

Novel Physical Design Methodology for Efficient Power Optimization in SoC's Server Partitions

Major Project Report

Submitted in partial fulfillment of the requirements
For the degree of

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

By
Chauhan Atmiya Jatanbhai
(16MECV03)



**Electronics & Communication Engineering Department
Institute of Technology**

Nirma University

Ahmedabad - 382 481

May, 2018

Novel Physical Design Methodology for Efficient Power Optimization in SoC's Server Partitions

Major Project Report

Submitted in partial fulfillment of the requirements
For the degree of

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

By
Chauhan Atmiya Jatanbhai
(16MECV03)

Under the Guidance of
Prof. Dr. Usha Mehta



Electronics & Communication Engineering Department

Institute of Technology

Nirma University, Ahmedabad - 382 481

May, 2018

Declaration

This is to certify that

1. 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.
2. Due acknowledgment has been made in the text to all other material used.

Chauhan Atmiya Jatanbhai

Certificate

This is to certify that the Major Project entitled “**Novel Physical Design Methodology for Efficient Power Optimization in SoC’s Server Partitions**” submitted by **Chauhan Atmiya Jatanbhai (16MECV03)**, 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 him 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.

Prof. Dr. Usha Mehta
Internal Guide

Prof. Dr. N. M. Devashrayee
PG Coordinator (VLSI Design)

Dr. D. K. Kothari
Head, EC Dept.

Dr. Alka Mahajan
Director, IT - NU

Date :

Place : Ahmedabad



Certificate

This is to certify that the Project entitled "Novel Physical Design Methodology for Efficient Power Optimization in SoC's Server Partitions" submitted by **Chauhan Atmiya Jatanbhai (16MECV03)**, towards the submission of the Project for requirements for the degree of Master of Technology in VLSI Design, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.

(External Guide)

Mr. Selvaraj Subramanian

Engineering Manager

Intel Technology India Pvt. Ltd.

Bangalore

(Mentor)

Mr. Naveen Kotha

SoC Design Engineer

Intel Technology India Pvt. Ltd.

Bangalore

Company Seal

Intel Technology India Pvt. Ltd.(Bangalore)

Date :

Place : Bangalore

Acknowledgment

First and foremost, sincere gratitude to my manager Mr. Selvaraj Subramanian. Also I want to thank Intel Technology India Private Limited, Bangalore for as-signing me such project and guide me through.

I would like to express my gratitude and sincere thanks to my mentor Mr. Naveen Kotha for his valuable guidance during the project work, he has given me valuable advices and support which I am very lucky to benefit from.

I would like to express my sincere gratitude to Dr. Alka Mahajan (Director and Head of Department NIRMA University, Ahmedabad) for his continuous guidance and support. I would like to take this opportunity to thank Dr. N. M. Devashrayee (Professor and Program Coordinator, M. Tech - EC (VLSI Design)), Internal Guide Prof. Dr. Usha Mehta and all the faculties for their vision, support, and encouragement to provide me with the opportunity to carry out my project work in such a renowned and esteemed organization.

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. I also owe my colleagues in the Intel, special thanks for helping me in this project.

- Chauhan Atmiya Jatanbhai (16MECV03)

Abstract

In System-on-Chips (SoCs), the most critical design constraint used to be Area and Performance but Power has also become an important constraint in recent times. Therefore, the power-aware design should be introduced in Physical Design at early stages of SoCs Server Partition where it has the highest benefits for power reduction.

To make the design power-aware, new low power techniques and standards are introduced. Introduction of Unified Power Format and implementation of low power techniques in Electronic Design Automation tools has helped the designers to provide power intent separated from the design functionality during initial stages of ASIC Physical Design flow, thus reducing the power dissipation. Implementation of Multisource CTS and Multibit Flip Flop technique in SoC's partitions reduces power consumption of design significantly.

This project concludes that by implementing low power techniques using Unified Power Format, Multisource CTS and Multibit Flip Flop it is possible to make the design power aware which consumes less power.

Table of Contents

Declaration	i
Certificate	ii
Internship Certificate	iii
Acknowledgment	iv
Abstract	v
List of Figures	ii
1 Introduction	1
1.1 Motivation	1
1.2 Objective	2
1.3 Overview	2
1.4 Preface	2
2 Literature Survey	5
2.1 Dynamic Power	5
2.2 Static Power	6
2.3 Advanced Low Power Techniques	7
3 Unified Power Format	11
3.1 Introduction	11
3.2 What does UPF contain?	12
3.2.1 Power Domain UPF:	12

3.2.2	Power Supply Port Nets UPF:	12
3.2.3	Level Shifter UPF:	13
3.2.4	Isolation Cell UPF:	14
3.2.5	Retention Flop UPF:	15
3.2.6	Always-ON Logic UPF:	16
3.2.7	Power Switch UPF:	18
3.2.8	Power State Table UPF:	19
3.3	Practical Implementation of UPF	20
4	Low Power ASIC Design Flow	21
4.1	Low Power ASIC Flow	21
5	Spyglass Low Power	25
5.1	Overview	25
5.2	Flow	26
5.2.1	Liberty Requirements:	28
5.2.2	Design Requirements:	28
5.3	Results	28
6	Efficient CTS for Power Reduction	33
6.1	Multisource CTS	33
6.2	Clock Tree Power	35
6.3	Multibit Flip-Flops	37
6.4	Criteria for using MBFF	37
6.5	Implementation	39
7	Result Analysis	43
7.1	Introduction	43
7.2	Multisource CTS	43
7.2.1	Latency	43
7.2.2	Timing	45

7.2.3	Power	45
7.2.4	Conventional vs Multisource CTS Comparison	45
7.3	Multibit Flip Flops	46
7.3.1	Cell Count	46
7.3.2	Power	47
7.3.3	Timing	47
7.3.4	SBFF vs MBFF Comparison	48
8	Conclusion and Future Work	49
8.1	Summary	49
8.2	Work Conclusion	49
8.3	Future Scope of Work	50
	References	51

List of Figures

2.1.1 Switching Power Dissipation [1]	6
2.1.2 Short-circuit Power Dissipation [1]	7
2.2.1 Static Power Dissipation [1]	8
2.3.1 Power Gating [Shutdown] [1]	8
2.3.2 Multi-Voltage [1]	9
2.3.3 MV Power Gating with State Retention [1]	9
2.3.4 Low-VDD Standby [1]	9
2.3.5 MV Power Gating with State Retention [1]	10
3.2.1 Power Domain [1]	13
3.2.2 Supply Ports Nets [1]	13
3.2.3 Level Shifter [1]	14
3.2.4 Level Shifter in Power Domain pdA [1]	14
3.2.5 Isolation Cell [1]	15
3.2.6 Isolation Cell in Power Domain [1]	15
3.2.7 Retention Flop [1]	16
3.2.8 Retention Flop in Power Domain [1]	16
3.2.9 Always ON Logic [1]	17
3.2.10 Power Switch [1]	18
3.2.11 Power Switch in Power Domain [1]	18
3.2.12 Power State Table [1]	19
3.3.1 Practical Implementation of UPF	20
4.1.1 Low Power ASIC Design Flow	22

5.2.1 Spyglass_lp Flow	26
5.2.2 Spyglass_lp Methodology [3]	27
5.3.1 Spyglass_lp Violations [3]	29
5.3.2 Description of Violations	29
5.3.3 Examples of Low Power Checks [3]	29
5.3.4 Schematic of Violation [3]	30
5.3.5 Report with Violations [3]	30
5.3.6 Report with Clean Violations [3]	31
6.1.1 Multisource CTS	34
6.1.2 Clock Mesh	35
6.1.3 Clock Spine	36
6.2.1 Short circuit current in Multisource CTS	36
6.3.1 Merging two 1-bit flip-flops into one 2-bit flip-flop. (a) Two 1-bit flip-flops (before merging) (b) 2-bit flip-flop (after merging)	38
6.4.1 Applying 2-bit FFs according to clock tree topologies during timing-driven placement)	39
6.5.1 2-bit Register with scan enabled	42
7.2.1 Clock path to a sequential in (a) Multisource CTS (b) Conventional CTS	44
7.2.2 Latency Comparison between Multisource CTS Conventional CTS	44
7.2.3 Timing Comparison between Multisource CTS Conventional CTS	45
7.2.4 Timing Comparison between Multisource CTS Conventional CTS	46
7.2.5 Conventional vs Multisource CTS Comparison	46
7.3.1 Conventional vs Multisource CTS Comparison	47
7.3.2 Power Comparison between MBFF SBFF	47
7.3.3 Timing Comparison between MBFF SBFF	48
7.3.4 SBFF vs MBFF Comparison	48

Chapter 1

Introduction

1.1 Motivation

In System-on-Chips (SoCs), the most critical design constraint used to be Area and Performance but Power has also become an important constraint in recent times. Therefore, power aware design should be introduced in Physical Design at an early stages of SoCs Server Partition where it has the highest benefits for power reduction. So, such design techniques are required that consume less power and also maintaining a comprehensive performance, leading to the introduction of Low power design techniques and methodologies to implement them.

The project help is understanding the whole Low Power ASIC Physical Design Flow and gives a chance to get hands-on on the tools used for design implementation. Also it deals with the implementation of methodologies needed while converging SoC Server Partition.

Also it gives an opportunity to work in a team and understand the industrial ways of implementation. It gives an experience in how to deal and interact with people and issues present at the location or outside.

1.2 Objective

The objective of this project is to develop a novel physical design methodology for the convergence of the design with less power consumption by implementing different low power techniques to meet the all specifications.

1.3 Overview

The power consumption in a conventional CMOS digital circuit can be divided into two types of power dissipation, Dynamic Power and Static Power Dissipation. These power can be further categorized into (i) Switching Power Dissipation (ii) Short-circuit Power Dissipation (iii) Leakage Power Dissipation. Switching power represents the power dissipated because of the signal transition at input charging or discharging the load capacitance, Short circuit power dissipation is because of short circuit current which is due to the lower transition rate at input signals. This will turn on both the PMOS and NMOS network simultaneously in CMOS logic. It will generate current from VDD to GND to generate short circuit power. The MOSFETs in CMOS logic normally will have some non-zero reverse leakage and sub-threshold current, which causes the leakage power consumption. Both switching and short circuit power dissipation occur during active state of circuit so it is called as dynamic power dissipation and the leakage power occurs during idle mode of circuit so it is called static power dissipation. The static power increases faster than dynamic power with the shrinking of the feature size. [1]

1.4 Preface

The report is organized such that the basic underlying concepts are described first before delving into more advanced topics.

Chapter 2 This chapter gives an overview and highlights of power dissipation in ASIC Design and different low power techniques which are popularly used in industries.

Chapter 3 This chapter gives overview of Unified Power Format in ASIC design flow and how its impact in making our design power aware. This chapter provides detail information of special power reduction cells used by UPF and practical implementation of UPF in SoC server partition.

Chapter 4 This chapter gives the overview of Low Power ASIC design flow.

Chapter 5 This chapter gives overview that special cells used by Unified Power Format placed properly or not in SoC server partition.

Chapter 6 This chapter gives overview of Multisource CTS and Multibit Flip Flop techniques and how we can reduce power using it.

Chapter 7 Result Analysis.

Chapter 8 Conclusion and Future Work.

Chapter 2

Literature Survey

This chapter gives overview of different types of power dissipation in ASIC design and how we can reduce power dissipation using different types of low power techniques.

2.1 Dynamic Power

Dynamic power dissipation occurs during active state of the circuit and a major contributor to power dissipation. There are two types of power dissipation occurs in this category.

