Physical Design Implementation of Leon Processor

C. V Hima[1], R. Rajaprabha[2], P. B Prajitha[2]

[1] PG Student, VLSI Design, Sri Shakthi Institute of Eng. and Technology, Coimbatore, India.

[2]Assistant Professor, Department of ECE, Sri Shakthi Institute of Eng. And Technology, Coimbatore, India.

ABSTACT

The design-cycle of VLSI-chips consists of different consecutive steps from high-level synthesis (functional design) to production (packaging).This project aims to implement a physical design flow from netlist to gdsii that starts from floorplan, placement, CTS, routing and ends with physical verification. The main objective of this project is to fix the violations those results in the implementation of Leon processor such as crosstalk, slew violations, congestion and other signal integrity issues. Results show that the proposed steps eliminate these issues and clear the physical verification checks such as DRC, LVS static and IR and antenna design rule. Further the resultant optimized design meets the timing constraints and minimizes area to obtain a design suitable for manufacture.

Key words- Signal Integrity, Crosstalk, CTS, Congestion, Slew violations.

1. INTRODUCTION

The design-cycle of VLSI-chips consists of different consecutive steps from high-level synthesis (functional design) to production (packaging). The physical design is the process of transforming a circuit description into the physical layout, which describes the position of cells and routes for the interconnections between them. The input of a physical design process is the netlist. The main concern in the physical design of VLSI-chips is to find a layout with minimal area, further the total wire length has to be minimized.

Static timing analysis is a method of validating the timing performance of a design by checking all possible paths for timing violations under worst-case conditions. To check a design for violations, Prime Time breaks the design down into a set of timing paths, calculates the signal propagation delay along each path, and checks for violations of timing constraints. It considers the worst possible delay through each logic element, but not the logical operation of the circuit under test. Static timing analysis checks the design only for proper timing, not for correct logical functionality. In static timing analysis, the word static alludes to the fact that this timing analysis is carried out in an input-independent manner. It locates the worst-case delay of the circuit over all possible input combinations. There are huge numbers of logic paths inside a chip of complex design.

To check a design for violations or say to perform STA there are three main steps design is broken down into sets of timing paths, calculates the signal propagation delay along each path and checks for violations of timing constraints inside the design and at the input/output interface. The paths analysed by STA includes data path, clock path, clock gating path and asynchronous path. Delay of the cell depends upon Library setup time, Library delay model, External delay, Cell load characteristic, cell drive characteristic, operating condition (PVT), wire load model, input skew, back annotated delay. Net delay refers to the total time needed to charge or discharge all of the parasitic (Capacitance / Resistance / Inductance) of a given net. So net delay is a function of net resistance, net capacitance and net topology.

If any of this parameter varies, the delay varies accordingly. Few of them are mutually exclusive and in that case it is considered the effect of only one parameter at a time. If that's the case, then for STA, calculated the delay in both the condition and then categorize them in worst (max delay) condition or the best condition (min delay). E.g. - if a cell has different delay for rise edge and fall edge. Then it is sure that in delay calculation we have to use only one value. So as per their value, we can categorize fall and rise delay of the entire cell in the max and min bucket. And finally it comes up with max delay and min delay.

The advantage of STA is that it performs timing analysis on all possible paths (whether they are real or potential false paths).However, it is worth noting that STA is not suitable for all design styles. It has proven efficient only for fully synchronous designs. Since the majority of chip design is synchronous, it has become a mainstay of chip design over the last few decades.

2. RELATED WORKS

Buffered clock trees are often desirable, but added at the expense of complicating the clock design. From [1], skew due to buffer mismatch is minimized by first clustering the clock nodes so that identical buffers can be used at a level, and balancing the higher-order loads of the clusters so that load dependent buffer delays are matched. Interconnect delays within clusters are concurrently balanced too, thereby generating a low-skew buffered clock tree design. While the two techniques we have presented are most effective when used concurrently, they are completely independent of each other. The clustering technique can be used to generate clusters of equal capacitive loading for any clock tree synthesis methodology. Similarly, the delay- and admittance-matching wire sizing technique can be used for constructing any buffered clock tree that uses equally-sized buffers at the same level.

