Design of a 16-Bit Harvard Structure RISC Processor in Cadence 45nm Technology

Chandran Venkatesan
Department of Electronics and Communication Engineering
KPR Institute of Engineering and Technology
Coimbatore, India
https://orcid.org/0000-0003-2412-9493

Thabsera Sulthana M
Department of Electronics and Communication Engineering
Bannari Amman Institute of technology
Sathyamangalam, Erode, India
thabse94@gmail.com

Sumithra M.G
Department of Electronics and Communication Engineering
KPR Institute of Engineering and Technology
Coimbatore, India
https://orcid.org/0000-0002-0504-2061

Suriya M
Department of Computer Science Engineering
KPR Institute of Engineering and Technology
Coimbatore, India
suriya.m@kpriet.ac.in

***Abstract*— The architecture of a MIPS (Microprocessor without Interlocked Pipeline Stages) based RISC or Reduced Instruction Set of Computers is a type of microprocessor which was designed by Harvard type data path structure to execute high speed using a small set of Instructions. This project explains the design and implementation of a 4-stage pipelining based low power processor. This feature leads to increase the reliability and speed of the system. The pipelining includes fetch, decode, execute and memory read/write operations. Low power was obtained by using clock gating technique. Clock gating is used to eliminate the unwanted clock usage when the module is not used. The main aim of the project is to design a 4-stage pipelined RISC processor starting from RTL to GDSII (Physical Design). The processor was coded by Verilog HDL language and implemented in Cadence Encounter Compiler tool. Calculated area, power, delay and clock gating using Cadence RTL compiler using slow and fast libraries of 45nm technology.**

Keywords*— RISC, MIPS, RTL (Register Transfer Logic), GDSII (Graphic Design System for Information Interchange a Gerber File), Cadence Encounter Compiler, Innovus, 4-stage Pipeline, Physical Design.

I. INTRODUCTION

Based on the Speed and low power the Microprocessors and Microcontrollers are designed under two categories. The one is Reduced Instruction Set of Computers (RISC) and another is Complex Instruction of Computers (CISC). In order to fulfill the hardware designers needs these architectures were developed. It is used in hardware or software specific quantity. CISC architecture is used to minimize or embedding the instruction count in each program. Basically, cost of the memory was decreased. Because the large number of programs needs more memory to store and this increases the cost. The advantage of CISC is each instruction can do a wide range of addressing modes. The operands are used in various locations of an Instruction Set [1]. But the instruction length and execution time leads a complex system.

RISC architecture has high power efficiency which is used in portable applications. The Instruction set length was fixed and the size has been reduced. RISC has more number of General Purpose Registers (GPRs), simplified architecture, reduced addressing modes which makes the faster execution of instructions. Compare with CISC design reaches a net gain performance and silicon consumption was reduced due to the simple design structure. In recent trend applications of VLSI (Very Large Scale Integration) requires the above features of a processor to market via applications [2]. RISC is simplified control unit and load-store architecture. Popularly used RISC processors are MIPS, SuperH, IBM and SPARC etc. Most the semiconductor industries are using the RISC architecture on “System on Chip” (SOC) based designs like ARC, Alpha, Am29000 and ARM which are mostly used in DSP applications[3].

MIPS architecture is mainly depends on the load/store architecture of RISC designs. The MIPS processors are all same type of architectures. But it varies in the implementation stages like pipelining, single or multiple etc. It executes operations in on chip registers faster than memory. Because the access time differs for register compare with the memory location. Due to the operation speed mobile phones, tablets and portable devices are using ARM RISC processor [4].