Switching Power Dissipation: Switching power dissipation occurs due to charging and discharging of load capacitance as shown in the below figure 2.1.1. The switching power is calculated using the equation:

$$P = \alpha C_L V_{dd}^2 f$$

Where, α = switching activity

f = Frequency

V_{dd} = Supply Voltage

C_L = Load Capacitance

Short-circuit Power Dissipation: When input signal transitions slowly, it will turn on both

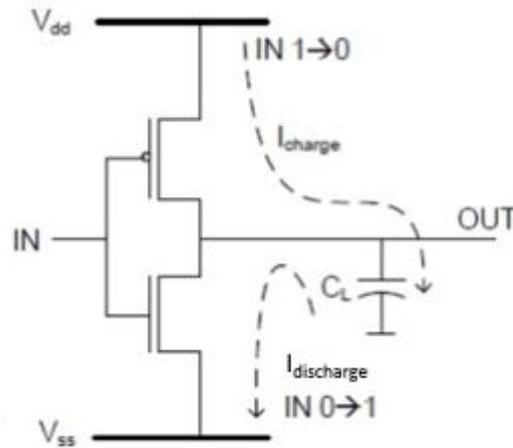


Figure 2.1.1: Switching Power Dissipation [1]

NMOS and PMOS simultaneously as shown in figure 2.1.2. This results in a conductive path from VDD to GND. Because of that, current will flow from VDD to GND. It will dissipate power which is called short-circuit power, which is calculated using:

$$P_{(sc)} = (\beta/12) (V_{dd} - V_T)^3 T f$$

Where, β = gain factor of transistor

V_{dd} = supply voltage

V_T = threshold voltage

T = rise/fall time of current (i_p)

2.2 Static Power

Static power dissipation occurs when the circuit is not in operating condition. Figure 2.2.1 shows different components responsible for static power dissipation. Four main components of leakage current are as follows:

Subthreshold Leakage: In the weak inversion region, the current that flows from the drain to

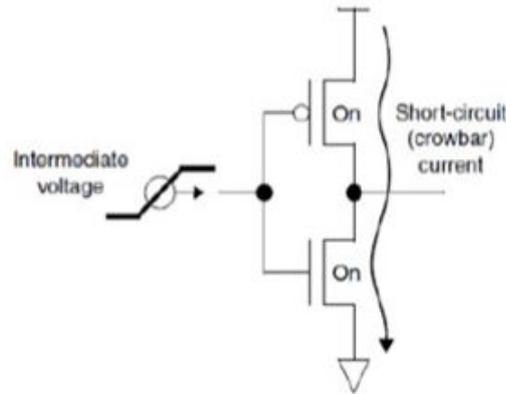


Figure 2.1.2: Short-circuit Power Dissipation [1]

source in a transistor is known as Subthreshold Leakage.

Gate Leakage (I_{GATE}): In a CMOS because of gate oxide tunneling and hot carrier injection the current that flows directly from the gate to substrate through the oxide is known as Gate Leakage.

Gate Induced Drain Leakage (I_{GIDL}): The current that flows from drain to substrate induced because of a high field effect in MOSFETs drain caused by a high V_{dg} is called Gate Induced Drain Leakage.

Reverse Bias Junction Leakage (I_{REV}): The main cause of leakage current is the minority carrier drift generation of E-H pair in depletion region of MOSFET.

2.3 Advanced Low Power Techniques

Power Gating [Shutdown]: All blocks operate on the same voltage but some blocks can be turned off when functionality is not needed as shown in the figure 2.3.1. This technique reduces leakage power.

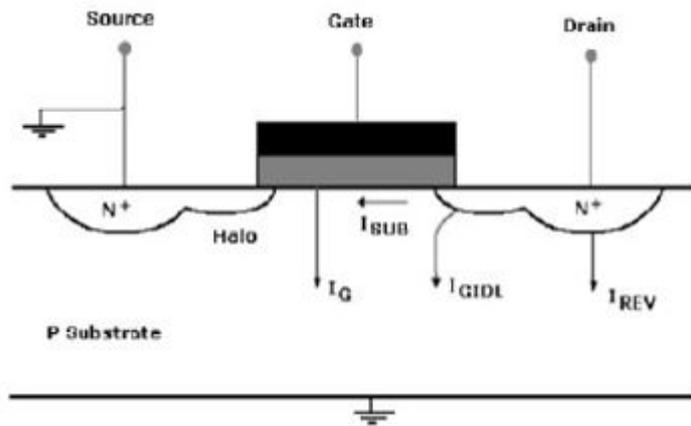


Figure 2.2.1: Static Power Dissipation [1]

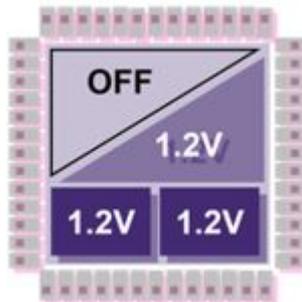


Figure 2.3.1: Power Gating [Shutdown] [1]

Multi-Voltage: Each block operates on different voltage as shown in the figure 2.3.2. Goal here is to reduce dynamic power by reducing the voltage of the blocks where the performance is low.

MV Power Gating with State Retention: In this low power technique we use retention registers. When the block functionality is not needed we turn off and when we turn on that block it has the same state when it was turned off. So it saves the state. 2.3.3

Low-VDD Standby: In this technique we are not cutting the voltage of the block but we are lowering to a value where transistor can retain the state. So here we are avoiding the use of retention register.2.3.4

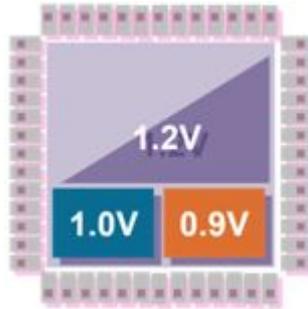


Figure 2.3.2: Multi-Voltage [1]

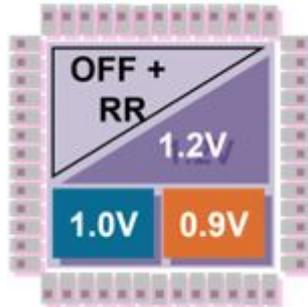


Figure 2.3.3: MV Power Gating with State Retention [1]

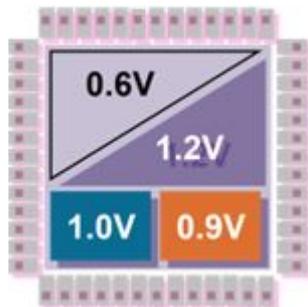


Figure 2.3.4: Low-VDD Standby [1]

Adaptive Voltage Frequency Scaling: In this technique power control module on chip determine the performance and requirement to increase or decrease the voltage in order to save dynamic power as well as frequency.2.3.5

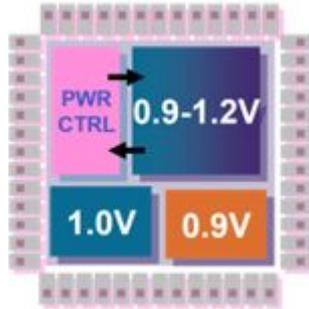


Figure 2.3.5: MV Power Gating with State Retention [1]

Chapter 3

Unified Power Format

This chapter gives overview of Unified Power Format in ASIC design flow and how its impact in making our design power aware. This chapter provides detail information of special power reduction cells used by UPF and practical implementation of UPF in SoC server partition.

3.1 Introduction

In order to introduce the low power strategies into designs Unified Power Format was introduced. Unified Power Format (UPF) is an IEEE standard for specifying power intent. An UPF committee was formed by the Accellera organization, chaired by Stephen Bailey of Mentor Graphics. The version 1.0 was approved to be published on February 26, 2007. This UPF was donated to IEEE in 2006. On March 26, 2009 the Standard for Design Verification of Low Power Integrated Circuits was published as IEEE Std 1801-2009. It is called as UPF 2.0. IEEE 1801-2013 was published in March 2013 and it is called as UPF 2.1. The latest version released was UPF 3.0, which got published in December 2015. [5]

We can implement power gating and voltage scaling using UPF. Mainly there are two reasons behind introducing UPF as separate standard instead of introducing as part of RTL. One is - today designers are reusing most of RTL. RTLs are updated to add in new functionality. Another reason is due to challenges associated to verification. If we introduce low power

intent as part of RTL then for every power intent change we have to verify both functional and also low power intent. Whereas if we introduce low power intent as separate part that once functional verification is done we dont need to check functionality again. Only needs to check low power intent.

3.2 What does UPF contain?

Unified Power Format contains Power Domains, Power Supply Ports, Power Supply Nets, Low Power Valid States, Operating Voltages and Special Cells like Power Switches, Level Shifters, Isolation Cells and Retention Flops. [1]

3.2.1 Power Domain UPF:

Power Domain is nothing but group of logic hierarchy that shares the same power supply. Power Domain is logical so we need corresponding physical region to place the cells within that region is called Voltage Area. 3.2.1

Command: `create_power_domain pdA element {A}`

3.2.2 Power Supply Port Nets UPF:

After creating the power domain we need to create supply ports and nets for the respective power domain which are created by using the following commands. 3.2.2

Command: `create_supply_port / create_supply_net`

Command: `connect_supply_net / set_domain_supply_net`

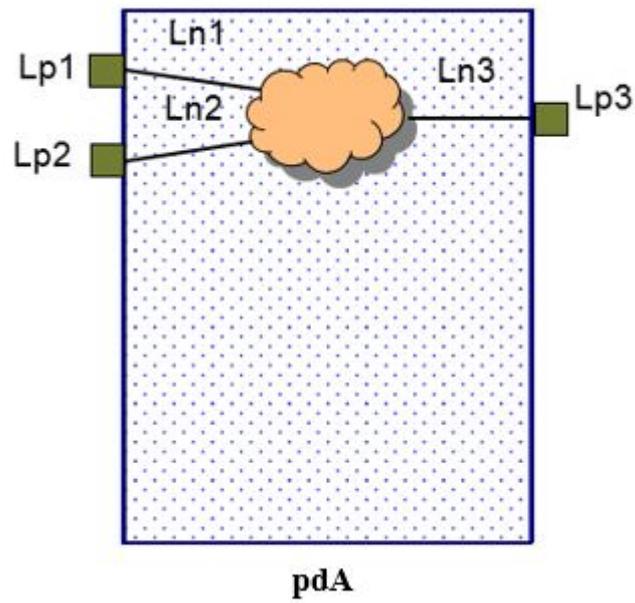


Figure 3.2.1: Power Domain [1]

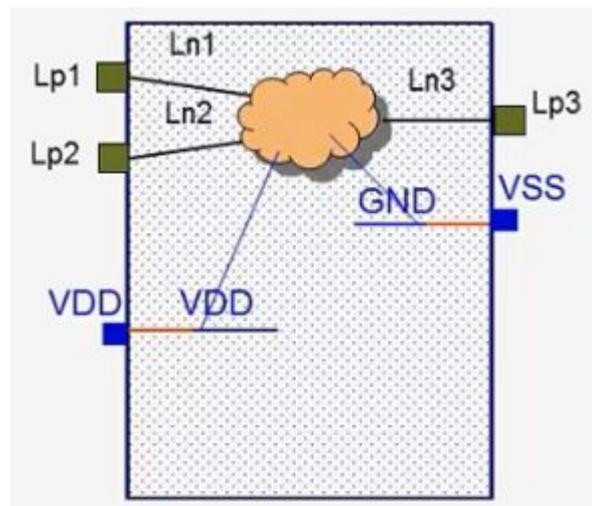


Figure 3.2.2: Supply Ports Nets [1]

