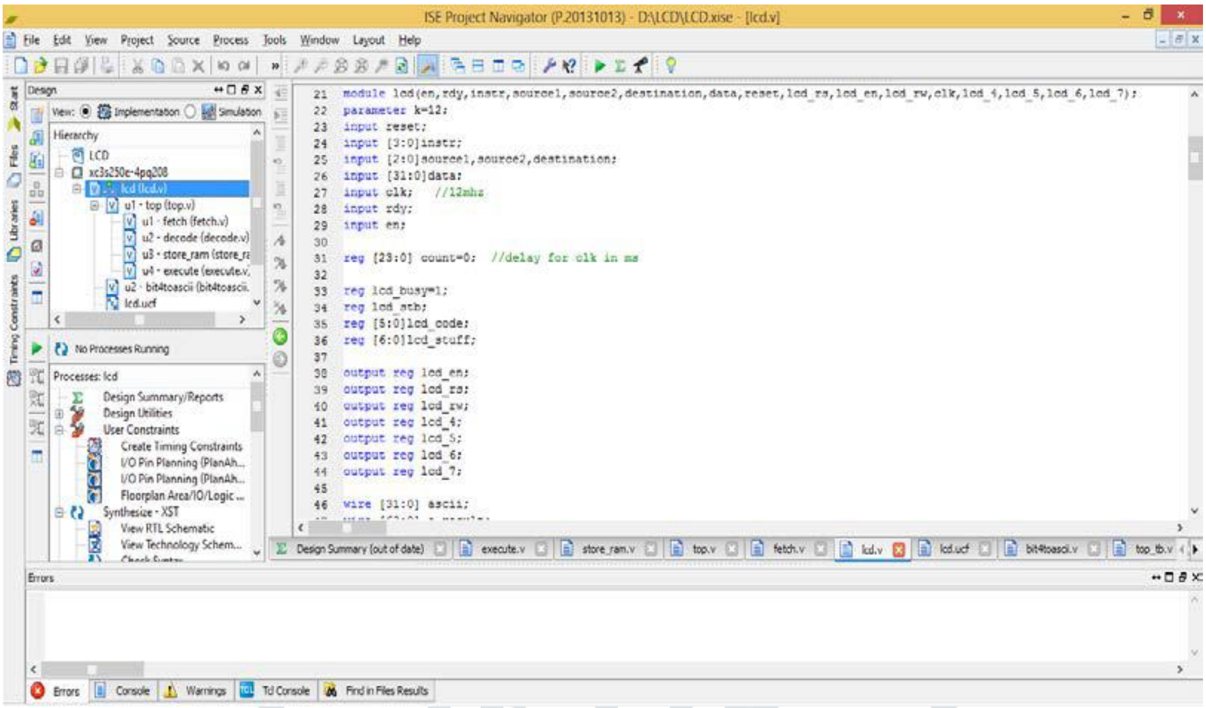
Name	Constraints	WNS	TNS	WHS	THS	TPWS	Failed Routes	LUT	FF	BRAM	DSP	Start	Elapsed	Status	Progress
synth_1	constrs_1													Not started	0%
impl_1	constrs_1													Not started	0%

Fig 5. XilinxVivado window

Vivado enables developers to synthesize their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. Vivado is a design environment for FPGA products from Xilinx, and is tightly-coupled to the architecture of such chips.

Vivado is an integrated design environment (IDE) with system-to-IC level tools built on a shared scalable data model and a common debug environment. Vivado includes electronic system level (ESL) design tools for synthesizing and verifying C-based algorithmic IP; standards-based packaging of both algorithmic and RTL IP for reuse; standards-based IP stitching and systems integration of all types of system building blocks; and the verification of blocks and systems. A free version WebPACK Edition of Vivado provides designers with a limited version of the design environment.