Crosstalk is a well–known phenomenon at all levels of electronic packaging from system level cables through wires on printed circuit boards and multi–chip–modules to chip level routing. It is an effect due to coupling capacitances and inductances between currents in electrical conductors. Crosstalk causes undesired signal noise to be coupled from an active line (aggressor) into a quiet line (victim). Depending on its magnitude, the induced noise onto the victim may influence the timing behavior of the victim signal by increasing its setup time by [2]. It may even cause failure by inducing false pulses or causing false signal levels which may be propagated through the circuit. With increasing integration density and reduced cycle times, these effects become more visible and more destructive, so they need to be handled more carefully. Crosstalk needs to be considered in particular on VLSI chips with sub–micron structures and today's large die sizes. As crosstalk is strictly a local phenomenon it is handled within detailed routing (local routing) rather than in global routing. The final arrangement of all wire segments is determined within detailed routing, where the crosstalk relevant parameters can be extracted. Moreover, the detailed router usually has sufficient freedom for the assignment of wire segments to channels to avoid the most critical coupling configurations. However, detailed routing is typically one of the most CPU time and memory intensive tasks in physical chip design. Therefore, the detailed router is guided by simple geometrical restrictions for crosstalk avoidance rather than by a complete complex electrical wire model.

Power gating has become one of the most widely used circuit design techniques for reducing leakage current. Its concept is very simple, but its application to standard-cell VLSI designs involves many careful considerations. The great complexity of designing a power-gated circuit originates from the side effects of inserting current switches, which have to be resolved by a combination of extra circuitry and customized tools and methodologies. The power is improved by reducing the sizes of switches, cutting transition delays, applying power gating to smaller blocks of circuitry, and reducing the energy dissipated in mode transitions. Power gating has also been combined with other circuit techniques, and these hybrids are also reviewed in [3].

3. PROPOSED SYSTEM

This project aims to implement a physical design flow from netlist to gdsii that starts from floorplan, placement, CTS, routing and ends with physical verification. The main objective of this project is to fix the violations those results in the implementation of Leon processor such as crosstalk, slew violations, congestion and other signal integrity issues. Results show that the proposed steps eliminate these issues and clear the physical verification checks such as DRC, LVC and antenna design rule to obtain a design that is suitable for manufacture. The main focus of this work is to achieve a good floorplan, to get timing clean, congestion free placement, maximum skew and insertion delay targets during CTS, timing fixes and to get a DRC clean, LVS clean design, fixing antenna violations, dynamic and static IR violations.

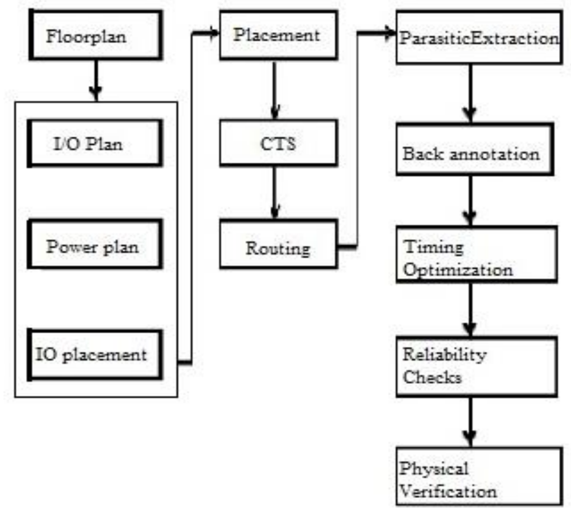


Fig 3.1 Physical design flow of leon processor

3.1 Floorplanning

The first step in the physical design flow is floorplanning. A floorplan of an integrated circuit is a schematic representation of tentative placement of its major functional blocks. Floorplanning is the process of identifying structures that should be placed close together, and allocating space for them in such a manner as to meet the sometimes conflicting goals of available space (cost of the chip), required performance, and the desire to have everything close to everything else. Floorplan is of sliceable and non-sliceable. Floor planning takes into account the macros used in the design, memory, other IP cores and their placement needs, the routing possibilities and also the area of the entire design. Floor planning also decides the IO structure, aspect ratio of the design. A bad floorplan will lead to wastage of die area and routing congestion.

