RTL To GDSII Implementation of Low Power Design

Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

In

Electrical Engineering

(VLSI Design)

By

Harshil N. Upadhyay

(12MECV31)

Department of Electrical Engineering

Institute of Technology

Nirma University

Ahmedabad-382 481

May 2014

RTL To GDSII Implementation of Low Power Design

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

In

Electronics & Communication Engineering

(VLSI Design)

By

Upadhyay Harshil N.

(12MECV31)

Under the Internal Guidance of

Dr.Nagendra Gajjar

and

External Guidance of

Mr.K G Varun

Department of Electronics & Communication Engineering

Institute of Technology

Nirma University

Ahmedabad-382 481

Declaration

This is to certify that

1. The thesis comprises of 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 acknowledgement has been made in the text to all other material used.

Harshil N. Upadhyay

Certificate

This is to certify that the Major Project entitled "*RTL To GDSII Implementation of Low Power Design*"submitted by *Harshil N. Upadhyay* (*12MECV31*), towards the partial fulfilment of the requirements for the degree of Master of Technology in Electronics and Communication Engineering Branch of Institute of Technology, Nirma University, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my 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 my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

----------	----------
Guide	
Dr.Nagendra Gajjar	Dr. N.M.Devashrayee
Professor	PG Co-ordinator (VLSI Design)
----------	----------
Dr. P.N Tekwani	Dr K Kotecha
Professor & Head, EE	Director, IT-NU

Date: _____ **Place: Ahmedabad**

Acknowledgement

I would have never succeeded in completing my work without the cooperation, Encouragement and help provided to me by various people.

Firstly, my sincere thanks to the Flow and Methodology team. Their guidance ,clarity of thought and support motivated me to bring this project to its present state.

I am highly indebted to Group manager **K G Varun**,for providing necessary information regarding the project and also for their constant guidance, supervision, kind co-operation, and invaluable support in all aspects. My thanks and appreciations also go to my colleagues and team members in developing the project and for providing me with a lively and energetic work environment.

I would like to express my sincere gratitude to **Dr. Ketan Kotecha** (Director, Nirma University, Ahmedabad) for his continuous guidance, support and enthusiasm. I would take this opportunity to thank **Dr. P. N. Tekwani** (Head of Department, Electrical Engineering), **Dr. N. M. Devashrayee** (Professor and Program Coordinator, M.Tech - EC (VLSI Design)), Internal Guide **Dr. Nagendra Gajjar** and all the faculties at **Nirma University (VLSI Design)**, for their vision and relentless effort, support, and encouragement to provide me with this excellent opportunity to carry out my project work in such a highly renowned and esteemed organization,Freescale Semiconductor. I am equally thankful to **Freescale Semiconductor** for providing me the invaluable exposure to the industry and the current market trends.

Finally, I would like to express my heartfelt thanks to my parents and colleagues for their blessings and for their constant love and support.

Upadhyay Harshil N.

(12MECV31)

Abstract

Design(cf5x_core) Consist of Four Modules cf5x_core_in, cf5x_core_out,spp_l2_cache, cf5_core_kbus, Design is operating at 200Mhz.

There is various Low power techniques available 1.Multple Supply Voltage(MSV) 2.Power Shut off (PSO) 3.Dynamic voltage frequency scaling (DVFS)

CPF enabled flow is used for logic synthesis and physical implementation.

Common Power Format(CPF) is use to capture the power intent for your design,a standardized format for specifying power-saving techniques early in the design process, to deliver an end-to-end low-power design solution.

Low power DVFS technique is implemented. Design is divided in two power domains pd_vddb_vss and pd_vdda_vss.power doamains are operating at the voltages. power domain pd_vddb_vss is operating at the 1.3v and its default power domain. power domain pd_vdda_vss is operating at the 1.1v and has the switched on and off capability.

power domain pd_vddb_vss has module cf5_core_kbus.power domain pd_vdda_vss has module cf5x_core_in, cf5x_core_out & spp_l2_cache.

Level Shifter are used for the signal that passes between power domain pd_vddb_vss and pd_vdda_vss because to up and down signal voltage level.

Isolation cells are used for the signal going from pd_vdda_vss to pd_vddb_vss to pass known value to power domain pd_vddb_vss when power domain pd_vdda_vss is switched off.

State Retention cells are used for sequential cells in power domain pd_vdda_vss to retain there previous state during the switched off state.

Power Switches are added in the power domain pd_vdda_vss to provide switched on and off capability.

Design is synthesized using CPF enabled flow.Level Shifter,Isolation and state retention cells are inserted at this stage

Design is Physically Implemented using the CPF enabled flow.Power Switches

are added at this stage.

Physical implementation steps are 1.floorplanning 2.power planning 3.placement 3.ClockTree Synthesize 4.Routing 5.Timing analysis 6.Signal Integrity 7.Power Analysis.

Freescale Semiconductor At A Glance

- A world leader in providing the semiconductor solutions that help our customers improve quality of life for everyone, both today and in the future.
- Freescale Semiconductor, Inc. is an American company that produces and designs embedded hardware, with 17 billion semiconductor chips in use around the world. The company focuses on the automotive, consumer, industrial and networking markets with its product portfolio including microprocessors, microcontrollers, digital signal processors, digital signal controllers, sensors, RF power ICs and power management ICs. In addition, the company offers software and development tools to support product development.
- Among the world's largest semiconductor companies
- A leading technology innovator
- A pioneer and visionary leader in sustainability
- President and CEO: Gregg Lowe
- The company is headquartered in Austin, Texas with design, research and development, manufacturing and sales operations in more than 20 countries.
- Freescale is currently ranked 7th among the semiconductor sales leaders in the United States and is ranked 16th worldwide.

- The company also holds an extensive patent portfolio, including approximately 6,100 patent families.
- Freescale filed to go public on February 11, 2011 and completed its IPO on May 26, 2011. Freescale is traded on the New York Stock Exchange under the ticker symbol FSL.
- Freescale is a leader in embedded processing solutions for the automotive, consumer, industrial and networking markets. From microcontrollers and microprocessors to sensors, analog ICs and connectivity, our technologies are fueling the next great wave of innovation.
- Earlier known as MOTOROLA and later in 2004 as FREESCALE.

Contents

List of Figures

Chapter 1

Introduction

RTL-GDSII flow is used for validation of DDK(Digital Design kit) cell sets using integration with tools and flows,to make sure that when these DDK's are going to integrate in SOC's, the flow will not break because of these DDK's. DDK low

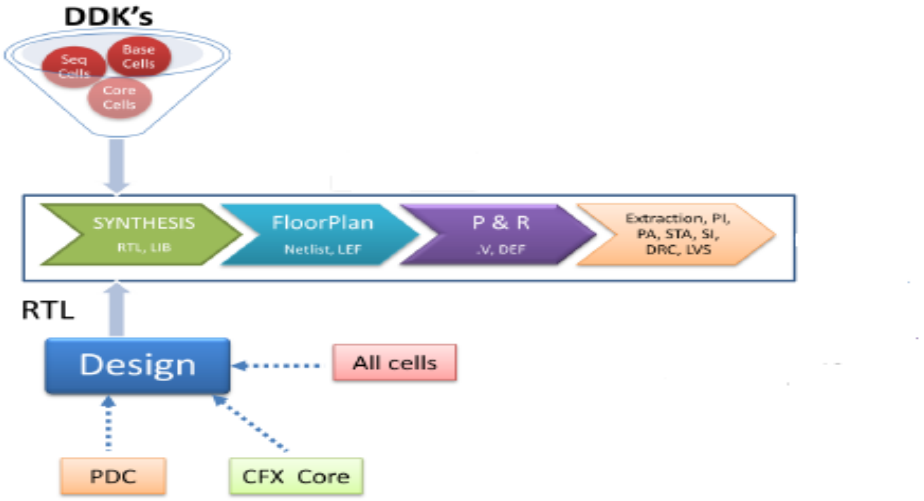

Figure 1.1: RTL To GDSII Flow

power cells used for validation so that when these cells integrate in final SOC.it will not break the RTL-GDSII flow down the line.

low power cells are Level Shifter,Isolation cells,Always on cells,Power Switch cell,state-retention power gated(SRPG) cell.

RTL to GDSII flow is mainly divided into the two part.

1.Logic Synthesis

2.Physical Implementation

CPF enabled Flow is used for both logic synthesis and Physical Implementation. CPF is a file which contains the information about which low power technique is used to implement the design.

1.1 Low Power Techniques

1.1.1 Multiple Supply Voltage

A Multiple Supply Voltage (MSV) design uses multiple supply voltages for the core logic.

In Figure 1.2 the top design and instance inst_A operate at voltage VDD1, while instance inst_B operates on voltage VDD2 and instance inst_C operates at voltage VDD3.

A portion of the design that operates at the same operating voltage (that is, uses the same main power supply) belongs to the power domain that corresponds to that operating voltage.

A steady state of the design is called a power mode.Pure MSV designs have only one power mode because the operating voltage of the power domains is not to change.

To pass signals between portions of the design that operate on different voltages, level shifters are needed.

The majority of cells in a power domain are driven by the same power supply, except for the level shifters which are driven by multiple power supplies.

Level shifters have two sets of power and ground pins, and are therefore associated with two power domains: a primary and a secondary power domain.

A power mode will also have a typical set of timing constraints associated

with it.

Operating Voltage (OV)	Instances Operating on OV	Libraries characterized for OV	Power Domains for OV
VDD1: 0.8	top, inst_A	lib1 lib2	PD1
VDD2: 1.0	inst_B	lib3	PD2
VDD3: 1.2	inst_C	lib4	PD3

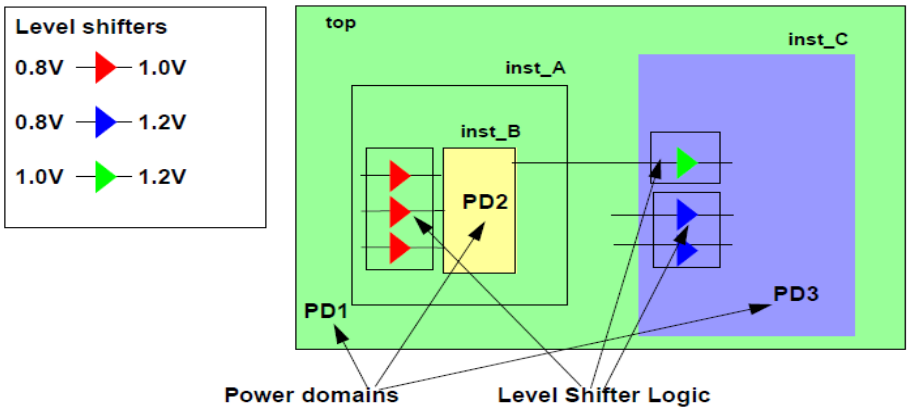

Figure 1.2: Example Of MSV Deign

1.1.2 Power Shut Off

A design using power shut off (PSO) implementation is a design of which some portions can be switched on and off as needed (or possibly) to save leakage and dynamic power.

The example design in Figure 1.3 has three power domains: The top-level of the design top and hierarchical instances inst_C and pm_inst, are always switched on they belong to domain PD1.

Hierarchical instances inst_A and inst_B are always switched on and off simultaneously.they belong to power domain PD2.

Hierarchical instance inst_D can be switched on and off independently from hierarchical instances inst_A and inst_B it belongs to power domain PD3

To prevent that unknown states in the power domains that are powered down propagate to the domains that remain powered on isolation cells are needed at the

boundaries of the power domains that are powered down.

To facilitate powered down blocks to resume normal operation, state retention cells can be used for some sequential cells to keep their previous state prior to power down.

To facilitate power down operation power switches are added in the power domain.

Power Mode	Power Domain		
	PD1	PD2	PD3
PM1	1.1V	1.1V	1.1V
PM2	1.1V	0.0V	1.1V
PM3	1.1V	0.0V	0.0V

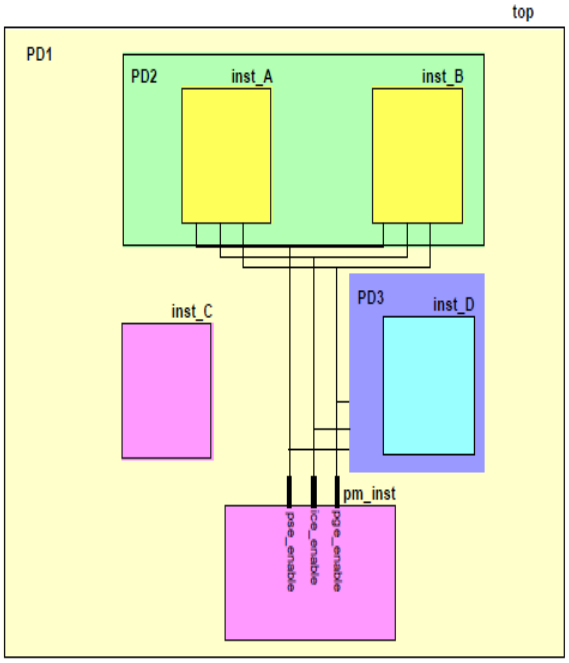

Figure 1.3: Example Of PSO Methodology

1.1.3 Dynamic voltage frequency scaling

Dynamic voltage frequency scaling (DVFS) reduces the power in the chip by scaling down the voltage and frequency when peak performance is not required.

