INTERNATIONAL JOURNAL · ENGINEERING RESEARCH ·

International Journal for Innovative Engineering and Management Research

A Peer Revieved Open Access International Journal

www.ijiemr.org

IJIEMR Transactions, online available on 25th Mar 2018. Link :http://www.ijiemr.org/downloads.php?vol=Volume-08&issue=ISSUE-03

Title: **PHYSICAL DESIGN IMPLEMENTATION OF TORPEDO SUB-SYSTEM USING 180NM TECHNOLOGY**

Volume 08, Issue 03, Pages: 232–244.

Paper Authors

KISHORE K, J LINGAIAH

ARJUN COLLEGE OF TECHNOLOGY AND SCIENCES

International Journal for Innovative Engineering and Management Research

A Peer Revieved Open Access International Journal

www.ijiemr.org

PHYSICAL DESIGN IMPLEMENTATION OF TORPEDO SUB-SYSTEM USING 180NM TECHNOLOGY

[1]KISHORE K, [2]J LINGAIAH

[1]M.tech Scholar , Department of Electronics and Communication Engineering, ARJUN COLLEGE OF TECHNOLOGY AND SCIENCES

[2]HOD & Associate Professor, Department of Electronics and Communication Engineering, ARJUN COLLEGE OF TECHNOLOGY AND SCIENCES

ABSTRACT

This project is an attempt to capture most of the VLSI industry requirements in a hands-on example. This project has been mentored by veterans in the industry. The goal here is to give the students hands-on experience in Physical Design aspect of VLSI. It is carefully crafted keeping in the mind the competency of students graduated from Indian colleges; thereby it becomes easy for them to start off. As we go through various stages in the project, we experience situations which emulate the challenges faced during the execution of a typical live project. By the end of the project execution, we are aware of current scenario in the industry and moreover, we would have experienced it. Hence, we are in a better position to manage it. This experience is what sets the character in us which is required to face the industry and makes us confident and competent enough to be called, technically, “Industry ready”. In this project we are using industry standard Design for Test and MBist to test the Standard logic and Macros. High count Macros to experience challenges in floor-plan. In Torpedo we are using sophisticated concurrent analysis engine MCMM which is current industry standard.This project is designed to experience all kinds of timing violations and analysis of timing reports. By this project we experienced different reliability issues that are faced by semiconductor industry and how we they tackled these issues using Design For Manufacturability. Hands-on experience on the industry standard Tools like IC Compiler, Calibre, PrimeTime.

Torpedo sub block includes 32 macros, 43275 standard cells with supply voltage of 1.8V and to get IR drop (VDD + VSS) less than 5% of 1.8V, working at an operating frequency of 400 MHz, it has total of 5 clocks 3 propagated and 2 generated, Design uses 5 metal layers. Fab: Jazz semiconductor and Technology node: 180nm

1. INTRODUCTION

1.1 Introduction

Very-large-scale integration (**VLSI**) is the process of creating an integrated circuit (IC) by combining hundreds of thousands of transistors or devices into a single chip.

VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device.

Before the introduction of VLSI technology most ICs had a limited set of functions they could perform. An electronic circuit might consist of a CPU, ROM, RAM and other glue logic. VLSI lets IC designers add all of these into one chip. VLSI is an important area of electronic and computer engineering. However, there are few textbooks available for undergraduate/postgraduate study of VLSI design automation and chip layout. VLSI Physical Design Automation: Theory and Practice fills the void and is an essential introduction for senior undergraduates, postgraduates and anyone starting work in the field of CAD for VLSI. It covers all aspects of physical design, together with such related areas as automatic cell generation, silicon compilation, layout editors and compaction. A problem-solving approach is adopted and each solution is illustrated with examples. Each topic is treated in a standard format: Problem Definition, Cost Functions and Constraints, Possible Approaches and Latest Developments.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 main concern in the physical design of VLSI-chips is to find a layout with minimal area, further the total wirelength has to be minimized. For some critical nets there are hard limitations for the maximal wirelength.

Special features: The book deals with all aspects of VLSI physical design, from partitioning and floorplanning to layout generation and silicon compilation; provides a comprehensive treatment of most of the popular algorithms; covers the latest developments and gives a bibliography for further research; offers numerous fully described examples, problems and programming exercises.