Xilinx ISE 14.7 software was used for implementation of the project on the FPGA development board.



```
module lcd(en,rdy,instr,source1,source2,destination,data,reset,lcd_rs,lcd_en,lcd_rw,clk,lcd_4,lcd_5,lcd_6,lcd_7);
parameter k=12;
input reset;
input [3:0]instr;
input [2:0]source1,source2,destination;
input [31:0]data;
input clk;   //12mhz
input rdy;
input en;

reg [23:0] count=0;  //delay for clk in ms

reg lcd_busy=1;
reg lcd_stb;
reg [5:0]lcd_code;
reg [6:0]lcd_stuff;

output reg lcd_en;
output reg lcd_rs;
output reg lcd_rw;
output reg lcd_4;
output reg lcd_5;
output reg lcd_6;
output reg lcd_7;

wire [31:0] ascii;
```

Fig 6.Xilinx ISE 14.7 window

III. IMPLEMENTATION

Hardware requirement:

The Verilog code of this project was implemented using sparten3E250 board.

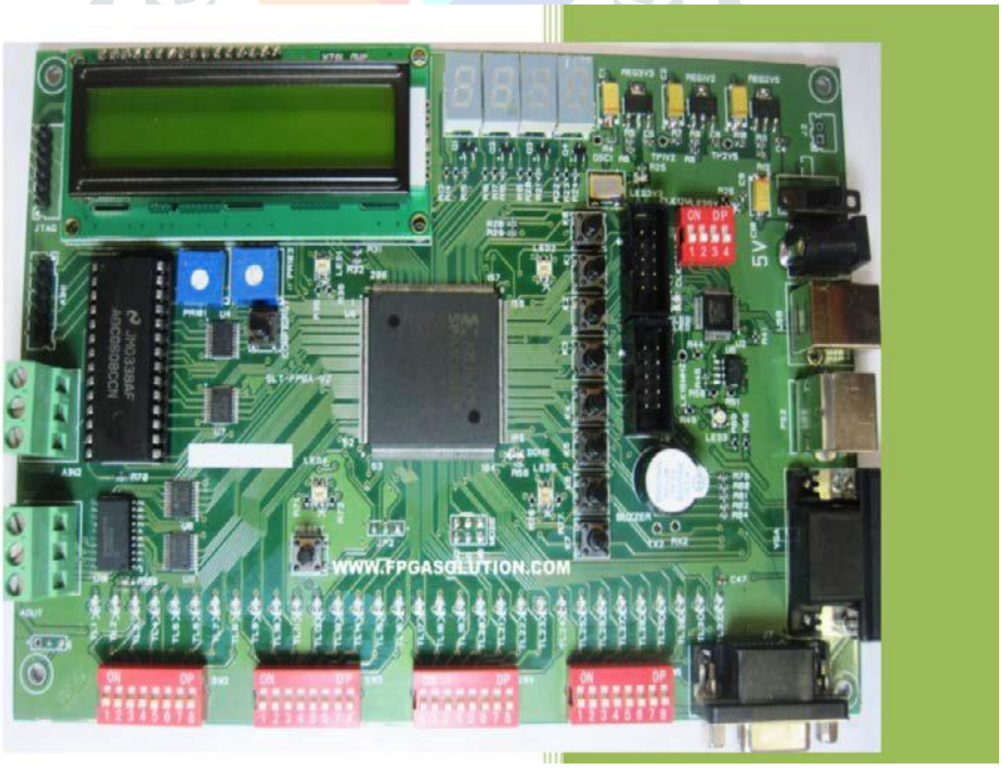


Fig 13. Spartan 3E250 Board

The key features of the VLSI Development board are:

- Spartan-3E XC3S250E FPGA has
  - Up to 172 user-I/O pins

- 208-pin FBGA package
- Over 5,000 logic cells
- 2-line, 16-character LCD screen
- One9-pin RS-232 ports
- 32 discrete LEDs and DIP switches
- PS/2 mouse or keyboard port
- Separate VGA display port
- Expansion connectors (16 free user I/O arranged in 10 pin FRC)
- Dual, 8-bit DAC
- 8-bit, 8 channel ADC
- USB interface (type B)
- 8 pushbuttons for trigger, input
- 4 common anode 7-segment LED display
- Buzzer
- RGB Led RG Led can used for traffic light processor
- Xilinx 2 Mbit Platform Flash configuration PROM
- On board USB Jtag

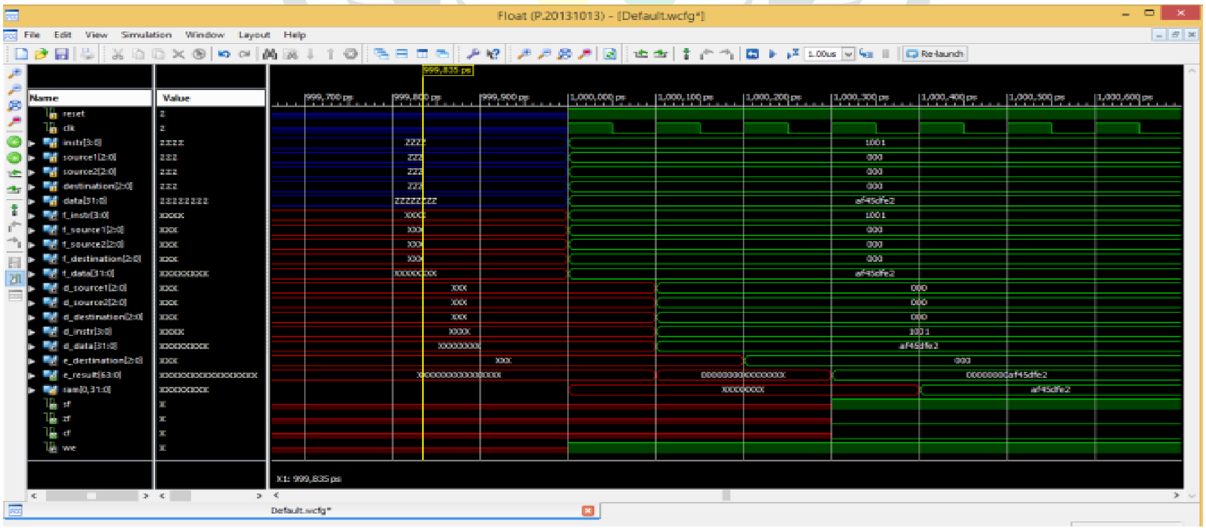
Configuration of LCD and DIP switches:

The LCD which is provided on Spartan3E250 development board is interfaced with FPGA chip using Verilog code in ISE design suite. After successful interfacing, FPGA displayed the result of executed instructions on LCD. The board has 32 individual bidirectional I/O's. Each I/O is connected with a surface-mount LED and a DIP switch. In our design we used DIP switches to provide digital input (i.e. logic 0 and logic 1) to the FPGA.

IV. RESULTS

Simulation of Processor

The simulation of the proposed architecture and the results are satisfactory as we can see in the simulation window that all modules and the 4 units are working properly and shows following four stage pipelined behaviour. The Fig 14 shows the simulation result of one instruction of our proposed design using Vivado. Form the figure it seen that the opcode of instruction which is given as an input is successfully fetched, decoded and executed also the result is stored in respective resister. From the simulation result we can calculate the execution period of each instruction.