A design using DVFS can be seen as a special case of an MSV design operating in multiple design modes.

- In a pure MSV design different portions of the design operate on different voltages and these portions remain operating at their respective operating voltage.

- In a DVFS design, in addition some portions can dynamically change to other voltages depending on the design mode or can even be switched off.

The example design in Figure 1.4 has three power domains: The top-level of the design dtmf_recvr_core and hierarchical instances pm_inst are always switched on they belong to domain AO. Its supply voltage is varying.

Hierarchical instance PLLCLK_INST is switched on belong to domain PLL.

Hierarchical instance TDSP_CORE_INST can be switched on and off ,belong to domain TDSP_CORE.

Level Shifter,Isolation,State Retention and Power Switch Cells are inserted as per the MSV and PSO Methodology.

Power Mode	Corresponding Power Domain		
	AO	PLL	TDSPCore
full	0.99	0.99	0.792
slow	0.99	0.99	0.0
sleep	0.792	0.99	0.0

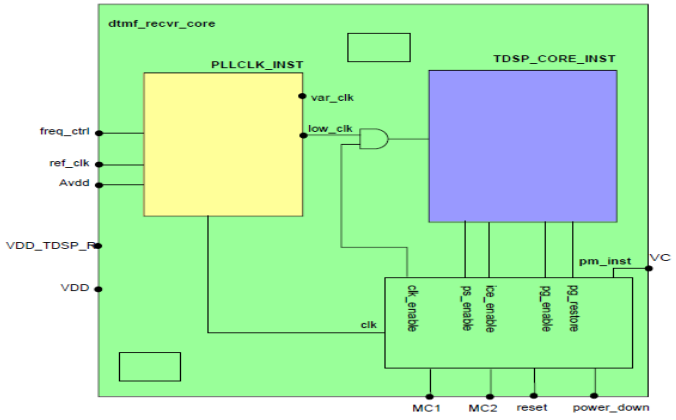

Figure 1.4: Example Of DVFS Methodology

1.2 Physical Implementation Flow

Physical implementation steps are 1.floorplanning 2.power planning 3.placement 3.ClockTree Synthesize 4.Routing 5.Timing analysis 6.Signal Integrity 7.Power Analysis.

1.2.1 Floorplanning

Floorplanning a chip or block is an important task of physical design in which the location, size, and shape of soft modules, and the placement of hard macros are decided.

Depending on the design style or purpose, floorplanning can also include row creation, I/O pad or pin placement, bump assignment (flip chip), bus planning, power planning, and more.

For example, floorplanning is very important when preparing the design for timing closure and detailed routing. Floorplanning, in conjunction with placement and trial routing, can be an iterative design process.

Floorplanning usually starts by preplacing blocks, modules, and submodules according to the prepared floorplan. All other modules or blocks not in the prepared floorplan are left outside the chip area.

1.2.2 Power Planning

Power planning and routing is composed of the following components:

- Adding a core ring
- Adding block rings
- Adding stripes to the core area
- Adding stripes over blocks within the design
- Creating a pad ring

- Connecting pad pins
- Routing standard cell pins
- Connecting block pins
- Connecting unconnected stripe
- Routing to power bumps

1.2.3 Placement

After floorplanning, place the cells in the design. Placement considers the modules that were placed during floorplanning and takes into account the hierarchy and connectivity of the design. It honors floorplanning constraints, including guides, regions, and fences.

Placement treats preroutes the same way it treats routing blockages: It places standard cell instances at legal locations where there should not be any DRC violations against preroutes or routing blockages.

Typically, you use preroutes for special nets that are floorplanned (predesigned) before placement,such as power, ground, and clock mesh nets, where you do not want any standard cells placed underneath. Placement Objective are:

Total wirelength: Minimizing the total wirelength is the primary objective. This not only helps minimize chip size, and hence cost, but also minimizes delay, which are proportional to the wirelength.

Timing: The clock cycle of a chip is determined by the delay of its critical path. Given a performance specification, a placer must ensure that no path exists with delay exceeding the maximum specified delay.

Congestion: While it is necessary to minimize the total wirelength to meet the total routing resources, it is also necessary to meet the routing resources within various local regions of the chips core area. A congested region might lead to excessive routing detours, or make it impossible to complete all routes.

1.2.4 Clocktree Synthesize(CTS)

CTS is the process of distributing clock signals to clock pins based on physical information. After placement of cells the tree of clock is synthesized.

CTS is of two types buffered and unbuffered

CTS is needed For synchronized data transfer between functional elements are synchronized by clock signals.

Clock tree design constraints

- Maximum capacitance
- Maximum Fanout
- Maximum buffer levels

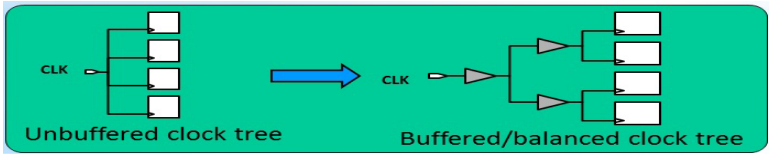

Figure 1.5: Buffered CTS

1.2.5 Routing

router performs concurrent signal integrity, timing-driven, and manufacturing aware routing (SMART routing) of cell, block, or mixed cell and block level designs.

Routing creates physical connection to all clock and signal pins through metal interconnection Metal layer has its own grid and preferred routing direction.

Objective :-

1Routed path must meet setup and hold timing and clock skew requirements.

2.Metal routes must meet physical DRC requirements.

Full routing consists of global and detailed routing. You can repeat detailed routing incrementally on a routed database.

1.Global Routing

During this phase, the router breaks the routing portion of the design into rectangles called global

routing cells (gcells) and assigns the signal nets to the gcells. The global router attempts to find the

shortest path through the gcells, but does not make actual connections or assign nets to specific tracks within the gcells.

It tries to avoid assigning more nets to a gcell than the tracks can accommodate. The detailed router uses the global routing paths as a routing plan.

The router can generate a map of the gcells, called a congestion map, that you can examine to see the approximate number of nets assigned to the gcells.

The congestion map uses colors to indicate whether there are too few, too many, or the correct number of nets assigned to the gcells. If the router assigns too many nets to a gcell, it marks the gcell as over-congested.

2.Detail Routing

During this phase, the NanoRoute router follows the global routing plan and lays down actual wires that connect the pins to their corresponding nets. The detailed router creates shorts or spacing violations rather than leave unconnected nets.

The router runs search-and-repair routing during detailed routing. During search and repair, it locates shorts and spacing violations and reroutes the affected areas to eliminate as many of the violations as possible.

The primary goal of detailed routing is to complete all of the required interconnect without leaving shorts or spacing violations.

1.2.6 Timing Analysis

The goal of timing analysis is to verify that a design meets timing requirements under a specified set of timing constraints, such as arrival and required times, operating conditions, slew rates, false paths, and path delays.

Performing timing analysis lets you determine how fast a design can run without incurring timing violations. You can use the results of timing analysis to fine tune and debug the speed-limiting, critical paths in a design.

Timing analysis includes the following features and capabilities:

Performs setup time analysis, which analyzes violating paths due to timing

Performs hold time analysis Performs analysis in ideal and propagated mode.

Chapter 2

Design Low Power Intent

Common Power Format (CPF) is used to capture the power intent of the Design. CPF is standardized format for specifying power-saving techniques early in the design process, to deliver an end-to-end low-power design solution to IC engineers.

CPF is a design specification language that addresses the limitation in traditional design automation tool flows by capturing the designer's intent for power management and enabling the automation of advanced power-lowering design techniques.

The CPF enables all design, verification, implementation - and technology-related power objectives to be captured in a single file and allows the application of that data across the design flow, providing a consistent reference point for design development and production.

CPF can be divided into two parts Technology CPF and Design CPF.

Technology CPF contains information such as

- Level shifter cells
- Isolation cells
- State-retention cells
- Switch cells

- Always-on cells

Design CPF contains information such as

- Power domain
  - Logical: hierarchical modules as domain members
  - Physical: power/ground nets and connectivity
  - Analysis view: timing library sets for power domains
- Power Logic
  - Level Shifter Logic
  - Isolation Logic
  - State-Retention logic
  - Switch Logic & Control Signals
- Power mode
  - Mode and transitions

CPF is a single source for all portions of the flow But not all commands are recognized by all tools.

2.1 Design Power Intent

Design(cf5x core) Consist of Four Modules cf5x_core_in, cf5x_core_out,spp_l2_cache, cf5_core_kbus, Design is operating at 200Mhz.

Low power DVFS technique is implemented. Design is divided in two power domains pd_vddb_vss and pd_vdda_vss.power domains are operating at different the voltages

1.pd_vddb_vss

2.pd_vdda_vss

power domain pd_vddb_vss is operating at the 1.3v and its default power domain.

power domain pd_vdda_vss is operating at the 1.1v and has the switched on and off capability.

A portion of the design that operates at the same operating voltage (that is, uses the same main power supply) belongs to the power domain that corresponds to that operating voltage.

Instances in both Power Domains are as below.

pd_vddb_vss - cf5_core_kbus

pd_vdda_vss - cf5x_core_in,cf5x_core_out and spp_l2_cache

Power and ground net For both power domains are as below.

pd_vddb_vss - vddb (voltage 1.3v) , vss(voltage 0.0v)

pd_vdda_vss - vdda (voltage 1.1v), vss (voltage 0.0v)

power net vdda is internal net of the design and its not directly connected to design power pin. its connected to the power net vdda_on (voltage 1.1v) through power switches in the power domain pd_vdda_vss.so when power switches are on they are connected and when power switches are off they are disconnected at this point power domain pd_vdda_vss is switched off.

for signals that are passing from pd_vdda_vss to pd_vddb_vss, up level shifters are used and there location are in the pd_vddb_vss

for signals that are passing from pd_vddb_vss to pd_vdda_vss, down level shifters are used and there location are in the pd_vdda_vss

signals that are passing from pd_vdda_vss to pd_vddb_vss isolation cells are added.isolation cells are added to provide constant 0 output to signals to power domain pd_vddb_vss.location of the isolation cells is in pd_vddb_vss .control signal of the isolation cell is iso_enable.

State Retention cells are used for sequential cells in power domain pd_vdda_vss to retain there previous state during the switched off state.control signal for the state retention cell is sr_enable.

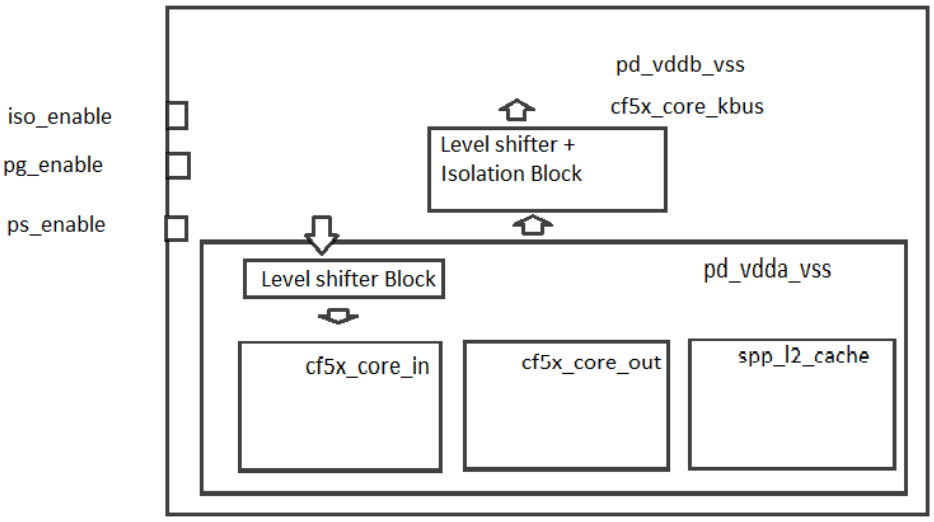

Figure 2.1: low power intent

Different Control signals for the design are as below.

Table 2.1: Control Signals

Power Domain	power switch	isolation cell	state retention cell
pd_vddb_vss	no control signal	no control signal	no control signal
pd_vdda_vss	so_enable	iso_enable	sr_enable

A steady state of a design in which some power domains are switched on and some power domains are switched off is called a power mode.In a power mode, each power domain operates on a specific voltage (nominal condition).Design is operating in two modes.