1.2 History

The history of the transistor dates to the 1920s when several inventors attempted devices that were intended to control current in solid-state diodes and convert them into triodes. Success came after World War II, when the use of silicon and germanium crystals as radar detectors led to improvements in fabrication and theory. Scientists who had worked on radar returned to solid-state device development. With the invention of transistors at Bell Labs in 1947, the field of electronics shifted from vacuum tubes to solid-state device.With the small transistor at their hands, electrical engineers of the 1950s saw the possibilities of constructing far more advanced circuits. However, as the complexity of circuits grew, problems arose. One problem was the size of the circuit. A complex circuit like a computer was dependent on speed. If the components were large, the wires interconnecting them must be long. The electric signals took time to go through the circuit, thus slowing the computer.

The invention of the integrated circuit by Jack Kilby and Robert Noyce solved this problem by making all the components and the chip out of the same block (monolith) of semiconductor material.

The circuits could be made smaller, and the manufacturing process could be automated. This led to the idea of integrating all components on a single-crystal silicon wafer, which led to small-scale integration (SSI) in the early 1960s, medium-scale integration (MSI) in the late 1960s, and then large-scale integration (LSI) as well as VLSI in the 1970s and 1980s, with tens of thousands of transistors on a single chip (later hundreds of thousands, then millions, and now billions (10^9)) The first semiconductor chips held two transistors each. Subsequent advances added more transistors, and as a consequence, more individual functions or systems were integrated over time. The first integrated circuits held only a few devices, perhaps as many as ten diodes, transistors, resistors and capacitors, making it possible to fabricate one or more logic gates on a single device. Now known retrospectively as small-scale integration (SSI), improvements in technique led to devices with hundreds of logic gates, known as medium-scale integration (MSI). Further improvements led to large-scale integration (LSI), i.e. systems with at least a thousand logic gates. Current technology has moved far past this mark and today's microprocessors have many millions of gates and billions of individual transistors.At one time, there was an effort to name and calibrate various levels of large-scale integration above VLSI. Terms like ultra-large-scale integration (ULSI) were used. But the huge number of gates and transistors available on common devices has rendered such fine distinctions moot. Terms suggesting greater than VLSI levels of integration are no longer in widespread use.

In 2008, billion-transistor processors became commercially available. This became more commonplace as semiconductor fabrication advanced from the then-current generation of 65 nm processes. Current designs, unlike the earliest devices, use extensive design automation and automated logic synthesis to lay out the transistors, enabling higher levels of complexity in the resulting logic functionality. Certain high-performance logic blocks like the SRAM (static random-access memory) cell, are still designed by hand to ensure the highest efficiency.

1.3 Structured Design

Structured VLSI design is a modular methodology originated by Carver Mead and Lynn Conway for saving microchip area by minimizing the interconnect fabrics area. This is obtained by repetitive arrangement of rectangular macro blocks which can be interconnected using wiring by abutment. An example is partitioning the layout of an adder into a row of equal bit slices cells. In complex designs this structuring may be achieved by hierarchical nesting.

Structured VLSI design had been popular in the early 1980s, but lost its popularity later because of the advent of placement and routing tools wasting a lot of area by routing, which is tolerated because of the progress of Moore's Law. When introducing the hardware description language KARL in the mid' 1970s, Reiner Hartenstein coined the term "structured VLSI design" (originally as "structured LSI design"), echoing Edsger Dijkstra's structured programming approach by procedure nesting to avoid chaotic spaghetti-structured program

Difficulties: As microprocessors become more complex due to technology scaling, microprocessor designers have encountered several challenges which force them to think beyond the design plane, and look ahead to post-silicon:

- **Process variation** – As photolithography techniques get closer to the fundamental laws of optics, achieving high accuracy in doping concentrations and etched wires is becoming more difficult and prone to errors due to variation. Designers now must simulate across multiple fabrication process corners before a chip is certified ready for production, or use system-level techniques for dealing with effects of variation.
- **Stricter design rules** – Due to lithography and etch issues with scaling, design rules for layout have become increasingly stringent. Designers must keep in mind an ever increasing list of rules when laying out custom circuits. The overhead for custom design is now reaching a tipping point, with many design houses opting to switch to electronic design automation (EDA) tools to automate their design process.
- **Timing/design closure** – As clock frequencies tend to scale up, designers are finding it more difficult to distribute and maintain low clock skew between these high frequency clocks across the entire chip. This has led to a rising interest in multicore and multiprocessor architec tures, since an overall speedup can be obtained even with lower clock frequency by using the computational power of all the cores.
- **First-pass success** – As die sizes shrink (due to scaling), and wafer sizes go up (due to lower manufacturing costs), the number of dies per wafer increases, and the complexity of making suitable photomasks goes up rapidly. A mask set for a modern technology can cost several million dollars. This non-recurring expense deters the old iterative philosophy involving several "spin-cycles" to find errors in silicon, and encourages first-pass silicon success. Several design philosophies have been developed to aid this new design flow, including design for manufacturing (DFM), design for test (DFT), and Design for X.

