International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 02 | Feb 2019 www.irjet.net p-ISSN: 2395-0072

Physical Design Implementation of Single Core 32 Bit RISC Processor on 28nm Technology

Feroz Ahmed Choudhary[1], Amay Shiva Naik[2], Dr. Rajashekhar C Biradar[3]

[1,2,3]*Dept. of Electronics and communication Engineering, REVA University, Bengaluru, India*

Abstract – Physical Design implementation means the layout assembling and connectivity of digital logic gates as per the design input file (called as netlist) of an Integrated chip say a processor, by meeting the design specifications like timing, power and area. Here in this design we have used the inputs given by the synthesis team and further physical design flow has been carried out in a proper physical design flow i.e. from import design, floor plan further placement till signoff using the Physical design tools. At each stage checks are done such as timing (hold & setup), quality of report, congestion, routing etc.

Key Words: **Import Design, Floor Plan, Placement, Place opt, Clock Tree Synthesis, Opt Clock Tree Synthesis, Routing, opt Routing, Signoff.**

1.INTRODUCTION

In physical design implementation of 32Bit RISC Processor here we started with Design netlist which contains information of the cells used, their interconnections, area, and other details, this design netlist is synthesized means constraints were applied to ensure the design meets the functionality and speed. Next step was floor planning in this die and core area are created with respect to aspect ratio and utilization factor and then based on the macros present we placed the macros in smart way so that in further stages there are no congestions. Then partitioning was done to divide the chip into small blocks after that power planning was done further before doing placement, all wire load models(WLM) were removed as placement used RC values from virtual route (VR) to calculate timing. Placement was done as pre-placement, in placement and post placement after placement optimization Clock tree synthesis is done to balance the skew and minimize the insertion delay after CTS Routing was carried out which is divided into two global and detailed routing as this is done further we have to do search and repair and cells later have to do ECO checks such as timing ECO, functional ECO, metal ECO, power ECO and Clock ECO. As all this physical design steps are done we have to do physical verification as it checks the correctness of the generated layout design. This includes DRC (design rule check), LVS (layout vs schematic), ARC (antenna rule checking), ERC (electrical rule checking). At last we perceive Graphic Database System (GDS II) file, it is database file format which is industry standard for data substitute of IC layout artwork. It is binary file representing planar geometric shapes, text labels and other information about layout in hierarchical form.

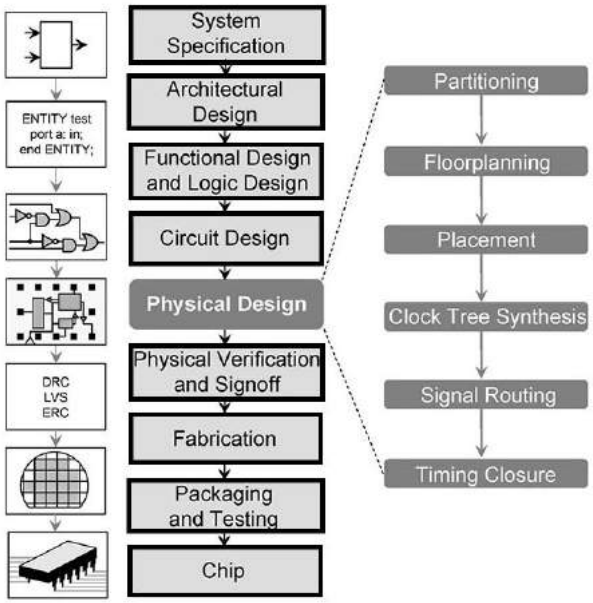

Fig -1: Physical Design Flow shown in ASIC Flow

1.1 Design Specifications

- Technology node: 28nm
- Layers: 9 Routing metal Layers
- Macro counts: 40
- No. of Clocks: 6 Clocks
- Total Cells: 0.52 Million
- Target clock Frequency: 416 MHz with 6 Clocks in Design
- Design Corners: 2
- Voltage Domain: 1

2. IMPORT DESIGN

Import design is the fundamental rung in Physical Design. In this stage all the requisite inputs & essential references are interpreted into the tool plus prime checks are done i.e. design, technology consistency.

2.1 Inputs Required in Import Design

Inputs Required Are Gate level netlist, Logical (Timing) & Physical views of standard cells & all other IPs used in the design, Timing constraints (SDC), Power Intent (UPF / CPF), FP DEF & Scan DEF, Technology file & RC Co-efficient files.