3.2.3 Level Shifter UPF:

Level Shifters are used between power domains which are operating at different voltages so that the signal levels are corrected between them. 3.2.3 and 3.2.4

Command: `set_level_shifter LSI -domain pdA -applies_to inputs -location parent`

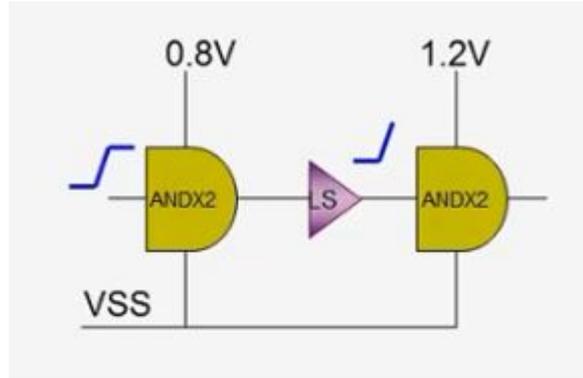


Figure 3.2.3: Level Shifter [1]

Command: set_level_shifter LSO -domain pdA -applies_to outputs -location

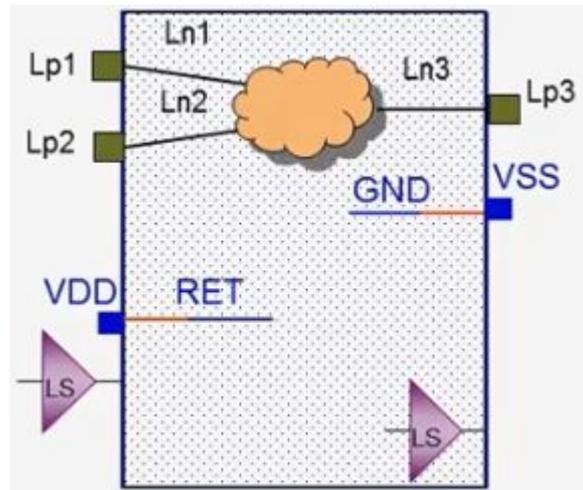


Figure 3.2.4: Level Shifter in Power Domain pdA [1]

3.2.4 Isolation Cell UPF:

Generally Isolation Cells are used at output of a gated domain so that known values reach to the Always On logic from output of gated domain. 3.2.5 and 3.2.6

Command: set_isolation ISO domain pdA clamp 0 isolation_power_net VDD isolation_ground_net GND applied_to outputs set_level_shifter LSI -domain pdA -applies_to inputs -location parent

Command: `set_isolation_control ISO -domain pdA -isolation_signal isolate_enable -location`

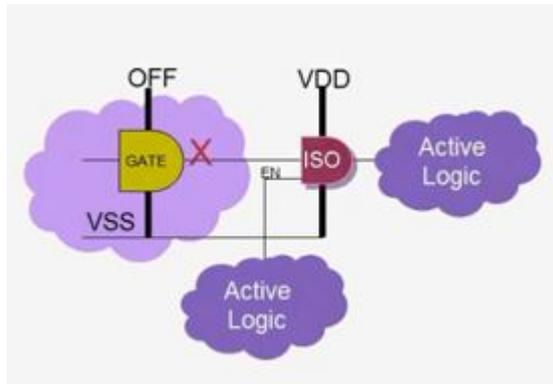


Figure 3.2.5: Isolation Cell [1]

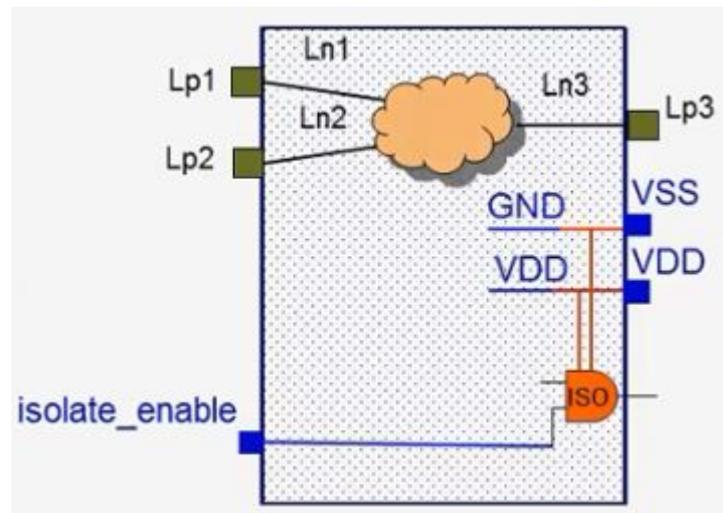


Figure 3.2.6: Isolation Cell in Power Domain [1]

3.2.5 Retention Flop UPF:

Before the power cutoff Retention flop copies the value of Flip-flop to Latch and when re-store signal enables again it copies the last saved values of Latch to Flip-flop. 3.2.7 and 3.2.8

Command: `set_retention RFF -domain pdA -retention_power_net VDD -retention_ground_net GND -elements A`

powered within shutdown block.3.2.9

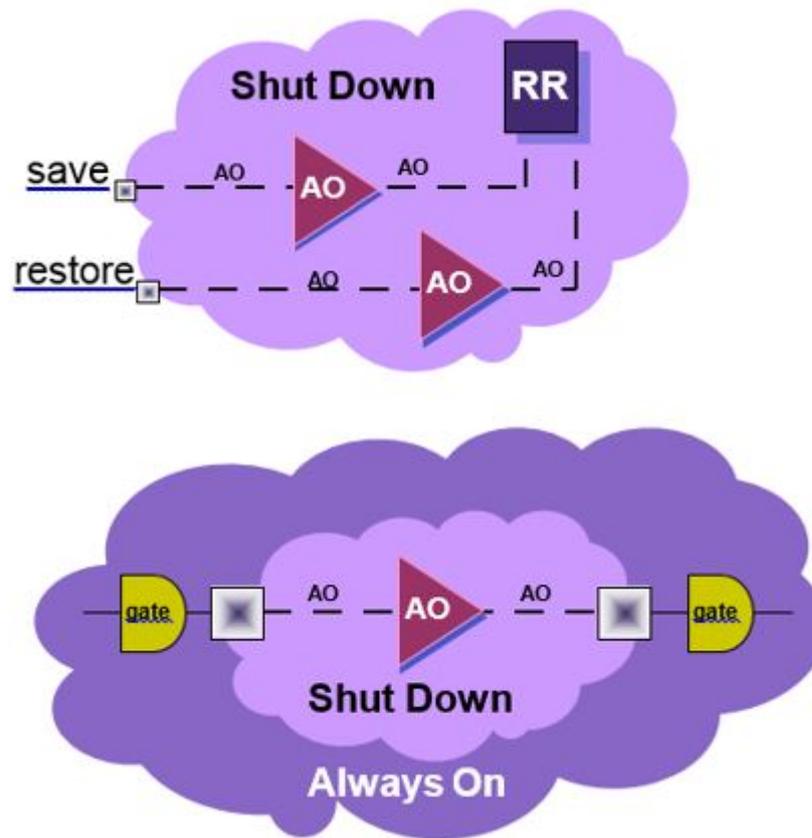


Figure 3.2.9: Always ON Logic [1]

Some logic needs to stay active during shutdown:

1. Paths to Enable pins of ISO/ELS
2. Power Switches
3. Retention Registers
4. Feedthrough Paths

3.2.7 Power Switch UPF:

Power Switches are generally used for cutoff power when functionality is not needed. Power Switches are not inserted in Synthesis. They are inserted in ICC2.3.2.10 3.2.11

Command: `create_power_switch SW1 -domain pdA -input_supply_port {VDD VDD} -output_supply_port {V_RET VIRTUAL_VDD} -control_port {sleep shut_down} -on_state {state0 VDD {!sleep}}`

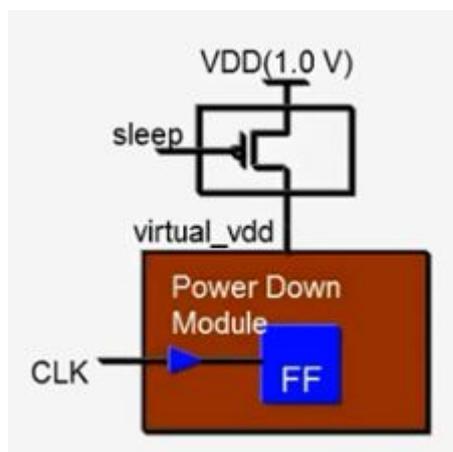


Figure 3.2.10: Power Switch [1]

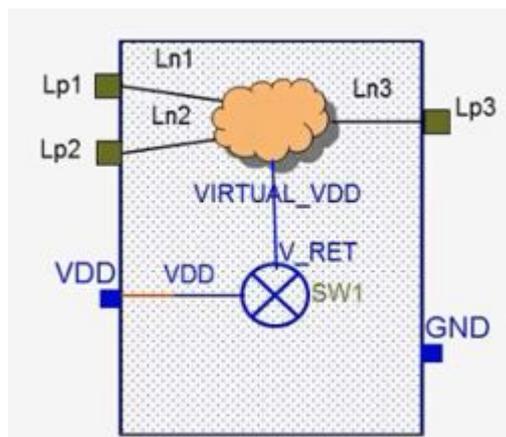


Figure 3.2.11: Power Switch in Power Domain [1]

3.2.8 Power State Table UPF:

We have created supply ports, supply nets, power domains, connected supply nets, defined what is primary supply of domain and also we defined low power strategies based on design requirements. Once we decide all things now we have to define Power State for each of the supply port until what is operating voltage the port will be operating or supply net will be operating.

Command: `add_port_state / create_pst / add_pst_state`

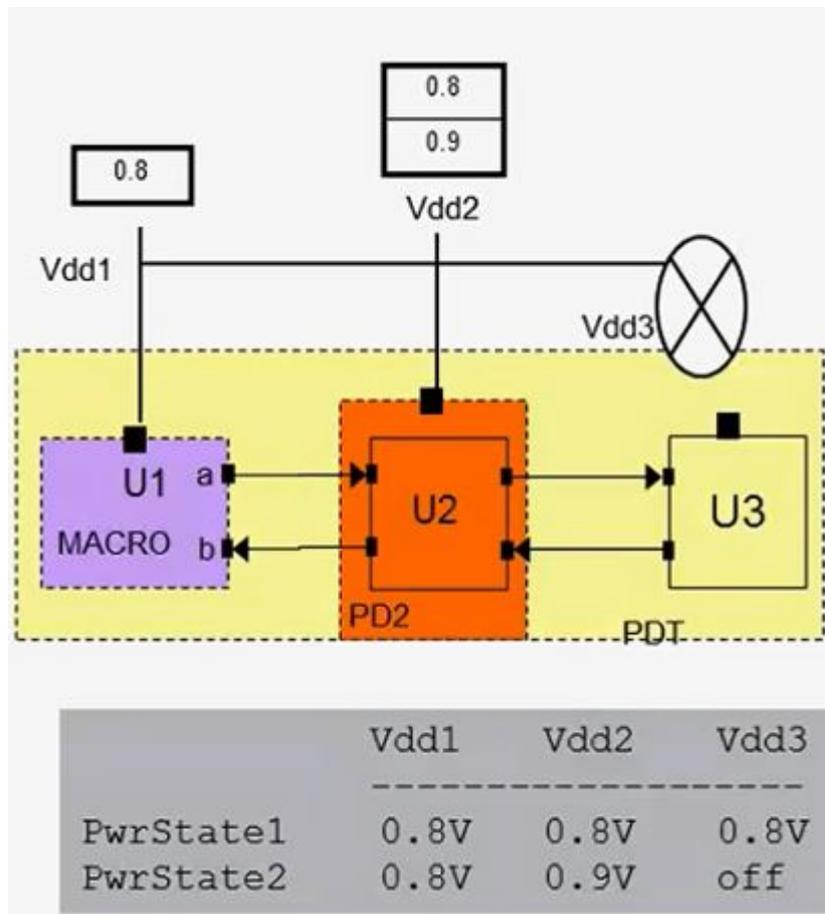


Figure 3.2.12: Power State Table [1]