2 DESIGN PLANNING

2.1 Physical Design Flow

INPUTS

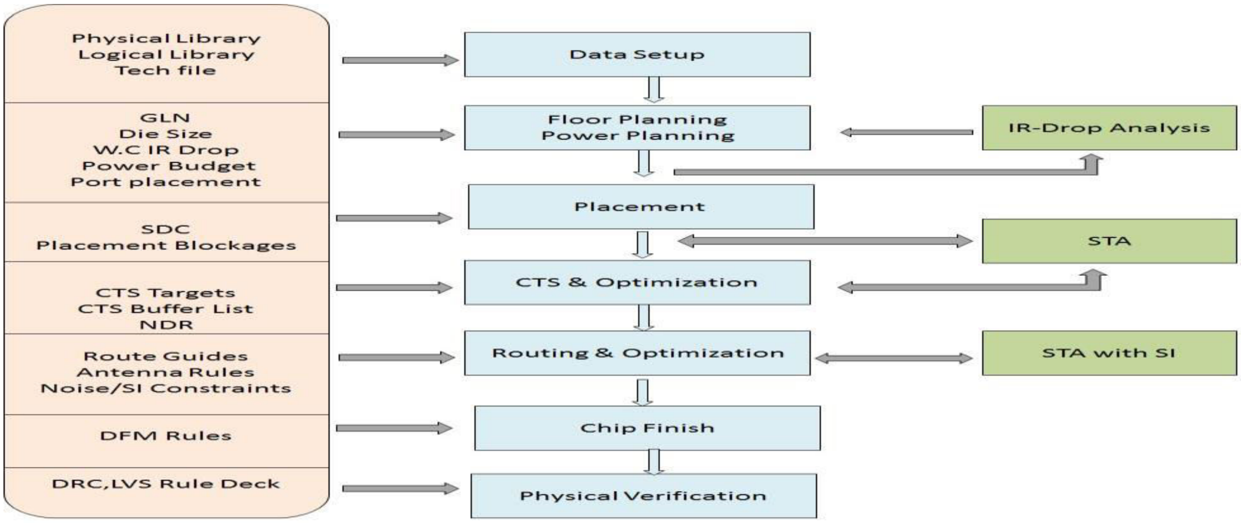


Fig 2.1 Physical Design Flow

Table2.1: Physical Design Specifications

Technology	180 nm
Library Vendor	Jazz Semiconductors
Operating Voltage	1.8 V
Clock Frequency	400 Mhz
Power Budget	300 mw
Max IR drop (VDD+VSS)	5 % (90 mw)
No. of Macros	32
No. of Std Cells	43000
Functional Clocks	**sys_clk** – 400 Mhz, **sys_rclk** – 400 Mhz

Test Mode Clocks	**scan_clk** – 100 Mhz
Die Area	5.9 mm sq
No. of Metal Layers	6
Standard Cell Height	5.04 um (9 track library)
Process Voltage Temperature	max_lib (Temp: 125, Voltage: 1.95 V, Process: slow), min_lib (Temp: -40, Voltage: 1.65 V, Process: fast), nominal_lib (Temp: 25, Voltage: 1.8 V, Process: nom).

2.2 Design Planning

In the design planning context, floor-planning is the process of sizing and placing hierarchical cells and functional blocks in a manner that makes later physical design steps more effective. Floor-planning in hierarchical flows provides a basis for estimating the timing of the top level. A timing budget allocates the clock cycle time to each block according to the top-level timing estimation.An effective floor-plan helps ensure timing closure in many ways, such as placing blocks to make critical paths short, preventing routing congestion that would lead to longer paths, and eliminating the need for over-the-top routing for noise-sensitive blocks. The challenge is to create a floor-plan with good area efficiency while leaving sufficient area for routing.