Objectives of floorplan are minimize the die size ,meet the timing requirements, power routing should meet the IR/EM targets, Further good floorplan would reduce the number of iterations during the Design/Timing closure.

3.2 Placement

Before the start of placement optimization all WLM are removed. Placement uses RC values from VR to calculate timing. VR is the shortest Manhattan distance between two pins. Pre-placement Optimization optimizes the netlist before placement, HFNs are collapsed. It can also downsize the cells. In-placement optimization re-optimizes the logic based on VR. This can perform cell sizing, cell moving, cell bypassing, net splitting, gate duplication, buffer insertion, area recovery. Optimization performs iteration of setup fixing, incremental timing and congestion driven placement. Post placement optimization before CTS performs netlist optimization with ideal clocks. It can fix setup, max trans/cap violations. It can do placement optimization based on global routing. It re does HFN synthesis. Post placement optimization after CTS optimizes timing with propagated clock. It tries to preserve clock skew. Objectives of placement includes minimize the all critical net delay,minimize the total estimated interconnect length and minimize the interconnect congestion.

3.3 CTS

The goal of CTS is to minimize skew and insertion delay. Clock is not propagated before CTS.After CTS hold slack should improve. Clock tree begins at .sdc defined clock source and ends at stop pins of flop. There are two types of stop pins known as ignore pins and sync pins. ‘Don’t touch’ circuits and pins in front end (logic synthesis) are treated as ‘ignore’ circuits or pins at back end (physical synthesis). ‘Ignore’ pins are ignored for timing analysis. If clock is divided then separate skew analysis is necessary. First is global skew achieves zero skew between two synchronous pins without considering logic relationship. Second one is local skew achieves zero skew between two synchronous pins while considering logic relationship. If clock is skewed intentionally to improve setup slack then it is known as useful skew.

In CTO clock can be shielded so that noise is not coupled to other signals. But shielding increases area by 12 to 15%. Since the clock signal is global in nature the same metal layer used for power routing is used for clock also. CTO is achieved by buffer sizing, gate sizing, buffer relocation, level adjustment and HFN synthesis. We try to improve setup slack in pre-placement, in placement and post placement optimization before CTS stages while neglecting hold slack. In post placement optimization after CTS hold slack is improved. As a result of CTS lot of buffers are added. Generally for 100k gates around 650 buffers are added.

3.4 ROUTING

After placement, the routing process determines the precise paths for nets on the chip layout to interconnect the pins on the circuit blocks or pads at the chip boundary. These precise paths of nets must satisfy the design rules provided by chip foundries to ensure that the designs can be correctly manufactured. The most important objective of routing is to complete all the required connections. The routing step adds wires needed to properly connect the placed components while obeying all design rules for the IC.This stage involves routing of nets connecting different standard cells through different metal layers. There are two types of routing in the physical design process namely global and detailed routing.Global routing -is used to provide instructions to the detailed router about where to route every net. It provides channels for interconnect to be routed. Global routing allocates routing resources that are used for connections. Global routing first partitions the routing region into tiles and decides tile-to-tile paths for all nets while attempting to optimize some given objective function. Detailed routing -is where we specify exact location of the wires/interconnects in the channels specified by the global routing. Metal layer information of the interconnect are also specified here. Detailed routing assigns routes to specific metal layers and routing tracks within the global routing resources. The primary goal of detailed routing is to complete all of the required interconnect without leaving shorts or spacing violations.

Objectives of routing are, reducing the routing wirelength and ensuring each net to satisfy its required timing budget, have become essential for modern chip design.