Table 2.2: Power Modes

Power Mode	pd_vddb_vss	pd_vdda_vss
mode0	1.3	1.1
mode1	1.3	0.0

2.2 Technology CPF

2.2.1 Specify Library Set

A group of library is attached with each power domain which contain the information about the standard cells.power domain is synthesized using the libraries attached with it.

library sets are defined in the technology CPF using the command define_library_set.

Four library sets are defined bc1,bc2,wc1,wc2.

library set bc1 and wc1 are for power domain pd_vddb_vss.as operating range of the design is -40 C to 125 C. set bc1 contains library characterized at -40 C and wc1 contains library characterized at 125 C.both sets are characterized at operating voltage 1.3v.

library set bc2 and wc2 are for power domain pd_vdda_vss.as operating range of the design is -40 C to 125 C.

set bc2 contains library characterized at -40 C and wc1 contains library characterized at 125 C.both sets are characterized at operating voltage 1.1v.

define_library_set -name bc1 -libraries Lib1_bc,Lib2_bc

define_library_set -name bc2 -libraries Lib3_bc,Lib4_bc

define_library_set -name wc1 -libraries Lib1_wc,Lib2_wc

define_library_set -name wc2 -libraries Lib3_wc,Lib4_wc

2.2.2 Level Shifter Cells

power domain has different power voltages but same ground voltage,so to pass signals both power domain power level shifter(LS) is used.

power level shifter is modeled using the define_level_shifter_cell command.

Up and Down level shifter are used with the valid location to:so for signal going from pd_vdda_vss to pd_vddb_vss level shifter are placed in domain pd_vddb_vss and vice versa for down level shifter.

Up level shifters has two power pins primary power pin is connected to vdda and secondary power pin is connected to vdda.Down Level Shifter has only single power pin connected to vdda.

Modeling of up Level Shifter

define_level_shifter_cell -cells XYZ -input_voltage_range 0.550:1.320 -output_voltage_range 0.550:1.320 -direction UP -input_power_pin VDDI -output_power_pin VDD -ground VSS -valid_location to

Modeling of Down Level Shifter

define_level_shifter_cell -cells XYZ -input_voltage_range 0.550:1.320 -output_voltage_range 0.550:1.320 -direction down -output_power_pin VDD -ground VSS -valid_location to

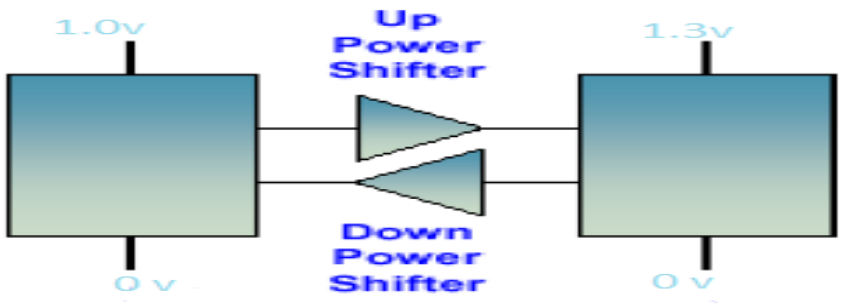

Figure 2.2: Level Shifter

2.2.3 Isolation Cells

power domain pd_vdda_vss is switchable domain,when pd_vdda_vss is witched off at this time for signal passing from pd_vdda_vss to pd_vddb_vss has unknown value,to make the value of that signal isolation cells are used.isolation cells which provide high output are used.

isolation cells enable pin is connected to pso_enable.

isolation cell with valid location ẗoïs used.so isolation cells will be placed in power domain pd_vddb_vss.

isolation cells has two power supplies primary power pin is connected to vdda and secondary power pin is connected to vddb.

isolation cells are modeled using the define_isolation_cell.

Modeling isolation cell.

define_isolation_cell -cells XYZ -power VDDR -ground VSS -enable ISO -valid_location to

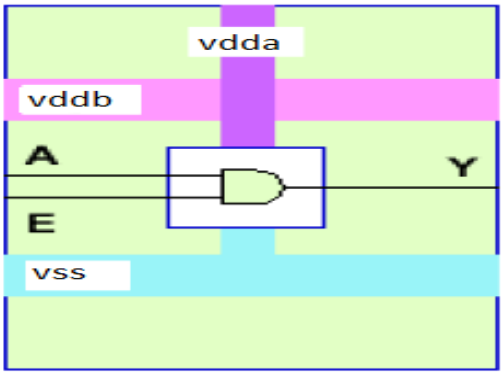

Figure 2.3: Level Shifter

2.2.4 State Retention Cells

State retention cells are used for sequential cells to keep their previous state prior to power down. State Retention Cells with Save and Restore Controls are used.the cell saves the current value when the save control pin is activated and the power is on, while the cell restores the saved value when the restore control pins is activated.SRPG cells are modeled using the

define_state_retention_cell -cells XYZ -cell_type my_retention_fsb2dprbq -clock_pin CK -restore_function B2 -save_function B1 -power_switchable VDD -power VDDR -ground VSS

save pin is connected to !sr_enable and restore pin is connected to sr_enable.

2.2.5 Power Switch Cells

Power Switch Cells are used connect and disconnect the power (or ground) supply from the gates in internal switchable.

Power Switch Cells control signal is ps_enable.

Power Switch Cells are modeled using the command define_power_switch_cell.

Modeling Power Switch cell

define_power_switch_cell -cells XYZ -stage_1_enable EN -stage_1_output ENX -type header -power_switchable VDDC -power VDDP -stage_1_saturation_current 2.677987e-03 -stage_1_on_resistance 4.854392e+02 -leakage_current 1.063105e-10

.

2.3 Design CPF

First step is to specify for which design CPF is created using the set_design

set_design cf5x_core

nominal condition are created which specifies operating voltage and state on/off.nominal conditions are associated with power domains for a given power mode.Three operating voltage present in our design so three nominal conditions are created.

create_nominal_condition -name nc_1p3 -state on -voltage 1.3

create_nominal_condition -name nc_1p1 -state on -voltage 1.1

create_nominal_condition -name nc_0p0 -state off -voltage 0.0

Library sets are than associated with nominal conditions.so power domain operating on specified nominal condition is synthesized using library associated with it.

update_nominal_condition -name nc_1p1 -library_set bc2

update_nominal_condition -name nc_1p32 -library_set bc1

Power and ground nets are created and there operating voltages are defined.Four nets are created vss,vdda_on,vddb and internal net vdda.

create_ground_nets -nets vss

create_power_nets -nets vdda -internal -voltage 1.1

create_power_nets -nets vdda_on -voltage 1.1

create_power_nets -nets vddb -voltage 1.3

Two power domains are created pd_vddb_vss and pd_vddb_vss using create_power_domain.powe domain pd_vddb_vss is default domain so instances which are not in the other domains are all included in this domain.For power domain pd_vdda_vss instances are also specified and its switched on and off capability also specified using shutoo_condition.

create_power_domain -name pd_vddb_vss -default

create_power_domain -name pd_vdda_vss -instances spp_l2_cache cf5x_core_out cf5x_core_in -shutoff_condition so_enable

power and ground nets are than associate with power domains.so each power domain will have it's power and ground net.net vddb and vss are assigned to pd_vddb_vss and net vdda_on are assigned to vdda and vss area assigned to pd_vdda_vss.

update_power_domain -name pd_vdda_vss -primary_power_net vdda -primary_ground_net vss

update_power_domain -name pd_vddb_vss -primary_power_net vddb -primary_ground_net vss

Create the level shifter rule for the signal passing from power domain pd_vddb_vss to pd_vdda_vss and also for signal passing from pd_vdda_vss to pd_vddb_vss.synthesis tool will insert level shifter according to this rule and uses level shifter defined in the technology file.

create_level_shifter_rule -name lsr_vddb_vdda -from pd_vddb_vss -to pd_vdda_vss -location to

create_level_shifter_rule -name lsr_vdda_vddb -from pd_vdda_vss -to pd_vddb_vss -location to

Create the isolation rule for the signal passing from power domain pd_vddb_vss to pd_vdda_vss which provides logic low to signal the signal when pd_vdda_vss is switched off.control signal of the isolation cells is iso_enable.synthesized tool will insert the isolation cells according to this rule.isolation with location 'to' is used.

create_isolation_rule -name isor_vdda_vddb -isolation_condition iso_enable -from pd_vdda_vss -to pd_vddb_vss -isolation_output low -location to

Create the State Retention Rule for power domain pd_vdda_vss so all the sequential logic will be replaced by the state retention logic.

create_state_retention_rule -name srr_vdda_vddb -domain pd_vdda_vss -restore_edge sr_enable -save_edge !sr_enable

create rule for power switches which are to be placed in domain pd_vdda_vss for switch on and off capability.

create_power_switch_rule -name SW1 -domain pd_vdda_vss -external_power_net vdda_on

Steady state of the design is called the power mode.power mode gives information about power domain and which nominal condition they are operating.two power modes are created with our design.

create_power_mode -name mode0 -default -domain_conditions pd_vddb_vss@nc_1p3 pd_vdda_vssnc_1p1 create_power_mode -name mode1 -domain_conditions pd_vddb_vssnc_1p3 pd_vdda_vssnc_0p0

Design Constraints are associated with power mode.

update_power_mode -name mode0 -sdc_files cf5x_core_encounter.sdc update_power_mode -name mode1 -sdc_files cf5x_core_encounter_ps.sdc

To check if the design functions correctly when slightly different operating conditions

multi-corner timing analysis is done for the worst and best case corner.

the timing and power characteristics of the cells depend on the operating conditions.Four Operating Corners are created.

create_operating_corner -name BC_PD1 -process 1 -temperature -40 -voltage 1.3 -library_set bc1

create_operating_corner -name WC_PD1 -process 1 -temperature 125 -voltage 1.2 -library_set wc1

create_operating_corner -name BC_PD2 -process 1 -temperature -40 -voltage 1.1

-library_set bc2

create_operating_corner -name WC_PD2 -process 1 -temperature 125 -voltage 0.99 -library_set wc2

Operating Corner	Operating Condition			Libraries
	process	temperature	voltage	
BC_pd_vdda_vss	1	0	1.1	Lib3_BC
WC_pd_vdda_vss	1	125	0.9	Lib3_WC
BC_pd_vdda_vss	1	0	1.3	Lib1_BC,Lib2_BC
WC_pd_vddb_vss	1	125	1.1	Lib1_WC,Lib2_WC

BC-Best Case WC–Worst Case

Figure 2.4: Library Table

When analyzing a view of the design, you need to indicate the corners at which the power domains are operating two views best case and worst case can be considered The design must function correctly in both.

create_analysis_view -name AV_BC -mode mode0 -domain_corners pd_vddb_vss-BC_PD1 pd_vdda_vss-BC_PD2

create_analysis_view -name AV_WC -mode mode0 -domain_corners pd_vddb_vss-WC_PD1 pd_vdda_vss-WC_PD2

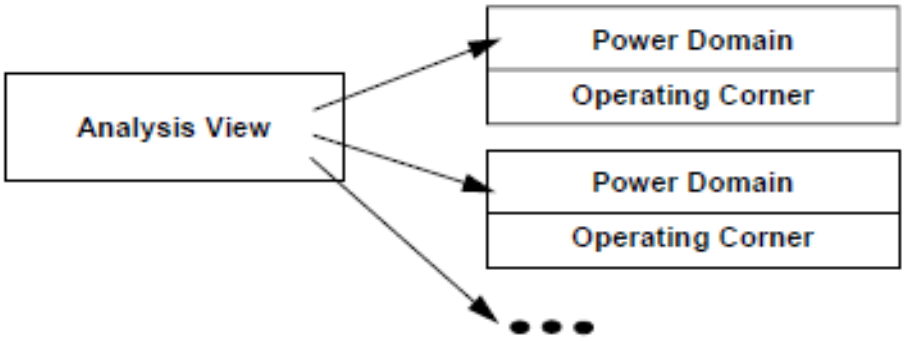

Figure 2.5: Analysis View

Chapter 3

Logic Synthesis

Logic Synthesis is the process by which an abstract form of desired circuit behaviour, typically Register Transfer Level(RTL), is turned into design implementation in terms of logic gates.

3.1 Synthesis Requirements

1.Technology Library Files (.lib)

Synthesis tools require liberty files, which contains information about delays, leakage, switching power of different types of standard cells. Delays are modelled in look up table form with two parameters of slew and load. Depending on different values of input transition (slew) vs output load (arises due to load of net plus capacitance at input pin), delays are calculated. If value of input transition or output load falls outside the range of LUT's values, tools extrapolate in order to determine the delay for certain standard cell.