2.2.1 Data Setup

Data required for floor-plan can be segregated under below categories

1) Synthesis data: It contains following data

Library or timing related information:

□Provide timing and functionality information for all standard cells (and, or, flip-flop …) □Provide timing information for hard macros (IP, ROM, RAM …) □Define drive/load design rules:

Max fan-out

Max transition

Max/Min capacitances are usually the same ones used by Design Compiler during synthesis these are specified with variables: target library link library.

Constraints file or SDC:

An ASCII text file (with the extension. sic) that contains design constraints and timing assignments in the industry standard Synopsys Design Constraints format. The constraints in a Synopsys Design Constraints File are described using the TCL.

Constraints may consists of clock definitions, IO delays, max transitions, Load values to the design, case analysis, constant value settings, Don't use settings, ideal net, ideal networks, max area and disable timing and exceptions.

GLN (Gate Level Netlist):

It contains gate level description of design where standard cell and macros are instantiated. It is given by synthesis.

2) Physical Data:

a) Physical Reference Libraries: Contain physical information of standard, macro and pad cells, necessary for placement and routing, and definition of placement unit tile which contains

- Height of placement rows
- Minimum width resolution
- Preferred routing directions
- Pitch of routing tracks

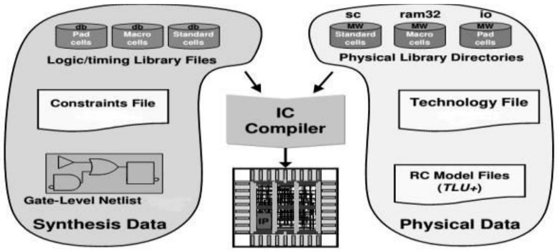


Fig 2.2 Synthesis data and Physical data

2.3 Multi Corner Multi Mode

Multi-corner multi-mode (MCMM) analysis is a technique intended to provide high confidence results for timing and other metrics without performing exhaustive simulation of all possible IC conditions. The analysis uses multiple design points to examine the effects of process and environmental variations as well as changes caused by shifts into different operating modes. Multi-corner analysis is intended to capture the effects of variation on the manufacturing process as well as voltage and temperature. Multi-mode analysis alongside has become more commonplace because of the demand for ICs that have different low-power modes, as well as test modes and various functional modes. By using traditional worst case and best case operating conditions no longer works because as the technology node shrinking, fixing setup or hold in one PVT no longer works, so we need method to analyze in all corners and mode so we are using MCMM.

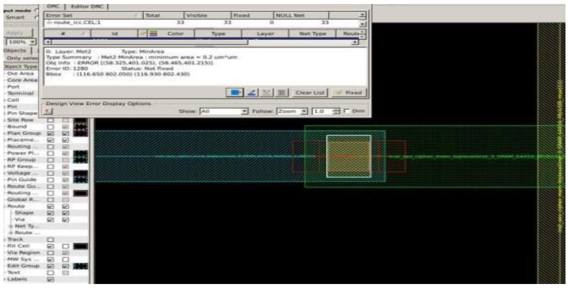
3. PHYSICAL VERIFICATION

3.1 Design Rule Check

Design Rules are a series of parameters provided by semiconductor manufacturers that enable the designer to verify the correctness of a mask set. Design rules are specific to a particular semiconductor manufacturing process. A design rule set specifies certain geometric and connectivity restrictions to ensure sufficient margins to account for variability in semiconductor manufacturing processes, so as to ensure that most of the parts work correctly.

In PD we check only sub-set of rules there in DRC rule deck, these sub-set of rules are related to metal layers and via's. These rules are present in Technology File. These rules include Min Spacing, Min Area, and Min Width etc. Some DRC errors we have encountered in our block are given below.