2.2 Key checks to Qualify Import Design

Here Checks of errors and warning have been done while reading netlist, timing constraints, UPF/CPF, black boxes, MV design (equivalent to LP checks) and assign & tri statements (Usually its checked & fixed after Synthesis)

2.3 Sanity Checks

Sanity checks essentially check the condition of netlist in terminology of timing, it also consists of read-through the issues allied to library files, timing constraints, IOs and optimization directives. Sanity checks which have been performed are as

1. Library checks: to check the missing cell information, missing pin information & if any Duplicate cells present or not.

2. Design checks: to check the Inputs with floating pins, nets with tristate drivers, nets with multiple drivers, Combinational loops, Empty modules, Assign statements
3. Constraint checks: to check all flops are clocked or not, unconstraint paths should not be present & Input and output delays.

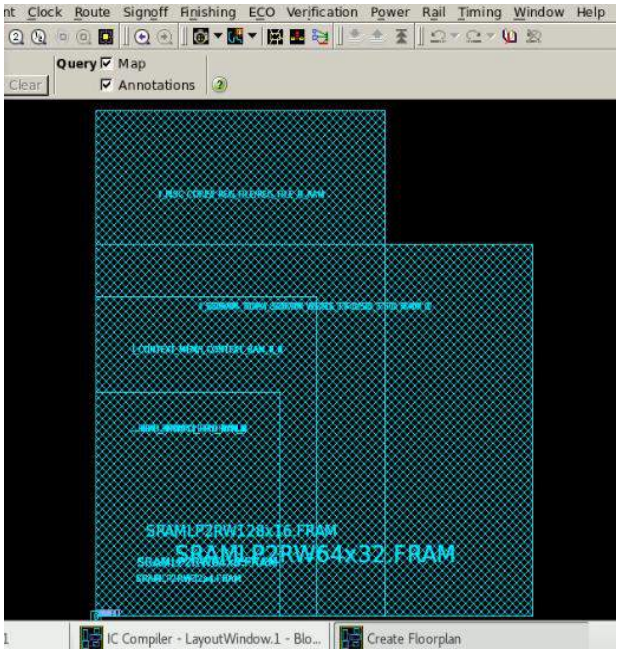

Fig -2: Imported Design

3. FLOORPLAN

This is the first major step in getting the layout done. Here floor plan determines your chip quality. At this step, we define the size of your chip/block, allocates power routing resources, place the hard macros, and reserve space for standard cells. A Satisfactory floorplan can make implementation method (place, CTS, route & timing closure) cake walk. On similar lines a bad floorplan can create all kind issues in the design (congestion, timing, noise, IR, routing issues). A bad floorplan will propel up the area, power & affects reliability, life of the IC and also it can increase overall IC cost (more effort to closure, more LVTs/ULVTs)

3.1 Deciding the Utilization factor & aspect ratio

a. Utilization factor decides the size of the block meaning suppose out of 100 percent we give utilization factor of 70 percent it means 70 percent will be used for placing macros and instances in the design and remaining 30 percent will be used for routing purpose to connect all the macros and instances in the design.

$$hip\ utilization = \frac{Area\ of [std\ cell + macro + (pad, padfiller, cornerpad)]}{Area\ of\ chip}$$

b. Aspect ratio gives shape of the block i.e. width/height example: say height of 1.4 and width of 1.0 is given hence it will form a rectangle shape design.

$$Aspect\ Ratio = \frac{W}{H}\ \frac{Horizontal\ Routing\ Resources}{Vertical\ Routing\ Resources}$$

