Development & Verification of Standard Cell

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

(VLSI Design)

By

Bhavsar Stefi Ashok

(13MECV01)

Electronics & Communication Engineering Branch
Electrical Engineering Department
Institute of Technology
Nirma University
Ahmedabad-382 481

MAY 2015

Development & Verification of Standard Cell

Major Project Report

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology
in
Electronics & Communication Engineering
(VLSI Design)

By

Bhavsar Stefi Ashok

(13MECV01)

Under the guidance of

External Project Guide:
Mrs. Sumana Pal
Staff Design Engg.
ARM Embedded Technologies Pvt. Ltd.,
Banglore.

Internal Project Guide:
Prof. Amisha Naik
Professor (EC Dept.),
Institute of Technology,
Nirma University, Ahmedabad.

Electronics & Communication Engineering Branch
Electrical Engineering Department
Institute of Technology
Nirma University
Ahmedabad-382 481
MAY 2015

Declaration

This is to certify that

a. The thesis comprises my original work towards the degree of Master of Technology in VLSI Design at Nirma University and has not been submitted elsewhere for a degree.

b. Due acknowledgment has been made in the text to all other material used.

- Bhavsar Stefi A.

Disclaimer

"The content of this paper does not represent the technology, opinions, beliefs, or positions of Company, its employees, vendors, customers, or associates."

Certificate

This is to certify that the Major Project entitled **"Development & Verification of Standard Cell "** submitted by **Bhavsar Stefi Ashok. (13MECV01)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design , Nirma University, Ahmedabad is the record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of our knowledge,haven't been submitted to any other university or institution for award of any degree or diploma.

Date: Place: Ahmedabad

Prof. Amisha .P .Naik
Internal Project Guide

Dr. N. M. Devashrayee
Program Coordinator

Dr. P. N. Tekwani
Head of EE Dept.

Dr. Ketan Kotecha
Director, IT

this page is left blank intentionally to accommodate company certificate.

Acknowledgements

I am grateful to my thesis supervisors **Dr. Amisha .P .Naik**, Associate Professor, EC Department, Nirma University and **Mrs. Sumana Pal**, Staff Design Engineer at ARM Embedded Technologies Pvt. Ltd. for their constant guidance and motivation.

I would also like to thank to **Dr. P. N. Tekwani** , Head of Electrical Engineering Department and **Dr. Ketan Kotecha**, Director IT, Nirma University.

I am deeply indebted to **Dr. N. M. Devashrayee**, Program Co-ordinator of M.Tech VLSI Design and **Dr. N .P .Gajjar**, Program Co-ordinator of M.Tech Embedded Systems for allowing me to undertake this thesis work and for their guidelines during the review process.

I also wish to thank **Mr. Dileep CR**, Design Engineer, **Mr. Kenchappa Hanasi**, Senior Layout Design Engineer, **Ms. Sripriya S.**, Design Engineer, **Mr. Manigandeshwaran M.**, Staff Design Engineer and all other Standard Cell team members at ARM for their constant help and support. Without their experience and insights, it would have been very difficult to do quality work.

Last, but not the least, no words are enough to acknowledge constant support and sacrifices of my family members because of whom I am able to complete the degree program successfully.

- Bhavsar Stefi

13MECV01

Abstract

ASIC(Application specific Integrated Circuit) must be optimized in terms of leakage power and delay. This report will explain standard cell optimization method used by company to get optimized cost product, which will be power delay product (power x delay) in most of the cases. It gives detailed explanation of different input files used for tuning by celltuner tool, which is used to perform sizing on the standard cell.

Company has decided to keep flat schematic in the source for each cell used in project so far. So we need to convert all hierarchical schematics to flattened ones which will not have any ARM internal symbols and everything will be defined in terms of NMOS and PMOS models provided by foundry. This report explains how we can use Prosizer tool for this purpose and also for schematic sizing.

Pretty unfied builders is the project to reduce the effort required to configure DARBuilder, CellBuilder and QABuilder for an entire Standard Cell (SC) platform. The goal is to have a minimal collection of "source" files which can be used "as is" for all the automated phases (ie builders) of a SC flow and all products in a particular platform.

Various Cell level Physical checks and Electrical Checks are performed to estimate changes in Standard Cell Library. Device failing on silicon is more risky than on design. So by providing a margin of failure various Electrical Checks : Flop Margin Analysis, Balanced Beta Ratio, Latch Node Stability, Level Shifter Analysis, Electro-Migration are performed and the design is allowed to operate under various PVT (Process ,Voltage & Temperature) conditions to ensure under what conditions design can withstand. For these a flow named “EDAR“ (Electrical Design Assurance Report) is developed by company. The report explains how we can use DARBuilder Tool for this purpose.

Contents

List of Figures

Abbreviation Notation and Nomenclature

ASIC ..... Application Specific Integrated Circuit
NFET ..... N-type Field Effect Transistor
PFET ..... P-type Field Effect Transistor
EDAR ..... Electrical Design Assurance Report
CP ..... Common Platform
TSMC ..... Taiwan Semiconductor Manufacturing Company
DRC ..... Design Rule Check
ECO ..... Engineering Change Order
EDA ..... Electronics Design Automation
SVS ..... Schematic Vs Schematic Check
LVS ..... Layout Versus Schematic
GDS-II ..... Graphic Database System
MT-CMOS ..... Multiple-Threshold Voltage CMOS
LBDOEE ..... Layout Based Design of Engineering Experiments
SVT ..... Standard Threshold Voltage
HVT ..... High Threshold Voltage
LVT ..... Low Threshold Voltage
CDL ..... Circuit Description Language
CBL ..... Cell Builder log
FMA ..... Flop Margin Analysis
BBR ..... Balanced Beta Ratio
EM ..... Electromigration
LNS ..... Latch Node Stability
SPEF ..... Standard Parasitic Exchange Format
DSPF ..... Detailed Standard Parasitic Format

Chapter 1

Introduction

The custom design approach proves to be prohibitively expensive, a wide variety of deign approaches have been introduced over the years to shorten and automate the design process. The automation comes at the price of reduced integration density and performance. The following rue tends to hold : the shorter the deign time , the larger is penalty incurred. The idea behind cell based design methodology come into picture .This is to reduce library implementation effort by reusing a limited library of cells. The advantage of this approach is that cells only need to be designed and verified once for a given technology and they can be reused many times, thus amortizing the design cost. One Among the cell based approaches is the standard cell approach.

The standard cell approach standardizes the design entry level at the logic gate. A library containing wide selection of logic gates over a range of fan in and fan out counts is provided.

ARM Standard Cell Libraries are available from the latest advanced 28nm nodes to established mainstream 250nm nodes. These are "best in class" libraries based on architectural analysis by experts in leading-edge processes and design styles. The libraries include models for successful implementation across the entire design flow, from synthesis to tape out. ARM processors and other IP use the ARM Standard Cell Libraries for reference and

benchmark designs. It maintains High Density , High performance and Low power library with Clock Gating Cells, Level Shifter and Isolation Cells. Multi-VT cells (MTCMOS), Retention cells, Cells with Digital voltage and frequency scaling (DVFS) support .

ARM Standard Cell Libraries are for use in ultra-low power designs in the MHz range up to multiple GHz in performance critical applications. Variety of architectures based on different track heights and cell designs cover a wide performance, power and area range .

All libraries are tested at various Process, Voltage and Temperature conditions and correlated with library models .[2]

1.1 Objective

The overall objective of the project is to support standard cell design flow. For that mainly two tasks were assigned.

In first task , the company has decided to have complete foundry primitive source for every project schematics used so far. Previously for each schematic there were flat schematics in sc arm , sc tsmc and sc cp libraries. TSMC and CP are foundries which are used by the most of the customers. These foundries have little bit different models of NMOS and PMOS. So we need to convert sc arm schematics to sc tsmc and sc cp. Sc arm library will have ARM internal NMOS and PMOS symbols, while sc cp and sc tsmc libraries will have NMOS and PMOS symbols according to TSMC and CP foundry conventions. If source does not have flat schematic for particular cell, then new flat schematic should be created in sc arm first, and then it should be converted to sc tsmc and sc cp libraries. But recently only sc arm is maintained with flat schematics and enhancement has been made in ARM specific tools to convert schematics from arm primitives to foundry primitives.

The second task assigned was generation of Design Assurance Report for engineering ref-

erence. Here various cell level electrical checks such as Balanced Beta Ratio Check, Level Shifter Check, Electromigration Check (EM), Latch Node Stability, Flop Margin Analysis are performed to confirm library standards finally a report is generated, and the company provides customers with a report that gives each product‘s status with respect to these checks. The above two tasks are explained in the report later.

Company has decided to maintain a unified source which would help project team, when one want to migrate from one builder to another. The task assigned was creating an regression suite for all possible extraction combination used so far.

1.2 Thesis Organization

The rest of the thesis is organized as follows.

Chapter 2, *Theoretical Overview* , gives theoretical background of power optimization techniques , concept of logical effort and PVT conditions .

Chapter 3 , *Standard Cell Optimization* , describes how cost product is calculated using tool Celltuner also how Celltuner gives out optimum sizes.

Chapter 4, *Schematic Sizing using Prosizer* , describes how prosizer is used for Sizing schematics for project support.

Chapter 5 , *Cell Level Physical Checks*, describes how DARBuilder is used for performing various cell level physical checks.

Chapter 6 , *Parasitic Extraction* , describes about parasitic extraction using different tools.

Chapter 7, *EDAR Regression* , describes how Cell level Electrical Checks are performed and analysis is made from the Results.

Chapter 8, *HSPICE Simulation* , describes how waveforms are interpreted when failures occurs while performing cell level electrical checks.

Finally **Chapter 9** , *Conclusion* , concluding.

Chapter 2

Theoretical Overview

2.1 Power Optimization Techniques

A standard cell library is a collection of pre designed layout of basic logic gates like inverters, buffers, ANDs, ORs, NANDs etc. All the cells in the library have same standard height and have varied width.These reference libraries are technology specific and are generally provided by ASIC vendor like TSMC, Artisan, and IBM etc. In addition to standard cell libraries, reference libraries contain I/O and Power/Ground pad cell libraries. The standard cell library is categorized into Base library and Low power Library. The low power library consists of : Integrated Clock gating cells, Power Gating cells (MT-CMOS), Level shifters, Retention Flops etc.