In recent days, power demand was increasing mainly due to latest growth of electronic products such as portable mobile phones, laptops and other devices needs high speed and low power consumption. The major drawback in portable devices was that takes high power which leads to less battery life and causes failure in silicon parts of the devices. So always there is a trade off in VLSI circuits between systems area, power and delay. This drawback has been reduced in this project during by-pass the pipelining stages, but it causes Dynamic power dissipation [5]. The power dissipation is mainly due to unwanted switching stages or more number of transitions present in the device.

$$P_{\text{avg power}} = P_{\text{dynamic power}} + P_{\text{static power}} \qquad (1)$$

$$P = \tfrac{1}{2} CV^2Nf \qquad (2)$$

The paper is divided as follows. Next section describes the Pipelined Harvard Architecture and Instruction Set of RISC. Section 3 explains the Data path Unit and Control

978-1-5386-9533-3/19/$31.00 ©2019 IEEE

Unit of RISC. Section 4 presents the implemented designs using cadence tool with the suitable results. Section 5 concludes the project and future work.

II. Instruction Set and Pipelined Architecture

The pipelined architecture of RISC is shown in below. To increase the performance of the design 4-stage pipelining is implemented. Clock gating method is an added advantage to reduce power. Because it turn off the clocks when the clock signal is not needed in the design. Mainly this design reduces the Dynamic power dissipation. For example, it may load up to four clock cycles so simultaneously do another process before finish the previous process [6]. Every output of the pipelining stage is the next state input. The microinstructions performed in this design were separated.

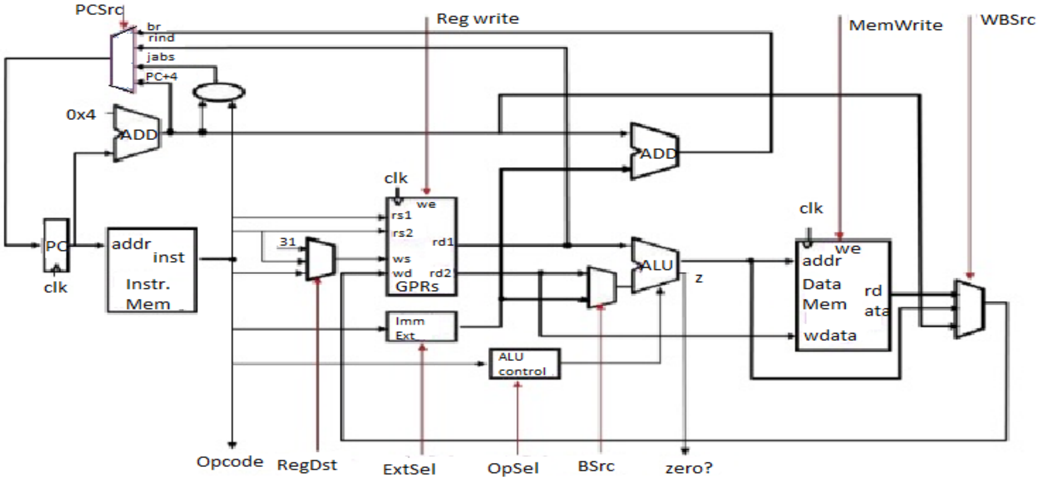


Fig.1. Pipelined RISC Architecture

A. Architecture of RISC

The RISC CPU architecture is a 16-bit 4-stage pipelined Instruction format microprocessor. Because of the load/store architecture operations only performed on registers rather than memory locations. It has Harvard architecture model to speed up the process. The Instruction set includes Jump and load/store type of instructions. Halt instruction is the terminated line of the Instruction memory and also the data memory of the processor.

B. Logical Blocks

Fig.1 explains the block diagram of 16-bit RISC processor. The architecture consists of five blocks. It includes Program Counter (PC), Arithmetic Logical Unit (ALU), Data path Unit, Control Unit and Memory Unit. The data path unit and control unit explained as follows. not revise any of the current designations.

C. Program Counter

A latch of 16-bits consists of the memory address from where the next instruction in machine language will fetch by the processor. It implements two types of operations. One is incrementing and another one is loading. The PC is the second largest sub-block in complexity. First sub-block is control unit of the processor. Most of the instructions in PC are incremented by follow the instruction set. Maybe a simple adder also used for implemented. But it increases the hardware structure and this leads to power dissipation in the circuit. Hence, this design consists of data memory only for load/store architecture [7].