2.HDL files (.v)

These type of files contain RTL design files, which are written in Verilog HDL or VHDL hardware descriptive languages.

3.Library Exchange Format Files (.lef)

These files contains physical information of various standard cells. It contains

cell name, various pins associated with the cells such as input, output and inout pins. It also has describes different metal layers and geometrical coordinates as well as shape of standard cells.

4.Captble file (.ctl)

These files contains information about net resistance and parasitic capacitance of each node, it contains interconnect RC values.

5.Constraints (.sdc)

It contains max and min delay information for output and input pins, max transition for clock pins. It contains phase shift details of all the clocks of designs.

Synthesis Modes

There are two modes of synthesis :- wireload model synthesis and physical layout estimator synthesis (PLE).

In wireload model synthesis, CAD tools have internal default wireloads and capacitance information. These tools infers this information to model wireload, which leads to incorrect results, when actual connectivity is established during back-end flow of design.

In physical layout estimator mode, CAD tools fetch .lef and .captble files as input. These files model actual values of load (R) and internal and parasitic capacitance, hence establish perfect physical connectivity during routing stage in backend flow.

3.2 Logic Synthesis Flow

Logic synthesis flow is given in a figure below :-

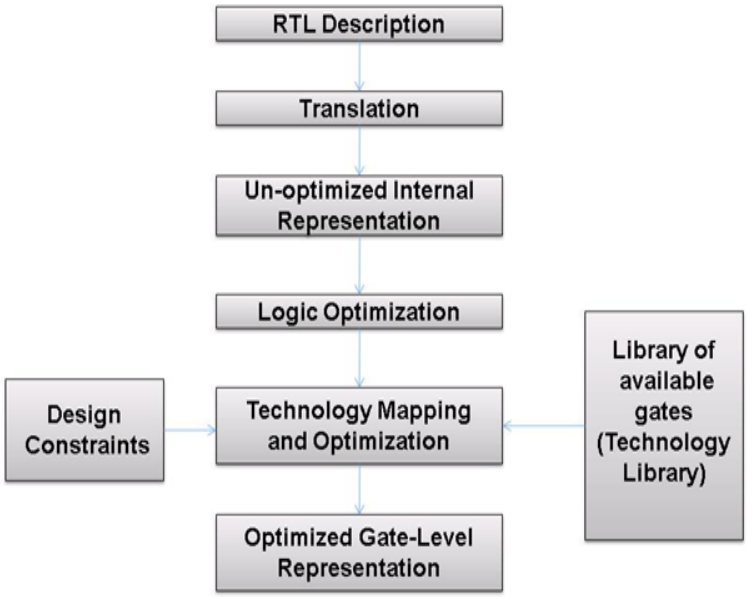

Figure 3.1: placement-Physical cell

3.2.1 Elaboration

(i) Elaboration:

Before Elaboration the design should be read in along with other inputs. Make sure that general compiler settings are done.

Elaboration is required on the top level design which automatically elaborates all its references. The *elaborate* command does following things:

- Builds Data structures and infers registers in the design.
- Performs High level HDL optimization, such as dead code removal.
- Identifies Clock Gating and operand isolation candidates.
- During elaboration, the tool reports the following :
- Unresolved references, that is the instances found with no corresponding module or library cell.
- Semantic problems, including unused ports, inconsistent resets.etc

- *check_design* command can be used to report all these things in detail.

3.2.2 Technology Independent Synthesis (Generic)

- A technology independent synthesis is done during this stage. Following are some of the optimizations done in this stage:

- Carry save arithmetic optimization

- Logic Pruning

- Resource Sharing

- Implementation Selection

- Redundancy Removal

- Mux Optimizations

- Common Sub-expression sharing

3.2.3 Technology mapping optimization

- The global mapping process in RC involves:

- - Structuring

- - Mapping

- - Target setting

- - Restructuring/Mapping as per target settings

3.2.4 Incremental Optiization

- Incremental Mapping Performs:
- - Critical Range Re-synthesis
- - DRC fixing
- - Buffering
- - Area Recovery

3.3 Optimization

Optimization is needed to satisfy the constrains applied to the design such as the area,timing and power optimization.variouse techniques for optimization are as below.

3.3.0.1 Area-based Optimization

(i) Analyze the timing reports prior to synthesis, i.e., at pre-synthesis diagnosis stage. If the I2C, I2O paths are already meeting timing, then area can be recovered from these paths. For this, under constrain these path groups before the global mapping stage:
Ex: The following command will relax the timing on I2O paths by 200 ps. So, now the effective clock-period for these paths are [Clock_period + 200].
path_adjust name UNDER_CON delay 200 from [all::all_inps] to [all::all_outs]
After global mapping phase, remove the path-adjust, so that timing reports would be accurate and with respect to the actual clock period.
(ii) Synthesize -to_generic effort medium
As per LEC requirement,it is preferred to use medium effort for synthesize to_generic. In cases where timing is critical ,the effort level can be set to high if and only if LEC is not an issue.Here ,RC uses more aggressive csa optimization algorithms for

better timing. This might result in larger area as well.

(iii) Set the attribute drc_first to false, in case if you have set it to true (default is false). This has to be set to false, so that RC does not fix DRCs aggressively & bump up the area.

(iv) Review and remove any preserve attributes (dont_touch) that are not needed. This will help RC to optimize the results better.

(v) Check if there are any datapath elements in the critical paths with preserve attributes.

(vi) Set the attribute dp_postmap_downsize to true on data path elements present in the design before incremental synthesis. This will perform architecture downsizing after mapping. This attribute is effective only in incremental optimization.

(vii) If timing is not critical in the design, you do not need to turn on the attribute tns_opto.

(viii) Area multiplier: The area of a cell can be modified by using the attribute area_multiplier. The default value for this attribute is 1.0. When set to a value less than default value can favor a cell to be picked up by the tool. For example, when applied on complex cells, it can bias the tool for mapping them on non-critical paths. To do so, one could set the area_multiplier value to less than one 1.0 for these kind of cells. However, for accurate area reporting, you should change the multiplier to 1.0 before issue any cell or area reporting command.

(ix) Check if proper complex libcells are selected when needed. Otherwise one can bias the tool to pick them up using area multiplier trick discussed above.

(x) If you have a max_delay constraint set on your design, RTL Compiler interprets the constraint more restrictively and may produce larger area and instance count. To improve area, check for these and convert them in to set_input_delay set_output_delay constraints that refererence to the appropriate clock.

3.3.0.2 Timing-based Optimization

Following are the important steps in addition to those mentioned in the flow, which will improve the performance:

(i) Do a detail analysis of timing reports in RC just prior to synthesis, i.e., just after loading the RTL into RC (Pre-synthesis Diagnosis). If there is some MACRO, which is in the critical path of reg-to-reg path groups and which has huge delay, then make a separate path group for all the paths passing through that MACRO. This will greatly improve the optimizations of all the other reg-to-reg paths not passing through that MACRO.This MACRO path can also be optimized in a better way by over constraining it.

(ii) At the beginning of global mapping, based on the libraries, logic structure and constraints RC will estimate a target slack for all cost groups. RC tries to meet these numbers. In the logfile look for the word target slack. If there is a cost group with large negative slack normally is a problem area. If the constraints are clean, one could set the initial target to 0 or some positive value using the initial_target attribute. This forces the mapper to do aggressive structuring and optimization on them to meet the target set.

set_attribute initial_target 0 [find / -cost_group name_cg]

Note : This may increase area.

(iii) Over constrain the paths, which are not meeting timing.

Ex: The following command will over constrain the reg-to-reg paths by 200 ps. So, the effective clock-period for these paths is now [Clock_period 200 ps]. Hence, the tool will try to meet its target in this effective clock-period.

path_adjust name OVER_CON delay -200 from [all::all_seqs] to [all::all_seqs]

This command is to be used before global mapping stage. After global mapping, remove the over constraint and do report timing, so that the timing reports are with respect to the actual clock-period.

(iv) If timing degradation is due to high fanout nets in the design, idealizing those

nets improves timing at the synthesis stage. In the later stages a place and route tool can build a better buffer tree for these nets. The script mentioned in the Tcl procedures section can be used to find out the high fanout nets and idealizing them.
(v) Usage of Low Vth cells significantly improves timing. But usage of these cells significantly increases the power numbers as these have high leakage power.
(vi) Set the attribute tns_opto to true. When set, it forces the tool to consider all the endpoints for the optimization.
Note: This may increase the area.
(vii) Make sure that the attribute drc_first is not set to true. By default it is set to false. If set to true the tool will give higher priority to design rule constraints than the timing constraints.
(viii) Set the attribute iopt_ultra_optimization to true .When set, it enables ultra optimization in incremental optimization to achieve best QOR with higher runtime.

3.4 Special Care abouts and Recommendations

1. Boundary Optimization

By default, RTL Compiler performs boundary optimization during synthesis for all subdesigns in the design. It controls boundary optimization on the subdesign and hierarchical pin inversion. To preserve the input and output pins of a subdesign, you can turn off the boundary optimization. In this case, no hierarchical pin inversion will be done either for this subdesign.
Note: To exclude individual pins from boundary optimization, use the preserve attribute
The boundary optimization can be turned off as below :
set_attr boundary_opto false [find / -subdesign ⟨ subdesign_name ⟩]

2. Retiming Improves the performance of the design by either optimizing

the area or the clock period (timing) of the design. Retiming moves the registers across the combinational logic to improve the performance without changing the input/output behavior of the circuit .Generally it is not recommended retiming on the whole design due to multiple reasons :

- You will lose traceability of the flops since all the retimed flops will become retime*reg. Verification could be a major issue for both formal and simulation. The retiming space explodes to the complete design and flops can get spread out e from one module to one or more others.
- There will be runtime increase since the Retiming flow would require an initial mapping during prepare phase, retiming step and a final mapping on the whole design.
- It is recommended to use Retiming only on selected subdesigns. The ideal subdesign would those which have been designed for pipelining or if not choose that would benefit from register spreading based on unbalanced paths paths which have positive slack on one side and negative slack on the other side. So moving the registers in the direction of positive slack would reduce the negative slack.
- On a Post-mapped design, you could try incremental retiming which makes use of the unbalanced paths. But on typical designs, if the negative slack cannot be improved by incremental retiming, then success would be limited.

3. Ungrouping

RC by default ungroups the user created hierarchies during synthesis to improve area and timing during synthesis. This can be controlled by setting the attribute auto_ungroup to both/none and must be specified before synthesis.

none Ungrouping will not be performed.

Both Ungrouping will be performed with an emphasis on both optimizing timing

and area.

You can also set the ungroup_ok attribute to false to control any subdesign/hierarchy not to ungrouped. This way, while achieving better QoR, you can also keep the hierarchy of interest intact.

Note: Ungrouping of user hierarchies happens during

- RTL optimization synthesize -to_gen -effort high
- Technology mapping synthesize -to_map -effort high

4. Information to be looked in log files

(i) For unresolved instances during elaboration. The command for checking unresolved instances is check_design unresolved

(ii) check if any sdc errors exist and verify the SDC summary report in the log file.

(iii) Warnings pertaining to sequential logic deletion.

(iv) Clock gating fanout statistics.

(v) At the beginning of the global mapping step, RC will estimate a target slack for each cost group. This estimated target is based on the libraries, the logic structure, and the constraints. RC will work toward this target number during the optimization process. In the logfile, search for the keyword target slack, as it will be printed before and after the global mapping step for each cost group in the design. A cost group with large negative target slack would normally indicate a problem area. Also check whether these targets are met after mapping or not.

(vi) On some designs RC might spend time in incremental optimization. To debug such cases one should have set the debug variable iopt_stats to 1 and the attribute information_level to 9.

(vii) Before attempting to run synthesis, the user should check the input data, pay attention to the warning messages and correct any obvious issues.

3.5 Design For Test

Scan chain is inserted for testing.sequential cells present in the design made the chain of sequential logic when test enable is on.

Two types of scan chain method is available.

1.Muxed Style

2.LSSD Style

sequential logic with internal multiplexer is required for scan chain so it can select system input and scan_in input based on the signal scan_enable Helps achieve

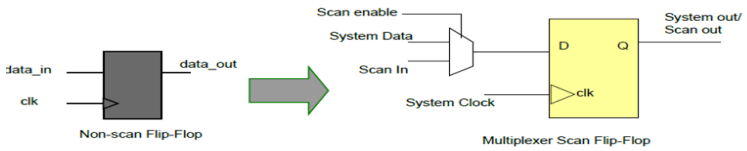

Figure 3.2: Sequential Logic

controllability and observability.The muxed scan style is the most commonly used in scan chain. Circuit after scan chain insertion is shown in the below figure. Two

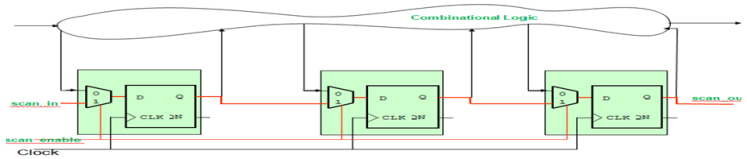

Figure 3.3: Scan Chain

Modes of operation performed during Testing.