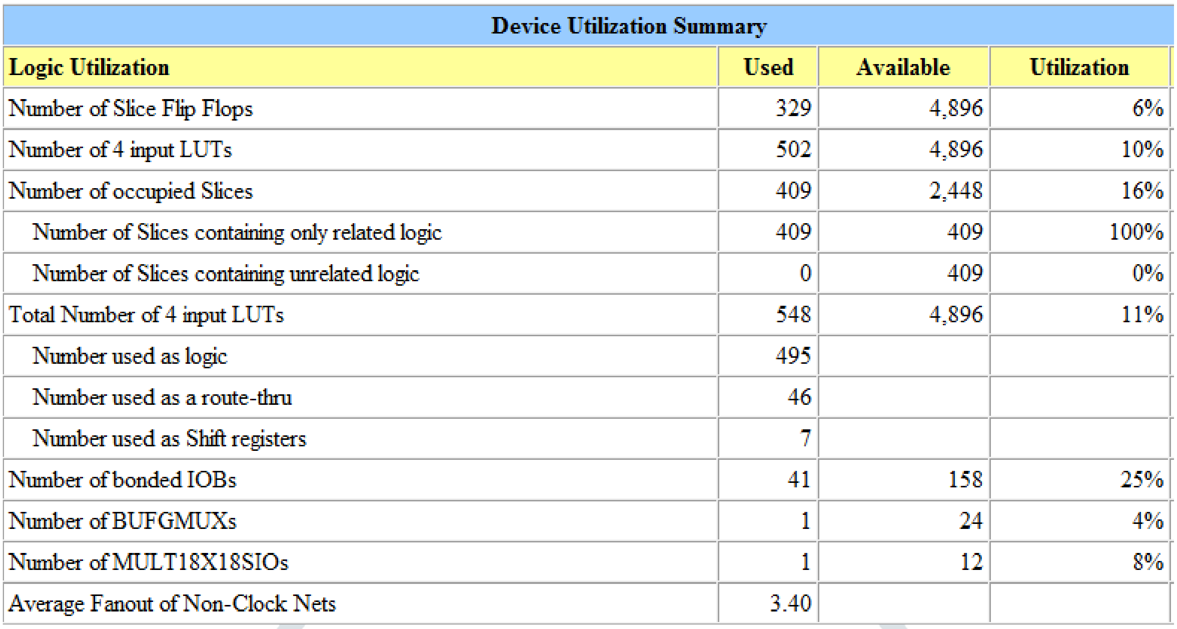
Name	Value
reset	Z
clk	Z
instr[3:0]	ZZZZ
source1[2:0]	ZZZ
source2[2:0]	ZZZ
destination[2:0]	ZZZ
data[31:0]	ZZZZZZZZ
f_instr[3:0]	XXXX
f_source1[2:0]	XXX
f_source2[2:0]	XXX
f_destination[2:0]	XXX
f_data[31:0]	XXXXXXXX
d_source1[2:0]	XXX
d_source2[2:0]	XXX
d_destination[2:0]	XXX
d_instr[3:0]	XXXX
d_data[31:0]	XXXXXXXX
e_destination[2:0]	XXX
e_result[63:0]	XXXXXXXXXXXXXXXX
ram[0,31:0]	XXXXXXXX
sf	X
zf	X
cf	X
we	X

Fig 14. Simulation of Four Stage Pipelined Processor

Device Utilization:

From the Table 2 we can conclude that the look up tables (LUTs) and Slices on the FPGA are sufficient for this design. The design summary which is created in the Xilinx Software is given.

Table 2: Device Utilization Summary



Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	329	4,896	6%
Number of 4 input LUTs	502	4,896	10%
Number of occupied Slices	409	2,448	16%
Number of Slices containing only related logic	409	409	100%
Number of Slices containing unrelated logic	0	409	0%
Total Number of 4 input LUTs	548	4,896	11%
Number used as logic	495		
Number used as a route-thru	46		
Number used as Shift registers	7		
Number of bonded IOBs	41	158	25%
Number of BUFGMUXs	1	24	4%
Number of MULT18X18SIOs	1	12	8%
Average Fanout of Non-Clock Nets	3.40		

Timing Parameters of our design:

The Table 3 is obtained in ISE Design Suite which gives the timing parameters of our design. From this Table it can be seen that there is no time violation and multicycle path in our design. The Hold time is defined as the minimum amount of time after the clock's active edge during which data must be stable and it is 0.722 ns in our design. It is observed that the overall delay of the circuit is 13.669 ns. The maximum frequency of this processor is 73.158 MHz.