D. Arithmetic and Logical Unit

The arithmetic and logical unit executes arithmetic and logical operations and also such as rotate and shift bit operations. The improved ALU consists of shift, arithmetic and rotate are the three sub-modules. It performs add operation and produces zero flag and sign flag depends on

the result shown. Shift module is necessary for Digital Signal Processing applications. It performs instructions such as shift and rotate operations. The ALU Control Unit is mandatory to give signals to ALU which operation is will be executed.

ALU Control				
AL U Op	Opcode (Hex)	ALU Cnt	ALU Operation	Instruction
10	xxxx	000	ADD	LW,SW
01	xxxx	001	SUB	BEQ,BNE
00	0002	000	ADD	D-type: ADD
00	0003	001	SUB	D-type: SUB
00	0004	010	INVERT	D-type: INVERT
00	0005	011	LSL	D-type: LSL
00	0006	100	LSR	D-type: LSR
00	0007	101	AND	D-type: AND
00	0008	110	OR	D-type: OR
00	0009	111	SLT	D-type: SLT

Fig.2. ALU Control Design

E. Data/Instruction Memory

Instruction Memory based on the RISC processor includes Instruction Set, Instruction format and pipelining stages.

More than 30 instructions are present in the Instruction Set.

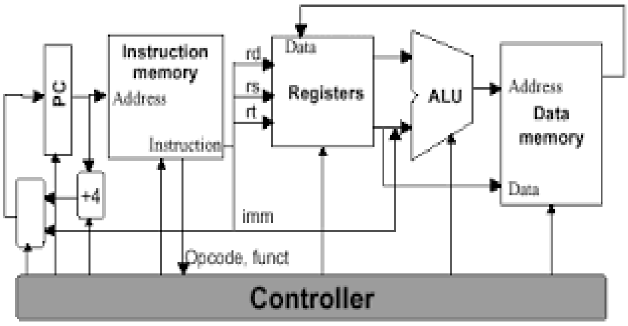


Fig.3. Block Diagram of Instruction Memory

The Instruction Format is shown in below fig.4.

Memory Access: Load

Op	Rs1	ws	Offset
4	3	3	6

Memory Access: Store

Op	Rs1	Rs2	Offset
4	3	3	6

Data Processing:

Op	Rs1	Rs2	ws	Offset
4	3	3	3	3

Branch:

Op	Rs1	Rs2	Offset
4	3	3	6

Jump:

Op	Offset
4	12

OP	
0000	Load Word
0001	Store Word
0002	Add
0003	Subtract
0004	Invert
0005	Logical Shift Left
0006	Logical Shift Right
0007	Bitwise AND
0008	Bitwise OR
0009	Set on less than
0010	Hamming Distance
0011	Branch on Equal
0012	Branch on NOT Equal
0013	Jump

Fig.4. Instruction Format of RISC

This section includes three Instruction Formats.

- Register Format (R-type)
- Immediate Format (I-type)
- Branch type Format (J-type)

Because various operations use different instructions and concluding that different formats are needed. The pipelining stage includes fetch, decode, execute and memory read/write operations. In a single path, we can do multiple operations by using pipelining [8].

III. Datapath Unit and Control Unit

Datapath Unit and Control Unit are the major sub-modules of the RISC processor.

A. Datapath Unit

Data path unit instantiates ALU, ALU Control Unit, Data Memory, GPRs (General Purpose Registers) and Instruction memory. It generates opcode as a output and this is an input to the control unit to perform operations. Registers file includes each 16-bit capable of 8 general purpose registers. It addressed using a 3-bit identifier for source and end. GPRs having 3-bit address ranges from 000 to 111[9]. The load/store instruction is mainly used to implement load the instructions and store the retrieve values are back to Data Memory to get the resultant outputs back from the design processor. In this unit performs read and write operations of the processor and generates output as a opcode to control unit.