- **Clock gating integrated cells :** Design Implementation. Clock being the highest frequency toggling signal contributes maximum towards the dynamic power consumption in the SoC even when the flops that are being fed by the clock are not changing their state. So, it is practical to gate the clock from reaching the set of registers or maybe some block in a design to save on the dynamic power consumption. In standby mode, only a sub-system of your entire SoC is working. Hence to save on the power consumption, one can employ clock gating. Instead of using an AND or an OR gate for clock gating which is vulnerable to glitchy output, design engi-

neers prefer to use the Clock Gating Integrated Cell (CGIC) to completely obviate the problem. Here's the circuit of a CGIC.

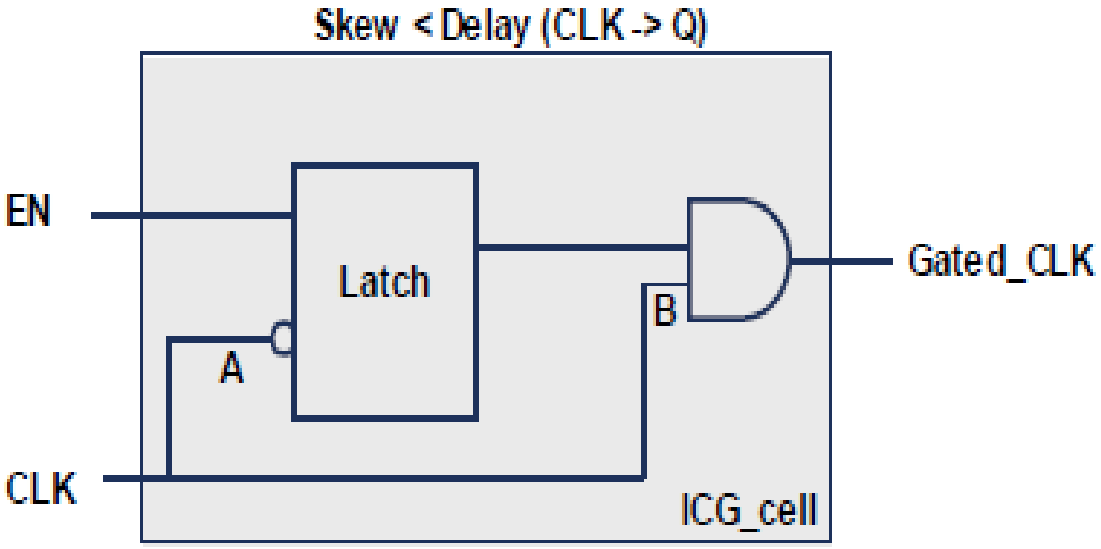

Figure 2.1: Integrated Clock Gating

- **Power Gating :** Power Gating is another effective implementation employed in Low Power Designs. Unlike Clock Gating, which saves the dynamic power, Power Gating also saves the leakage power. As we move from micron (i.e. greater than 90nm) technology nodes to sub-micron (i.e. less than 90nm) technology nodes, leakage power dissipation dominates the dynamic power dissipation. It is therefore employed very frequently in modern SoCs. Consider any CMOS digital logic circuit consisting of Pull-Up Network (made from PMOS transistors) and Pull-Down Network (made from NMOS transistors), as shown in the Figure 2.2. At any point of time, if a direct path would exist from the power supply (VDD) and the ground (GND), the circuit would continue to dissipate leakage power . One can gate the power and ground terminals from the circuit when it is not intended to be used which is accomplished by a power gate. The circuit is shown in Figure 2.3.

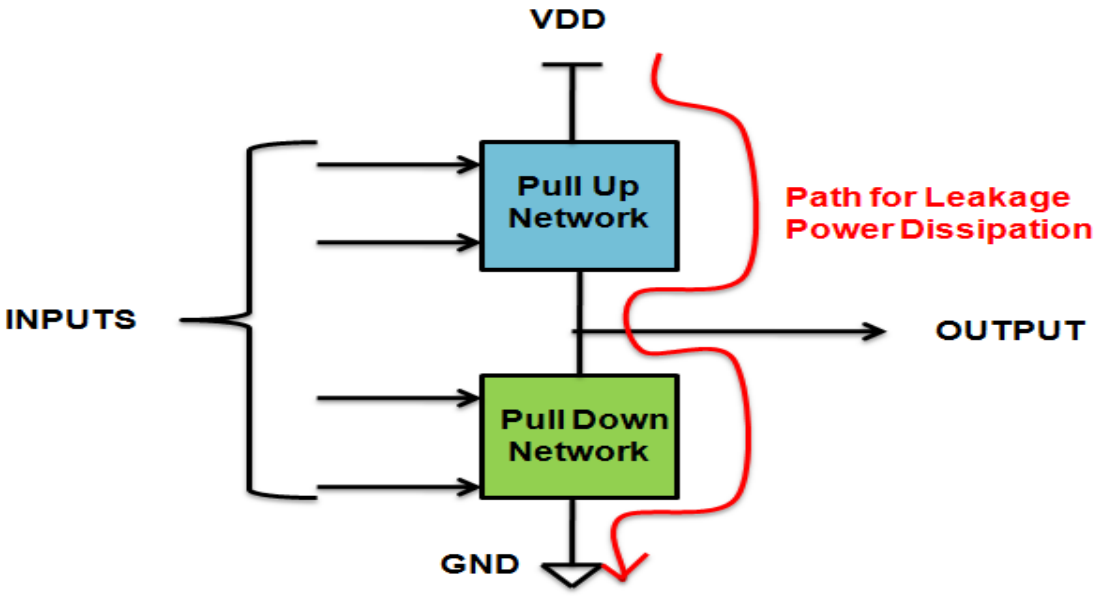

Figure 2.2: Pull-up Pull-down network

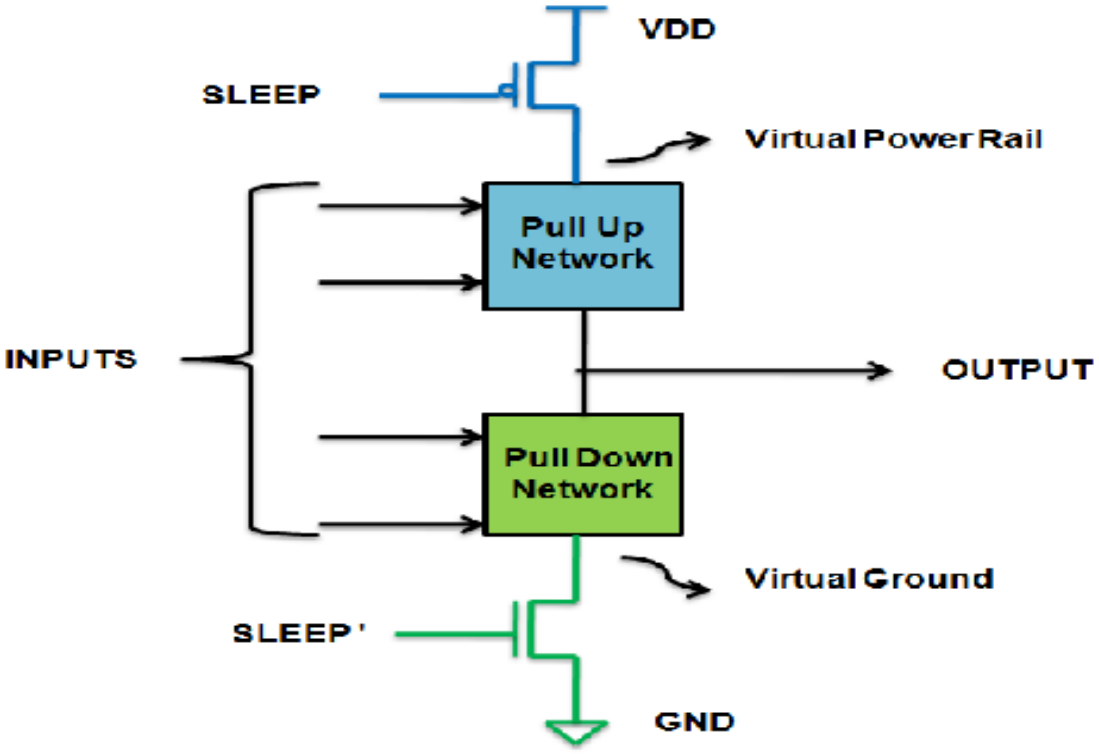

Figure 2.3: Power Gating

- **Retention Flops :** Power Gating Cell helps in minimizing the leakage power consumption of an SoC. The basic idea is to cut the direct path from the battery (VDD) to ground (GND). Though efficient in saving the leakage power, the implementation discussed suffers from one major drawback! It does not retain the state! That means, once power of the SoC is restored, the output of the power gated cell goes to 'X'. with no surety whether it is logic 1 or a logic 0 . If this X propagates into the design, the entire device can go into a metastable state. The below circuit has two parts.
  - The one inside the red oval is same as the normal power gating structure.
  - The one inside green box (on the right) is the additional circuitry required to enable this device to retain it's state.