3.3 Practical Implementation of UPF

By using the following synopsis commands we can create Power Domain, Ports, Nets, Isolation Cells, Level Shifters and Retention Flops as shown in below figure.3.3.1

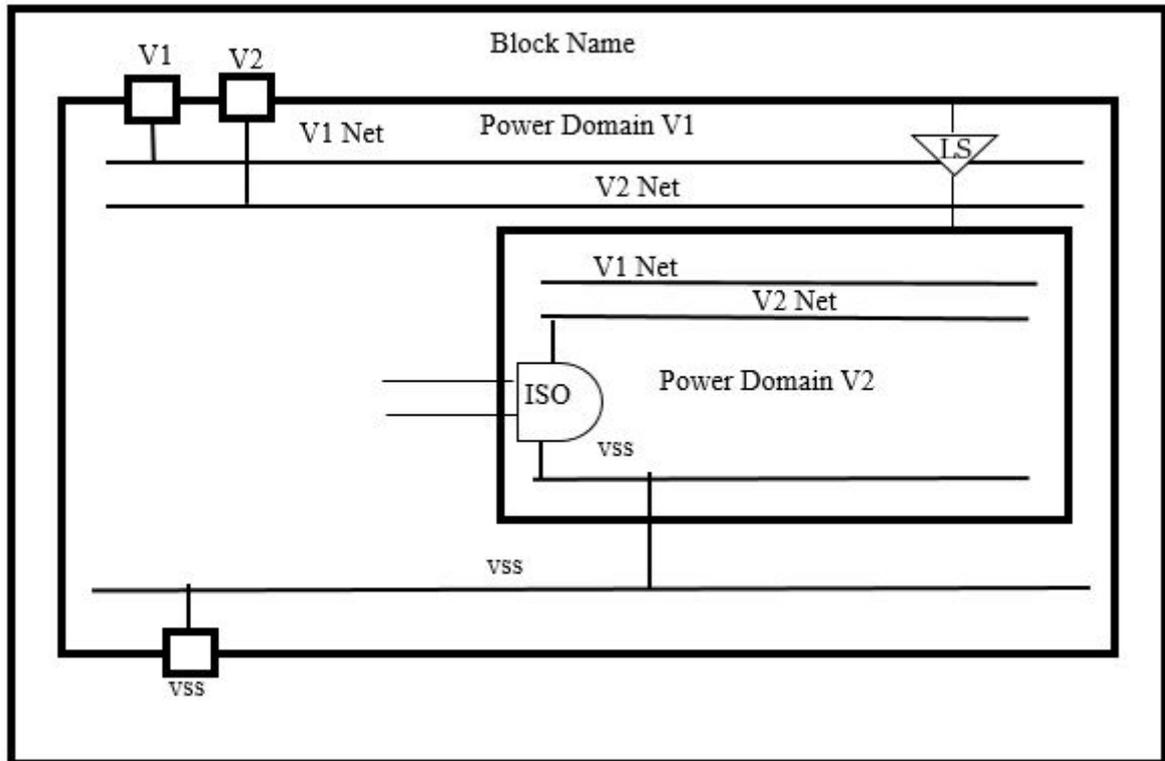


Figure 3.3.1: Practical Implementation of UPF

```
create_power_domain "Power Domain v1"  
create_supply_port -domain "<domain>" "v1"  
create_supply_net -domain "<domain>" "v1 Net"  
connect_supply_net "<domain>" -port "<v1>"  
set_isolation_control -domain "<domain>" -isolation_signal "<signal name>" -location  
"<self/parent>"  
set_level_shifter "LS" -domain "PDV1" -applies_to_inputs -location "input"
```

Chapter 4

Low Power ASIC Design Flow

This chapter gives the overview of ASIC design flow. Once we have RTL and power intent info (UPF) of a design ready, the next step is to check whether the design is appropriate (as desired) and is implementable or not. To check that, the design goes through various stages which all together are termed as Design Flow, where the design from a lower abstraction level moves towards a higher abstraction one. Each stage in the Design flow requires the use of an individual tool and a suitable executable environment according to each tool.

4.1 Low Power ASIC Flow

The flow begins with the RTL i.e. register transfer level description logic of the design, along with a UPF description given separately for the design to define the power intent. The RTL and UPF are put in different files as its easier to maintain and modified them separately.

Next step is Synthesis. Process of transforming HDL designs to technology specific gate level netlist is known as synthesis. [6] In this, first the tool reads HDL files and check for syntax. Also UPF file is given as low power intent. After that the tool converts all these to technology independent cells representation. After which the design is optimized in terms of area, power and time and further mapped with technology cells and netlist. [6]

After this, again low power rules are checked between synthesized netlist and UPF and also checks for low power intent are done. After this, the next step is Formal Equivalence. In simple words, Formal Equivalence means verification between two representations. Here one input is RTL and UPF. Another input is synthesized netlist and UPF. It will check both these representations are equal or not.

Physical implementation is next step. Here, tool reads the synthesized gate level netlist and UPF prime power intent files, and performs physical implementation (floorplan to routing) according to the files, generating a modified gate level netlist, a complete power and ground (PG) netlist, and an updated UPF file. The UPF file contains the UPF information plus the modifications done to low-power structures because of physical implementation, such as placing power switches.

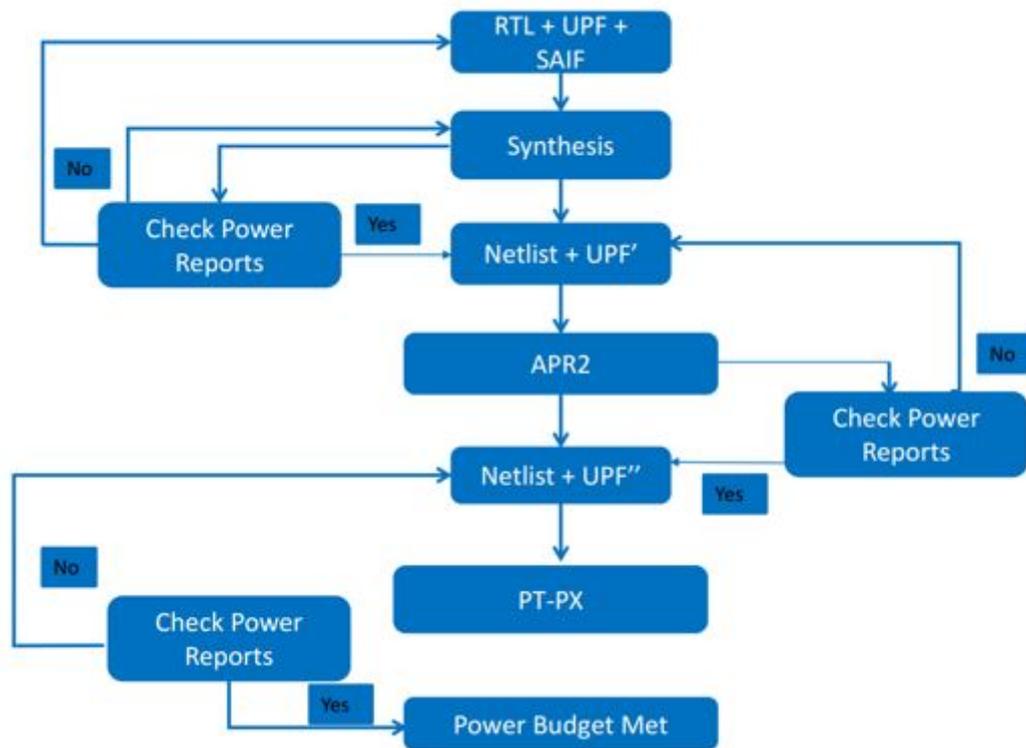


Figure 4.1.1: Low Power ASIC Design Flow

In physical implementation part, first step is floor planning. Floor planning is process of positioning blocks in partition area. It is an important process of creating and developing physical model of the design in the form of an initial optimized layout. Based on area of design and hierarchy, a suitable floorplan is decided upon for different blocks like cells, hard IPs, macros. Also there proper placement, I/O ports and blockage areas are defined. [4] Circuit Partitioning is also done to reduce the complexity, making routing easier and also tools can handle it easily. Next step is power planning. Power planning means connect all power and ground pins to power and ground rails. Normally power routes are not modified during detail routing.

After this step, next step is placement. In this step, all standard cells and macros are placed on their location. Placement of blocks may result in an unrouteable design, i.e., routing might not be possible in space left after block placement. In such case, another iteration is carried out for placing the blocks. In order to minimize the iterations cycles of the placement, an approximate estimation of the routing space is used during the phase of placement.

After this next step is routing. Routing is the process of creating physical connections based on logical connectivity. Signal pins are connected using routing metal interconnects. Routed metal must meet clock skew, timing, max trans/cap requirements and also physical DRC requirements. [4]

After this next step is checking Power consumed by the design whether it is in specified power budget or the design is consuming more power which is more than the specified budget. This power reports are checked by using the PT-PX tool which provides efficient reports of power consumed by the specific design.

Chapter 5

Spyglass Low Power

This chapter gives overview that special cells used by Unified Power Format placed properly or not in SoC server partition. If the specials cells are not places properly or missed then Spyglass tool will show errors in reports generated by it.

5.1 Overview

Power aware design has become critical for wireless as well as wired applications and advances in device technology are presenting new challenges in power minimization. In order to get power efficient designs, low power techniques need to be incorporated from the beginning. The quality of RTL code is very important for downstream optimizations targeting lower power designs. Spyglass LP helps design engineers apply low power techniques from the start when the design is being coded in RTL. By having the RTL designed for low power, you get even better results with downstream power optimization tools. Furthermore, with its built-in knowledge-base of low power design techniques, Spyglass LP effectively makes every engineer in the design team into a low power design expert. Estimating power consumption and designing for low power at RTL is very difficult because power usage is dependent on the actual structure. Spyglass built-in fast synthesis engine quickly synthesizes the RTL into the detailed structure level required to accurately determine areas of focus for low power design needs. With the detailed structural level information available to it, Spyglass can accurately identify logic

structures that should be modified for more efficient power utilization. [3].

5.2 Flow

Spyglass low power checks for the following things in the design at various stages as shown in the figure 5.2.1. Power Intent Consistency Checks Performs syntax and semantic checks on the UPF that help validate the consistency of the UPF before beginning the implementation.