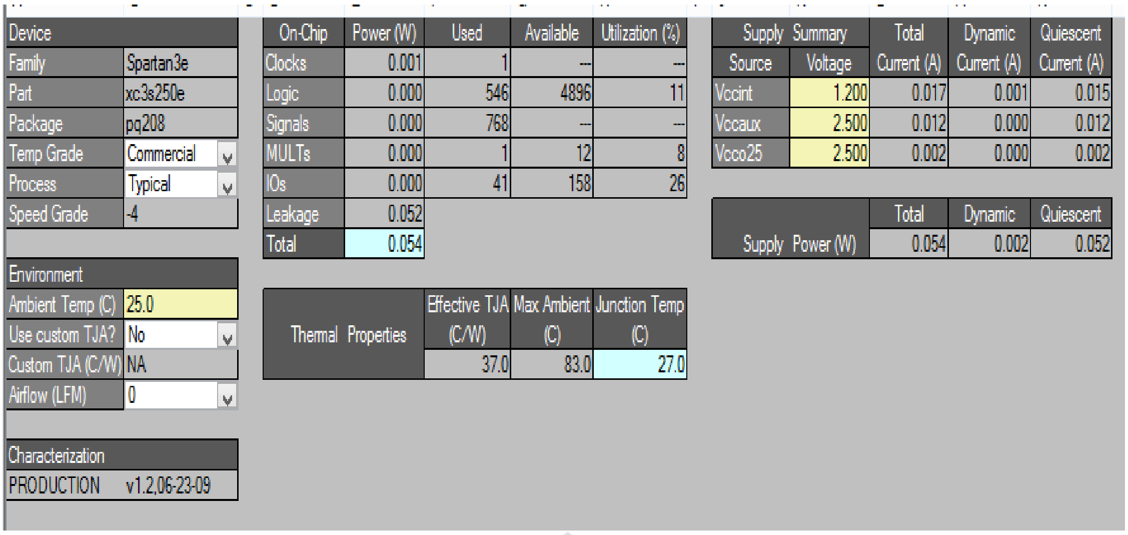
Table 3: Timing Parameters Result

Design Parameters	Result
Time violation reported	None
Multicycle Paths	None
Design Speed	12 MHz
Minimum Period	13.669 ns
Maximum Frequency	73.158 MHz
Setup Path	69.661 ns
Hold Path	0.722 ns
Component Switching Limits	80.138 ns

Power Utilization & Thermal Properties:

The Table 4 illustrates the On-Chip power utilization during the synthesis of proposed design and from the table it seen that the total power utilization is 0.054W also there is no individual power utilization by logical elements, Multiplexers (MULTs) and I/O’s.

Table 4: Power Utilization &Thermal Properties



Device	
Family	Spartan3e
Part	xc3s250e
Package	pq208
Temp Grade	Commercial
Process	Typical
Speed Grade	-4

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
PRODUCTION	v1.2,06-23-09

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.001	1	--	--
Logic	0.000	546	4896	11
Signals	0.000	768	--	--
MULTs	0.000	1	12	8
IOs	0.000	41	158	26
Leakage	0.052			
Total	0.054			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	37.0	83.0	27.0

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.017	0.001	0.015
Vccaux	2.500	0.012	0.000	0.012
Vcco25	2.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.054	0.002	0.052

Comparison of parameters:

The Table 5 shows the comparison of different parameters of our design with the published research papers. On comparing the parameters it can be seen that maximum frequency of our design is 73.158MHz which better than the other designs. Also the total power utilization and number of LUTs of our design during synthesis is better than previous designs.