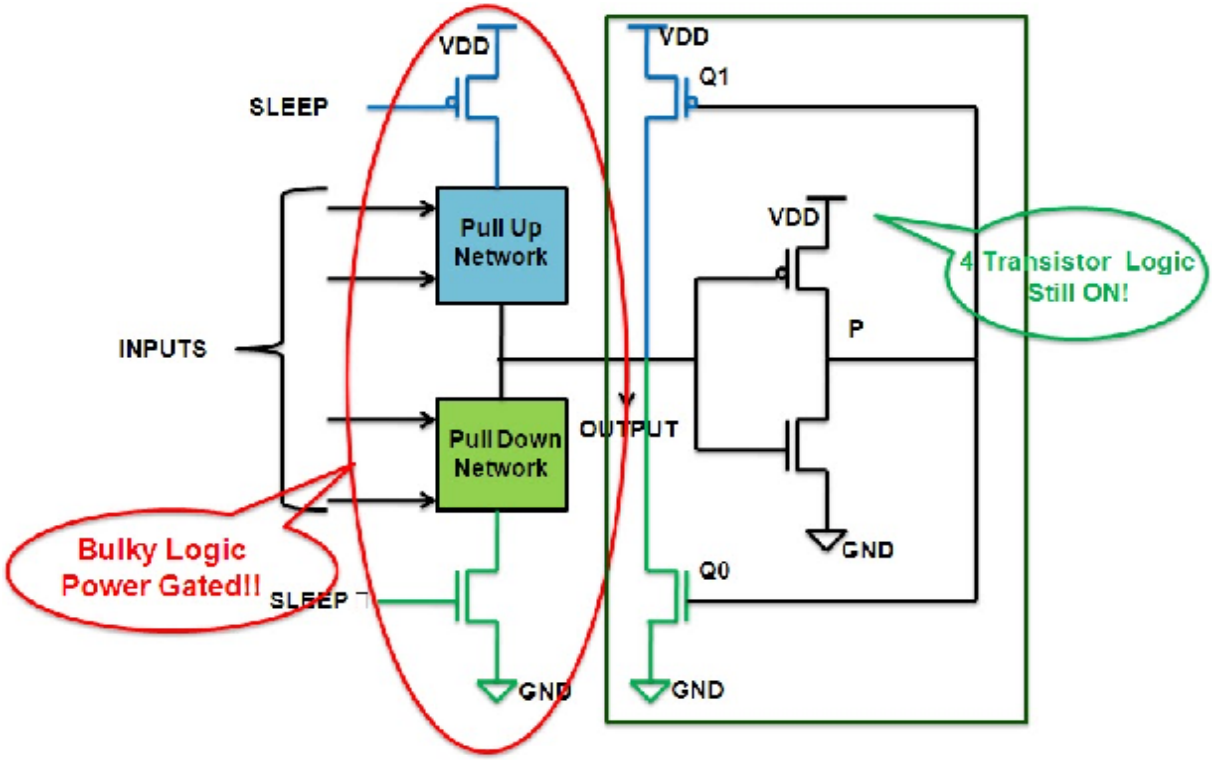

Figure 2.4: Retention Flop

Operation: Before going into the SLEEP mode, the device had the output as logic 1. After entering the SLEEP mode (power off), the sleep transistors come into action and cut the power and ground rails of the device and hence save the leakage power.

But the logic on the right (in green rectangle) is still ON! The output of the inverter would now become OUTPUT', i.e., logic 0. This would in turn enable the PMOS transistor Q1 and output would be restored back to logic 1. Same is true when the output would be logic 0 before power gating. In that case the NMOS transistor Q0 would come into action to help the output node retain it's data.

All this while, when the device is in sleep mode, the output node would continue to leak. By adding the additional circuitry, as demonstrated, we are basically trying to create a feedback loop, which again helps in retaining the state. The hit, of course, is the leakage power of 4 transistors. However, the standard cell logic (in red oval) is usually bulky. Even a simple 2-input NAND gate has 4 transistors itself. And higher order input would have more! Same technique can be applied to any sequential device like a Flip Flop, latch or even a clock gating integrated cell.

- **Level Shifters :** Level shifters are used in multi VDD design, Because in multi VDD design different blocks are working on different voltages. So when a signal passes from one voltage domain to another voltage domain the level shifter is needed, particularly when a signal passes from low voltage domain to high voltage domain. The level shifter will convert one voltage level from to another voltage .

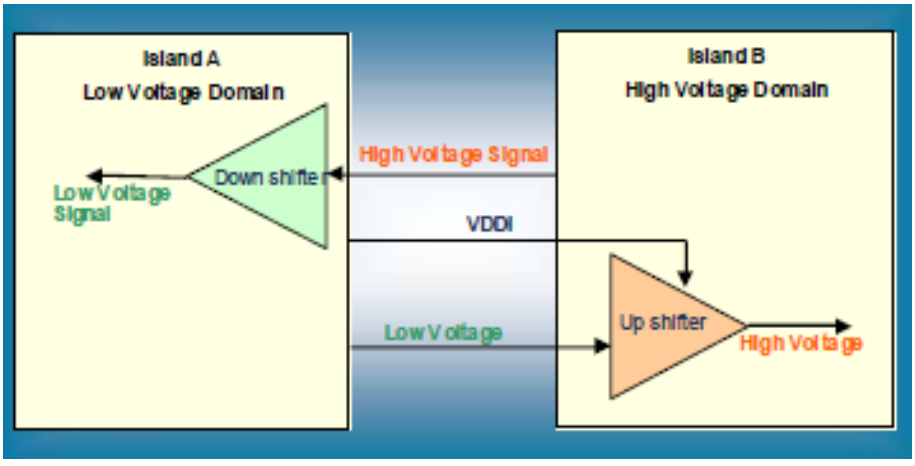

Figure 2.5: Level Shifter

2.2 Logical effort

The The section concept of Logical effort on which Prosizer tool works on the logical. This concept can be used to estimate the delay of the circuit. It is better to get overview of logical effort theory so that we can get better understanding of the tool. Logical effort theory is explained below.
The method of logical effort is an easy way to estimate the delay in an MOS circuit. The method can be used to decide the number of logic stages on a path and also what should be the size of the transistors. Using this method we can do a simple estimations in the early stages of design, which can be a starting point for more optimizations.

The logical effort of a gate tells how much worse it is at producing output current than an inverter, given that each of its inputs may contain only the same input capacitance as the inverter. Reduced output current means slower operation, and thus logical effort number for a logic gate tells how much more slowly it will drive a load than an inverter would.Equivalently, logical effort is how much more input capacitance a gate presents to deliver the same output current as an inverter.[3]

The model describes delays caused by the capacitive load that the logic gate drives and by the topology of the logic gate. Clearly, as the load increases, the delay increases, but delay also depends on the logic function of the gate. Inverters, the simplest logic gates, drive loads best and are often used as amplifiers to drive large capacitance. Logic gates that compute other functions require more transistors, some of which are connected in series, making them poorer than inverters at driving current. Thus a NAND gate has more delay than an inverter with similar transistor sizes that drives the same load. The method of logical effort quantifies these effects to simplify delay analysis for individual logic gates and multistage logic networks.[3]

The first step in modeling delays is to isolate the effects of a particular integrated circuit

fabrication process by expressing all delays in terms of a basic delay unit particular to that process.2T is the delay of an inverter driving an identical inverter with no parasitic . Thus we express absolute delay as the product of a unit less delay of the gate d and the delay unit that characterizes a given process:

$$dabs = d\tau \tag{2.2.1}$$

The delay incurred by a logic gate is comprised of two components, a fixed part called the parasitic delay p and a part that is proportional to the load on the gate‘s output, called the effort delay or stage effort f. The total delay, measured in units of τ , is the sum of the effort and parasitic delays :

$$d = f + p \tag{2.2.2}$$

The effort delay depends on the load and on properties of the logic gate driving the load. We introduce two related terms for these effects: the logical effort g captures properties of the logic gate, while the electrical effort h characterizes the load. The effort delay of the logic gate is the product of these two factors:

$$f = gh \tag{2.2.3}$$

The logical effort g captures the effect of the logic gate‘s topology on its ability to produce output current. It is independent of the size of the transistors in the circuit. The electrical effort h describes how the electrical environment of the logic gate affects performance and how the size of the transistors in the gate determines its load driving capability. The electrical effort is defined by:

$$h = C_{out}/C_{in} \tag{2.2.4}$$

As number of transistors is increased, we get better performance but at the same time leakage power is also increased. This is not desirable. So we need to study the causes of leakage power and methods of reducing static power consumption.[3]

2.3 Leakage Current Reduction in CMOS VLSI Circuits

The rapid increase in the number of transistors on chips has enabled a dramatic increase in the performance of computing systems. However, the performance improvement has been accompanied by an increase in power dissipation; thus, requiring more expensive packaging and cooling technology . Historically, the primary contributor to power dissipation in CMOS circuits has been the charging and discharging of load capacitance, often referred to as the dynamic power dissipation. This component of power dissipation is quadratically proportional to the supply voltage level. Therefore, in the past, chip designers have relied on scaling down the supply voltage to reduce the dynamic power dissipation. Maintaining the transistor switching speeds requires a proportionate downscaling of the transistor threshold voltages in lock step with the supply voltage reduction. However, threshold voltage scaling results in a significant amount of leakage power dissipation due to an exponential increase in the sub-threshold leakage current conduction.

Borkar in predicts a 7.5 fold increase in the leakage current and a five-fold increase in total energy dissipation for every new microprocessor chip generation.

There are three main sources for leakage current:

- Source/drain junction leakage current
- Gate direct tunneling leakage
- Sub-threshold leakage through the channel of an OFF transistor

The junction leakage occurs from the source or drain to the substrate through the reverse-biased diodes when a transistor is OFF. The magnitude of the diode‘s leakage current depends on the area of the drain diffusion and the leakage current density, which is in turn determined by the process technology.

The gate direct tunneling leakage flows from the gate through the “leaky“ oxide insulation to the substrate. Its magnitude increases exponentially with the gate oxide thickness Tox and supply voltage VDD. According to the 2001 International Technology Roadmap for Semiconductors, high-K gate dielectric reduced direct tunneling current is required to control this component of the leakage current for low standby power devices. [4]

The sub-threshold current is the drain-source current of an OFF transistor. This is due to the diffusion current of the minority carriers in the channel for a MOS device operating in the weak inversion mode (i.e., the sub threshold region.) For instance, in the case of an inverter with a low input voltage , the NMOS is turned OFF and the output voltage is high. Even when VGS is 0V , there is still a current flowing in the channel of the OFF NMOS transistor due to the VDD potential of the VDS. The magnitude of the sub-threshold current is a function of the temperature, supply voltage, device size , and the process parameters out of which the threshold voltage (Vth) plays a dominant role . In current CMOS technologies , the sub-threshold leakage current is much larger than the other leakage current components . This current can be calculated by using the following equation:

$$I_{DS} = K(1 - e^{(-V_{DS}/V_T)})e^{(V_{GS}-V_T+V_{DS})/nV_T} \tag{2.3.1}$$

Where K and n are functions of the technology. Clearly, decreasing the threshold voltage increases the leakage current exponentially. In fact decreasing the threshold voltage by 100 mv increases the leakage current by a factor of 10. Decreasing the length of transistors increases the leakage current as well. Therefore, in a chip, transistors that have smaller threshold voltage and/or length due to process variation contribute more to the overall leakage. Although previously the leakage current was important only in systems with long inactive periods (e.g., pagers and networks of sensors), it has become a critical design concern in any system in today‘s designs.