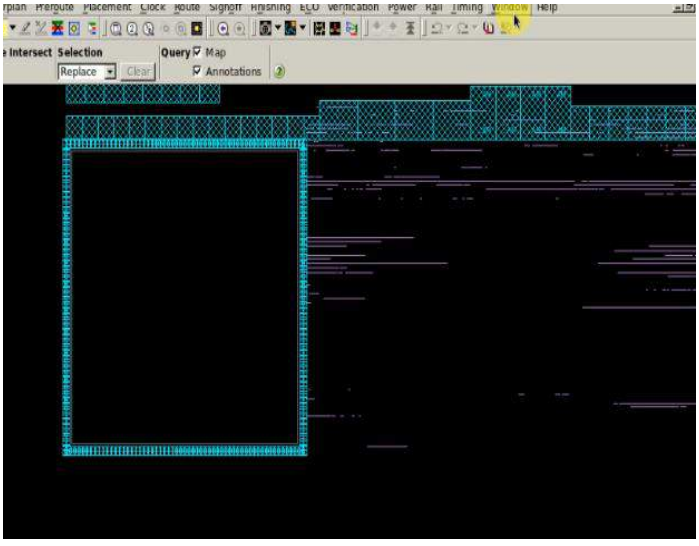

Fig -2: Created Floorplan with aspect ratio and utilization factor

c. After utilization and aspect ratio we go for pin placement. In pin placement we have to place pins legally

3.2 Macros Placement

Method to place macros which have been followed in this design as follows**.**

i. place macros around chip periphery. If you don't have reasonable rationale to place the macro inside the core area, then place macros around the chip periphery. Placing a macro inside the core can invite serious consequence during routing due to a lot of detour routing, because macros are equal to a large obstacle for routing. Another advantage to placing the hard macros around the core periphery is it's easier to supply power to them, and reduces the change of IR drop problems to macros consuming high amounts of power.

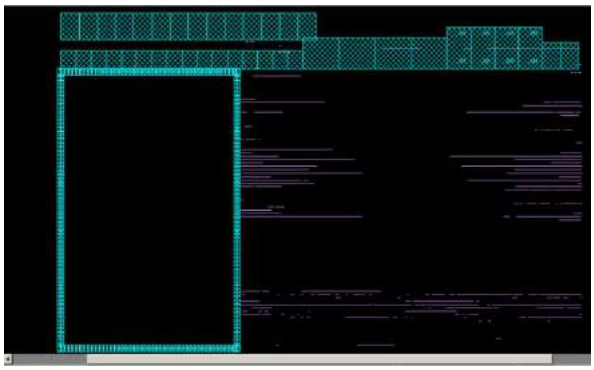

Fig -4: Before macro placement

ii. Consider connections to fixed cells when placing macros. When you decide macro position, you have to pay attention to connections to fixed elements such as I/O and preplaced macros. Place macros near their associate fixed element. Check connections by displaying flight lines in the GUI.

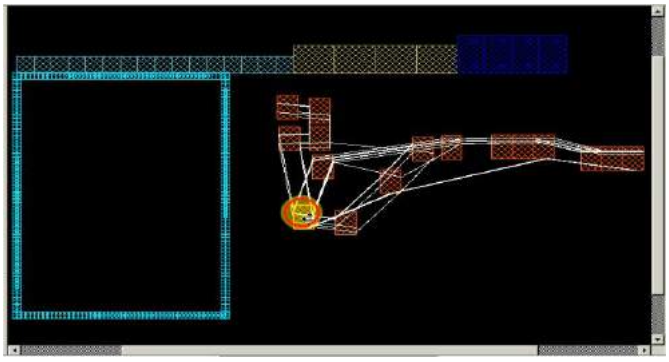

Fig -5: Fly line analysis

iii. Orient macros to minimize distance between pins. When you decide the orientation of macros, you also have to take account of pins positions and their connections.

iv. Reserve enough room around macros. For regular net routing and power grid, you have to reserve enough routing space around macros. In this case estimating routing resources with precision is very important. Use the congestion map from trial Route to identify hot spots between macros and adjust their placement as needed. The space between macros is given by equation

$$macro\ spacing = \frac{number\ of\ pins * pitch}{total\ number\ of\ metal\ layers}$$

v. Reduce open fields as much as possible. Except for reserved routing resources, remove dead space to increase the area for random logic. Choosing different aspect ratio (if that option is available) can eliminate open fields.

vi. Reserve space for power grid. The number of power routes required can change based on power consumption. You have to estimate the power consumption and reserve enough room for the power grid. If you underestimate the space required for power routing, you can encounter routing problems 5. After macro placement we will place physical cells like endcap and well tap cells