Min Area DRC



Error Set	Total	Visible	Fixed	NULL Net
route_icc.CEL;1	33	33	0	33

```
0:  Layer: Met2              Type: MinArea
Type Summary   : Met2 MinArea : minimum area = 0.2 um*um
Obj Info   : ERROR [(58.325,401.025), (58.465,401.215)]
Error ID: 1280              Status: Not Fixed
Bbox      : (116.650 802.050) (116.930 802.430)
```

Show: All   Follow: Zoom   1.0   Dim

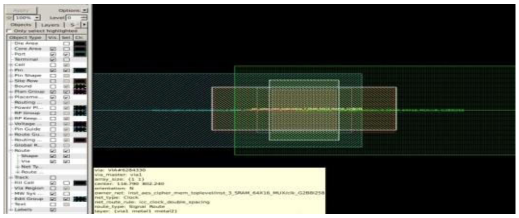
Fig 3.1 Min Area of M2 at Via

Error: Min area of metal 2 should be 0.2 um*um.

Description: While jumping from non-adjacent metals, In this case M1 to M4 we have to put Via1—>M2-->Via2-->M3--Via3. If the M2 and M3 area is not meeting min area then there will be DRC violation at that VIA.

Cause: Via enclosing definition in technology file is wrong. It says the enclosing height is 0.06 and enclosing width is 0.01 and Via min width is 0.26. Now length of enclosing metal is 0.28 and height is 0.38, area = 0.28*0.38=> 0.1064 < 0.2.

Solution: We created M3 shape with same name as Via and using **resize_object -area 0.2** we get M3 segment of area 0.2 um*um and place it over DRC violating VIA.



```
via: VIA#6284330
via_master: via1
array_size: {1 1}
center: 118.790 802.240
orientation: N
owner_net: inst_aes_cipher_mem_toplevel/inst_3_SRAM_64X16_MUX/clk_G2B8I258
net_type: Clock
net_route_rule: icc_clock_double_spacing
route_type: Signal Route
layer: {via1 metal1 metal2}
```

Fig 3.2 After Fixing Min area Error

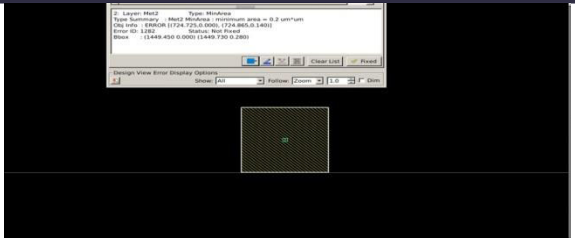
Min Area at Port Terminal

Error: Min arear of metal 2 should be 0.2 um*um

Discription: At port terminals M2 min area is not satisfied.

Cause: While placing port terminals min area is not taken care and this min area is taken care during routing, but some ports like **dmem_bus*** are unconnected. So, these ports are giving min area DRC error.

Solution: Select those ports which are violating min area and increased the height of terminal to 0.75 um using **resize_object -height 0.75** so that it satisfies Min area of 0.2 um*um.



```
2: Layer: Met2              Type: MinArea
Type Summary   : Met2 MinArea : minimum area = 0.2 um*um
Obj Info   : ERROR [(724.725,0.000), (724.865,0.140)]
Error ID: 1282              Status: Not Fixed
Bbox       : (1449.450 0.000) (1449.730 0.280)
```

Fig 3.3 Min Area DRC at Port Terminal Before Fixing

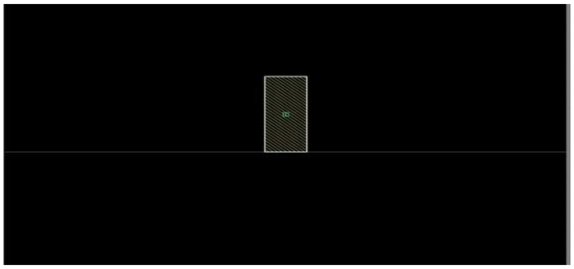
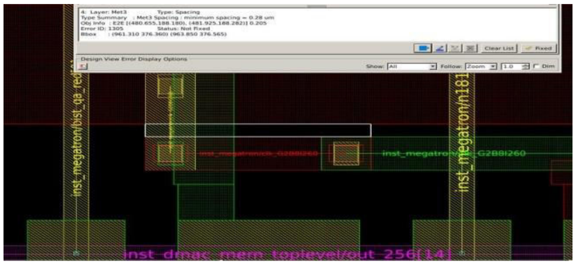