2.4 PVT

PVTs are inter-chip variation which depend largely on external factors like: the ambient temperature ; the supply voltage and the process of that particular chip at the time of manufacturing.

- **Variation in Process :** There are millions of devices (standard cells); and probably billions of transistors packed on the same chip. You can expect every single transistor to have the same process or the channel length. If manufactured chip is with worst process, it means that the channel length tends to deviate towards the higher side. This variation may be more for some transistors and less for some. It can be a ponderous task to quantify this variation between the transistors of the device, and is often modeled as a percentage deviation from the normal[5].

- **Variation in Voltage :** All the standard cells need voltage supply for their operation. And voltage is usually 'tapped' from the voltage rail via interconnects which have a finite resistance. In two parts of the chip, it is fairly probable for the interconnect length to be different, resulting in a finite difference in the resistance values and hence the voltage that the standard cells actually receive. So the voltage received by standard cell might be different.

- **Variation in Temperature :** Some parts of the chip can be more densely packed or might exhibit more active switching as compared to the other parts. In these regions, there is a high probability of the formation of localized 'HOT SPOTS' which would result in increased temperature in some localized areas of the chip. Again, this difference might be order of a few degree centigrade, but can be significant [5].

2.5 Summary

Chapter provides theoretical background of Standard Cell libraries, delay estimation in circuit, power minimization techniques, PVT conditions.

Chapter 3

Standard Cell Optimization

3.1 Overview

Leakage power and critical path delay are very critical parameters for standard cell. It is always desired to have less delay and leakage power. Now these parameters directly depend on sizes of NMOS and PMOS, which are sensitive to critical path delay. So we have to size our transistors such a way that they give optimized power and delay. For this purpose celltuner tool is used. We will see the working of tool and procedure of sizing in next section.

3.2 Celltuner

Celltuner is a tool which is used to perform sizing. For that first we will define our cost function. In most of the case it will be power delay product (power x delay). Celltuner varies sizes of the transistors in steps using bisection algorithm and gives the size combination to Hspice. Hspice simulation is performed on that size combination and the cost product is measured for each combination. Celltuner plots a graph of cost product values related to each combination. It chooses the optimum point of the graph and gives respective width combination as an output. It will generate .CSV file, which will have all the statistical details about cost product.

3.3 Procedure

Celltuner uses different input files for performing sizing. Needed information is provided to celltuner through these files. Some files have project related information while other files has project independent information. The detailed explanation of each file is given below :

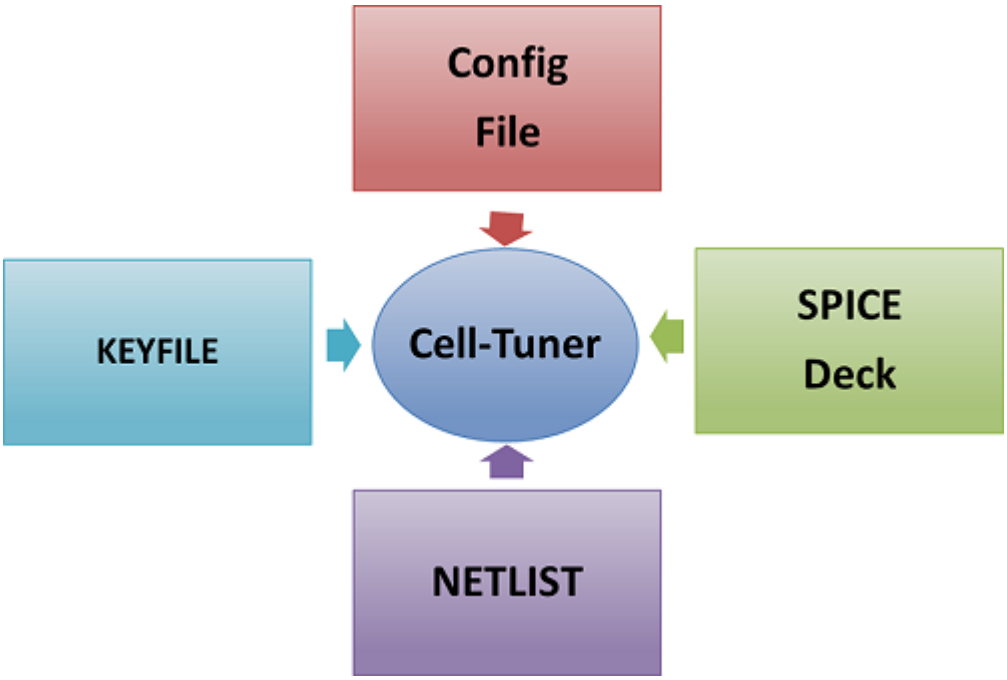

Figure 3.1: Celltuner Flow

3.3.1 Configuration Settings

The Configuration file consists of Process related setting , architecture related settings and Setup related settings. Following are the required configurations in .config file:

# Cell family name defined here
ctFamilyName=XOR2
# Cell family drive strengths listed here
ctDriveStrengths=X1M_A12TR_C35
# Tuning objective

ctObjective= pwrdly, pwr1.5dly, pwr2dly

Device step size used in simulation

ctStepWp=5.00000e-08

ctStepWn=5.00000e-08

Transistor groupings defined here, syntax is as follows:

P transistors : N transistors : (beta range)

ctGroups=

XMXPna0 XMXPA1 : XMXNA1 XMXNna0 : BetaR_tgate_d,

XMXPA0 : XMXNA0 : BetaR_inv_d_d,

Transistor size ranges defined here, hold for all drives

ctXtors=

XMXPna0 XMXPA1 : pfet_width_min 0.65,

The values of the variables used in the .conf file will be passed through these files. Cell related information like drive strength (current capability of driving the output), beta ratio (ratio of width of PMOS to the width of NMOS) of different instances, transistor grouping are provide to Celltuner through this file. Here, we will group those transistors which can be tuned together. Company follows some guidelines in which range of beta ratio, transistor groupings are mentioned for each cell.

3.3.2 Key File

This file is used to replace instance names with general recognizable names. This will be helpful while analyzing results. Also keyfile is used to generate .csv data. The example of key file is shown below:

#result vdd_energy rdelay max_ar max_af adelay max_br max_bf bdelay

Input A inv(p): XMXPA0

Input A inv(n): XMXNA0

Input B inv(p): XMXPB0

Input B inv(n): XMXNB0

TG1 (p): XMXPna0

TG1 (n): XMXNA1

For example, if XMXPA0 is used in result then we may not get idea about inverter PMOS. But if Input A inv(p) is mentioned instead , then user can easily understand. So after regression celltuner will pick instance names from this file and replace them with general names and display result in .CSV file. vdd_energy ,rdelay, max_ar, max_af, adelay, max_br , max_bf, bdelay are power and delay parameters which are measured by HSPICE for each size combination.

3.3.3 Netlist

An example netlist is shown in chapter 4 , section 4.2 .

3.3.4 Ct-deck file

Example of ct deck cannot be shown as it is confidential information. This file controls the HSPICE simulations. First supply voltages (VDD, VSS etc) are defined. Input stimulus (vA) is defined. The values will be passed to the variables of this file through header files in .conf which will have cell related information. It supplies transient analysis parameters (target and trigger values) to HSPICE. At last cost function is defined, which will be measured by HSPICE for each width combination.

3.4 Result

After measuring cost function for each width combination, Hspice gives result to celltuner. Celltuner plots the graph for each combination and finds the optimum point of the graph, which has desired width combination. The graph is shown below. It shows total number

of combinations which have been tried by the celltuner. Various parameter values are displayed. In second line as we can see, it has found the best possible width combination for which cost product will be optimized.

As shown, it measures three cost functions depending on the power proportion. The values of power and delay for different cost functions are shown in three different colours.Where X-axis is Delay and Y-axis is Dynamic energy. As we can see, the optimum point of the convex hull will have minimum value of cost product. So celltuner will take sizes of that combination and display it in .CSV file.

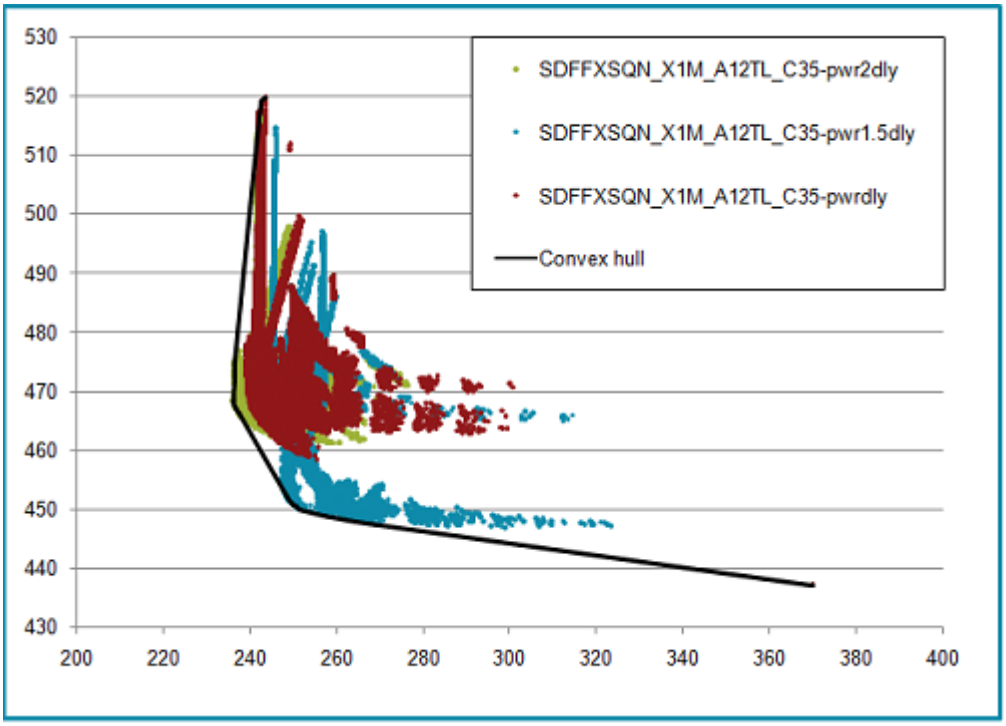

Figure 3.2: Graph generated by celltuner

3.5 Summary

This chapter explains how celltuner is used to obtain optimized devices sizes providing optimized cost product.