1.Serial Shift In Mode

2.Serial Shift Out Mode

Scan chain is formed in the serial shift in mode and sequential logic selects the scan_in input so all the flip flops will be in the known state.

Circuit is bring back to normal functional operation,After that again scan_enable is on and data is shifted out and compared with expected data.

Scan in and scan enable,scan output port are added in the design.Two scan chains are created one for each domain.

1.DFT_sdi_0

2.DFT_sdi_1

3.DFT_sdo_0

4.DFT_sdi_1

5.test_enable

6.test_mode

Internally generated clock gives DFT violations because that clock become uncontrollable.Internally generated clock is due to the clock gating cells

test_mode signal is defined to control the test clock. For circuit like given below.

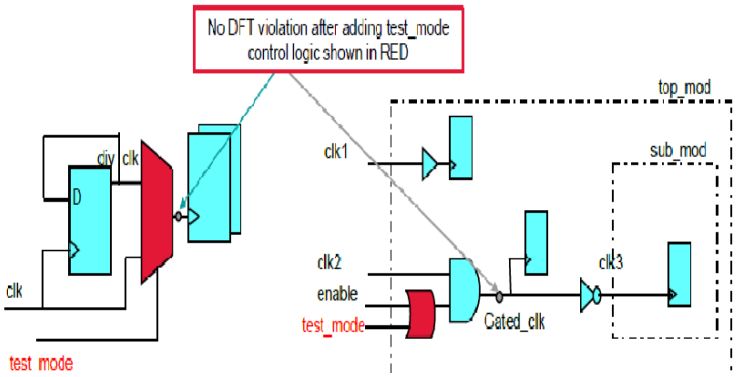

Figure 3.4: DFT Violation

3.6 Implementation Flow

synthesis is done using the cpf enabled flow.Lef file and cap tables are also provided for PLE interconnect mode which gives the better timing result. CPF is used to capture the low power Intent of the design.

level shifter and isolation and state retention cells are defined in the technology CPF.

Synthesis tool insert the level shifter,isolation and state retention cells using the information specified in the CPF. synthesis implementation flow is shown in the figure.

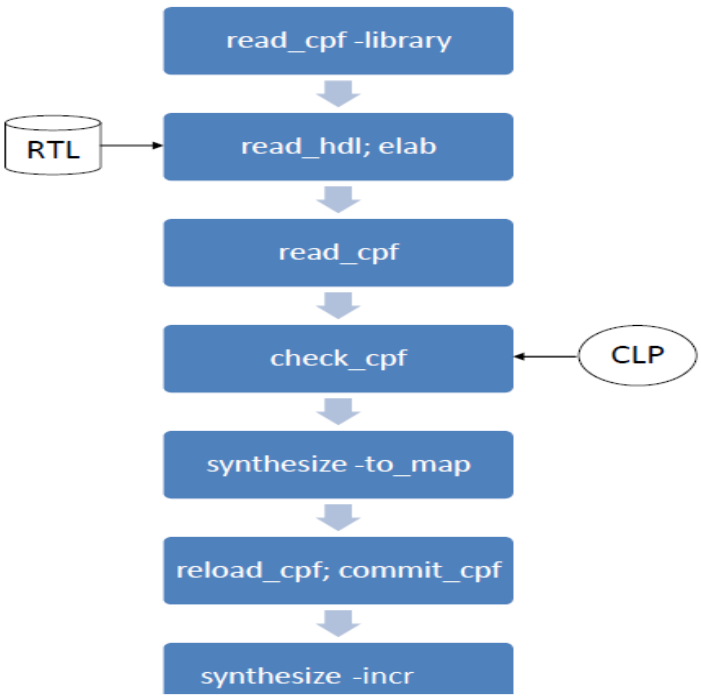

Figure 3.5: Synthesis flow

1.Libraries specified in the CPF file is loaded in the tool.

2.Design(cf5x_core) RTL is loaded in the tool.

3.Enable Clock gating cell insertion

4.Elaboration

5.check for any unresolved part of the design

6.CPF is loaded in the tool

7.CPF is checked for any errors using the CLP tool interface.

8.specify DFT style

9.Generic Synthesis and Technology Mapping are performed at this stage.

10.commit cpf-level shifter,isolation and state retention cells are added at this stage.

11.Check for any DFT violation and solve it.

12.auto connect scan chain.

13.increametal synthesis for timing optimization.

Block Diagram of the synthesized design is shown in the figure.

Synthesized design is shown in the figure.

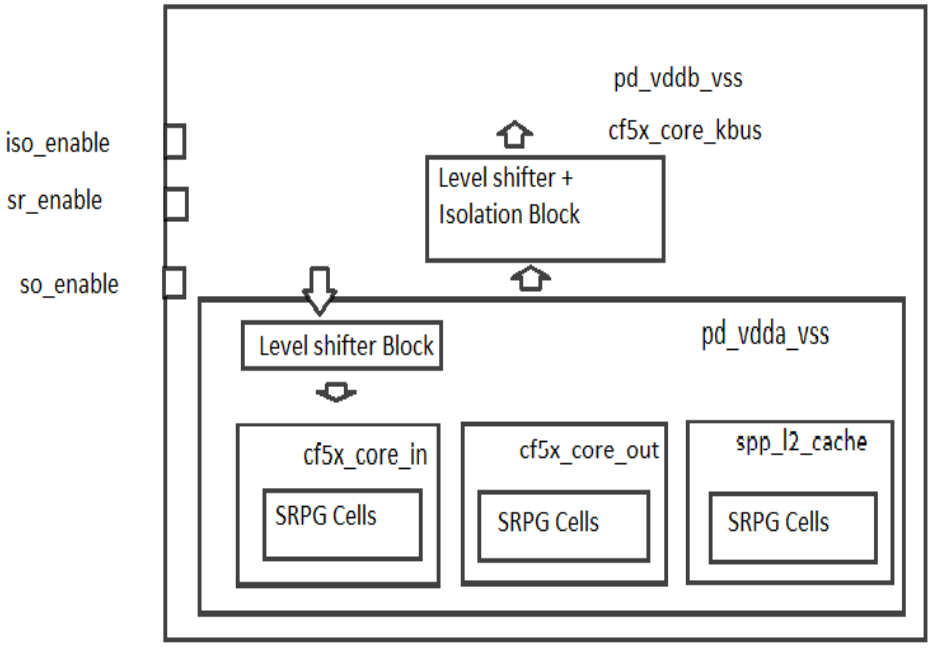

Figure 3.6: Synthesized Design Block Diagram

Chapter 4

Physical Implementation

4.1 Power Planning

Power Planning is used to connect the power pins of standard cell or macro to the power net present in the design.

Power Planning is done By creating power ring and stripes.

There are two options for creating the power ring. 1.Around the core 2.Around the Die

Power ring is then used for connection with power rail.

Power stripe is used to connect the power pins of standard cell or macros which power pin is not aligned to power rail.

4.1.1 Power Ring

The power ring has been created to all around the core to get the proper connection from the power pads. These rings can be specifying as planar rings, intermingled rings, overlapping rings, etc.

It was observed that the I/O core power supply cells have pins on higher metals (metal6 and metal7). In order to make good connections to the rings, same metal layers (metal6 and metal7) are used for power routing.

In overlapping rings scenario where a pair of dual power rings between I/Os and the core is created. The power rings are created using metal7 through to metal4. The metal7 is on top of the metal5 and they are used for the horizontal edges, whilst the metal6 is on top of the metal4 and they are used for the vertical edges.

The main reason behind having dual rings, i.e. two metal layers on each side of the core region, is to have stripes connected in those places where I/O core supply cells connect to the ring. If there was only a single ring, the I/O connections could block the stripes, which connect to the ring.

By using the dual power rings, the width of the rings can be reduced, hence saving overall chip area. For other type of rings please see the solution linked.

Five Power Rings are created.

1.pwer rings of vddb and vss are created around core with distance from core is 2.5 um and each other net is 2.5 um.Metal 2 is used for top and bottom part ,metal3 is used for right and left part.power ring width is 4 um.

2.power ring of vddb is created around power domain pd_vdda_vss with distance from power domain 0.6 um and width 1 um.

3 . power ring of vdda_on is created around power domain pd_vdda_vss with distance from power domain 0.5um and width 1 um.

4.power ring of vdda is created inside created around power domain pd_vdda_vss with distance from power domain 0.5 um and width 1 um.

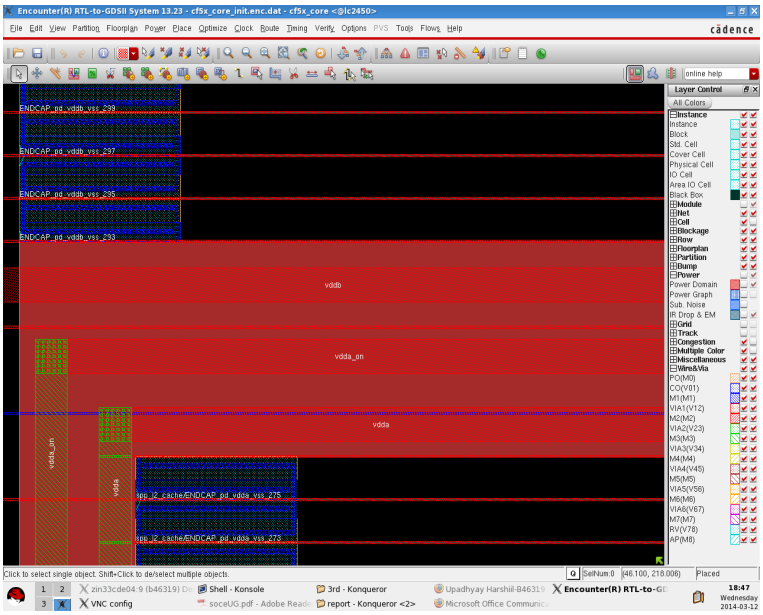

Figure 4.1: Power Ring

These power nets are connected by rails using special routing.In alternate rows, vdd and vss are routed, which facilitates standard cell placement during placement stage.

4.1.2 Power Stripes

Power mesh is created to supply the uniform power into the core. The density of the mesh can be determined using the core power consumption of the chip. While calculating the total power consumption of the core few assumptions has been taken.

4.1.3 Power Net Connect

In a design global power and ground nets must be connected throughout the design in order to perform optimization and complete planning. . This adds a new global or special net connection to the specific global net. This should be run after design import and prior to power planning. It establishes the logical power connectivity.

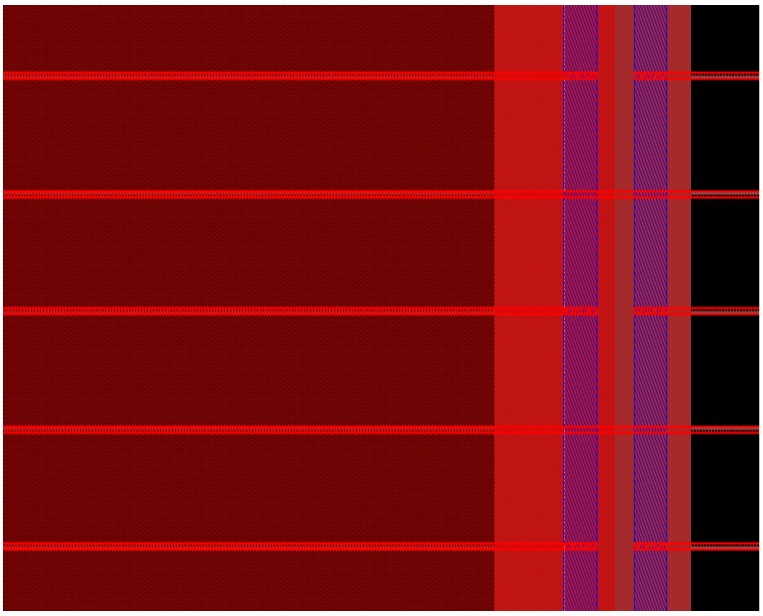

Figure 4.2: Power rail

Connect pins in a single instance to a global net. Connect multiple inst pins to a global net Connect one or more existing nets to the global net

secondary power pins of level shifters placed in pd_vddb_vss is connected with logically connected with power net vdda_on.

secondary power pins of state retention cells present in the power domain pd_vdda_vss is logically connected with power net vdda_on.

power pin VDCC of power swith cell is connected to power net vdda ,pin VDDP is cnnecte to power net vdda_on.

power rails of power nets vdda and vss are created in the power domain pd_vddb_vss and power rails of power net vdda and vss are created in power domain pd_vdda_vss.