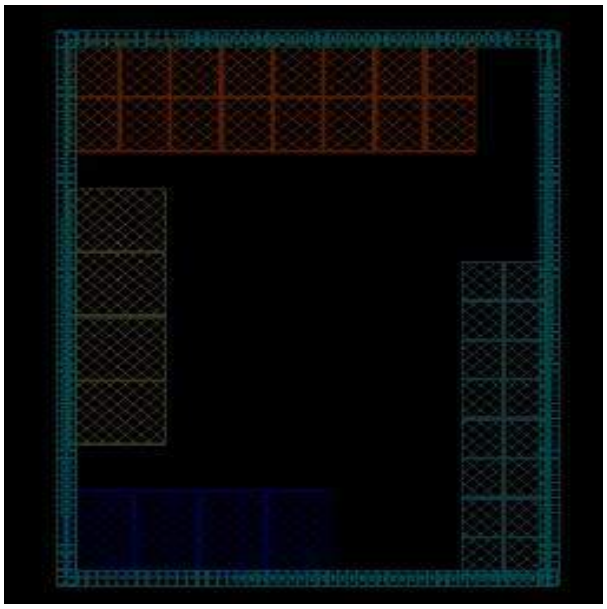

Fig -6: Macros Placed

Further after placing the macro we have added keep out margin to macros, physical cells and cut rows in the design.

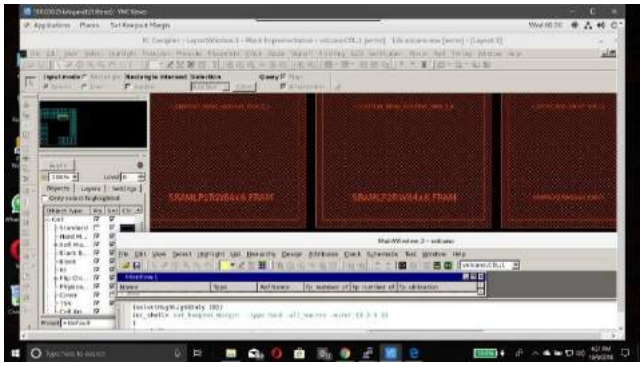

Fig -7: added keep out margin on macros

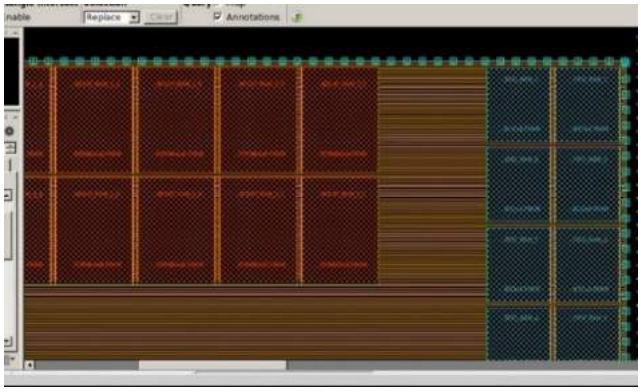

Fig -8: Cut rows applied in the design

4. PLACEMENT

Placement is the process of placing standard cells in the rows created at Floorplanning stage. The goal is to decrease the total area and interconnects cost. The trait of routing is highly determined by the placement.

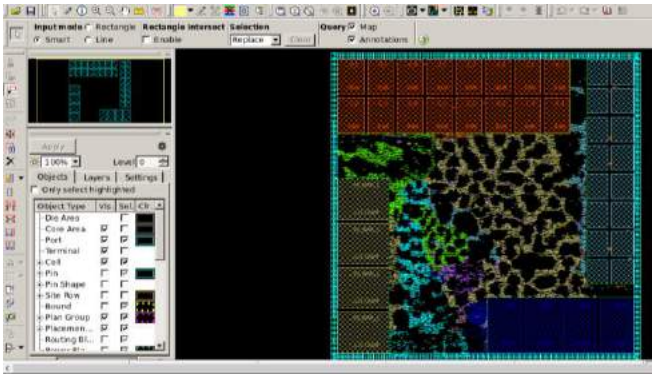

Fig -9: Standard cells placed in the core area

One way to overcome the complication concern is to perform placement in several controllable steps as discussed below.

Global Placement: Global placement aims at generating a coarse placement solution that may violate some placement constrains (e.g., there may be overlaps among modules) while maintaining a global view of whole netlist. here the Objective is to lessen the interconnect wire lengths.