Fig 3.4 Min Area DRC at Port Terminal After Fixing

Min Spacing DRC Error



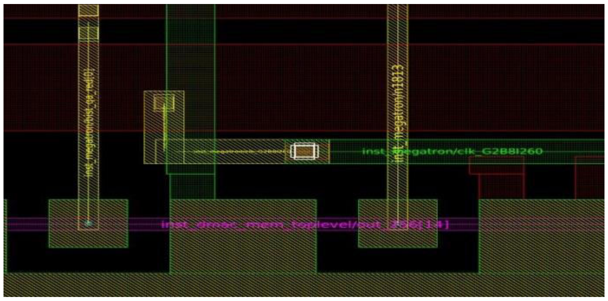
```
4: Layer: Met3              Type: Spacing
Type Summary   : Met3 Spacing : minimum spacing = 0.28 um
Obj Info   : E2E [(480.655,188.180), (481.925,188.282)] 0.205
Error ID: 1305              Status: Not Fixed
Bbox       : (961.310 376.360) (963.850 376.565)
```

Clear List
Fixed
Design View Error Display Options
Show: All
Follow: Zoom
1.0
Dim
inst_megatron/bist_qa_red
inst_megatron/clk_G2B8I260
inst_megatron/clk_G2B8I260
inst_megatron/n181
inst_dmac_mem_toplevel/out_256[14]

Fig 3.5 Min Spacing Before Fixing

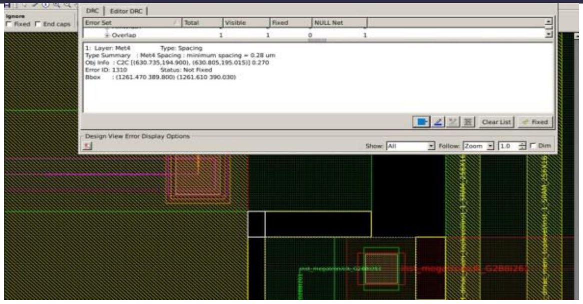
Error: Min Spacing between M3 – M3 should be 0.28 um

Solution: Above Figure show the image of error, to fix this error we extended M2 to M4 and placed Via between M2 and M4



inst_megatron/bist_qa_red0
inst_megatron/n1813
inst_megatron/clk_G2B8I260
inst_dmac_mem_toplevel/out_256[14]

Fig 3.6 Min Spacing After Fixing



Error Set	Total	Visible	Fixed	NULL Net
Overlap	1	1	0	1

```
1:  Layer: Met4            Type: Spacing
Type Summary   : Met4 Spacing : minimum spacing = 0.28 um
Obj Info  : C2C [(630.735,194.900), (630.805,195.015)] 0.270
Error ID: 1310             Status: Not Fixed
Bbox      : (1261.470 389.800) (1261.610 390.030)
```

Fig 3.7 Min spacing DRC

Error: Min spacing between M4 should be 0.28 um

Solution: We tried to fix this error manually but it is violating NDR. So we selected that particular net and removed the net routing using **remove_net_routing** and re-routed that particular net using **route_zrt_eco -reroute any_nets,** the router reroutes any nets freely to fix the DRC violations.

3.2 Layout Vs Schematic

A successful Design rule check (DRC) ensures that the layout confirms to the rules designed/required for faultless fabrication. However, it does not guarantee if it really represents the circuit you desire to fabrication. This is where an LVS check is used.

LVS checking software recognizes the drawn shapes of the layout that represent the electrical components of the circuit, as well as the connections between them. Thisnetlistis compared by the "LVS" software against a similar schematic or circuit diagram'snetlist

LVS Checking involves following three steps:

1. **Extraction**: The software program takes a database file containing all the layers drawn to represent the circuit during layout. It then runs the database through many area basedlogic operationsto determine the semiconductor components represented in the drawing by their layers of construction. Area basedlogical operationsuse polygon areas

   as inputs and generate output polygon areas from these operations. These operations are used to define the device recognition layers, the terminals of these devices, the wiring conductors and via structures, and the locations of pins (also known as hierarchical connection points). The layers that form devices can have various measurements performed to and these measurements can be attached to these devices. Layers that represent "good" wiring (conductors) are usually made of and called metals. Vertical connections between these layers are often called vias.
2. **Reduction**: During reduction the software combines the extracted components into series and parallel combinations if possible and generates anetlistrepresentation of the layout database. A similar reduction is performed on the "source" Schematic netlist.
3. **Comparison**: The extracted layout netlist is then compared to the netlist taken from the circuit schematic. If the two netlists match, then the circuit passes the LVS check. At this point it is said to be "LVS clean."