3.5 Parasitic extraction

Parasitic extraction is an significant component of the Back annotation cycle time**.** Parasitic extraction is calculation of the parasitic effects in both the designed devices and the required wiring interconnects of an electronic circuit detailed device parameters, parasitic capacitances, parasitic resistances and parasitic inductances, commonly called parasitic devices parasitic components, or simply parasitic. The major purpose of parasitic extraction is to create an accurate analog model of the circuit, so that detailed simulations can emulate actual digital and analog circuit responses. Digital circuit responses are often used to populate databases for signal delay and loading calculation such as timing analysis, circuit simulation and signal integrity analysis. Analog circuits are often run in detailed test benches to indicate if the extra extracted parasitic will still allow the designed circuit to function.

3.7 Back annotation

The process of putting delays from a given source for the cells in a netlist during netlist simulation is called back annotation. Normally the values of the delays corresponding to each cell in the netlist would come from the simulation library ie, verilog model of library cells. But those delays are not the actual delays of cells, as each of them is instantiated in a netlist in different surroundings, different physical locations, different loads and different fan in. The delay of two similar cells in the netlist at two different physical locations in a chip can be significantly different depending upon above said factors. Therefore in order to have actual delays for the cells in the netlist, an SDF is written out, by a EDA tool can be a synthesis tool or a layout tool etc. which contains the delays of each instance of each library cell in the netlist, under the circumstances the cell is in. During simulations or Static Timing Analysis, each cell in the netlist gets its corresponding delay read, or more technically 'annotated' from the SDF file.

3.7 Physical verification

Physical verification checks the correctness of the generated layout design. The major checks performed here are DRC,LVS ,antenna rule checking,EM and IR analysis.

DRC- checks determine if the layout satisfies a set of rules required for manufacturing. The most common of these are spacing rules between metals, minimum width rules, via rules etc. There will also be specific rules pertaining to the technology. An input to the design rule tool is a ‘design rule file’. LVS- is another major check in the physical verification stage. Here the layout created is verified such that it is functionally the same as the schematic/netlist of the design-that have correctly transferred into geometries your intent while creating the design. Antenna - antenna effect or “plasma induced gate oxide damage” is a manufacturing effect. i.e, this is a type of failure that can occur solely at the manufacturing stage. This is a gate damage that can occur due to charge accumulation on metals and discharge to a gate through gate oxide. EM analysis-is the gradual displacement of metal atoms in a semiconductor. It occurs when the current density is high enough to cause the drift of metal ions in the direction of the electron flow, and is characterized by the ion flux density. This density depends on the magnitude of forces that tend to hold the ions in place, i.e., the nature of the conductor, crystal size, interface and grain-boundary chemistry, and the magnitude of forces that tend to dislodge them, including the current density, temperature and mechanical stresses.IR analysis-voltage drop in power (and bounces in ground) due to electrical parameters (resistance, capacitance, inductance) of power (ground) network.

4. RESULT ANALYSIS

Placement:

Issue-standard cell placement in the narrow region between macros, creating congestion after placement.Solution-placement blockages (buffer only) have been added in the floorplan database and did placement again.

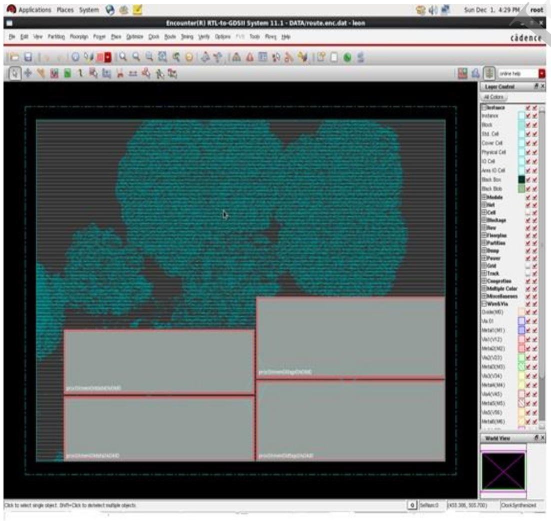

Encounter(R) RTL-to-GDSII System 11.1 - DATA/route.enc.dat - leon
cadence
Layer Control
World View