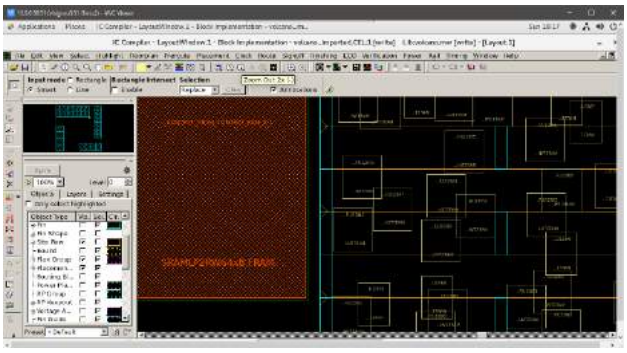

Fig -10: coarse placement of standard cell in the design

Legalization: Legalization makes the rough solution from global placement legal (i.e., no placement constraint violation) by moving modules around locally.

Detailed Placement: Detailed placement further improves the legalized placement solution in an iterative method by rearranging a small group of modules in a local region while keeping all other modules fixed. Here the Objective is to meet design constraints such as Timing/Congestion and to conclude standard cell placement.

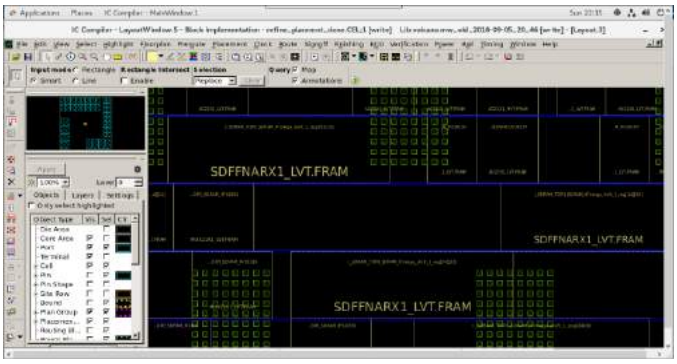

Fig -11: Detailed and legalized placement of standard cell

5. CLOCK TREE SYNTHESIS

Clock tree synthesis is a course of action which makes sure that the clock gets circulated evenly to all the sequential elements in a design.

CTS is the procedure of insertion of buffers or inverters along the clock paths of ASIC design in order to achieve zero/minimum skew or balanced skew. The goal of CTS is to reduce skew and insertion delay. Apart from these, useful skew is also added in the design by way of buffers and inverters.

5.1 Pre Clock Tree Synthesis

Setup slack was meeting whereas there where hold violation present in the design as can be seen in the figure below.

```
File  Edit  View  Search  Terminal  Help
I_PCI_TOP/preq_n  (PCI_TOP)                0.00       0.28 r
U539/Y (NBUFFX32_HVT)                     0.10 *     0.37 r
I_PCI_TOP/IN4 (PCI_TOP)                   0.00       0.37 r
I_PCI_TOP/U645/Y (AND2X2_HVT)             0.20 *     0.58 r
I_PCI_TOP/pack_n (PCI_TOP)                0.00       0.58 r
U612/Y (INVX16_HVT)                       0.09 *     0.67 f
U1059/Y (INVX32_HVT)                      0.10 *     0.77 r
pack_n (out)                              0.02 *     0.79 r
data arrival time                                    0.79

clock v_PCI_CLK (rise edge)               7.50       7.50
clock network delay (ideal)               0.50       8.00
output external delay                    -3.00       5.00
data required time                                   5.00
-------------------------------------------------------------
data required time                                   5.00
data arrival time                                   -0.79
-------------------------------------------------------------
slack (MET)                                          4.21


  Point                                                   Incr       Path
  ------------------------------------------------------------------------
  clock PCI_CLK (rise edge)                               0.00       0.00
  clock network delay (ideal)                             0.00       0.00
  I_PCI_TOP/I_PCI_CORE/pad_out_buf_reg[21]/CLK (SDFFARX1_LVT)
                                                          0.00       0.00 r
  I_PCI_TOP/I_PCI_CORE/pad_out_buf_reg[21]/Q (SDFFARX1_LVT)
                                                          0.18       0.18 f
  I_PCI_TOP/pad_out[21] (PCI_TOP)                         0.00       0.18 f
  U1086/Y (INVX4_HVT)                                     0.06 *     0.24 r
  U1085/Y (INVX32_HVT)                                    0.07 *     0.31 f
  pad_out[21] (out)                                       0.04 *     0.34 f
  data arrival time                                                  0.34

  clock v_PCI_CLK (rise edge)                             0.00       0.00
  clock network delay (ideal)                             0.50       0.50
  output external delay                                   1.00       1.50
  data required time                                                 1.50
  ------------------------------------------------------------------------
  data required time                                                 1.50
  data arrival time                                                 -0.34
  ------------------------------------------------------------------------
  slack (VIOLATED)                                                  -1.16
```