Chapter 4

Schematic Sizing using Prosizer

4.1 Prosizer Overview

Prosizer is an automated tool which is used for schematic sizing because source library (sc arm) is technology independent, schematics in source library don't have sizing parameters. To dump sizes to schematic it has size from cdl option and size from LBDOEE result file. It uses one prosizer technology file (.tcl) in which we can mention needed details about source schematic library, target library, cdl-directory etc. cdl file is similar to netlist file a sample netlist file and LBDOEE result file is shown below section.

For schematic conversion previously GENIE script was separately used to convert arm primitive schematics to foundry schematics. But now GENIE has been integrated to Prosizer that does both schematic conversion and sizing. The only thing we need to provide for schematic conversion is the GENIE configuration file in the Prosizer technology file (.tcl). Prosizer uses the below two options as input for copying sizes to schematics :

- **Size from CDL file :** It will first pick a cdl file from mentioned cdl directory and find a match for that cdl in mentioned source library, then it copies sizes from that cdl and dumps a flat schematic in target library.
- **Size from LBDOEE Result file :** The lbdoee result file contains various Beta ratios

for different reference cells like nand ,nor ,inverter and also logical effort for each cell, based on the concept of logical effort it will copy sizes from that result file to schematics and dumps a flat schematic in target library.

A sample CDL file and LBDOOE file is shown below :

```
#########################
.SUBCKT INV_X1M_A12TR_C35 Y VDD VNW VPW VSS A
*.PININFO A:I Y:O VDD:B VNW:B VPW:B VSS:B
    MPY Y A VDD VNW pch_lvt_mac l=35.0n w=310n
    MNY Y A VSS VPW nch_lvt_mac l=35.0n w=300n
.ENDS
#########################
inv_x1b 1.332
inv_x2b 1.332
nand2_x1b 0.718
nand2_x2b 0.641
nor2_x1b 2.512
nor2_x2b 2.395
nand2_le_m 1.430
nand3_le_m 1.909
#########################
```

To convert from arm primitive to foundry primitive Prosizer uses GENIE configuration file as an input. As discussed before PMOS and NMOS symbols are different for different foundry .There is an offset in the bulk terminal of PMOS and NMOS after conversion, to adjust the bulk offset GENIE configuration file is used. A sample GENIE configuration file is shown below:

```
#########################
deviceNameMapTbl ["nfet4"] ["SVT"] = "nfet"
deviceNameMapTbl ["nfet4"] ["HVT"] = "hvtnfet"
deviceNameMapTbl ["nfet4"] ["LVT"] = "lvtnfet"
```

foundryDeviceXYOffset = 0.0625 * 4

foundryDeviceGTermExtend = 0.0625

foundryBulkNodeOffset = 0.0625

foundryDeviceBulkNodeWireShift = 0.0625

########################

A sample Prosizer technology file (.tcl) with details about project mentioned in this file is shown below :

########################

ps_option –source-lib <source library name >

ps_option –source-path < source library path >

ps_option –target-lib <target library name>

ps_option –target-path <target library path >

ps_option –ap-tech <arm primitive tech. file >

ps_option –cds-tech-lib <project tech. library name >

ps_option –cds-tech-path <project tech. library path >

ps_option –doee-result-file <LBDOEE .result file >

ps_option –size-from-cdl 1

ps_option –source-cdl-dir < cdl directory path >

ps_option –genie-config-file <genie configuration file >

########################

If any new topology is introduced in the source then very specific constraint is needed for that particular cell. We can specify this using constraints parameter via the prosizer tcl file. The constraint parameter are show below :

########################

ps_set parameter variation [list ALL BUFCAP] BUFCAP

ps_set parameter variation [list ALL INVCAP] INVCAP

ps_set parameter variant [list ALL [list AND] 2 M] _ALT

########################

The cell name appears to be : INV_X1M_A12TR_C35 , where INV is the function name

, X1M stands for drive strengtht , A12T stands for 12 Track library, R stands Regular VT , C35 stands for technology. If the cell name doesn't have VT information then adding following switch to prosizer setup will have that specified VT as output VT in the target library .

##########

ps_set parameter vt_type [list] SVT

##########

As discussed before PMOS and NMOS symbols are different for different foundry. The instances are named according to threshold Voltage (VT) it carries. The source library carries Standard VT while the foundry may have different VT it can be HVT or LVT. Different VT library will have different symbol name so to adjust the PMOS and NMOS symbol names following mapping is used in prosizer technology file:

##########

ps_option –source-tech-model-map

[list nch_mac [list nch_lvt_mac]

pch_mac [list pch_lvt_mac]

nch_svt_mac [list nch_hvt_mac]

pch_svt_mac [list pch_hvt_mac]]

##########

As shown above we can mention techlib file , Arm primitive tech file , source schematic library , target library , input cdl directory etc. Finally Prosizer dumps a flat schematic in target library. The models of NFET and PFET are defined in techlib.

4.2 Result

We can use cadence virtuoso tool for viewing the schematics . Source schematic and Schematic after conversion are shown below .

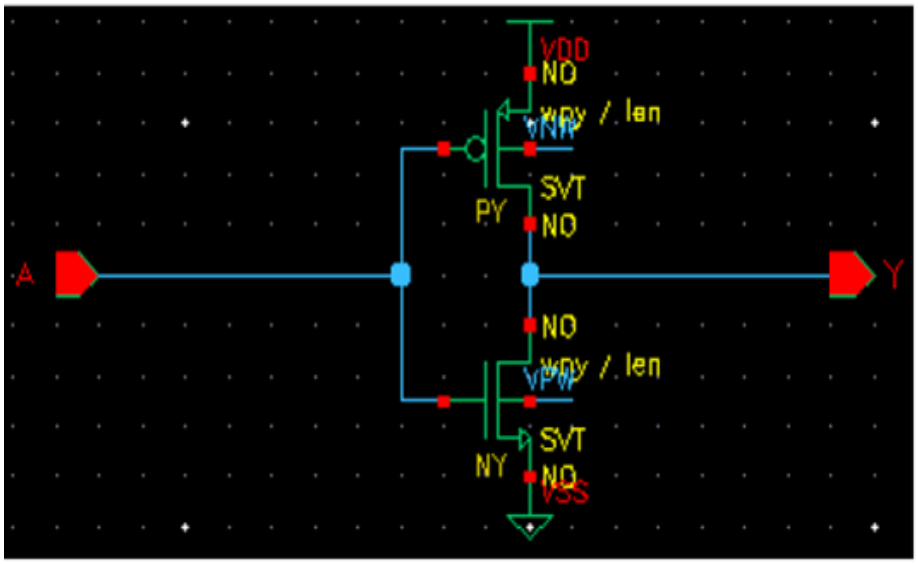

Figure 4.1: FLAT INV schematic in sc arm library

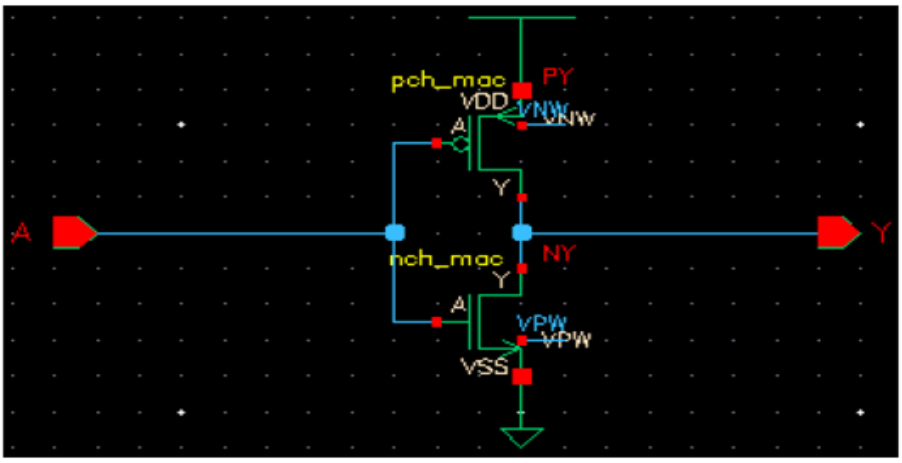

Figure 4.2: FLAT INV schematic in Target library

4.3 Summary

Chapter describes in detail about how Prosizer uses concept of logical effort for schematic sizing using CDL and LBDOEE file as input.

Chapter 5

Cell Level Physical Checks

5.1 DARBuilder Overview

Various Cell level Physical checks and Electrical Checks are performed to estimate changes in Standard Cell Library. DARBuilder is a tool which is used for performing various cell level Physical checks & Electrical checks. The Cell Level checks are LVS/SVS/Cadence Symbol.

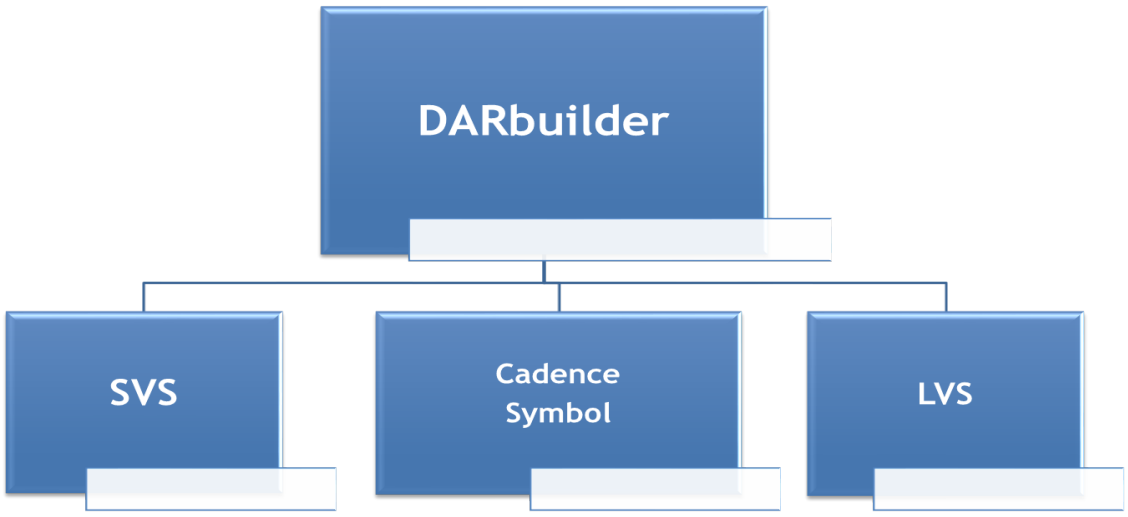

Figure 5.1: Cell Level Physical Checks

The tools takes in CBL file as input. In the CBL file we can turn on those checks which we require to perform on the Standard Cell library. A list of xml files are given as input to the tool in the CBL file. Detailed explanation of SVS and LVS is ias below:

5.2 LVS (Layout versus schematic)

A successful Design rule check (DRC) ensures that the layout conforms to the rules designed/required for faultless fabrication. However, it does not guarantee if it really represents the circuit you desire to fabricate. This is where an LVS check is used. Layout Versus Schematic comparison compares the layout and schematic cell views. LVS is used to ensure that your layout is identical to the source schematic. A complete LVS flow is shown in figure below .