4.1.4 problems

4.1.4.1 power-ground short

First tried to make power ring at M2 layer on right and left hand side and at M1 layer on top and bottom side of vdd and vss.

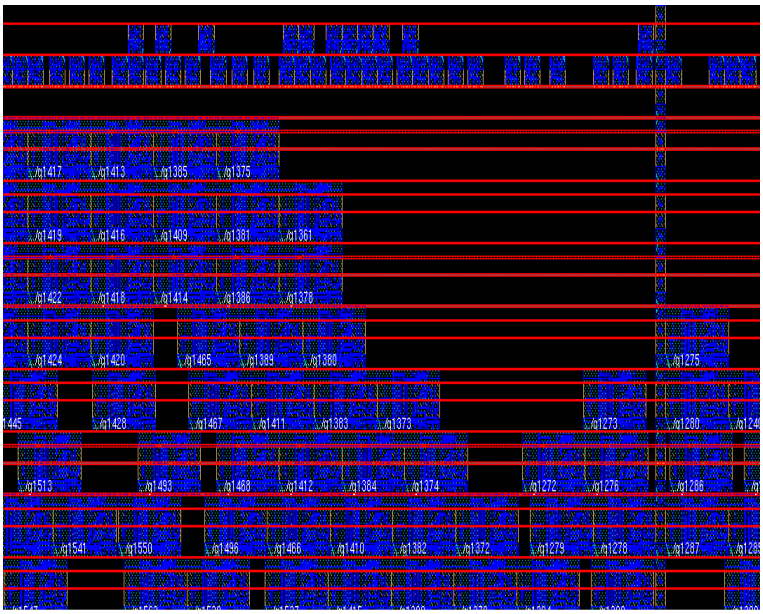

Figure 4.3: Power stripe

Made the power rail at M1 layer.but its giving number of shorts between power and ground. after analyzing the internal structure of the double height standard cell,it is found that level Shifter has internal connection of vss at M1 layer.

Due to which power and ground are shorted internally.

So to solve this problem power rail are made at M2 Layer.

4.1.4.2 tangling poawer net

A power net which is open and has no close connection are known as the the tangling net. so for default power domain power rail another power ring of vdd is made around the another power domain.

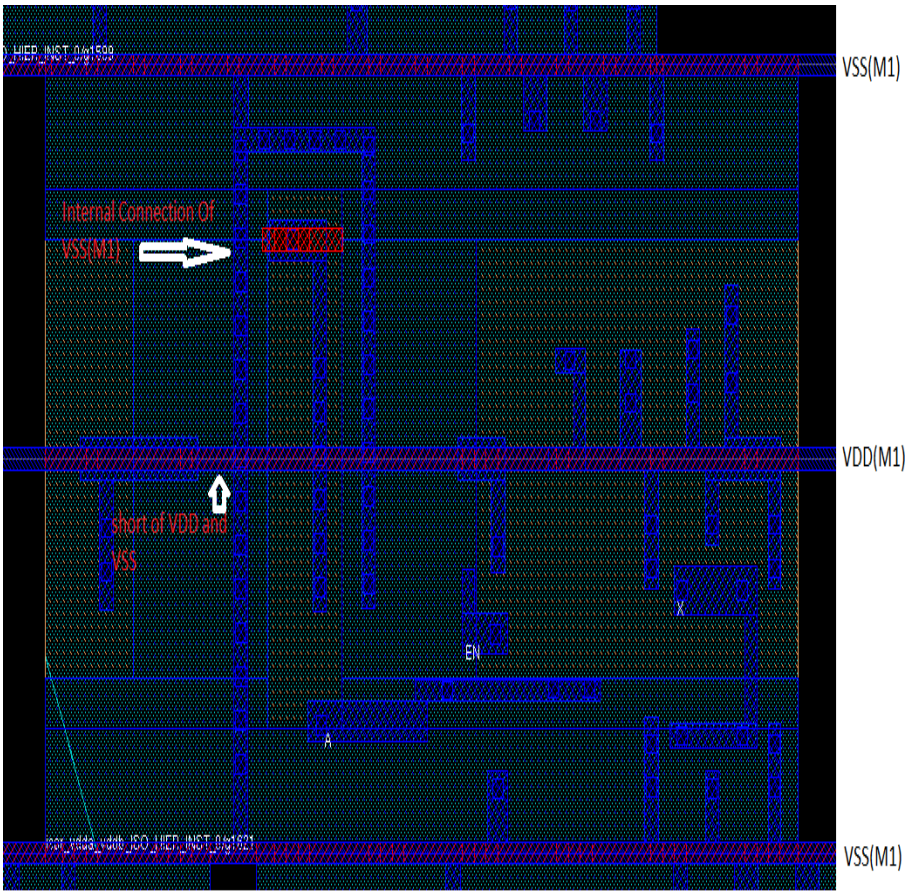

Figure 4.4: Power-ground short

4.2 Floorplan

4.2.1 Overview

Floorplanning a chip or block is an important task of physical design in which the location, size, and shape of soft modules, and the placement of hard macros are decided. Depending on the design style or purpose, floorplanning can also include row creation, I/O pad or pin placement, bump assignment (flip chip), bus planning, power planning, and more. For example, floorplanning is very important when preparing the design for timing closure and detailed routing. Floorplanning, in conjunction with placement and trial routing, can be an iterative design process.

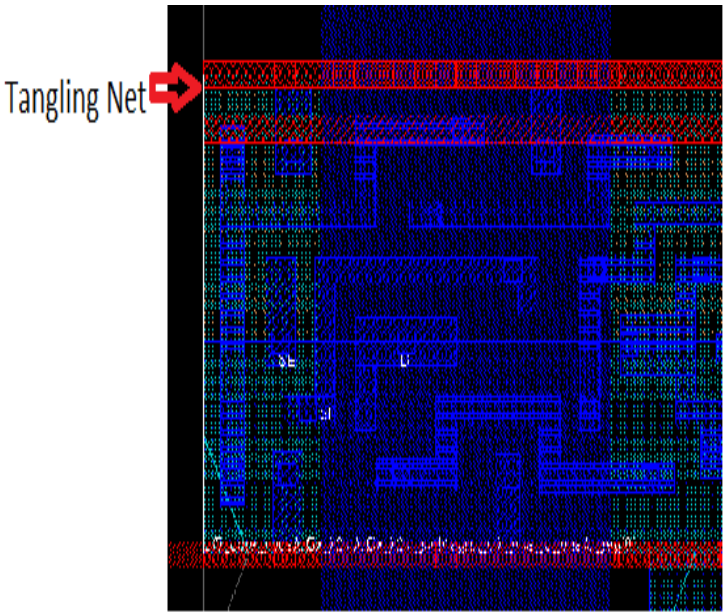

Figure 4.5: tangling power net

4.2.2 Common Floorplan Sequence

Floorplanning usually starts by preplacing blocks, modules, and submodules according to the prepared floorplan. All other modules or blocks not in the prepared floorplan are left outside the chip area. The following steps describe the most common sequence for floorplanning:

1. Importing the design.

2. Studying the design's connectivity.

3. Performing the minimum amount of floorplanning based on the chip design floorplan, or do no floorplanning at all.

4. In some cases, no floorplanning is required. For example, a front-end designer

might want to predict the quality of the design's netlist by initially placing the entire design without any floorplanning. This iteration provides a good indication of how the blocks should be located and arranged together with the larger modules. After a few iterations, it should be clear how to position the blocks and modules in the floorplan.

5. Running placement and Trial Route to view placement and routing congestion. Optionally, running resize floorplan to enlarge or shrink the die after placement and routing.

6. In this case, floorplanning is done to detail the pre-placement of all blocks, most likely done by a back-end designer to gauge the feasibility of a prepared floorplan.

7. The placer places all remaining blocks that were not preplaced in the floorplan, and alsorecognizes the floorplan object, such as power and ground routes.

8. If you are at the design's top-level in the display area and want to generate a guide for asubmodule, ungroup the top module until you have reached the submodule.

9. Using the full chip placement to refine block (hard macro and blackbox) locations. (Optional) Based on the full chip placement results-placement density and routing congestion, running resize floorplan to enlarge or shrink the die.

10. View the placements of blocks to determine if you need to change the alignment ororientations.

11. Looking for congestion in modules and change heavily congested modules' placementdensity to a lower percentage (using the createDensityArea text command).

12. (Optional) If you made any changes in step 5, or especially step 6, rerun placement.

4.3 Viewing the Floorplan

In the design display area, the objects to the left of the core area are the top-level modules,which can be moved and reshaped. The objects to the right of the chip area are the blocks,which can be moved but not reshaped.

In Floorplan view, you can view the block pins and connection flight lines by clicking on a block or module. Flight lines show the connections and number of connections between the selected module or block to any other modules and blocks. Figure below displays the connections between modules and macros.

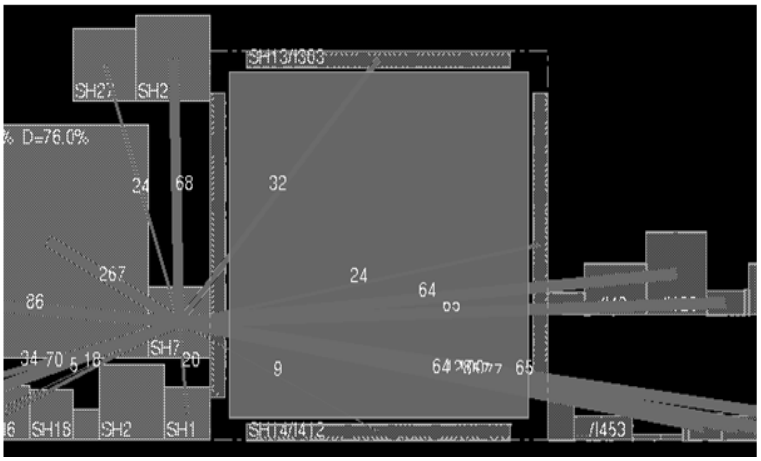

Figure 4.6: floorplan view

4.4 Module Constraint Types

The entire design size is initially calculated during design import, and each module size is calculated. The size of the modules are determined by either the core utilization or the core width and height specifications. The imported design modules can have one of the following constraint types:

1. None:- The module is not pre-placed in the core design area. The contents of the module are placed without any constraints.

2. Guide:- The module is preplaced in the core design area. A module guide represents the logical module structure of the netlist. The purpose of a module guide is to guide placement to place the cells of the module in the vicinity of the guide's location. After the design is imported, but before floorplanning, you can locate module guides on the left side of the core area, which appear as pink objects (by default) in the Floorplan view.

When a module is preplaced in the core design area, it snaps to a standard cell row in the vertical direction and to a metal 2 pitch in the horizontal direction (the default). This default can be changed to snap to the manufacture grid.

3. Fence:- The module is a hard constraint in the core design area.After specifying a hierarchical instance as a partition, the constraint type status of amodule guide is automatically changed to a fence.

The physical outline of a fence module is rigid, and the design for the module is self contained within the rigid outline. Only child instances must be contained within the partition physical outline; non-child blocks or modules that do not belong to the partition are excluded, and should not be pre-placed within another partition. This restriction is a hard restriction for third party back-end tools where the placement file for a partition does not match the partition netlist.

Module constraints display a target utilization (TU=%) value to represent their physical design size. This is an estimation of module utilization for the given size of the module where only standard cell and hard macro areas are considered; floorplan constraints, such as placement blockages, are not considered. This value is calculated by the standard cells area plus the hard macros area, divided by the module area. The initial TU values are calculated during design import.

4.4.1 Implementation

Floorplanning implementation steps are as belowe.

1.specify the floorplan with the utilization factor 0.6 and height to width ratio 1

2.specify core area from die boundary 40 um all the side.

3.space between core and die boundary is used for power ring ,in/out pad placement and pin placement.

4.Move the Power Domain into the core area.