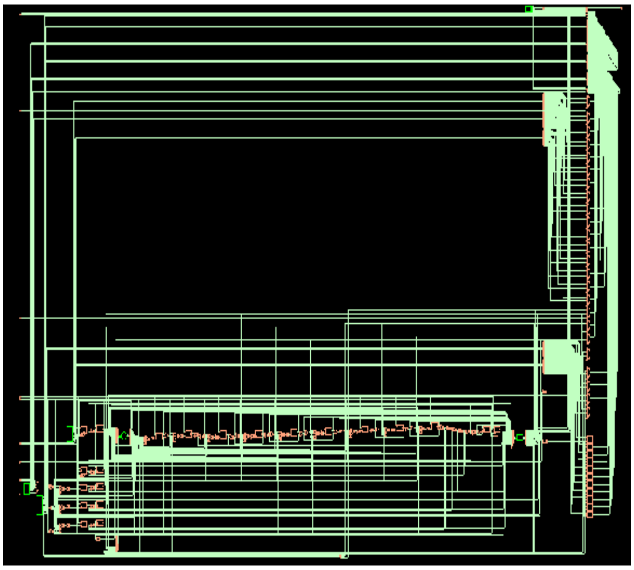


Fig.5. RTL structure of Data path Unit

The RTL structure of the data path unit is shown in fig.5

B. Control Unit

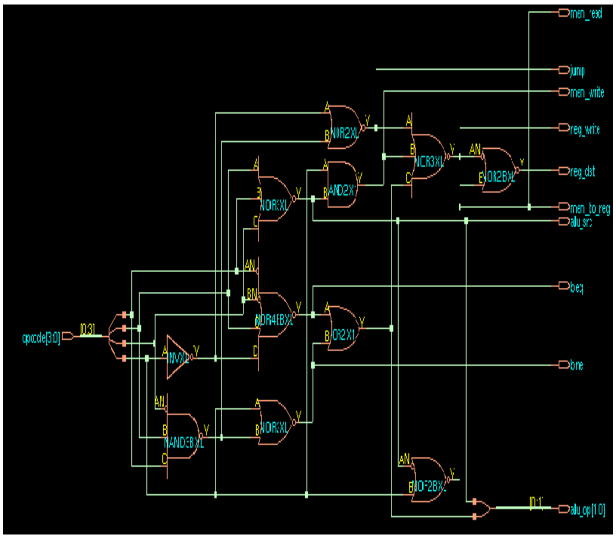


Fig.6. RTL structure of Control Unit

The RTL structure of control unit is shown in fig.6. It gets input as opcode from Data path Unit. Based on the opcode it executes operations. It passes the signals from the memory register to identify which operations should be performed. Control unit is the first sub-block of the RISC processor. The memory register passes the output is written in the MEM/WB register add with the control of WB [10].

The write back module should take responsible for write the data that which computes from out of the EX/MEM register and write it to the another one register in the register file. The control executes all opcode instructions from the Data path Unit and also a default memory is there for the unavailable data output. The next section includes the simulation results.

IV. Schematic Design and Simulation Results

The verilog code for 16-bit MIPS based RISC processor was compiled using Cadence RTL compiler and Simvision tool is used to check the simulated outputs. Schematic and Simulated waveform is shown in below fig.7, fig.8 and fig.9