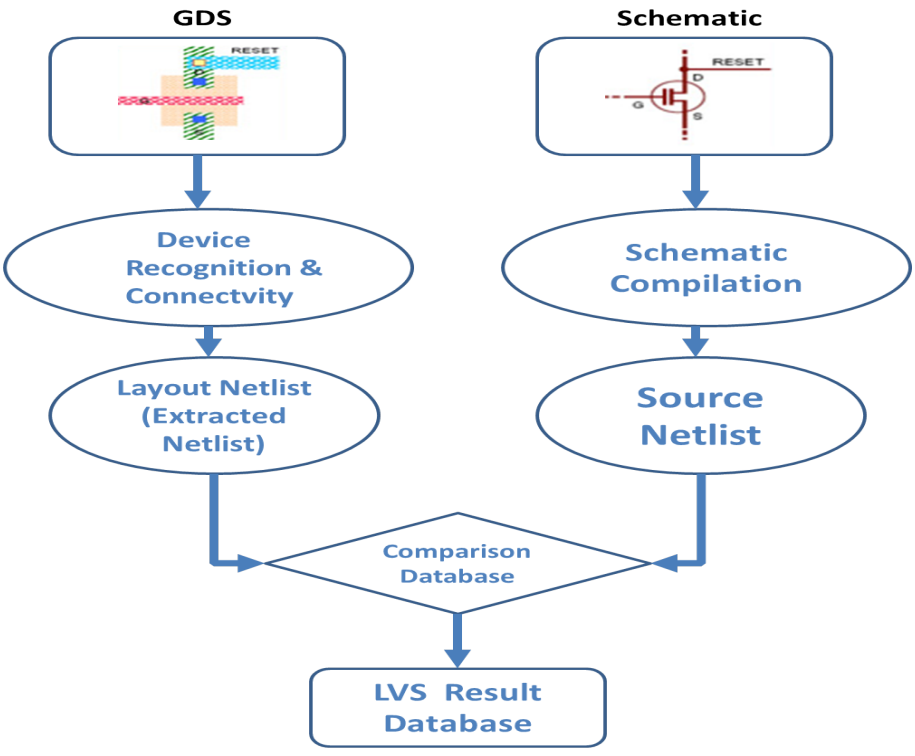

Figure 5.2: LVS Flow

LVS recognizes the drawn shapes of the layout that represent the electrical components of the circuit, as well as the connections between them. LVS works by generating a new net list for each circuit. The netlist is compared by the "LVS" tool against a similar schematic or circuit diagram's netlist. If any discrepancies are found, LVS will display them. Before LVS can be run on layouts, the layout must be extracted.[6]. Typical errors encountered during LVS include:

- **Shorts**: Two or more wires that should not be connected have been and must be separated.
- **Opens**: Wires or components that should be connected are left dangling or only partially connected.
- **Component Mismatches**: Components of incorrect type have been used (e.g. a low Vt MOS device instead of a Standard Vt MOS device)
- **Missing Components**: An expected component has been left out of the layout.
- **Parameter Mismatch**: Components in the netlist contain properties. The LVS tool can be configured to compare these properties to a desired tolerance. If this tolerance is not met, then the LVS run is deemed to have a Property Error. A parameter that is checked may not be an exact match, but may still pass if the lvs tool tolerance allows it. (example: if a resistor in a schematic had resistance=1000 (ohms) and the extracted netlist had the a matched resistor with resistance=997(ohms) and the tolerance was set to 2%, then this device parameter would pass as 997 is within 2% of 1000.

5.2.1 Result

Tool will generates output file, LVS Output File provides a lot of useful information about a cell, including the number of devices, nets, etc. within the cell. It also lists some results that can be useful in tracking down errors that caused LVS not to pass.

5.3 SVS (Schematic versus schematic)

SVS (Schematic versus schematic) check is used to ensure that Standard Cell schematic is an exact topographical and parametric match to the project schematic. Complete SVS flow is shown in the figure below.

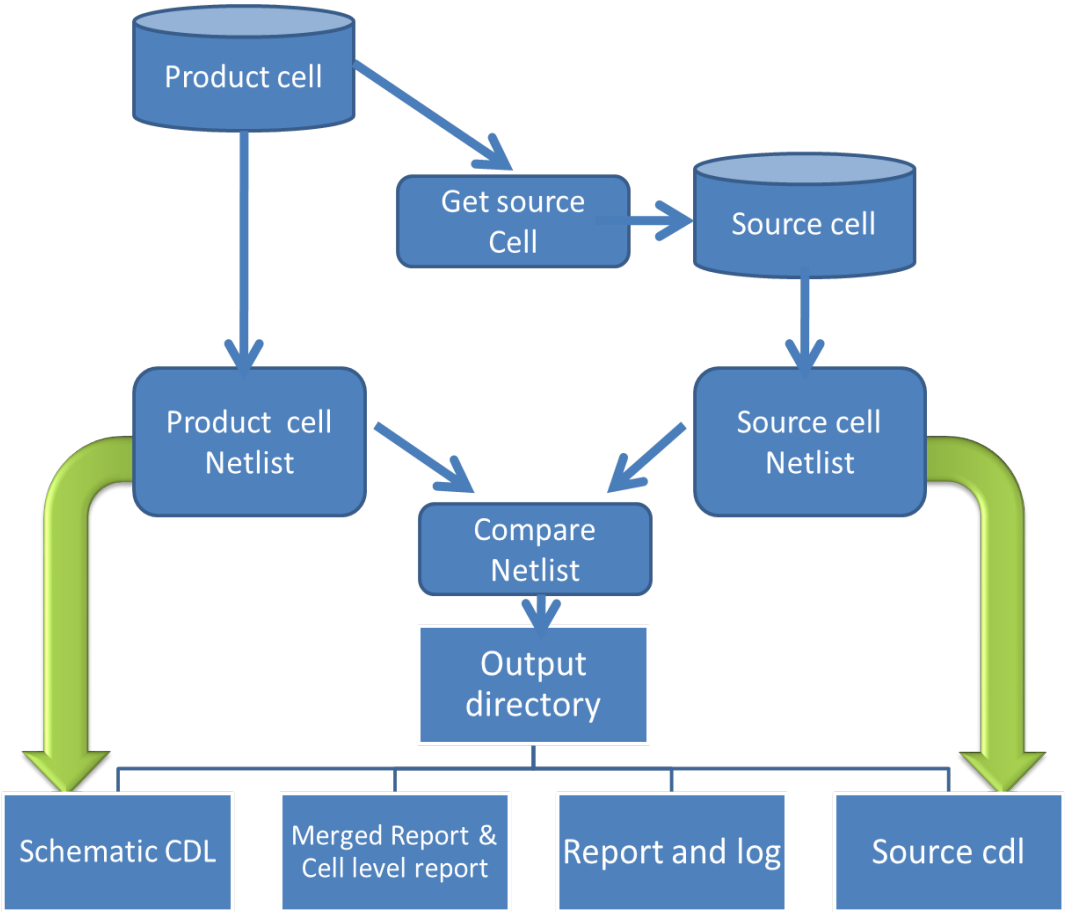

Figure 5.3: SVS Flow

After sizing schematic using Prosizer, we need to make sure whether the sized schematics are exactly same as Source schematics or not. This comparison is known is SVS (Schematic vs. Schematic) check. SVS check is CDL dependent, DARBbuilder generates CDL from the schematics. Various xml's are given as input to the tool in CBL file. The Calibre tool then compares the source CDL and the schematic CDL and gives the result PASS or FAIL. DARBuilder generates a cell level and top level report file giving information about the cell having Passed or Failed. If there exist some error in converted schematic (using Prosizer), the mismatch will be highlighted in the Merged report file , (ex) Internal nets connectivity errors, Instance name mismatch, Incorrect Vt mapping are captured using SVS check. The example .cbl file is shown below :

##########################

PropFoundry=tsmc

PropTechnology=cln28hpm

PropTechLibraryName=cln28hpm

PropLibraryName=sc1_planar_fast

PropLogFile=sc1_planar_fast_svs.log

PropReportFile=sc1_planar_fast_svs.rep

PropNoCDL=false

PropNoCDLHerculesExtract=true

PropNoLibrary=true

PropNoCDLCalibreExtracted=true

PropNoLVS =true

PropNoSVSCheck=false

PropConfigFileList= < XML file list >

##########################

As shown above, there are various options for various checks. We can enable required option for the SVS check.

5.3.1 Result

Tool will generate the report for SVS check which will have detailed information about instances, input, output nodes etc.

5.4 Cadence Symbol Check

The purpose of this check is to check if the pins for Standard Cell in the symbol views are located on the grid. This check is activated by:
##########################
PropNoCadenceSymbolCheck=false
##########################
The grid step are configured in the xml , the default is 0.0625. The check results are shown as below:
Checked cell: INV_X1M_A9TR_C600
param Grid_Pin: 0.0625
CHECK GRID_PIN on VPW : PASS
CHECK GRID_PIN on Y : PASS
CHECK GRID_PIN on A : PASS
CHECK GRID_PIN on VDD : PASS
CHECK GRID_PIN on VSS : PASS
CHECK GRID_PIN on VNW : PASS

5.5 Summary

This chapter describes how DARBuilder is used to perform various Physical Checks to validate functional topology equivalence.