Fig 4.2 (b) Result after placement

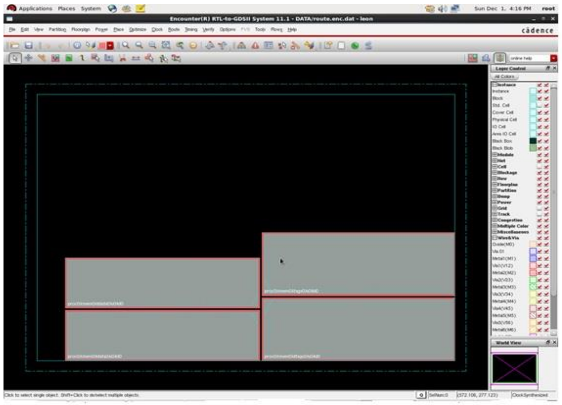

Encounter(R) RTL-to-GDSII System 11.1 - DATA/route.enc.dat - leon
cadence
Layer Control
World View

Fig 4.1 (a) Result after floorplan

Slew violation:

Issue-too long transition from one logic level to another that depends upon input capacitance and node resistance. Solution-check out the length of the wire throughout the design in the placement database and buffer the long nets to avoid slew violation.

Timing summary after placement

```
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+---------+---------+---
------+
|     Setup mode     |   all   | reg2reg | in2reg  | reg2out | in2out  |
clkgate |
+--------------------+---------+---------+---------+---------+---------+---
------+
|           WNS (ns):| -4.171  | -4.171  |  0.679  | -0.012  |   N/A   | -
1.453   |
|           TNS (ns):| -4110.6 | -4110.6 |  0.000  | -0.012  |   N/A   | -
4.446   |
|    Violating Paths:|  4002   |  4001   |    0    |    1    |   N/A   |
4    |
|          All Paths:|  12269  |  11463  |   321   |   486   |   N/A   |
34    |
+--------------------+---------+---------+---------+---------+---------+---
------+

+----------------+-------------------------------+------------------+
|                |             Real              |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |    61 (61)       |  -18.880   |     69 (69)      |
|   max_tran     |   586 (13008)    |   -7.607   |   592 (13024)    |
|   max_fanout   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 39.599%
Routing Overflow: 0.00% H and 0.28% V
------------------------------------------------------------
Reported timing to dir ./timingReports
Total CPU time: 17.15 sec
Total Real time: 18.0 sec
Total Memory Usage: 474.855469 Mbytes
```

Post CTS-timing summary

```
###############################################################
#  Generated by:      Cadence Encounter 11.10-p003_1
#  OS:                Linux i686(Host ID sietcadence)
#  Generated on:      Sun Dec  1 17:58:43 2013
#  Design:            leon
#  Command:           timeDesign -postCTS
###############################################################

------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+---------+---------+---
------+
|     Setup mode     |   all   | reg2reg | in2reg  | reg2out | in2out  |
clkgate |
+--------------------+---------+---------+---------+---------+---------+---
------+
|           WNS (ns):| -4.248  | -4.248  |  0.289  | -1.056  |   N/A   | -
1.838  |
|           TNS (ns):| -4040.3 | -3953.4 |  0.000  | -86.914 |   N/A   | -
5.954  |
|    Violating Paths:|  3911   |  3725   |    0    |   186   |   N/A   |
4    |
|          All Paths:|  12273  |  11463  |   566   |   486   |   N/A   |
34    |
+--------------------+---------+---------+---------+---------+---------+---
------+

+----------------+------------------------------+------------------+
|                |              Real            |       Total      |
|    DRVs        +------------------+-----------+------------------|
|                |  Nr nets(terms)  | Worst Vio |  Nr nets(terms)  |
+----------------+------------------+-----------+------------------+
|   max_cap      |    62 (62)       |  -19.016  |    62 (62)       |
|   max_tran     |   611 (13150)    |   -7.751  |   611 (13163)    |
|   max_fanout   |      0 (0)       |     0     |      0 (0)       |
+----------------+------------------+-----------+------------------+

Density: 40.201%
Routing Overflow: 0.00% H and 0.37% V
------------------------------------------------------------
```