During LVS we got following errors

1) Short: we got this LVS error because after putting antenna diode during antenna fixing we not routed or connected antenna diodes VDD and VSS, so we fixed this by connecting respective pin through derive_pg_connection command

2) Bad Component Subtype: we got this error because whatever diode name in the spice netlist and in GDS are defferent

4. SIGN-OFF STATIC TIMING ANALYSIS

4.1 Sign-off Static Timing Analysis

IC Compiler is an implementation engine, and StarRC and PrimeTime are signoff engines. In general, the implementation engine and the signoff engine are well correlated with each other. However, small variations can occur because

- PrimeTime and StarRC, as signoff tools, are more accurate.
- PrimeTime has reduced signal integrity pessimism.
- IC Compiler might have implementation margins.
- IC Compiler might have fewer corners.

These differences can cause timing violations during final signoff.

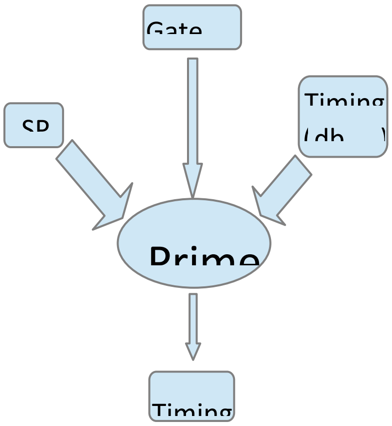


Fig 4.1 Block Diagram of STA

Primetime supports the use of timing models to represent chip sub modules. A timing model contains information about the timing characteristics, but not the logical functionality, of a submodule.

Primetime can generate a timing model from a sub module netlist, and then use that model in place of the original netlist for timing analysis at higher levels of hierarchy. This technique makes whole chip analysis run much faster.

To perform stage delay calculation accurately and efficiently, Primetime uses models to represent the driver, RC network, and capacitive loads on the net. An ideal model would produce exactly the same delays and slews as a SPICE simulation at the output of the driver and at the input of each receiver.

To perform static timing analysis, Primetime must accurately calculate the delay and slew (transition time) at each stage of each timing path. A stage consists of a driving cell, the annotated RC network at the output of the cell, and the capacitive load of the network load pins. The goal is to compute the response at the driver output and at the network load pins, given the input slew or waveform at the driver input, using the least amount of runtime necessary to get accurate results.

To perform stage delay calculation accurately and efficiently, Primetime uses models to represent the driver, RC network, and capacitive loads on the net. See below Figure. An ideal model would produce exactly the same delays and slews as a SPICE simulation at the output of the driver and at the input of each receiver.

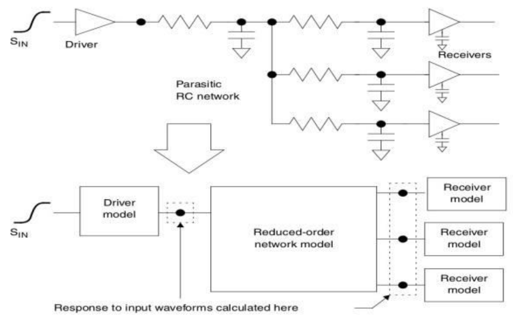


Fig 4.2 RC Network

The driver model is intended to reproduce the response of the driving cell's underlying transistor circuitry when connected to an arbitrary RC network; given a specific input slew. The reduced-order network model is a simplified representation of the full annotated network that has nearly the same response characteristics as the original network. PrimeTime uses the Arnoldi reduction method to create this model.

Additional Features in PrimeTime

- ✓ Advanced modeling capabilities with Interface Logic Models (ILM) and Extracted Timing Models (ETM)
- ✓ Graphical User Interface (GUI) enabling timing analysis and design visualization using schematics, histograms, tables, and tree graphs
- ✓ Save and restore
- ✓ What-if ECO analysis

Delay Annotation

PrimeTime supports SDF and SPEF to calculate interconnect delays. Standard Delay Format (SDF) contains only interconnect delay information; it doesn't contain any parasitic information. PrimeTime works faster with SDF, but less accuracy compared to SPEF**.**