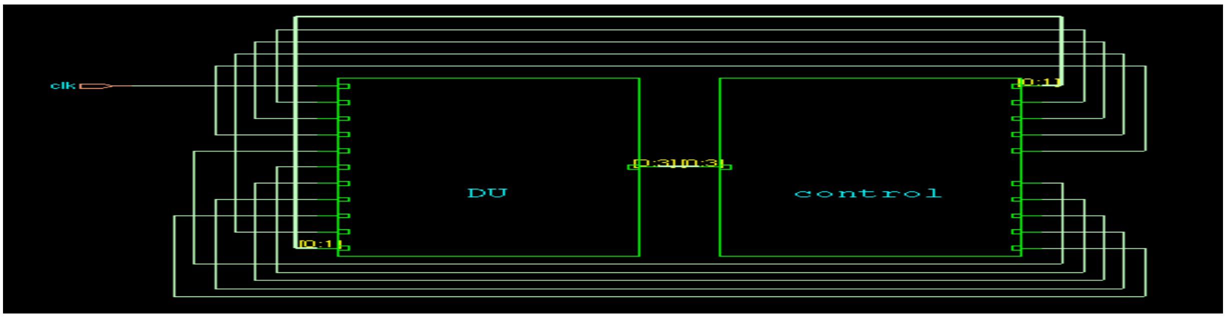


Fig.7. Block Diagram of RISC

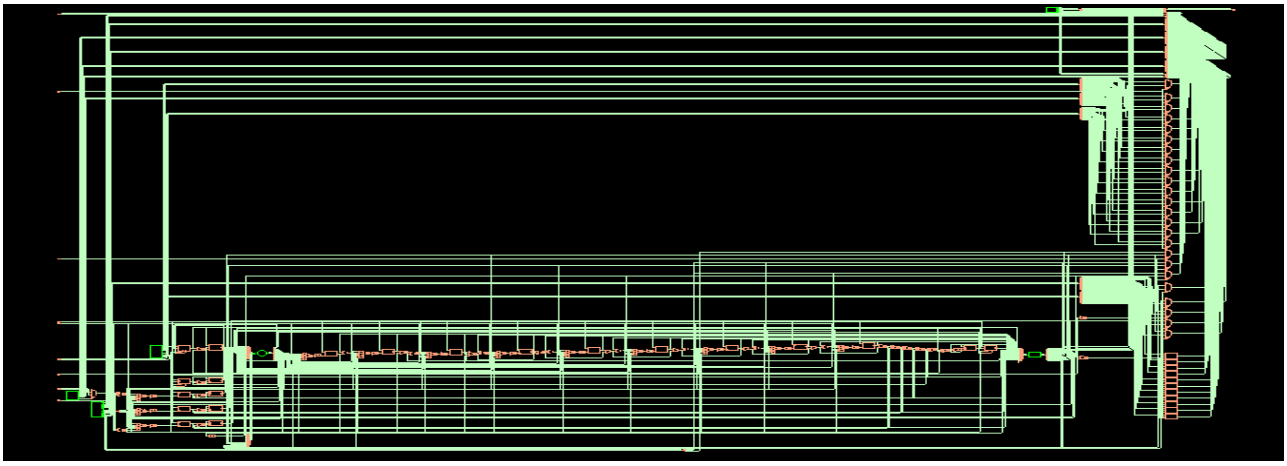
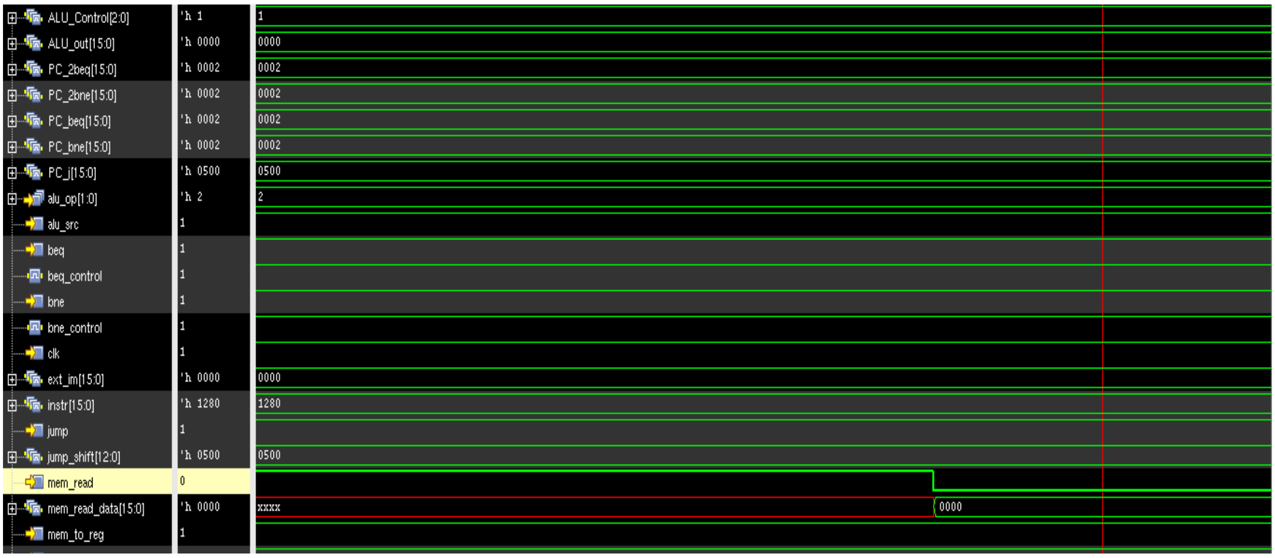