OCV-Timing summary

```
###############################################################
#  Generated by:       Cadence Encounter 11.10-p003_1
#  OS:                 Linux i686(Host ID sietcadence)
#  Generated on:       Sun Dec  1 18:04:42 2013
#  Design:             leon
#  Command:            timeDesign -postCTS -prefix ocv
###############################################################


------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+---------+---------+---
------+
|     Setup mode     |   all   | reg2reg | in2reg  | reg2out | in2out  |
clkgate |
+--------------------+---------+---------+---------+---------+---------+---
------+
|           WNS (ns):| -4.248  | -4.248  |  0.289  | -0.019  |   N/A   | -
1.838  |
|           TNS (ns):| -3953.4 | -3953.4 |  0.000  | -0.019  |   N/A   | -
5.954  |
|    Violating Paths:|  3726   |  3725   |    0    |    1    |   N/A   |
4    |
|          All Paths:|  12273  |  11463  |   566   |   486   |   N/A   |
34    |
+--------------------+---------+---------+---------+---------+---------+---
------+
+----------------+------------------------------+-------------------+
|                |             Real             |       Total       |
|    DRVs        +------------------+-----------+-------------------|
|                |  Nr nets(terms)  | Worst Vio |  Nr nets(terms)   |
+----------------+------------------+-----------+-------------------+
|   max_cap      |     62 (62)      |  -19.016  |     62 (62)       |
|   max_tran     |   611 (13150)    |   -7.751  |   611 (13163)     |
|   max_fanout   |      0 (0)       |     0     |      0 (0)        |
+----------------+------------------+-----------+-------------------+

Density: 40.201%
Routing Overflow: 0.00% H and 0.37% V
```

Post route-timing summary

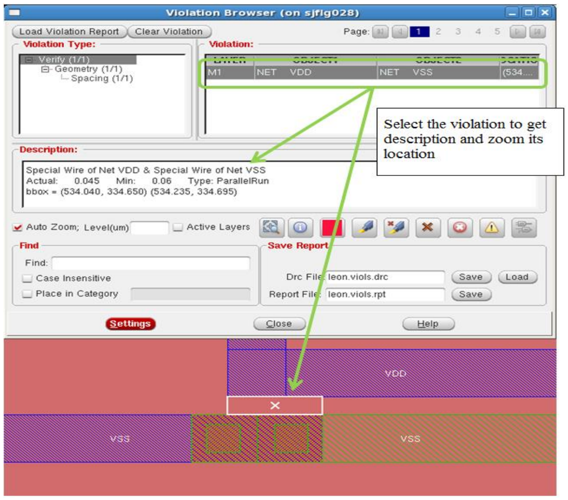
```
###############################################################
#  Generated by:      Cadence Encounter 11.10-p003_1
#  OS:                Linux i686(Host ID sietcadence)
#  Generated on:      Sun Dec  1 18:48:41 2013
#  Design:            leon
#  Command:           timeDesign -postRoute
###############################################################

------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+---------+---------+---
------+
|     Setup mode     |   all   | reg2reg | in2reg  | reg2out | in2out  |
clkgate |
+--------------------+---------+---------+---------+---------+---------+---
------+
|           WNS (ns):| -0.114  | -0.114  |  0.738  |  0.035  |   N/A   |
0.090  |
|           TNS (ns):| -1.066  | -1.066  |  0.000  |  0.000  |   N/A   |
0.000  |
|    Violating Paths:|   19    |   19    |    0    |    0    |   N/A   |
0     |
|          All Paths:|  12273  |  11463  |   566   |   486   |   N/A   |
34     |
+--------------------+---------+---------+---------+---------+---------+---
------+

+----------------+------------------------------+-------------------+
|                |             Real             |       Total       |
|    DRVs        +------------------+-----------+-------------------|
|                |  Nr nets(terms)  | Worst Vio |  Nr nets(terms)   |
+----------------+------------------+-----------+-------------------+
|   max_cap      |      0 (0)       |   0.000   |      0 (0)        |
|   max_tran     |     5 (173)      |  -0.907   |     5 (173)       |
|   max_fanout   |      0 (0)       |     0     |      0 (0)        |
+----------------+------------------+-----------+-------------------+

Density: 41.458%
------------------------------------------------------------
```