Fig -13: setup time met & hold time violated

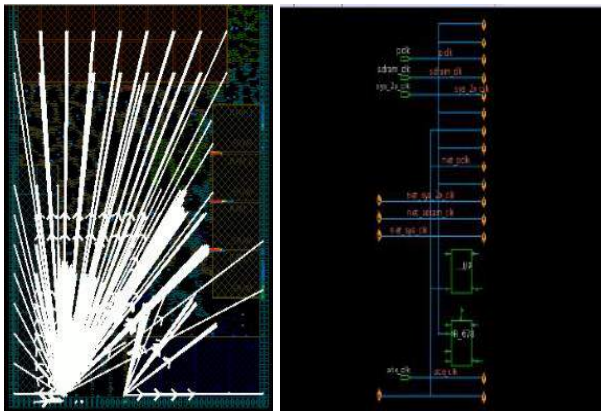

Fig -14: clock propagated randomly & clock schematic before post CTS

5.2 Post Clock Tree Synthesis

After running post CTS scripts provided by the top level we get well-structured clock tree and adding proper buffers and inverters to solve the hold violation, results shown below.

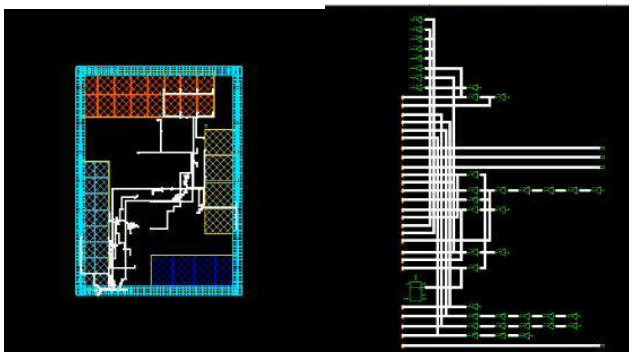

Fig -15: *HTree clock and its schematic with added buffer and inverter to remove hold violation*

As the post CTS was competed the setup and hold timings where meeting, results shown below

```
U537/Y (NBUFFX16_HVT)                     0.21 *     1.25 f
I_PCI_TOP/IN4 (PCI_TOP)                   0.00       1.25 f
I_PCI_TOP/U645/Y (AND2X1_HVT)             0.17 *     1.42 f
I_PCI_TOP/U1166/Y (DELLN2X2_HVT)          0.51 *     1.93 f
I_PCI_TOP/pack_n (PCI_TOP)                0.00       1.93 f
U630/Y (INVX16_HVT)                       0.08 *     2.00 r
U1059/Y (INVX32_HVT)                      0.10 *     2.10 f
pack_n (out)                              0.02 *     2.12 f
data arrival time                                    2.12

clock v_PCI_CLK (rise edge)               7.50       7.50
clock network delay (ideal)               0.50       8.00
output external delay                    -3.00       5.00
data required time                                   5.00
-------------------------------------------------------------
data required time                                   5.00
data arrival time                                   -2.12
-------------------------------------------------------------
slack (MET)                                          2.88


-------------------------------------------------------------------------
clock PCI_CLK (rise edge)                               0.00       0.00
clock network delay (propagated)                        0.57       0.57
I_PCI_TOP/I_PCI_CORE/d_out_p_bus_reg[6]/CLK (SDFFARX1_LVT)
                                                        0.00       0.57 r
I_PCI_TOP/I_PCI_CORE/d_out_p_bus_reg[6]/Q (SDFFARX1_LVT)
                                                        0.15       0.72 r
I_PCI_TOP/ptrdy_n_out (PCI_TOP)                         0.00       0.72 r
U128/Y (NBUFFX2_HVT)                                    0.07 *     0.78 r
U129/Y (NBUFFX32_HVT)                                   0.10 *     0.88 r
U917/Y (DELLN2X2_HVT)                                   0.51 *     1.39 r
U130/Y (INVX4_HVT)                                      0.06 *     1.46 f
U997/Y (INVX32_HVT)                                     0.06 *     1.51 r
ptrdy_n_out (out)                                       0.00 *     1.51 r
data arrival time                                                  1.51

clock v_PCI_CLK (rise edge)                             0.00       0.00
clock network delay (ideal)                             0.50       0.50
output external delay                                   1.00       1.50
data required time                                                 1.50
-------------------------------------------------------------------------
data required time                                                 1.50
data arrival time                                                 -1.51
-------------------------------------------------------------------------
slack (MET)                                                        0.01
```