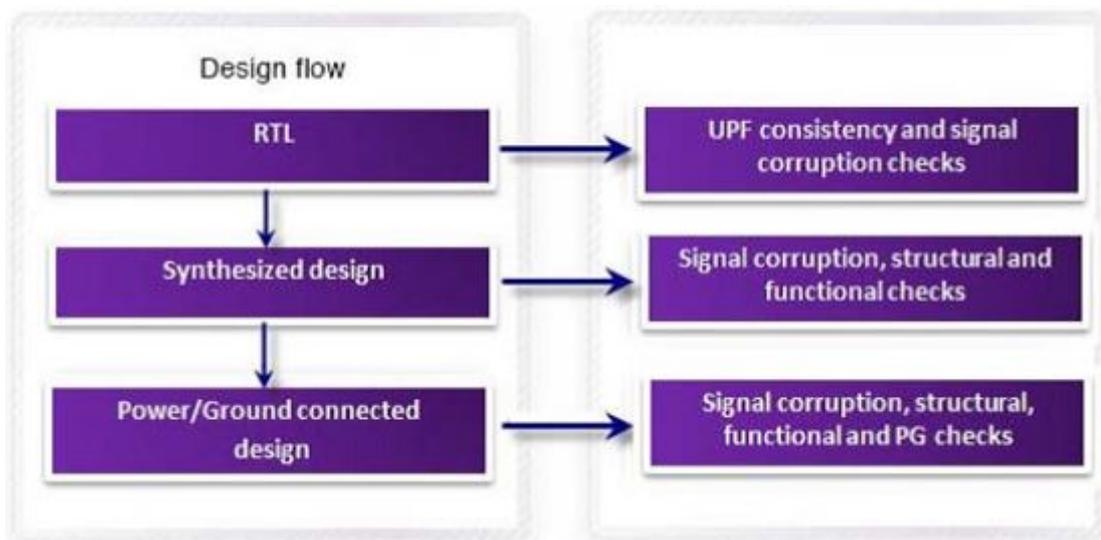


Figure 5.2.1: Spyglass_Lp Flow

Signal Corruption Checks Detects violating power architecture at the gate-level netlist. Structural Checks Validates insertion and connection of special cells used in low power design such as isolation cells, power switches, level shifters, retention registers, and always-on cells throughout the implementation flow. Power and Ground (PG) Checks Check the PG consistency against the UPF specification for power network routing on physical netlists. Functional Checks Validates the correct functionality of isolation cells and power switches. Spyglass LP takes in an RTL (Verilog, VHDL and SVD), netlist (Verilog) or post-layout netlist of the design. It reads the Liberty DB file for resolving, elaborating the design, recognizing special cells and annotating power connections. It accepts the power intent specified in the UPF. [3] As an out-

put, VC LP creates a log file, an error and warnings report for all violations related to low power static rule checks. The tools strong TCL infrastructure helps in debugging these violations. You can also use the Spyglass LP GUI to debug your design violations.

Spyglass can be used to have low power check at various stages in physical design for early low power analysis as following.

1. RTL power verification
2. Post Synthesis
3. Post Routing

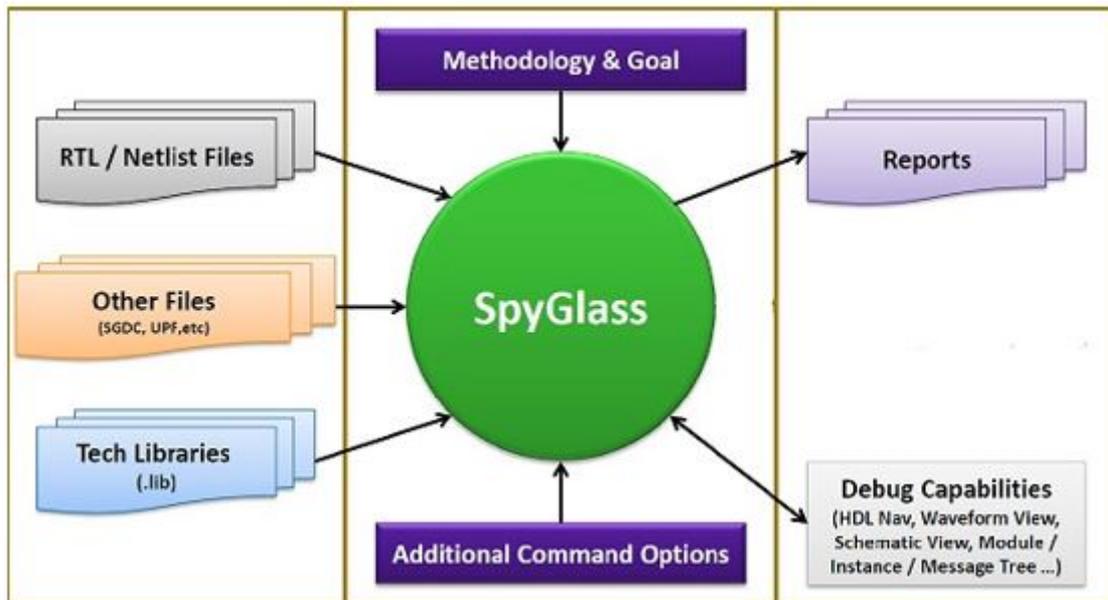


Figure 5.2.2: Spyglass_lp Methodology [3]

In physical design at two stages, low power checks is required. Post Synthesis and Post Routing. The flow for spyglass is as shown in figure 5.2.2 Inputs require for spyglass low power is synthesized netlist and UPF. And other inputs are used like tech libraries to extract the information as definition of power supply nets, ports, pins and associated data pins of standard cells. With required inputs spyglass generates reports through which all the violations related to low power can be analyzed. As methodology and goals there are many rules and check for low

power in the spyglass, from all those checks and rules all the needed rules are being applied as goals and methodology in spyglass. By analyzing reports, wherever required low power methods can be applied and violations can be also resolved for the design.

5.2.1 Liberty Requirements:

Spyglass_lp requires industry standard liberty files (compiled.db files) for gate level netlists (with or without PG routing). Low power cells in the design need to have specific library requirements.

5.2.2 Design Requirements:

Spyglass_lp supports RTL (Verilog, VHDL, MX, and SVD) and PG netlists

5.3 Results

As shown in figure 5.3.1, early analysis can be done by spyglass reports. The second column represents the rules which is violated in design. The description of this rule is shown in the figure 5.3.2. Like that there are certain rules for low power checks. By looking at this rule, there is missing isolation cell at output port in design. Example for some rules is given in figure 5.3.3. For ease of analysis schematic for the same can be viewed in GUI as shown in below figure 4.6.

As shown in the figure 5.3.4 there is a voltage domain crossing to the port which is why there should be isolation strategy from port. Like that there are more number of violations are being analyzed from the reports. Spyglass_lp generates simple report which has all the information regarding all the low power checks, violations, how many number of violations, what are the rules which is violating, description of those rules. Everything is included in this simple

report as shown in figure 5.3.5 as shown in figure there are two simple reports, first one is with violations, and by analyzing this report all the violations are checked and by modifying upf all violations has to be cleaned.

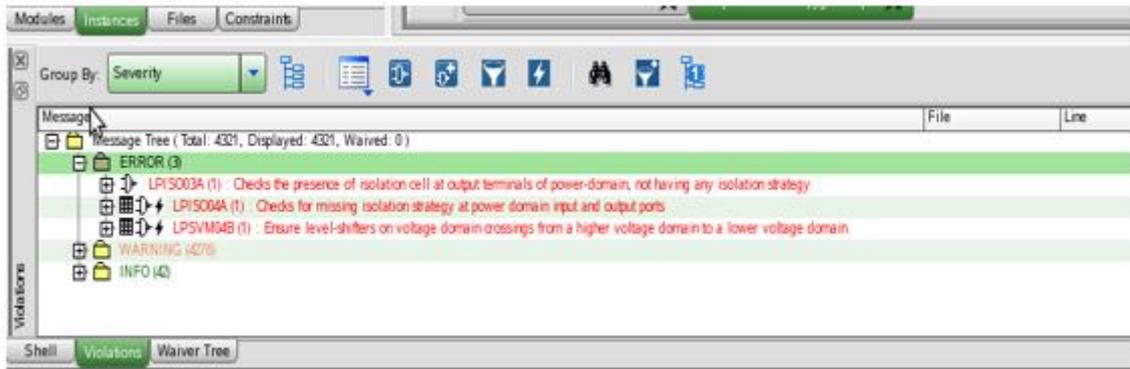


Figure 5.3.1: Spyglass_lp Violations [3]

<p>LPIS003A Reports missing isolation cell at output terminals of power domain, not having any isolation strategy</p>	<p>LPIS004A Reports missing isolation strategy at power domain output ports</p>
------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------

Figure 5.3.2: Description of Violations

level_shifter_data_pin	pin	Data pin of level shifter	LPSVM04
level_shifter_enable_pin	pin	Enable pin of Level shifter	LPSVM04
level_shifter_type	cell	Direction of level shifting, eg L=>H	LPSVM04
output_voltage_range	pin	Output Voltage Range	LPSVM04
pg_function	pg_pin	Present in output power pin	LPPLIB17
pg_pin	cell	Declare PG pin	LPPLIB*
pg_type	pg_pin	power vs ground, always on supply	LPPLIB*
power_down_function	pin	Power switch enable function	LPPLIB17

Figure 5.3.3: Examples of Low Power Checks [3]

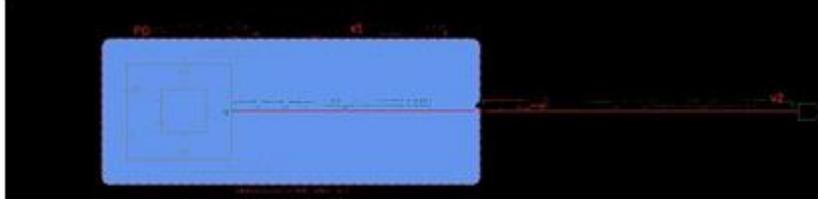


Figure 5.3.4: Schematic of Violation [3]

```

+++++
Severity  Rule Name          Count Short Help
=====
ERROR     LPISO03A           841  Checks the presence of isolation cell
                                at output terminals of power-domain,
                                not having any isolation strategy
ERROR     LPISO04A           219  Checks for missing isolation strategy
                                at power domain input and output ports
ERROR     LPSVM04A           840  Ensure level-shifters on voltage domain
                                crossings from a lower voltage domain
                                to higher voltage domain.
WARNING   LPSVM08B           841  Checks the presence of isolation cell
                                at output terminals of power-domain,
                                not having any isolation strategy
WARNING   NoContAssign       215  Continuous assignment statement present
                                in technology-mapped netlist
INFO     LPISO03A           1    Checks the presence of isolation cell
                                at output terminals of power-domain,
                                not having any isolation strategy

```

Figure 5.3.5: Report with Violations [3]

Severity	Rule Name	Count	Short Help
WARNING	NoContAssign	230	Continuous assignment statement present in technology-mapped netlist
INFO	LPIS003A	1	Checks the presence of isolation cell at output terminals of power-domain, not having any isolation strategy
INFO	LPIS003B	1	Checks the presence of isolation cell at excluded output terminals of power-domain
INFO	LP_CROSSING_DATA	1	This rule populates crossing information from power format intent specified in SGDC/CPF/UPF.
INFO	LP_DECOMPILE_CONSTR	2	Reports user-specified LowPower constraints interpretation details.

Figure 5.3.6: Report with Clean Violations [3]

Chapter 6

Efficient CTS for Power Reduction

In this chapter, the techniques used for this project will be described in detail. The techniques used are Multisource Clock Tree Synthesis, Clock Tree Aware Multibit Flip Flop. These methods are implemented on SoC partitions which are less dense compared to cores and IPs. These techniques can be implemented on any complex design based on the convergence and trade off between timing, area and power.

6.1 Multisource CTS

A multisource clock tree is a hybrid containing the best aspects of a conventional clock tree and a pure clock mesh. It offers lower skew and better on-chip variation (OCV) performance than a conventional clock tree; lower clock tree power/area; and a shorter, easier flow compared to a pure clock mesh implementation.