Fig.8. Schematic structure of 16-bit RISC



```
ALU_Control[2:0]     'h 1      1
ALU_out[15:0]        'h 0000   0000
PC_2beq[15:0]        'h 0002   0002
PC_2bne[15:0]        'h 0002   0002
PC_beq[15:0]         'h 0002   0002
PC_bne[15:0]         'h 0002   0002
PC_j[15:0]           'h 0500   0500
alu_op[1:0]          'h 2      2
alu_src              1
beq                  1
beq_control          1
bne                  1
bne_control          1
clk                  1
ext_im[15:0]         'h 0000   0000
instr[15:0]          'h 1280   1280
jump                 1
jump_shift[12:0]     'h 0500   0500
mem_read             0
mem_read_data[15:0]  'h 0000   xxxx    0000
mem_to_reg           1
```

Fig.9. Output Waveform of RISC Design

The RISC processor is synthesized using Cadence RTL compiler with slow libraries of Cadence 45nm technology. The results of area, power and delay is shown in Table 1 and Fig.10.

TABLE I AREA, POWER AND DELAY OF RISC

Instance	Cells	No. of Clocks	Area (um^2)	Power (nW)	Delay (ps)	Freq (MHz)	Library
RISC	1271	4	15020	1290751.491	5000	200	slow
Control Unit	11	0	50	1007.331	498	2080	slow
Instruction Memory	177	0	995	5767.010	1179	848	slow
ALU	401	0	2223	1035.128	501	1996	slow
Data Memory	233	4	3183	565173.780	4932	202	slow
GPRs	308	4	3929	602329.481	4168	239	slow
Data path Unit	129	4	1075	116380.729	4123	242	slow