Fig -14: *setup and hold timings met*

6. ROUTING AND TIME CLOSURE

In routing stage, metal and vias are used to create the electrical connection in layout so as to complete all connections defined by netlist this are carried out in two

types of routing one is global routing and the other is the detailed routing. In case of the global routing the interconnections between the standard cells connects loosely, while in case of the detailed routing the detouring of connections are reduced. During the placement of the standard cells also trial route is done to estimate the static timing analysis weather the design met the timing *constraints*[10]. in detail routing router runs search and repair routing here it locates shorts and opens and spacing violations so, it reroutes the effected area to eliminate violations. As the fixes are done and timings are met this procedure nothing but called as enginnering change order(ECO) hence we conclude the physical design flow by handing the GDS II file to the further ASIC flow stage.

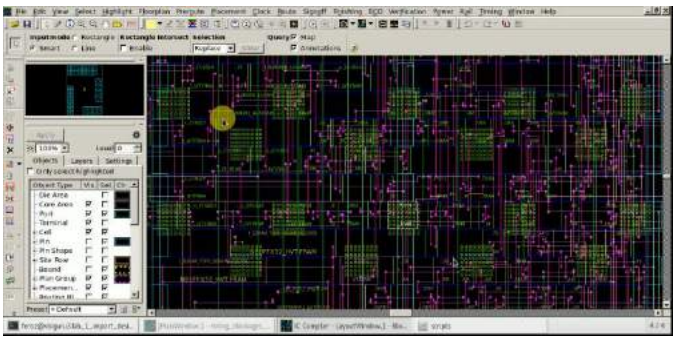

Fig -16: *Routing stage*

7. CONCLUSION

The physical design implementation of the RISC processor is done by achieving the each and every quality check during the floorplan, placement, clock tree synthesis, routing. As we did CTS to meet the skew, duty cycle, latency, pulse width and the clock tree power compared to the statistical analysis[8]. An Engineering Change Order (ECO) is done to ensure the design to meet the required specifications and timing.

REFERENCES

1. Devaraconda Dinesh and R. Manoj Kumar Physical Design Implementation of 16 Bit Risc Processor Vol 9(36), DOI: 10.17485/ijst/2016/v9i36/102911, September 2016
2. Ibrahim D.16-bit microprogrammable microcomputer with writable control store". IEEE Trans. Computers. 2011 Nov; 39(11):1385–90.
3. Kahng A, Lienig J, Markov I, Hu J. VLSI physical design: From graph partitioning to timing closure. Springer, 2011, 27.
4. Mehrotra, Alok, Van Ginneken, Lukas P, Trivedi, Yatin. Design flow and methodology for 50M gate ASIC, IEEE Conference Publications, 2003.
5. Sharda PK, Jain GP. Design and implementation of 5 Stages pipelined architecture in 32 Bit RISC Processor. International Journal of Emerging Technology and Advanced Engineering. IJETAE, 2012 April; 2(4):340–6.
6. neeraj Jain. VLSI design and optimized implementation of a risc processor using xilinx tool. International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE). 2012 Dec; 1(10):52–6.
7. Reaz MBI, Amin N , Ibrahimy MI , Mohd-Yasin F, Mohammad A. Zero skew clock routing for fast clock tree generation. Canadian Conference on Electrical and Computer Engineering. (CCECE). 2008 May; 4–7:23–8.
8. 10. Tsai JL, Chen TH and Chen CCP. Zeroskewclock tree optimization with buffer insertion/sizing and wire sizing. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. April 2004; 23(4):565–72.
9. 11. KV Rao, A Angeline, VSK Bhaaskaran. Design of a 16 bit RISC processor. Indian Journal of Science and Technology. August 2015; 8(20):1–7.