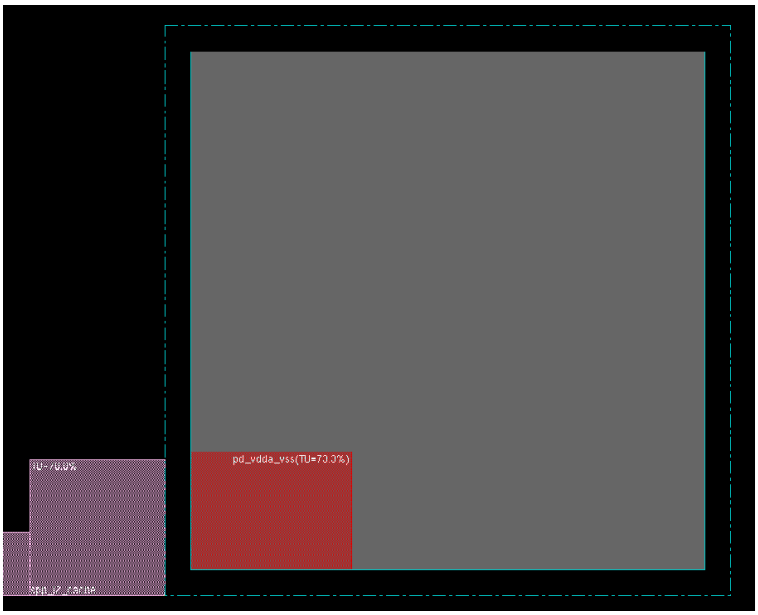

Figure 4.7: Initial floorplan

5.Create the fence around the both side of the power domain. Fence include the all level shifter cell.level shifter cell has two power supplies. Advantage of creating fence is that the number of the power stripe is limited to fence area.stripe has not to be made over entire core area.

6.Placement blockage is created around the power domain to remove minspacing and placement of cell between the two power rings.

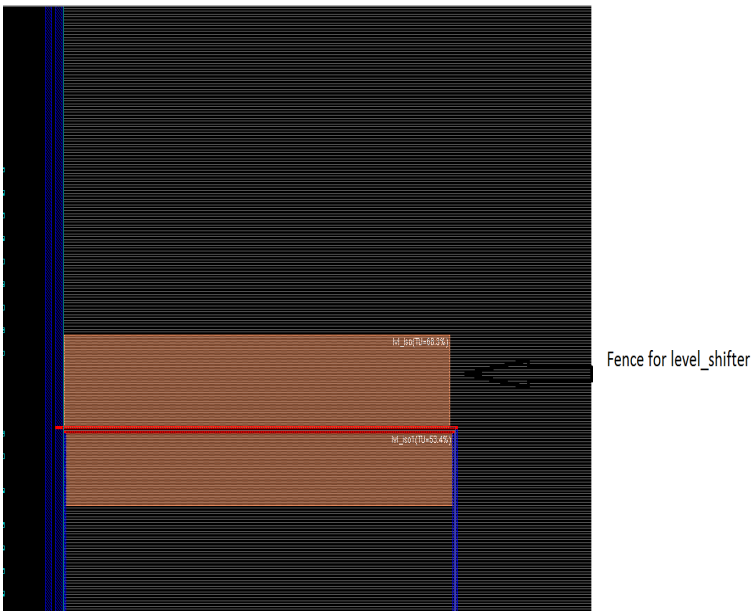

Figure 4.8: floorplan-fence

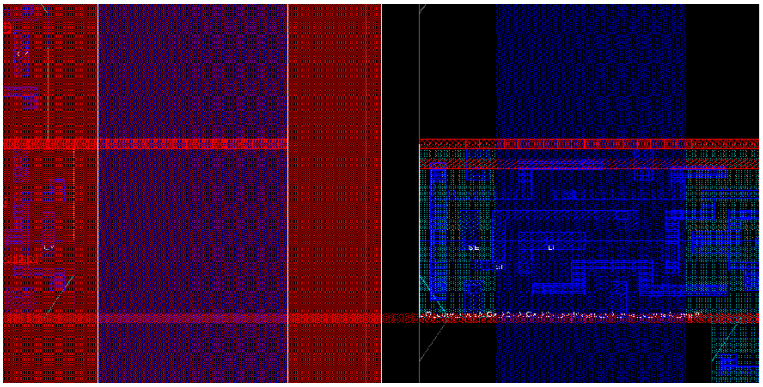

Figure 4.9: floorplan-minspacing problem

minspacing problem is solved using the placement blockage.

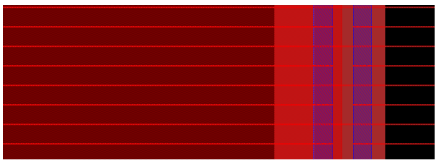

Figure 4.10: floorplan-placement_blockage

4.5 placement

After floorplanning, place the cells in the design. Placement considers the modules that were placed during floorplanning and takes into account the hierarchy and connectivity of the design. It honors floorplanning constraints, including guides, regions, and fences.

Placement treats preroutes the same way it treats routing blockages: It places standard cell instances at legal locations where there should not be any DRC violations against preroutes or routing blockages.

Typically, you use preroutes for special nets that are floorplanned (predesigned) before placement,such as power, ground, and clock mesh nets, where you do not want any standard cells placed underneath. Placement Objective are:

Total wirelength: Minimizing the total wirelength is the primary objective. This not only helps minimize chip size, and hence cost, but also minimizes delay, which are proportional to the wirelength.

Timing: The clock cycle of a chip is determined by the delay of its critical path. Given a performance specification, a placer must ensure that no path exists with delay exceeding the maximum specified delay.

Congestion: While it is necessary to minimize the total wirelength to meet the total routing resources, it is also necessary to meet the routing resources within

various local regions of the chips core area. A congested region might lead to excessive routing detours, or make it impossible to complete all routes.

4.5.1 Placement of the Physical cell

After power planning, physical cells such as welltap and endcaps cells are placed.

Welltap is used to tie bulk in pmos to vdd and bulk in nmos to vss, to maintain the biasing of device.so that they don't drift too much (especially towards the middle of the chip) and cause latchup.

Figure below shows the placement of welltap cells. End cap cells do not have

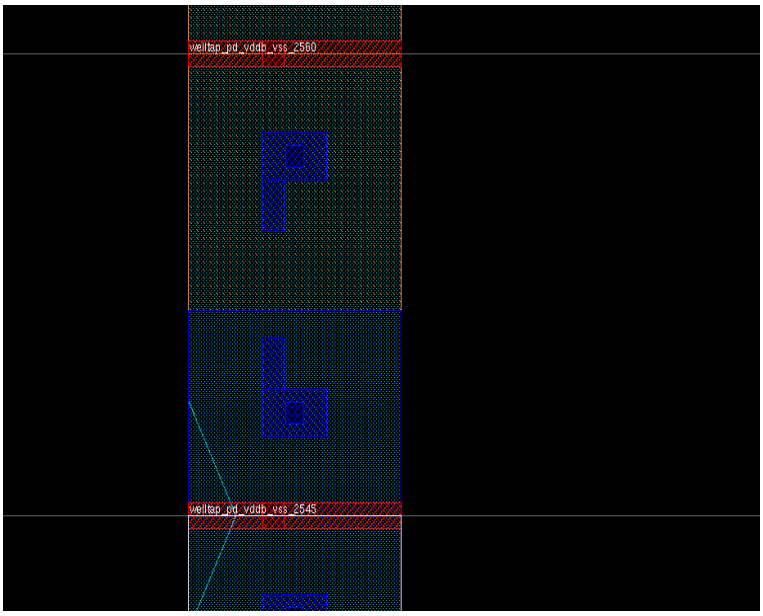

Figure 4.11: placement-Welltap cell

signal connectivity. They connect only to the power and ground rails once power rails are created in the design.

They also ensure that gaps do not occur between the well and implant layers.This prevents DRC violations by satisfying well tie-off requirements for the core rows.Each end of the core row, left and right, can have only one end cap cell specified. However, you can specify a list of different end caps for inserting horizontal end cap lines, which

terminate the top and bottom boundaries of objects such as macros.

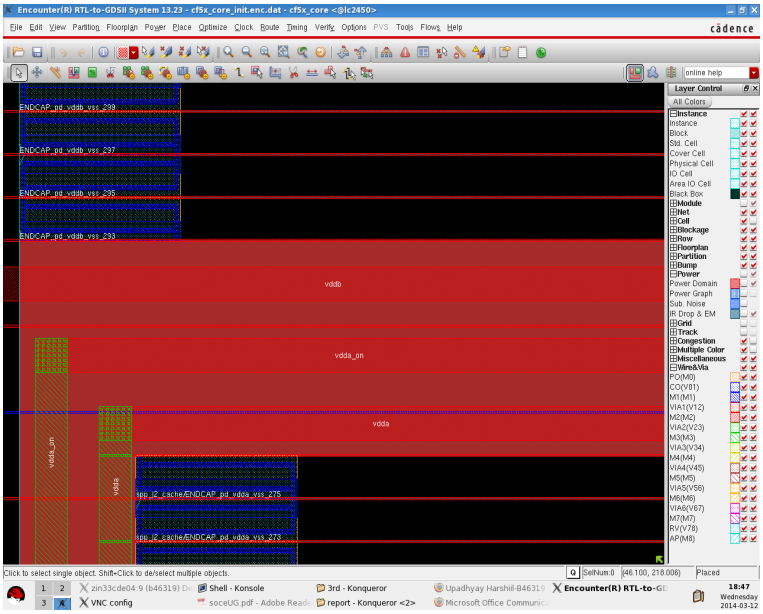

Figure 4.12: End cap cell

Power switches are added in the power domain pd_vddb_vss for switched off capability.

There are two styles of this methodology:

1.Ring style: All switches are inserted outside the domain.

2.Column style: All switches are inserted inside the power domain.

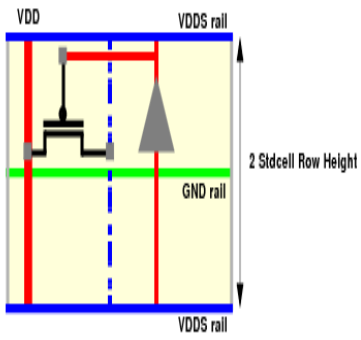

Figure 4.13: Column Style

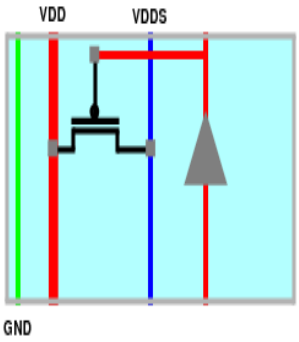

Figure 4.14: Ring Style

4.6 Clock Tree synthesis

Clock tree synthesis is Designed to meet clock timing constraints, such as clock skew, latency (insertion delay), and the transition time.

CTS is the process of insertion of buffers or inverters along the clock paths in order to achieve zero/minimum skew or balanced skew. The goal of CTS is to minimize skew and insertion delay. Buffer and Inverter Selection

Buffers and inverters to be added into the design during the clock tree synthesis must have equal raise and fall time delay.

If not, it will affect the time period/duty cycle of clock which in turn changes the operating frequency.

Input to Clock Tree Synthesis

1..Clock Tree Specification File

The first step is to create the clock tree constraints in a specification file. This file defines the minimum and maximum delay for the tree, the maximum skew, and other options to control how the tree is to be built.

Generate the clock tree specification file automatically from the SDC constraints by mapping following commands to clock tree specification file.

create_clock: AutoCTSRootPinin CTS

set_clock_transition: SinkLeafTran/BufMaxTran(Default: 400ps)

set_clock_latency: MaxDelay(Default: clock period) MinDelay(Default: 0)

set_clock_uncertainty: MaxSkew(Default: 300 ps)

level of created clock tree is 27 connected to 26701 sinks.

2017 Buffers are added to satisfy the clock tree specifications.

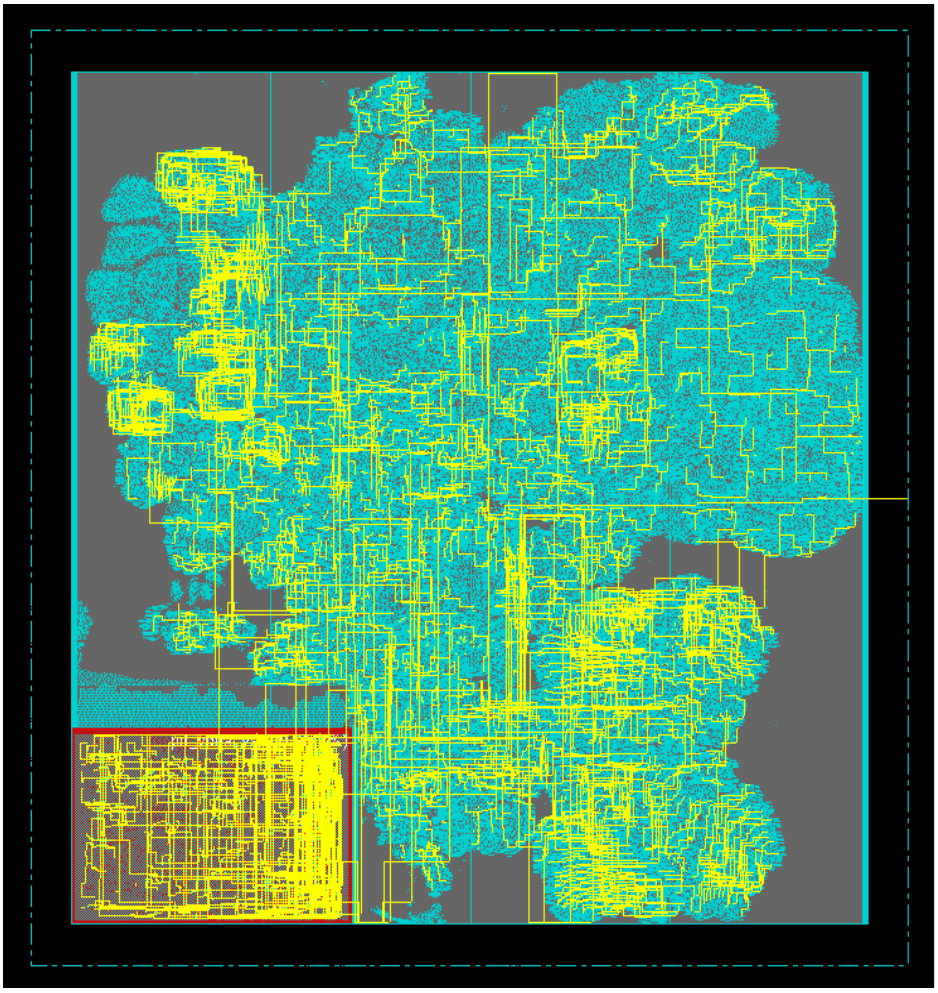

Figure 4.15: Clock Tree synthesis

4.6.1 CTS Power Reduction

In the clock tree synthesis Buffers or inverters are generally placed before the clock gating cells.

buffers and inverter consumes power even when clock gating cell is off and it increases the dynamic power consumption of the design because they are always on.