It is a custom clock structure that has more tolerance to on-chip variation and has better performance across corners than traditional clock tree structures. A renewed emphasis on high-frequency clock design has heightened interest in multisource clock-tree synthesis (CTS). Multisource CTS represents a new clock-distribution technology that fills the methodology gap between conventional CTS and pure clock mesh. Fig. 6.1.1 shows Multisource CTS. Whereas pure clock mesh delivers the best possible clock frequency, skew, and OCV results, and whereas

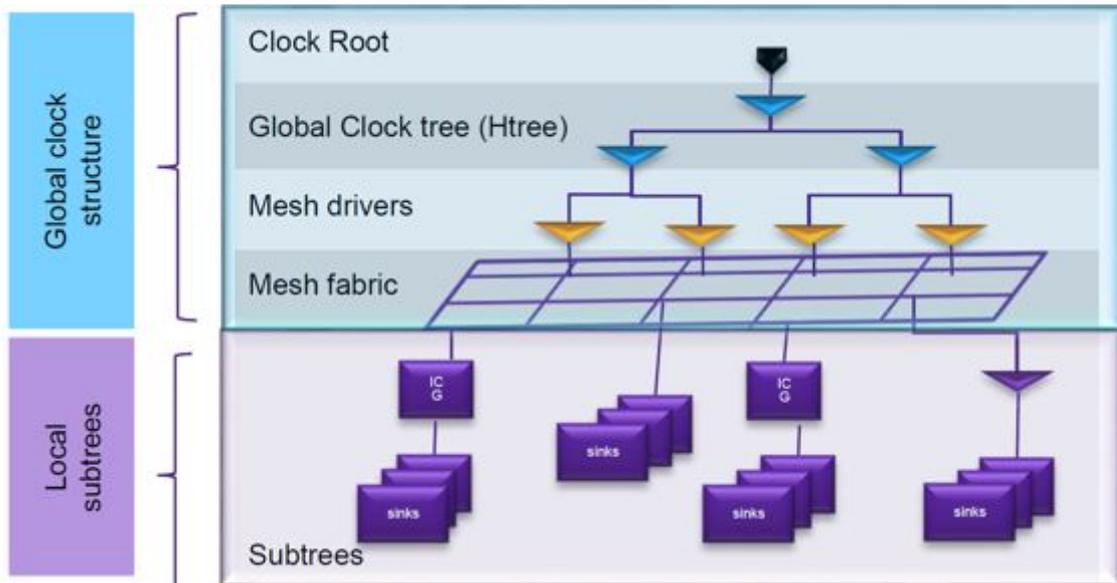


Figure 6.1.1: Multisource CTS

conventional CTS delivers the lowest power consumption and the easiest flow, multisource CTS offers a compromise between the two methods while favouring the OCV tolerant nature of pure clock mesh. As a result, a larger set of designs can access the considerable benefits garnered from mesh technology.

A custom clock tree generally consists of two parts:

1. Global Clock Distribution: Clock mesh driven by Htree, Clock Straps etc.
2. Local Clock Distribution: Optimized by merging and splitting clock cells, while preserving subtree levels in the input structure.

Creating clock straps is the initial step of Multisource CTS. These are straight metal shapes in a single routing layer. These consist of two forms:

Clock Mesh: It is a two-dimensional grid in a horizontal and a vertical layer, where the straps are connected by vias at the intersection points, as shown in Fig.6.1.2

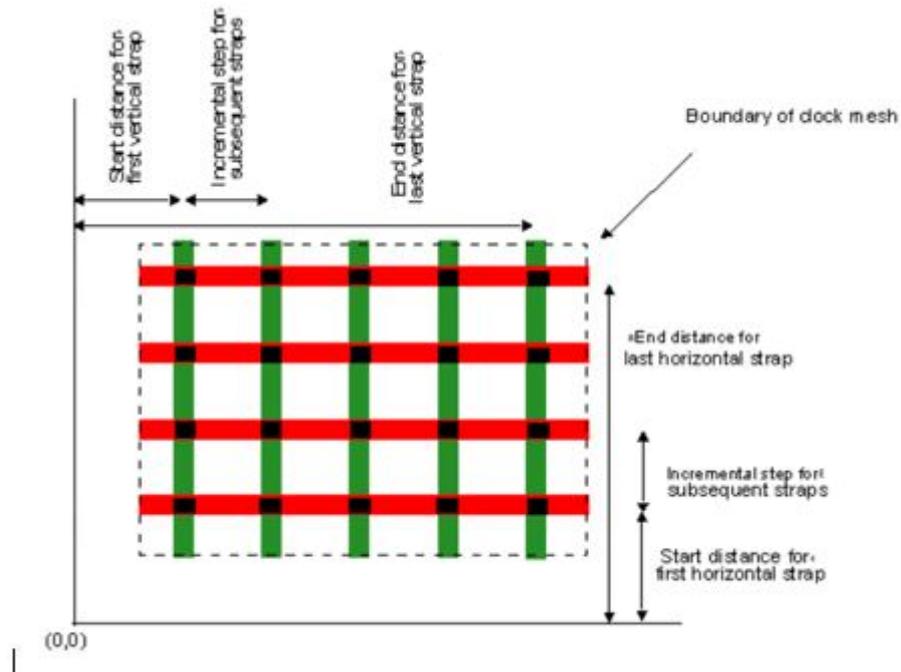


Figure 6.1.2: Clock Mesh

Clock Spine: It can be either a one- or two-dimensional structures. One-dimensional spines are straps in a single direction. Two-dimensional spines consists of one-dimensional spines connected to multiple stripes in the orthogonal direction. Stripes connected to one spine do not connect to stripes of a different spine and the minimum distance between the stripes of different spines is called the backoff, as shown in Fig.6.1.3

6.2 Clock Tree Power

Clock mesh power consumption is contributed by short circuit current due to skew from global clock tree, mesh drivers and clock mesh fabric.

Mesh causes a large increase of power consumption, in particular due to shorted buffers. It is observed that skew distribution of pre-mesh tree is important in determining the amount of short-circuit power. Fig.6.2.1 shows short circuit current in Multisource CTS. Short circuit

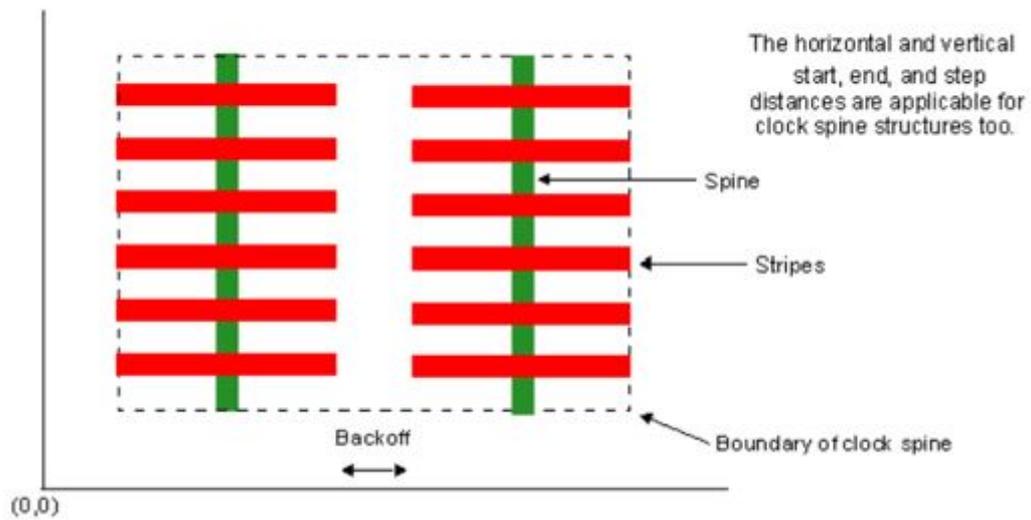


Figure 6.1.3: Clock Spine

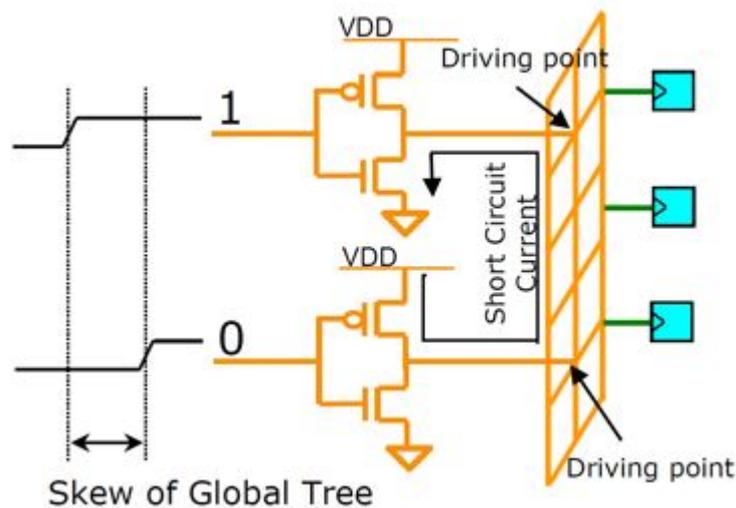


Figure 6.2.1: Short circuit current in Multisource CTS

power can be reduced by building a symmetric and balanced global clock tree (Htree) which ensures tight skew across the corners at input of mesh drivers. Mesh Fabric can be reduced by varying number of mesh straps based on load distribution and lower loads in sparser mesh fabric and hence fewer mesh drivers.

6.3 Multibit Flip-Flops

Given a design, we can reduce its power consumption by replacing some flip-flops with fewer multi-bit flip-flops. During clock tree synthesis, less number of flip-flops means less number of clock sinks. Thus, the resulting clock network would have smaller power consumption and uses less routing resource. Besides, once smaller flip-flops are replaced by larger multi-bit flip-flops, device variations in the corresponding circuit can be effectively reduced. As CMOS technology progresses, the driving capability of an inverter-based clock buffer increases significantly. The driving capability of a clock buffer can be evaluated by the number of minimum-sized inverters that it can drive on a given rising or falling time.

From Fig.6.3.1, total power consumption can be reduced because the two 1-bit flip-flops can share the same clock buffer. Inverters in flip-flops tend to be oversized according to the manufacturing rules. In ultra-deep sub-micron technology, the driving strength of clock drivers are higher so that clock drivers can drive more than one flip-flop. Merging single-bit flip-flops into one multibit flip-flop can avoid duplicate inverters and lower the total dynamic power consumption. In multibit flip-flops the inverters driving the master and slave latches are shared among all bits, MBFF shared clock, set/reset signal, scan_enable signal which help to reduce area, power and improve internal scan connection and clock routing significantly.

6.4 Criteria for using MBFF

Multibit flip-flops cells are capable of decreasing the power consumption because multibit flip-flops shared inverter driving master and slave latches. MBFF can minimize clock skew at the same time because of improved clock routing. Fig.6.4.1 shows replacing 2-bit FFs according to clock tree topologies. To obtain these benefits, the ASIC design must meet the following requirements. The single-bit flip-flops that we want to replace with multibit flip-flops must have same clock condition and same set/reset condition.