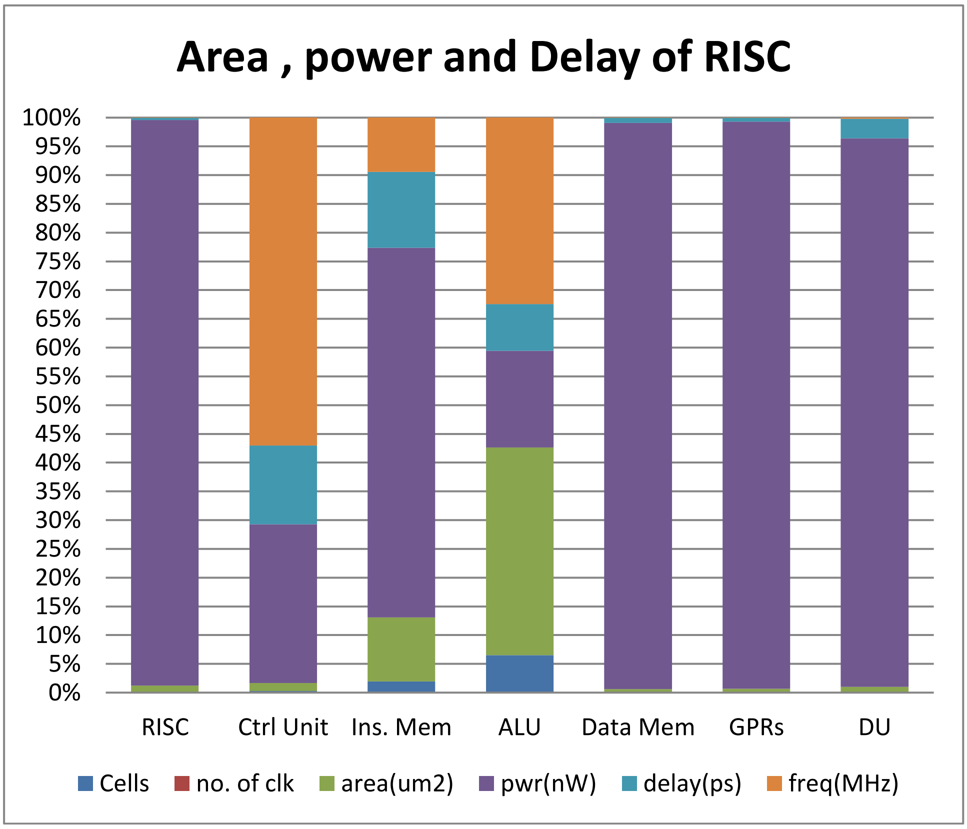


Fig.10 Area, Power and Delay of RISC

In pre-layout steps the processor netlist design was generated. The synthesized netlist file of verilog and corresponding constraints design file (.sdc) are imported in the Cadence SOC Encounter window to generate design layout from automated standard cells, pre CTS, post CTS, placement and routing. After write the sdf file nano route was generated. Save the netlist and Encounter design. This is the final step of the layout design shown in fig.11.