Chapter 6

Parasitic Extraction

6.1 Introduction

Currently with the technology scaling, the parasitic effects of the interconnects have become dominant influencing the performance of VLSI circuits. For the timing verification with high precision, fast and accurate parasitic extraction of interconnects is required. For high-performance circuit design, it is important to make timing verification at the early stage of physical design. This can ensure a faster design closure and reduce the time to market. So, after the placement of cells, we can perform a static timing analysis (STA) to find out the signal path violating the timing constraints. Parasitic extraction method takes in the locations of cell pins, and constructs the virtual routes for signal nets with LPE(Layout parasitic Extraction) method. The extracted netlist is needed for various purposes including circuit simulation, static timing analysis, signal integrity, power analysis and optimization, and logic to layout comparison.

Extraction often helpful to make an (informal) distinction between designed devices, which are devices that are deliberately created by the designer, and parasitic devices, which were not explicitly intended by the designer but are inherent in the layout of the circuit.

Primarily there are three different parts to the extraction process. They are designed device extraction, interconnect extraction, and parasitic device extraction. These parts are inter-related since various device extractions can change the connectivity of the circuit, e.g., resistors (whether designed or parasitic) convert single nets into multiple electrical nodes. Usually one level of interconnect extraction is used with designed device extraction to provide a circuit for simulation or gate-level reduction, and a second level of interconnect extraction is used with parasitic device extraction to provide a circuit for timing analysis. Below section describes effect of Interconnect resistance and capacitance.

6.2 Interconnect Capacitance and Resistance Extraction

Interconnect wires are responsible for signals and power from one end to other end . Interconnect resistance and wires obstruct these signals/Powers in circuit. Effects of these are : 1] Impact on Delay 2] Energy consumption 3] Power Distribution 4] Introduction of noise source which affects reliability. A signal propagation delay in a silicon design consist of two parts :

- Gate delay
- Interconnect delay

The impact of these interconnect parasitic on a path delay may vary significantly from one path to another. For one path, it may be very less because the cell delay is dominating and for other it may be high because the interconnect delay is dominating . Which ever the delay is dominating plays a important role in design.

6.3 Extraction Corners

To find out the impact of resultant delay there are models with help of RC corners. There are five parasitic extraction corner : 1] C-Best 2] C-worst 3] RC-best 4] RC-worst 5] Typical . Each of the corner is explained as below:

- **C-Best :** It has minimum capacitance. Also know as cmin corner. Interconnect resistance is larger than typical corner. This corner results in smallest delay path with short nets and can be used for minimum path analysis.
- **C-worst :** Refers to corner which results in maximum capacitance. So also known as Cmax corner. Interconnect resistance is smaller than typical corner. This results in larger delay paths with short nets and can be used for max-path analysis.
- **RC-Best :** Refers to corner which minimizes interconnect RC product. Also know as RC-min corner . Corner has smallest path delay for paths with long interconnects and can be used for min-path analysis.
- **RC-Worst :** Refers to the corner which maximizes interconnect RC product. Also know as RC-max corner. Corner has largest path delay with long interconnects and can be used for max-path analysis.
- **Typical :** This refers to interconnect resistance and capacitance.

6.4 CalibreXRC

LVS extracts the intended devices and parasitic extraction extracts the unintended resistances and capacitances that occur. Using a single-tool flow for verification and extraction provides results that are consistent across the flow. I have used CalibreXRC and Calibre LVS feature as a part of project, Calibre LVS and Calibre xRC use the same data and rule files and Calibre xRC reads Calibre LVS structures directly. The Tool supports both SPEF and DSPF extraction. The input to tool is GDSII layout database and process technology information. The process information can take the form of parasitic data tables, dielectric constants, permitivities, or even an existing LPE file [7] . The extraction flow works as below:

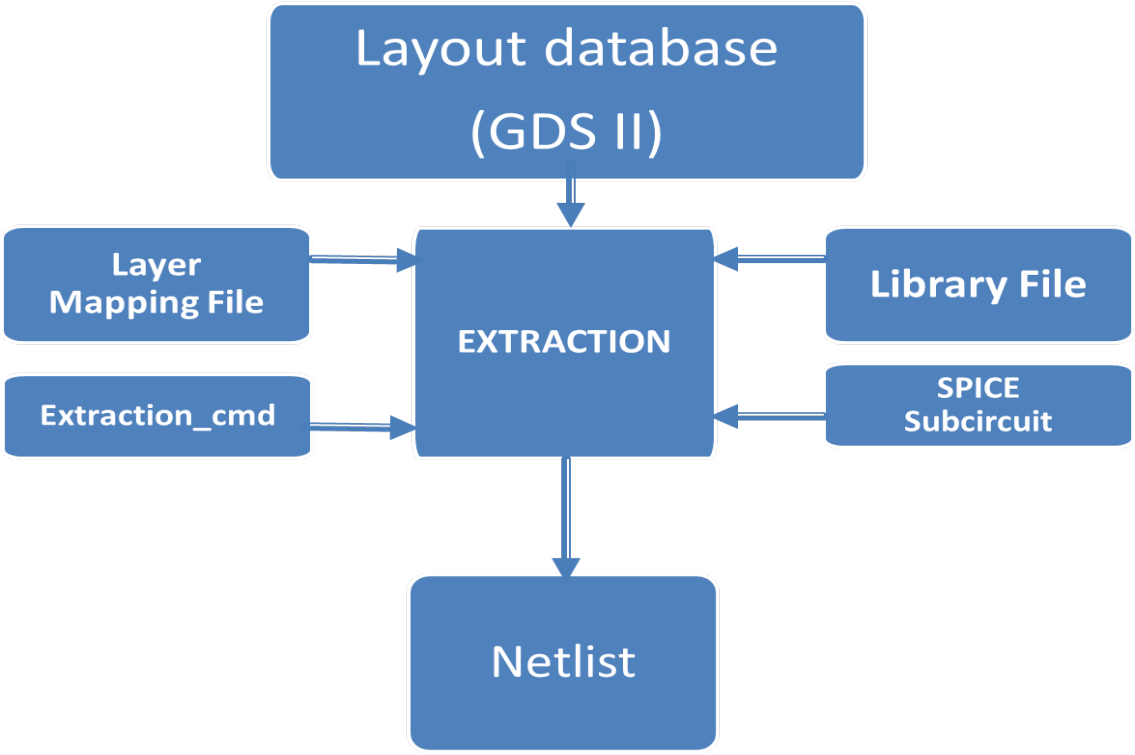

Figure 6.1: Extraction flow

6.5 Summary

The chapter explains how Parasitic extraction is carried out and how extracted information helps for static timing analysis, signal integrity, power analysis and optimization, and logic to layout comparison.

Chapter 7

EDAR Regression

7.1 Significance of EDAR Regression

EDAR stands for Electrical Design Assurance Report. After creating foundry schematics using Prosizer, various Cell level Physical checks and Electrical Checks are performed to ensure various Physical and Electrical properties of the standard cell . These Electrical Checks are performed under various PVT(Process ,Voltage and Temperature) conditions. For every PVT condition the observations are recorded. It is to make sure whether the circuit is able to with stand under the specified PVT Conditions or not. Analysis is made for Failing cells .

7.2 Cell Level Electrical Checks

The following are the Electrical Checks performed on the foundry schematics. The detailed explanation of these electrical checks is as below :

- **Flop Margin Analysis (FMA) :** Because of extreme process variation and temperature, the risk of a flop failing in silicon is more than on design. So by providing a limit on the number of expected failures for the flops it can help the customer to avoid use of certain cells and costly mask set revisions. The presence of an internal

race condition is one of the most likely failure mechanisms in a flop. By directly simulating a flop we can predict it failure rate and then it can be tested to ensure a value better than 1 failure out of many .

- **Balanced Beta Ratio Check (BBR) :** Cells should have balanced rise and fall delays, so that they meet specific target . After calculating the rise delay and fall delay , the cell should meet the following criteria , if the following criteria is not meet the cell fails for BBR check.
  (rise delay - fall delay)(rise delay + fall delay) < 5%
  BBR check is performed to ensure that cells with balanced rise/fall properties meet a specific target.

- **Latch Node Stability Check (LNS) :** Sequential elements flip their state when subjected to voltage variations on the power rail or ground rail. When these fast variations on power and ground rails are induced in cells containing sequential elements; the cell output is observed for glitches. The glitching simulated on the power and ground rails consists of large variations above and below the applied nominal voltages. Simulations are performed with the sequential element holding, both low logic value and high logic values, as well as at two different nominal supply voltage conditions. The cell output is monitored for glitches and variation of 10% of VDD beyond the expected value of the output node is said to be the threshold for pass/fail .

  LNS check is performed to ensures that nodes in the latch are retained in a proper state even in presence of power supply noise.

- **Level Shifter Range of Operation :** Design assurance for level shifters confirms their functionality across a wide range of conditions and also indicates their relative tolerance to process variation (when statistical simulation is supported by foundry models). The check measures the rising and falling delays through both the level shifter and a reference circuit. Both circuits are subject to local variation and various voltage conditions (upshift, downshift, no shift).

A proprietary variation metric is calculated for each resultant delay distribution; the metric for each level shifter is compared to the metric calculated for the reference circuit. If the level shifter metric is less than or equal to the 1.05 times the reference metric at each condition, it is said to have passed the check (i.e. a 5% tolerance is applied to the reference limit). Additionally, if the absolute value of the metric indicates that process variation has little-to-no impact upon delay for both the level shifter and reference circuit, the level shifter is considered to have passed this test.

- **ElectroMigration :** Current density is the primary factor influencing electromigration. By increasing the wire width, current density is reduced and susceptibility to EM is reduced. There is one exception to this rule and that is when the wire width falls below the average grain size of the interconnect material . This apparent contradiction is caused by the position of the grain boundaries, which in such narrow wires lie perpendicular to the width of the whole wire .[8]

  For EM check the current density guidelines are provided by foundries that help customers‘ designs meet a fixed, length of time, in hours, that electrical power is applied to a device is target free of Electromigration (EM)-related failures. A passing status for the DAR EM check indicates that a cell has wire widths and lengths that are in with these foundry EM guidelines. Simulations are performed at the foundry-recommended EM PVT condition. The EDAR EM check uses spice simulation to find EM violations.

7.3 EDAR Flow

For running EDAR regression DARBuilder is used. It uses .cbl file in which different options related to electrical checks ar avialable. We can enable required option for EDAR and run DARbuilder. It takes spice netlist and spice decks as input. It creates an output directory

for each of electrical check specified in the .cbl file. Each electrical check will generates results for different process corners. For each electrical check DARbuilder generates :

- Cell level report file which gives information about the the cell name and a status field indicating one of the following : PASS, FAIL .
- For each process corner it generates .dat files which gives information about Delay Sigma Fall, Delay Sigma Rise, Delay Mean Fall, Delay Mean Rise, COV.