Table 5: Comparison of parameters

Parameters	16-Bit low power RISC processor using VHDL[7].	32 –Bit RISC CPU based on MIPs[8].	32-Bit RISC processor using Xilinx[9].	**Proposed 32-Bit RISC Processor**
Delay(ns)	16.732	52.719	18.243	**13.609**
Frequency(MHz)	59.767	18.970	54.81	**73.158**
Total Power(W)	0.098	_	0.829	**0.054**
Number of Slices	3820	4283	471	**738**
Number of LUTs	3227	23545	_	**548**

V.

CONCLUSION, FUTURE SCOPE

Conclusion:

The presented architecture of 32-bit 4 stage pipelined processor is implemented on Spartan-3E XC3S250E. The design is implemented by using Verilog hardware description language and synthesized using Xilinx Vivado as well as Xilinx ISE. The design is verified by using I Sim simulator. The total available logic elements are around 250K and are sufficient to implement the design. The design just utilizes only 10% of the available logic elements for the performance of 12MHz, this indicates proposed design is efficient.

Future Scope:

The proposed architecture can be modified by adding few additional instructions and features. The additional features include use of stack pointer, program counter, flag registers, efficient multiply and Accumulate (MAC) unit and floating-point unit. The architecture can be designed for even low power.

The project can be further extended for higher number of bits, in future it can be used as programmable RISC processor for DSP applications by using flash memory interface.

Limitations:

The project can be fully simulated in the EDA tools but it was hard to find a specific FPGA board to implement and synthesize the designed project. The limitations are stated below

- Floating point data type is not supported by the processor.
- All arithmetical operations are performed except Division.
- Area and Floor planning was not a simple task as it required extensive modification in the design.

VI. REFERENCES

[1] Luker, Jarrod D., Prasad, Vinod B. (2001) “RISC system design in a FPGA” MWSCAS 2001, vol 2, pp. 532-536, 2001.

[2] Deepak Kumar, K. Anusudha, (2013) “RISC processor design in Xilinx” International Journal of advance research in Electrical, Electronics and Instrumentation Engineering. Vol 2 pp. 1406-1411

[3] Application note for FarhatMasood, RISC and CISC Computer Architecture, available at http://arxiv.org/ftp/arxiv/papers/1101/1101.5364.pdf, retrieved from: 11/07/2012

[4] RC Cofer and Ben Harding, (2005) “FPGA Soft Processor Design Considerations”, Available at http://www.eetimes.com/General/PrintView/4014791,online published, date 10/12/2005.

[5] W.M Medany, K.Kooheji,“Design and Implementation of a 32bit RISC processor”, Available at http://www.ursi.org/proceedings/procGA08/papers/DP1p2.pdf

[6] Morris, Kevin (2014-02-25). "Xilinx vs. Altera, Calling the Action in the Greatest Semiconductor Rivalry".Electronic Engineering Journal.

[7] Surendra Bajai, “Power and Delay Optimization of Customized 16-Bit Low Power RISC Processor Using VHDL”, IJLTEMAS, vol II, issue X, octomber 2013.

[8] N.Alekya, P.Ganesh Kumar, “DESIGN OF32-BIT RISC CPU BASED ON MIPS ” ,Journal of Global Research in Computer Science,Vol. 2, no.9, September 2011.

[9] Galani Tina G., Riya Saini, R.D.Daruwala, “Design and Implementation of 32 – bit RISC Processor using Xilinx”, IJETEE,vol. 5, issue 1, july 2013.

[10] “Xilinx UG230 Spartan-3E FPGA Stater Kit Board User Guide”, Available at http://www.xlinx.com/support/documentation/boards_and_kits/ug230.pdf

[11] “Vivado Design Suite User Guide:Implimentation-Xilinx” Available at http://www.xlinx.com/support/documentation/sw_manuals/xilinx2018_2/ ug904-vivado-implementation.pdf