4.2 DESIGN SUMMARY

Table 4.1 Design Summary

Design	WNS	TNS	Cell Count	Area	Buf/Inv	Utilization
floor-plan	-2899.07	-241034.547	43275	5137848	5662	65.39 %
placement	-1.645	-13219.260	48537	5285021	9967	71.56 %
CTS	-2.903	-24178.838	49034	5347308	10736	74.14 %
CTS OPT	-2.187	-22654.240	56642	5466194	19111	79.07 %
Clock Route	-2.119	21794.223	57574	5477563	19921	79.54 %
Routing	-2.159	-22453.443	57574	5477563	19921	79.54 %
Route OPT	-2.090	-22657.756	60855	5532429	23202	81.81 %
Chip-finish	-2.075	-22752.412	61156	5533926	23202	81.87 %

Even if you annotated all the delays of the design with SDF, you might want to annotate parasitics to perform certain design rule checks, such as maximum transition or maximum capacitance. In Torpedo we are using SPEF for better accuracy.

ECO Timing Violation Fixing

Setup:

Ways to fix setup:

- Gate Resizing
- Buffer insertion
- Using two inverters instead of one buffer

We first analyzed each and every cell delay, input transition in the violating timing report. Then we took cell with large delay and tried to get alternate drive strength of that cell using primetime command **estimate_eco -type size_cell -max.** This Command Quickly computes and displays the ECO alternatives based on current timing values of a particular stage. The estimate takes into account the incoming transition times, output loading, fanout loading, detailed parasitics if present, and driver characteristics of the candidate cells. Using **size_cell** resize that cell to required drive strength cell.We can ask tool to do this by using **fix_eco_timing -type setup.** We have to provide buffer list to the tool to use while buffer insertion.

Hold:

Hold violations can be fixed by inserting delay buffers in data path, but we have to make sure that inserting buffers should not violate setup time. We can use **estimate_eco -type insert_buffer -min** to get alternative cells and using **insert_buffer** command we can insert delay buffer to fix hold violation.

PrimeTime SI: PrimeTime can generate a timing model from a sub-module netlist , and then use that model in place of the original netlist for timing analysis at higher levels of hierarchy. This technique makes whole chip analysis run much faster.To perform stage delay calculation accurately and efficiently, PrimeTime uses models to represent the driver, RC Network and capacitive load on the net. An ideal model would produce exactly the same delays and slews as a SPICE simulation at the output of the driver and at the input of each receiver.To perform Static Timing Analysis, PrimeTime must accurately calculate the delay and slew (transition time) at each stage of each timing path. A stage consists of a driving cell, the annotated RC network at the output of the cell and the capacitive load of the network load pins. The goal is to compute the response at the driver output and at the network load pins, given the input slew or waveform at the driver input using the least amount of runtime necessary to get accurate results.The driver model is intended to reproduce the response of the driving cell's underlying transistor circuitry when connected to an arbitrary RC network given for a specific input slew. The reduced order network model is a simplified representation of the full annotated network that has really the same response characteristics as the original network.PrimeTime uses the Arnoldi reduction method to create this model.

CONCLUSION AND FUTURE SCOPE

CONCLUSION

The project “**Physical Design Implementation of Torpedo Sub-system Using 180nm Technology**” has been successfully designed and tested. Integrating features of all the hardware components used have developed it. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, using highly advanced IC's and with the help of growing technology the project has been successfully implemented.

FUTURE SCOPE

This futuristic approach as to be implemented in a real time analysis specifically related to any kind of communication or any kind of digital and analog signal applications or any kind of electronic gadget applications so that it can be utilized in any particular application.

REFERENCES

[1]."The History of the Integrated Circuit". Nobelprize.org. Retrieved 21 Apr 2012.

[2] "Digital Electronics - A Modern Approach by B K Jain". Retrieved 2 May 2017.

[3]. "A Survey Of Architectural Techniques for Managing Process Variation", ACM Computing Surveys, 2015

[4]. Baker, R. Jacob (2010). CMOS: Circuit Design, Layout, and Simulation, Third Edition. Wiley-IEEE. p. 1174. ISBN 978-0-470-88132-3. http://CMOSedu.com/

[5]. Weste, Neil H. E. & Harris, David M. (2010). CMOS VLSI Design: A Circuits and Systems Perspective, Fourth Edition. Boston: Pearson/Addison-Wesley. p. 840. ISBN 978-0-321-54774-3. http://CMOSVLSI.com/