Activate the check which need to be performed in cbl file the detail explanation of cbl file is in the previous chapter. The following switches are added to the cbl file for EDAR regression :

##########################

PropNoBalancedRatioCheck= false

PropNoLatchNodeStabilityCheck=false

PropNoEM=false

PropNoFlopMargining=false

PropNoLVShifterRangeOp=false

PropNetListDir= <SPICE NETLIST PATH> PropEdarBalancedBetaCells = CLK .* | AND2_X.*| OR .*

PropEdarFlopMarginingCells =.*FF.*

PropEdarLatchNodeCells =.*FF.*|.*LAT

PropEdarEMCells = .*

PropEdarLevelShifterCells =.*LVLU.*

##########################

The spice decks are mapped in an xml file. Example of spice deck cannot be shown as it is confidential information. Details about this file is mentioned below.

The deck controls HSPICE simulations. It defines supply voltages (VDD, VSS etc), Input stimulus, DUT, transient analysis parameters (target and trigger values) .

The most important required parameters , Spice Model, Voltage Conditions ,Temperature Conditions, Hspice Options, Monte Statement, all timing parameters namely clock period, Initialization time, stable data delay, target slew and process corners etc. are mentioned in an template xml file which is included in the cbl setup.

The tool takes this data values from xml files and then process it for various corners with given specifications .

After the run is completed analysis is carried out for failing cells by observing waveforms using HSPICE simulations.

Chapter 8

HSPICE Simulation

Sometimes measurement calculation fails due to some discrepancies in the decks. These discrepancies are interpreted by observing waveforms. HSPICE simulation does this. I have regressed various HSPICE decks as part of project, an example simulation is shown here. As Example of spice deck cannot be shown as it is confidential information. To have an overview a SPICE deck contains the following information :

- Circuit (all devices listed with their node connections)
- Sub circuit (a method to create hierarchy in circuits)
- Device Models (Model defining device performance)
- Stimuli (Signals applied to the circuit)
- Analysis (type of simulation: DC, AC, transient, etc.)
- Spice options (controlling simulation algorithm and accuracy)
- Simulation output (what information and in what format to be saved)

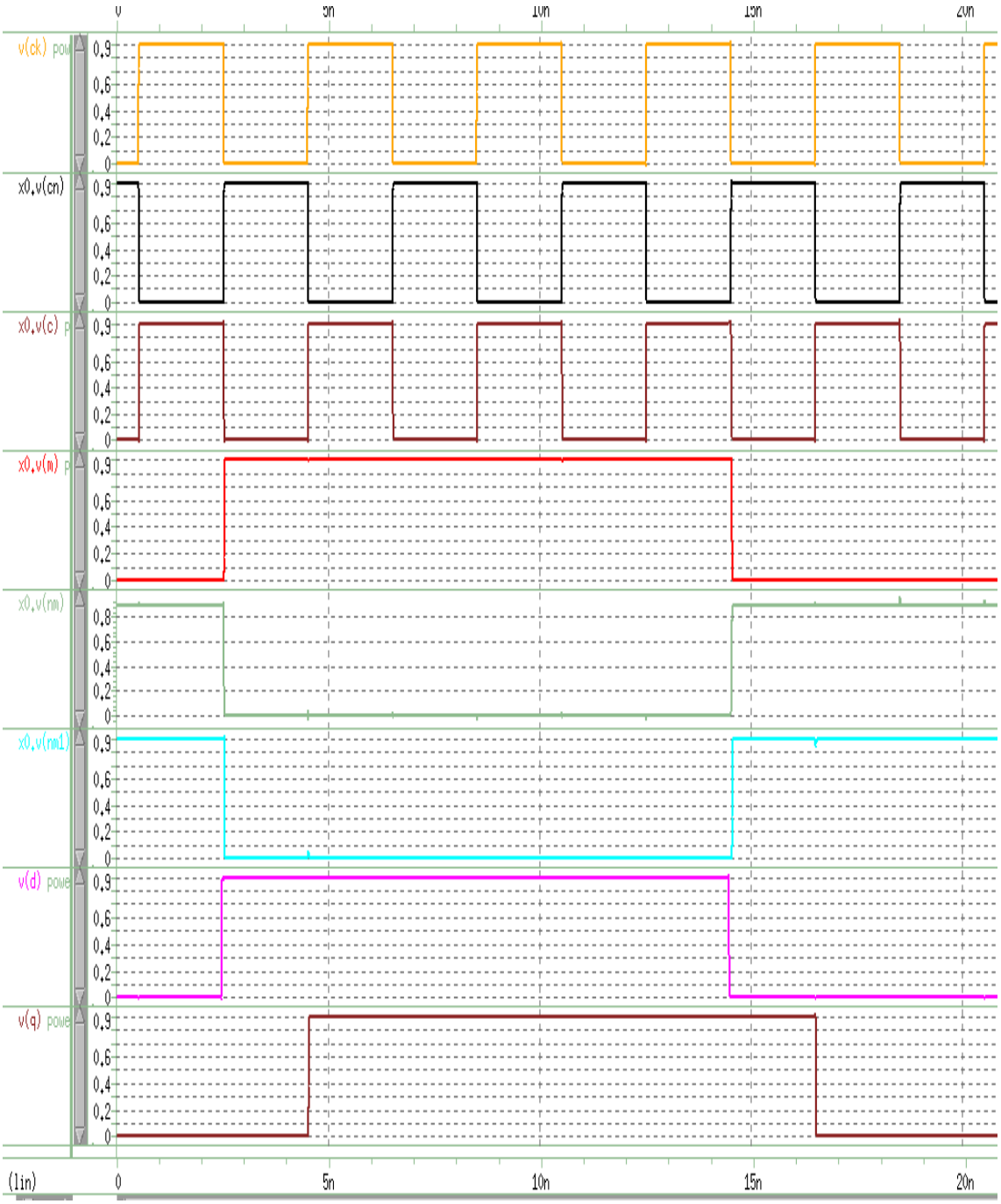

Figure 8.1: Master Slave D Flip Flop waveform

Chapter 9

Conclusion

Leakage power and critical path delay are very critical parameters for standard cell. It is always desired to have less delay and leakage power. Now these parameters directly depend on sizes of NMOS and PMOS, which are sensitive to critical path delay. So we have to size our transistors such a way that they give optimized power and delay. For this purpose celltuner tool is used.

After completing schematic conversion of all projects, source has flat schematics. So if customer wants both, flat schematic and layout, then company can ship without any delay. At the same time if customer comes for re-categorization in future, then using mapping file, company will know that which topology was shipped at the time of project and make additional changes as per customer requirements.

The "Genie" porting approach has been developed with the intention of reducing the manual effort and increasing the productivity when we migrate an existing design to a new process node. The example used as a is to migrate from CMOS32LP process to the CLN40LP process node. The "Genie" approach scales both the layout and schematics for a given design. The main objective is to reduce the amount of time spent by layout engineers and circuit designers in migrating the layouts and schematics.

SVS(schematic versus schematic) and LVS are very important from verification point of view. There is possibility of error in Schematics/Layout drawn by human, so verification is necessary before it is shipped to the customers.

After LVS is clean detailed parasitic information is extracted to know whether any such parasitic does not hinder timing path of the circuit.

Design failing on silicon is more risky, so by providing a tolerance to the design it is allowed to operate under various PVT conditions. To make sure whether the design is able to with stand under the specified PVT Conditions or not. EDAR flow is used to maintain the library standards.

If any cell fails during EDAR or due some discrepancies in SPICE decks, waveforms are observed using HSPICE simulation.

References

[1] ARM SPECIFIC DOCUMENTS

[2] ARM Power Management kits
http://www.arm.com/products/physical–ip/logic –ip/standard –cell–libraries.php

[3] http://www.eng.utah.edu/ cs5830/handouts/Sutherland_Ch1.pdf

[4] http://www.itrs.net/links/2001itrs/PIDS.pdf

[5] http://vlsi–soc.blogspot.in/2013/03/ocv –vs–pvt.html

[6] Cadence Help Document
http://www.egr.msu.edu/classes/ece410/mason/files/guide–LVS.pdf

[7] Corporation, M.G.(2014).CalibrexRC TM User's Manual.
http://ispc.ustc.edu.cn/ztbd/jszt/201305/P020130507526544108551.pdf

[8] Geden, B. (2016). Understand and Avoid Electromigration (EM) IR-drop in Custom IP Blocks, (November 2011),1–6.

[9] HSPICE ® User Guide,(September–2008).
http://cseweb.ucsd.edu/classes/wi10/cse241a/assign/hspices_a.pdf

[10] A. N, Chandorkar, Logical Effort Calculation ,
http://www.cdeep.iitb.ac.in/nptel/Electrical& Comm/Engg/VLSI/Design/TOC–l21.html

[11] Design Framework II SKILL TM Functions Reference, (June-2000).

[12] Cadence ® User Interface SKILL Functions Reference, (July-2002).

[13] Kamakoti, V., Balachandran, S. CAD for VLSI Design - II.
http://nptel.ac.in/courses/IIT–MADRAS/CAD_for_VLSI_Design_II/pdf/nptel–cad2–16.pdf

[14] Gupta, B., Nakhate, S.(2012). TRANSISTOR GATING: A Technique for Leakage Power Reduction in CMOS Circuits, 2(4), 321–326.

[15] Cadence Community
www.support.cadence.com

[16] http://www.mentor.com/products/ic_nanometer_design/verification
–signoff/circuit–verification/calibre–xrc/faqs

[17] Jin-Fu Li,Chapter 4 Low Power VLSI Design
http://www.ee.ncu.edu.tw/ jfli/vlsi2/lecture/ch04

[18] Kumar, M., Arya, S. K, Pandey, S.(2010).LEVEL SHIFTER DESIGN FOR LOW POWER APPLICATIONS, 2(5), 124–132.

[19] DETERMINISTIC CLOCK GATING LOW POWER DESIGN, (2007).
http://ethesis.nitrkl.ac.in/4404/1/Deterministic_Clock_Gating_for_Low$power_VLSI.pdf$