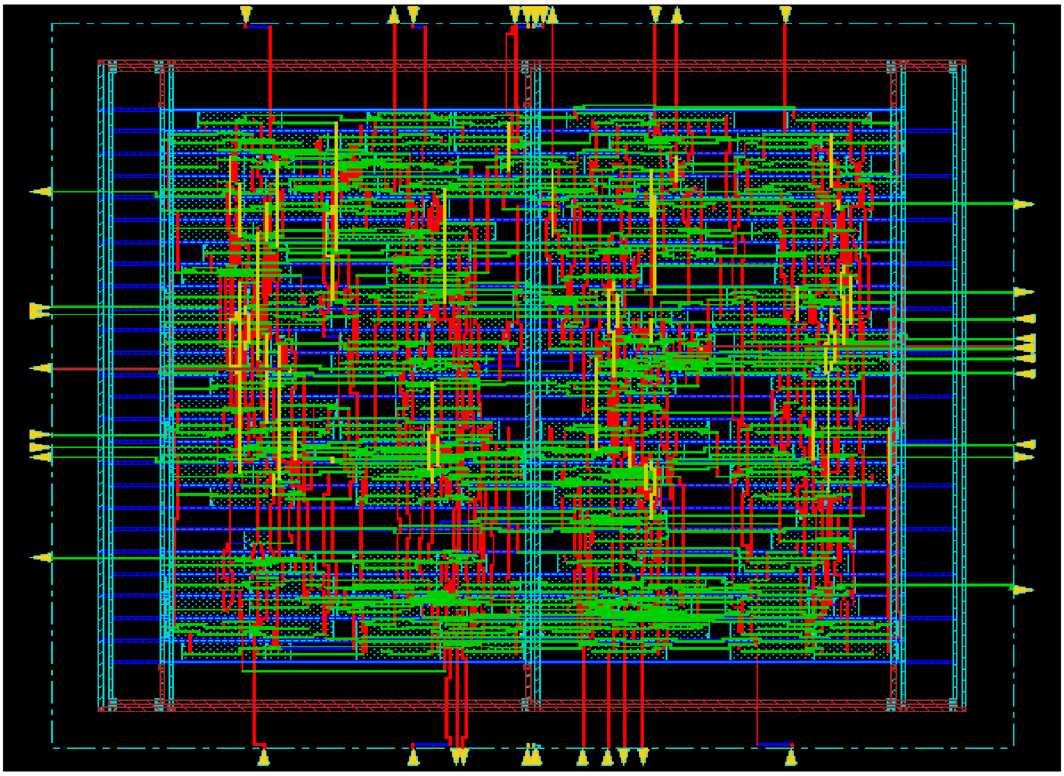


Fig.11. Layout of 16-bit MIPS based RISC Processor

CONCLUSION AND FUTURE SCOPE

A 16-bit RISC Processor has been implemented with Harvard architecture and 4-stage pipelining structure in one clock cycle. This design can be used for portable devices such laptops, mobiles and tablets etc., The RISC architecture is simulated and synthesized using Cadence RTL Compiler. The Layout is implemented in Cadence SOC Encounter. In future, this design bits may be increased upto 64-bits. We can compare those RISC structures in various technology libraries like slow, fast and typical includes different technologies so that can improve the area, power and delay of the designs.

REFERENCES

1. Manjula H R and Rekha S, "64-Bit RISC Processor Design using Verilog," in International Journal of Ethics in Engineering & Management Education, vol.2, Mar. 2015, pp. 4-6.
2. Dr.Nirmal Kumar K R, Chandran V, Valamathi R S, "Bitstream Compression for High Speed Embedded Systems Using Separated Split LUTs", in Journal of Computational and Theoretical Nanoscience 15(special),pp.1-9
3. Indu M and Arun Kumar M, "Design of Low Power Pipelined RISC Processor, "International Journal of Advanced Research in Electrical,Electronics and Instrumentation Engineering, vol.2, August 2013, pp. 3747–3756.
4.
5. Chandran V, Elakkiya B, "Energy Efficient and High-Speed Approximate Multiplier Using Rounding Technique", Journal of VLSI Design and Signal Processing 3(2,3).
6. Samiappa Sakthikumaran, Salivahanan S and Kanchana Baaskaran V S, "16-Bit RISC Processor Design for Convolution Applications," in IEEE- International Conference on Recent Trends in Information Technology, Jun. 2011 , pp. 394–397.
7. Chandran V, Ali K S, Gnanaprakash V, "Energy Efficient and High-Speed Rounding-Based Approximate Multiplier".
8. Mohit N. Topiwala and Saraswathi N, "Implementation of a 32-Bit MIPS based RISC Processor using Cadence," in IEEE International Conference on Advanced Communication Control and Computing Technologies(ICACCCT), 2014, pp. 979–983.
9. Mrudul S. Ghaturle and Prof. Kadam R D,"Review Paper on 32-Bit RISC Processor with Floating Point Arithmetic," in International Research Journal of Engineering and Technology, vol.4,Feb. 2017, pp. 1083–1086.
10. Rajesh Kumar B, Ravisaketh and Santha Kumar, "Implementation of a 16-Bit RISC Processor for Convolution Application," in Advanced in Electronic and Electric Engineering, vol.4,2014, pp.441–446.
11. Ramandeep Kaur and Anuj, "8-Bit RISC Processor using Verilog HDL," in Anuj et al Int. Journal of Engineering Research and Applications, vol.4, Mar. 2014, pp.417-422.