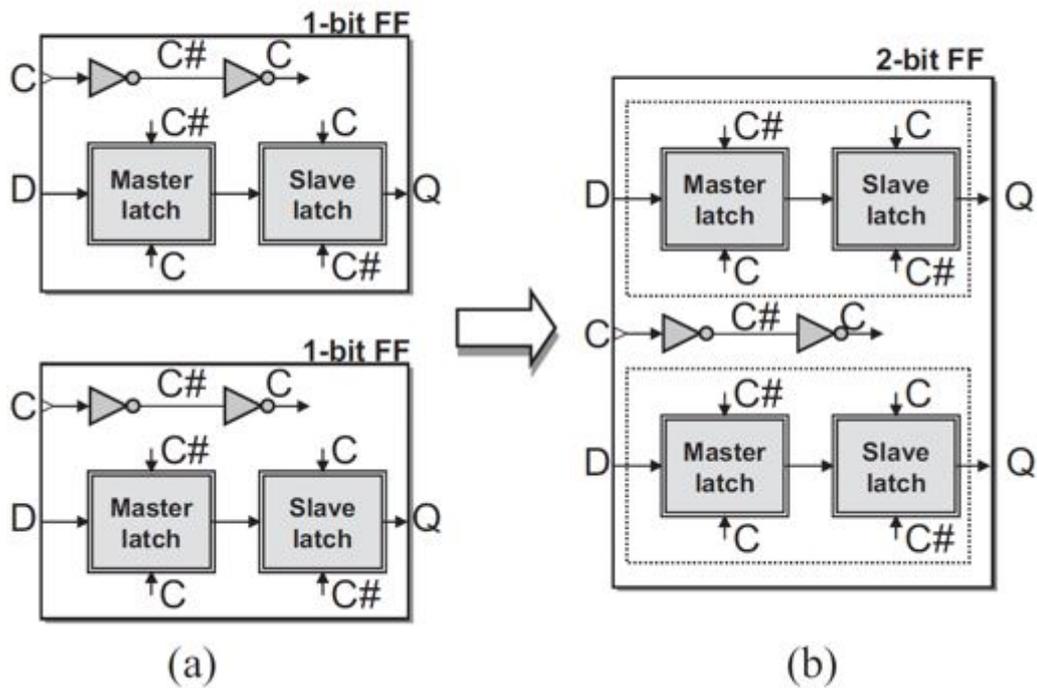


Figure 6.3.1: Merging two 1-bit flip-flops into one 2-bit flip-flop. (a) Two 1-bit flip-flops (before merging) (b) 2-bit flip-flop (after merging)

The cells are selected from the library by matching all the characteristics of a single and multiple bit flip flops. The single bit flip flops are mapped to corresponding multibit in a file so that, when the tool tries to swap the single bit flip flops, it is able to select the appropriate match for it in the multibit flip flop collection. The region on which the correct match should be checked can also be decided by the designer so that the placement of cells can be effective. The placement legality on the particular site row also should be taken care when the newly create multibit flip flops are placed. The multibit cells are comparatively bigger in size than the single bit cells which makes it consume more space in a particular location but lesser area compared to each individual cell. So, the pin congestion of the cell of should be taken care when doing the placement.

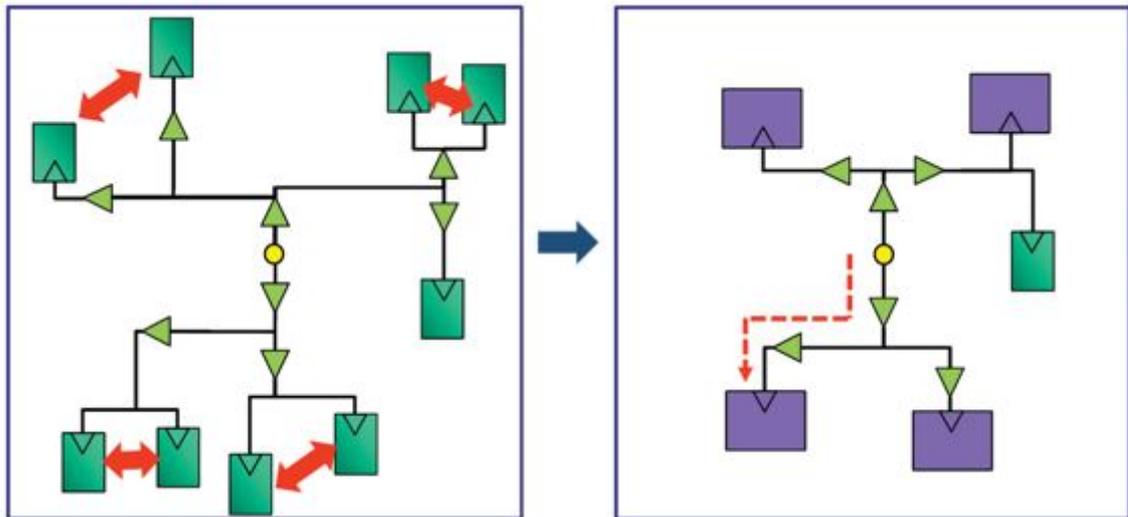


Figure 6.4.1: Applying 2-bit FFs according to clock tree topologies during timing-driven placement)

6.5 Implementation

To implement Multibit flip flop, the following procedure is followed:

1. Exclude the cells from getting swapped with its corresponding multibit version. The cell types include RTL specified cells, scan cells, macros, enable registers, debug cells, and design specific cells.
2. Set the bounding bow for the pair search. used.
3. Set the maximum capacitance difference allowed for clustering the cells.
4. The size of the merged cell should be controlled based on timing or power requirements. For timing, the smallest cell with Cmax bigger than the original cell is taken whereas for power the smallest cell with largest Cmax smaller than the original cell is taken.
5. The single-bit registers groups should be identified that can be replaced by multibit registers and then modify the netlist accordingly or generate a banking script file which can then be sourced back into the tool to replace single-bit cells with multibit cells. This can be achieved with the below two files:

FILE FORMATS

Format for specifying input map file

You can specify the input map file sequence as shown:

```
reg_group_name {list_of_reference_of_single_bit_flops }
```

```
bits {number_of_instances ref_multibit_flop } { ... }
```

```
bits {number_of_instances ref_multibit_flop } { ... }
```

Where `reg_group_name` specifies the name of the register group which can be mapped, `list_of_reference_of_single_bit_flops` specify the list of references of single-bit registers. Bits specifies the number of single-bit registers in a group to be replaced, `number_of_instance` specifies the number of multibit registers to be used, and `ref_multibit_flop` specifies the reference multibit library cell to be used.

For example:

```
reg_group_1 {REGX1 REGX2 REGX4}
```

```
2 {1 MREG2}
```

```
3 {1 MREG2}
```

```
4 {1 MREG4}
```

```
5 {1 MREG4}
```

```
6 {1 MREG2} {1 MREG4}
```

```
reg_group_2 {REGNX1 REGNX2 REGNX4}
```

```
2 {1 MREG2N}
```

```
3 {1 MREG2N}
```

```
4 {1 MREG4N}
```

```
5 {1 MREG4N}
```

```
6 {1 MREG2N} {1 MREG4N}
```

`reg_group_1 {REGX1 REGX2 REGX4}` means that single-bit registers whose reference is either `REGX1`, `REGX2`, or `REGX4` can be grouped together. `4 {1 MREG4}` specifies

that a group of four single-bit registers can be replaced by one cell whose reference is MREG4. Similarly, 6 {1 MREG2} {1 MREG4} means a group of six single-bit registers can be replaced using one instances of MREG2 and one instance of MREG4.

Format for specifying slack threshold file

This file can be used to exclude the cells based on the timing requirement.

You can specify the slack threshold file sequence as shown:

Name: threshold_number

Where name is either the name of a timing path group or the keyword others, and threshold_number is a floating-point number.

For example:

to_memory: -1

rsg_aclk: 0

others: 2

In this example, registers in timing path group to_memory having setup slack value less than -1 are ignored for banking. Similarly, registers in the timing path group rsg_aclk are ignored for banking if they have setup slack value less than 0, and registers from the rest of the timing path groups in the design are ignored for banking if their setup slack is less than 2.

6. New multibit cell from a list of registers or latches are created in the current design. All the single-bit cells in list are replaced by one multi-bit cell.
7. It should be verified that the cells in the list exist in the design and have valid locations, and do not have a "dont_touch" or "fixed" attribute.
8. The order of the specified cells determines the pin connection order of the new multi-bit

cell. For example, a net connected to the third specified cell in the list will be connected to the third bit of the multibit cell inserted by the command.

9. If the multi-bit library cell has a larger bit-width than the total bit-width of the specified cells, the pins of the unused bits of the cell are left dangling. If a pin of a specified cell does not have a corresponding pin in the multibit library cell, the pin is disconnected.
10. For the final merging of cells, the multibit register is created, the single bit cells are disconnected from the nets and then those nets are re-connected to the multibit registers.
11. Legalize placement of the multibit flip flops.



Figure 6.5.1: 2-bit Register with scan enabled

Fig.6.5.1 shows a 2-bit register with scan mode enabled. Unlike the single bit register, the d1 and d2 pins are the input data pins and the q1 and q2 pins are the output data pins but the clk driven is the same. This results in less branching of clock tree resulting in lower power consumption.

Chapter 7

Result Analysis

7.1 Introduction

In this chapter, the results obtained from the methodology and techniques used are explained. Tabular and graphical results are produced for each techniques showing the percentage improvement in the proposed techniques. A remarkable result was obtained after the completion of the design implementation. The timing, area and power were the major criteria taken into consideration.

7.2 Multisource CTS

7.2.1 Latency

The clock source latency (insertion delay) is the time it takes for the clock signal to propagate from its actual ideal waveform origin to the clock definition point in the design. The clock network latency is the time it takes a clock signal to propagate from the clock definition point to a register clock pin. Fig.7.2.1 shows clock path to sequential for Multisource Conventional CTS.

Table 7.2.2 describes the latency comparison between Multisource CTS and conventional

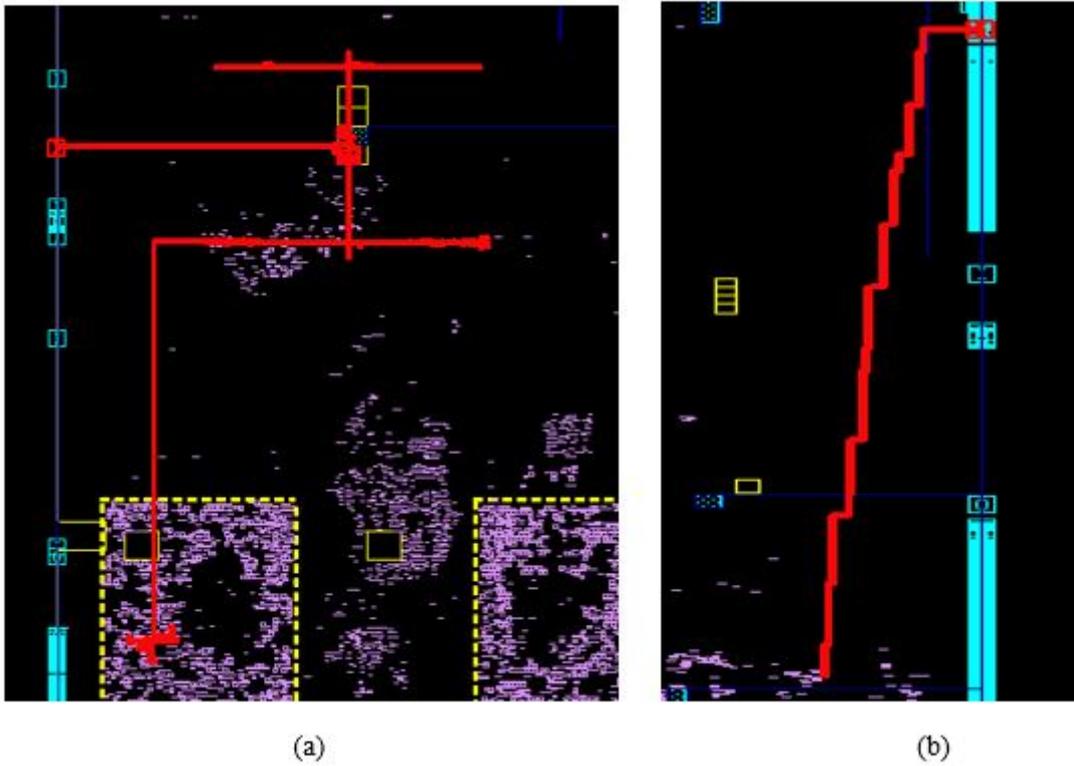


Figure 7.2.1: Clock path to a sequential in (a) Multisource CTS (b) Conventional CTS

CTS. The latency is less than half in case of Multisource CTS. The latency in a rectilinear partition for conventional CTS is large as the placing of clock port location is very challenging.