Select the violation to get description and zoom its location

Special Wire of Net VDD & Special Wire of Net VSS
Actual: 0.045 Min: 0.06 Type: ParallelRun
bbox = (534.040, 334.650) (534.235, 334.695)

Fig 4.2(a) DRC violation

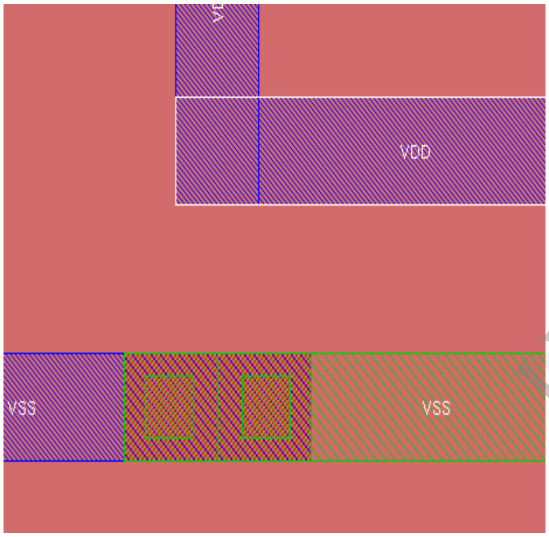


Fig 4.2 (b) Result after fixing DRC

5. CONCLUSION

The physical design flow of a Leon processor from netlist to gdsii that starts from floorplan, placement, CTS, routing and ends with physical verification was implemented. The violations those resulted in the implementation of Leon processor such as crosstalk, slew violations, congestion and other signal integrity issues are fixed by inserting additional buffers as well as using double width double spacing rule. For this implementation Cadence Encounter tool is used. Results show that the implemented design clears the physical verification checks such as DRC, LVC and antenna design rule to obtain a design that is suitable for manufacturing.

REFERENCES

[1] R. Chaturvedi and J. Hu, "Buffered clock tree for high quality IC design," *in Proc. Int. Symp. Qual. Electron. Des., 2004,* pp. 381–386.

[2] Tilmann stohr, Markus Alt, Asmus Hetzel, Jergen Koehl. 'Avoidance, reduction and analysis of crosstalk on VLSI chips', *IBM Entwicklung GmbH, Sch¨onaicher Straße 220, 71032 B¨oblingen, Germany.*

[3] Youngsoo shin and Jun seomunkaistkyu-myung choi *Samsung Electronics and takayasu sakurai,* 'Power Gating: Circuits, Design Methodologies, and Best Practice for Standard-Cell VLSI Designs' University of Tokyo.

[4] "Cadence SOC Encounter user guide." *http://www.cadence.com/products/di/ first encounter/pages/default.aspx*

[5] "Signal electro-migration analysis and fixing research in IC Compiler." *http://www.synopsys.com.cn/information/snug/2011/ signal-electro-migration-analysis-and-fixing-research-in-ic-compiler-2*

[6] "Understand and avoid electromigration (EM) & IR-drop in custom IP blocks." *http://www.synopsys.com/Tools/Verification/CapsuleModule/CustomSim-RA-wp.pdf*

[7] "Synopsys Design Compiler user guide." http://www.synopsys.com/Tools/ Implementation/RTLSynthesis/DCUltra/pages/default.aspx

[8] "Signal and design integrity whitepaper." *http://http://w2.cadence.com/whitepapers*