	Transition	Source Latency (ps)	Network Latency (ps)	Total latency (ps)
Multisource CTS	28.31	-91	190.03	99.03
Conventional CTS	30.56	-91	376.99	285.99

Figure 7.2.2: Latency Comparison between Multisource CTS Conventional CTS

7.2.2 Timing

In Conventional Clock Tree, we perform timing analysis using signoff static timing engines and similar timing engines embedded within place and route tools. In Clock Mesh and multi-source CTS, we perform timing analysis in mesh fabrics using circuit simulation. The standard is for automation within the place and route tool to launch the simulation run and then annotate the timing values onto the design for subsequent static timing reports and analyses.

	Number of violating paths(NVP)	Worst Negative slack (WNS) (ps)	Total Negative Slack (TNS) (ps)
Multisource CTS	107	-9.21	-530.32
Conventional CTS	359	-56.39	-1055.72

Figure 7.2.3: Timing Comparison between Multisource CTS Conventional CTS

7.2.3 Power

Power consumption is more in Conventional CTS, since clock routing has too many buffers, means there is a lot of capacitance to drive by the clock buffers. The clock buffers thus consume a tremendous amount of power driving the large capacitance of a clock tree. Since the clock mesh also consume much power, the power comparison is almost equal in both the cases as shown in Table 7.2.4

7.2.4 Conventional vs Multisource CTS Comparison

The bar graph in Figure 7.2.5 shows the percentage difference of latency, skew, WNS and power for Conventional and Multisource CTS. The graph clearly proves that, the Multisource CTS is way more advantageous than conventional CTS in timing perspective.

	Internal Power (uW)	Switching Power (uW)	Leakage Power (uW)	Total Power (uW)
Multisource CTS	1.17e+05	7.86e+04	8.49e+03	2.04e+05
Conventional CTS	1.21e+05	8.85e+04	7.65e+03	2.18e+05

Figure 7.2.4: Timing Comparison between Multisource CTS Conventional CTS

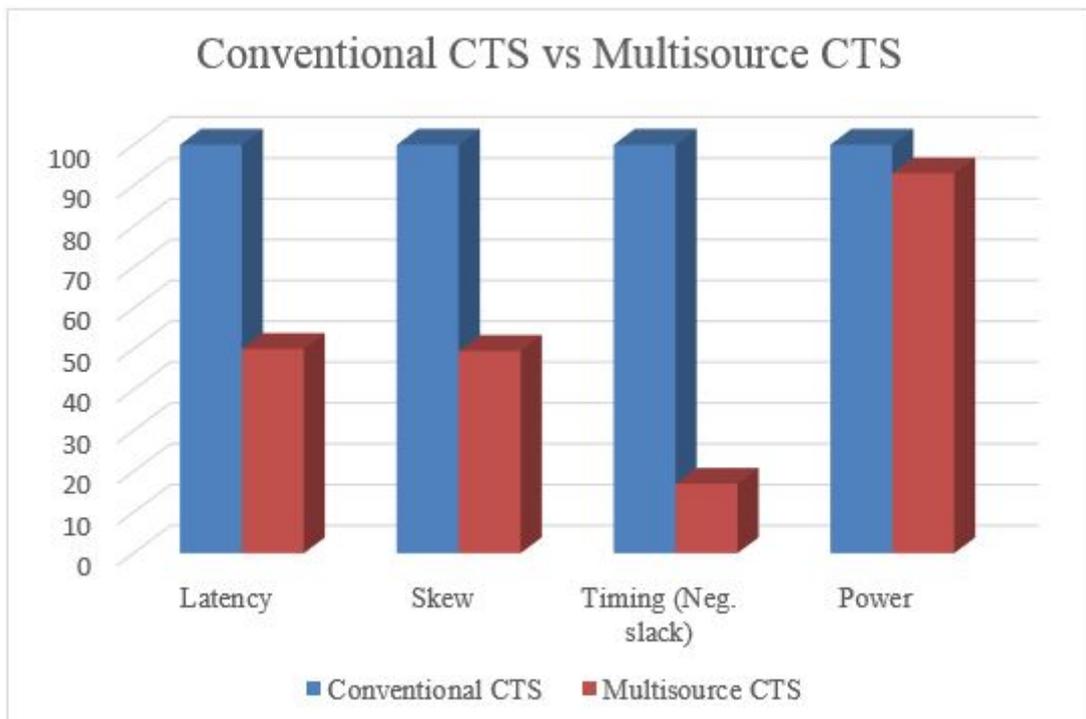


Figure 7.2.5: Conventional vs Multisource CTS Comparison

7.3 Multibit Flip Flops

7.3.1 Cell Count

One of the major advantages of using Multibit Flip flop is the sequential count reduction by merging the flip flops resulting in area reduction. Table 7.3.1 clearly shows the advantage of MBFF over SBFF by reducing the overall cell count and sequential cell count.

	Cell Count	Sequential Count
MBFF	85147	12004
SBFF	90566	18106

Figure 7.3.1: Conventional vs Multisource CTS Comparison

7.3.2 Power

Power consumption in a design is directly proportional to the number of cells present. So the power reduction is obtained due to the reduction of cells and lesser number of routes as shown in Table 7.3.2.

	Internal Power (uW)	Net Switching power(uW)	Leakage power(uW)	Total Power(uW)
MBFF	0.98e+05	6.6e+04	7.13e+03	1.7e+05
SBFF	1.17e+05	7.86e+04	8.49e+03	2.04e+05

Figure 7.3.2: Power Comparison between MBFF SBFF

7.3.3 Timing

As expected, timing improvements are less as the design with SBFF has more flexibility to move inside the design resulting in efficient placement. But when it comes to combinational cells, the count for MBFF design will have more to meet the timing as shown in Table 7.3.3

	Number of violating paths(NVP)	Worst Negative slack (WNS) (ps)	Total Negative Slack (TNS) (ps)
MBFF	107	-9.21	-530.32
SBFF	126	-12.39	-755.72

Figure 7.3.3: Timing Comparison between MBFF SBFF

7.3.4 SBFF vs MBFF Comparison

The bar graph in Figure 7.3.4 shows the percentage difference of cell count Sequential cell count, power WNS. The graph clearly proves that, the MBFF is way ahead in comparison to SBFF in case of area improvement and Power.

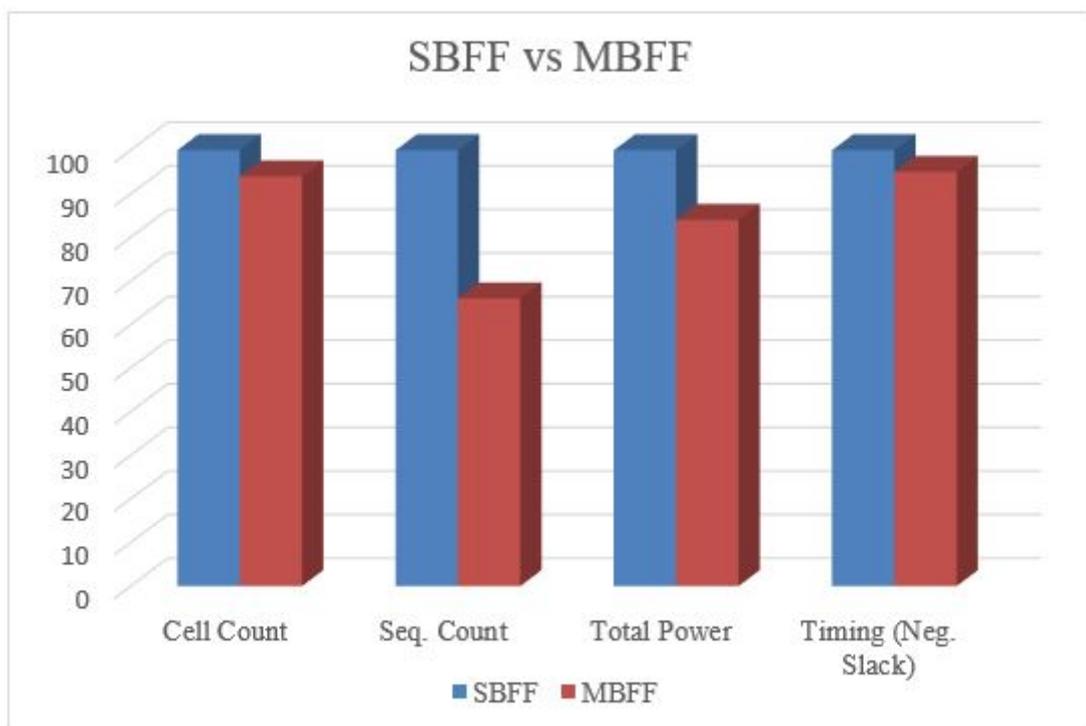


Figure 7.3.4: SBFF vs MBFF Comparison

Chapter 8

Conclusion and Future Work

8.1 Summary

The problem statement was to achieve reduction in power consumption with minimal on chip variation, latency, skew without any trade off. This was achieved as per proposed in the project. The methodologies used were Multisource Clock Tree Synthesis, Implementation of Multibit Flip Flop and use of UPF. The results obtained were as expected and more fruitful. Experience with these methodologies will enable designers to make the most optimal design choice given the design goals: power consumption, flow ease, and time-to-market pressure.

8.2 Work Conclusion

Even though the techniques were related to Clock Tree Synthesis, the whole physical design flow play a major role in the convergence of these techniques. Use of Multisource CTS over Conventional CTS shows reduction in power consumption in design.

Using Multibit Flip Flop is an effective and efficient implementation methodology to reduce the power consumption by merging single bit flip flop. Experimental results on the technology used indicate that MBFF is very effective and efficient method in deep sub micro design

to reduce power, save area and improve routing of clock tree.

8.3 Future Scope of Work

Multisource CTS can be improved by reducing the complexity of design required to implement the technique in the SoC level. In this project, the single bit flip flop was swapped with 2-bit registers. Bits more than two can also be implemented based on the design handled. In congested designs, higher level of bits can be used so that the area consumption is reduced.

References

- [1] Synopsys Solvent.
- [2] Intel Internal Documents
- [3] Spyglass_lp user guide by Atrenta and Spyglass GUI
- [4] Place and Route using IC compiler user guide manual by Synopsys.
- [5] IEEE Standard for Design and Verification of Low-Power Integrated Circuits.
- [6] RTL-to-Gate Level Synthesis using Design Compiler user guide manual by Synopsys.
- [7] Harvey Toyama, Multi-Source CTS Delivers Flexible High Performance and Variation Tolerance,
- [8] The International technology Roadmap for Semiconductors, 2007.1.
- [9] Static Timing Analysis for Nanometer Designs - A Practical Approach by J. Bhasker, Rakesh Chadha.
- [10] S. Tam, S. Rusu, U. N. Desai, R. Kim, J. Zhang, and I. Young, Clock generation and distribution for the first IA-64 microprocessor, IEEE J. Solid-State Circuits, vol. 35, no. 11, pp. 1545-1552, Nov. 2000.