To reduce the power consumed by the buffer and inverter ,place them after the clock gating cell . It will decrease the dynamic power consumption of the design.

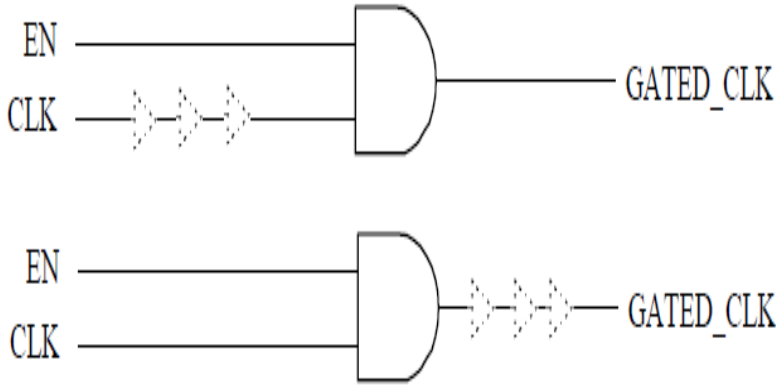

Figure 4.16: Improved Clock Tree synthesis

4.6.2 CTS violations

1. Max Transition Time Violation

Total Number of violations : 193

Reason : Transition issue caused by clock gating cells which have very high fan-out.

Solution: clock gating cell cloning it reduce the load on the single clock gate cell and improves its transition time.

increase in the clock gating cells : 53

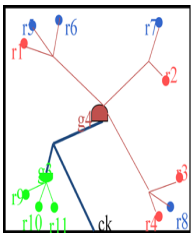

Figure 4.17: Before Clonning

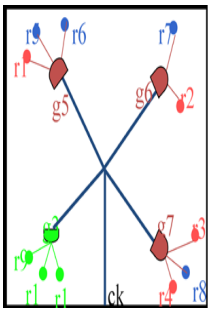

Figure 4.18: After Clonning

2. Clock skew Violations Total Number of violations : 465

Reason : non-uniform distribution of flops in the computing module which has very high number of flip flops.

Solution : flops that are driven by same clock are grouped together by placing them side-by-side in two columns using Structured Data path (SDP).

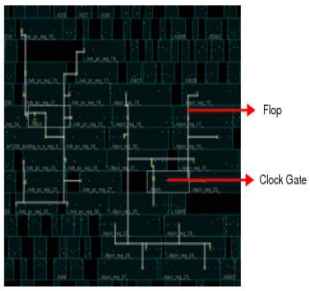

Figure 4.19: Structured Data Path

4.7 Routing

Router performs concurrent signal integrity, timing-driven, and manufacturing aware routing (SMART routing) of cell, block, or mixed cell and block level designs.

router is part of the block implementation and the top-level implementation stages of the EDI System flow. Run the router early in the design flow to identify and fix routability problems or avoid them altogether. You can run the router in non-timing-driven mode after the default parasitic extraction step to establish a baseline for future steps. If the design is congested or unroutable, stop and resolve problems before continuing.

Full routing consists of global and detailed routing.

1.Global Routing

2.Detail Routing

4.7.1 Global Routing

During this phase, the router breaks the routing portion of the design into rectangles called global routing cells (gcells) and assigns the signal nets to the gcells. The global router attempts to find the shortest path through the gcells, but does not make actual connections or assign nets to specific tracks within the gcells. It tries to avoid assigning more nets to a gcell than the tracks can accommodate. The detailed router uses the global routing paths as a routing plan.

The router can generate a map of the gcells, called a congestion map, that you can examine to see the approximate number of nets assigned to the gcells. The congestion map uses colors to indicate whether there are too few, too many, or the correct number of nets assigned to the gcells. If the router assigns too many nets to a gcell, it marks the gcell as over-congested. You can also read the Congestion Analysis Table in the EDI System log file to learn the distribution and severity of the congestion after global routing.

4.7.2 Detail Routing

During this phase, the NanoRoute router follows the global routing plan and lays down actual wires that connect the pins to their corresponding nets. The detailed router creates shorts or spacing violations rather than leave unconnected nets. You can run detailed routing on the entire design, a specified area of the design, or on selected nets. In addition, you can run incremental detailed routing on a database that has already been detail routed. The router runs search-and-repair routing during detailed routing. During search and repair, it locates shorts and spacing violations and reroutes the affected areas to eliminate as many of the violations as possible. The primary goal of detailed routing is to complete all of the required interconnect without leaving shorts or spacing violations.

During detailed routing, the router divides the chip into areas called switch boxes (SBoxes), which align with the gcell boundaries. The SBoxes can be expressed in terms of gcells; for example, a 5x5 SBox is an SBox that encompasses 25 gcells. The SBoxes overlap with each other and their size and amount of overlap might vary during search-and-repair iterations. The router also runs postroute optimization as part of detailed routing. During postroute optimization, it runs more rigorous search and repair steps. Detailed routing stops automatically if it cannot make further progress on routing the design. The routed data is saved as part of the EDI System database.

4.7.3 Check Design

Check design is used to ensure your design meeting the following conditions.

1.It is fully placed and the placement is legal, without any overlaps.

2.Fix Geometry(DRC) related problems.

3.Power is routed.

4.7.4 Check LEF File

You can avoid violations and save time if you ensure your LEF files are optimized for routing. Check the following statements and edit the values if necessary.

1.MINSIZE

The router does not support specifying MINSIZE without specifying AREA. MINSIZE allows a geometry that is smaller than AREA.

2.UNITS

The router does not support a value of 100 for DATABASE MICRONS in the UNITS statement. If the LEF file specifies DATABASE MICRONS 100, than issue it at the time of initialization.

3.MANUFACTURINGGRID

The router requires that you define the manufacturing grid.

4.MACRO

To improve pin access, ensure that all standard cell macros are defined as CLASS CORE. You must use real shapes, not block-style abstracts, for the shapes on the layers where you expect the router to connect to pins of standard cell macros.

5.VIA

The TOPOFSTACKONLY keyword is unnecessary if there are LEF LAYER AREA statements, because the router automatically derives TOPOFSTACKONLY vias based on the AREA statements. If a default via satisfies the AREA statement, the router tags it internally as a TOPOFSTACKONLY via. If there is no AREA statement for a routing layer, the router looks for TOPOFSTACKONLY vias that go up to the next metal layer. If TOPOFSTACKONLY vias exist, it derives the AREA rule from those vias- -the smallest area of the bottom layer metal of all such vias becomes the AREA rule. This feature provides backward compatibility with LEF files that do not have AREA rule support.

Design is routed with the Metal layers from M1 to M8 and routing is performed in the timing driven mode.

4.8 STA

Static timing analysis is a complete and exhaustive verification of all timing checks of a design. Other timing analysis methods such as simulation can only verify the portions of the design that get exercised by stimulus. Verification through timing simulation is only as exhaustive as the test vectors used. To simulate and verify all timing conditions of a design with 10-100 million gates is very slow and the timing cannot be verified completely. Thus, it is very difficult to do exhaustive verification through simulation.

Static timing analysis on the other hand provides a faster and simpler way of checking and analyzing all the timing paths in a design for any timing violations. Given the complexity of present day ASICs4, which may contain 10 to 100 million gates, the static timing analysis has become a necessity to exhaustively verify the timing of a design.

The design functionality and its performance can be limited by noise. The noise occurs due to crosstalk with other signals or due to noise on primary inputs or the power supply. The noise impact can limit the frequency of operation of the design and it can also cause functional failures.

Thus, a design implementation must be verified to be robust which means that it can withstand the noise without affecting the rated performance of the design. Verification based upon logic simulation cannot handle the effects of crosstalk, noise and on-chip variations.

STA is performed after every stage described in the Physical Implementation flow.

4.8.1 Timing Calculation

Timing is calculated for the four type of paths

1.input to register

2. register to register

3.register to output

4.input to output

Timing calculation of one such path is shown here

Start :

power_domain1/platform/spp2_faraday_plat/CORTEXM4INTEGRATION/ahb_lmem/ahb_ca
(clocked by bus_sync_div2_mode_clk trailing)

End :

power_domain1/platform/spp2_faraday_plat/CORTEXM4INTEGRATION/uCORTEXM4/u
(clocked by cm4_sync_div2_mode_clk leading)

Latency is the time required by clock to reach from source point to destination point .

launch clock bus_sync_div2_mode_clk coming from ccm (Hard Block) and has a latency of 2.300

capture clock cm4_sync_div2_mode_clk coming from ccm(Hard Block) and has a latency of 2.300

capture clock latency is also known as the other End Arrival Time.

Phase shift is clock period.

Clock Uncertainity is jitter in the clock.

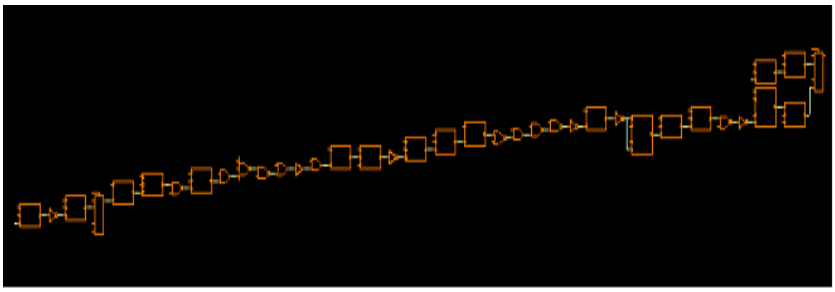

Figure 4.20: reg to reg path

Now Required time is calculated as

Required Time = Other End Arrival Time(Capture Clock Latency) (2.300) - Setup(0.146)+ Phase Shift (5.500)- Uncertainty (Of Capture Clock)(0.600) =7.054 Arrival time is calculated as

Arrival Time = Clock Fall Edge(2.750 + Clock Latency(LaunchClock)(2.300) + DataPath_Delay(2.083) = 7.133

Slack is calculated as

Slack =Required Time Arrival Time

= -0.07900 BeginPoint Arrival Time = Clock Fall Edge (2.750) + Clock Latency(2.300) = 5.050

Slack calculation can be shown graphically as below

Figure 4.21: slack-Graphic Representation

4.8.2 Timing Optimization

Every Path should have positive slack or zero slack for the proper operation. some timing optimization technique is given below.

1.Upsize the standard cell with the high fanout.

2.add Buffer to path to support higher fanout.

3.Down Size the standard cells connected to high fanout standard cell.

4.Replace the one cell with two or more standard cell to improve timing.

5.Swap the pins to reduce the path length.

6.create feed through path to reduce the length.

7.Clock gate clonning

8.Manually move Instants to reduce the path length.

9. Use the skew to meet the timing.

Chapter 5

Conclusion

To Test Low power cells,different low power methodology is tried but with MSV teachnique only level shifter cells are tested and with PSO technique only isolation cells and state retention cells are tested.To include all the low power cells in one design DVFS technique is used.

Challenge during implementation is to cover all the low power standard cells present in the DDK and to develop CPF for the same design.

Bibliography

[1] Cadence RTL Compiler user Guide.

[2] Cadence Encounter User Guide.

[3] Freescale internal reference.

[4] Advanced LP Flow Using RC

[5] LP implementation Using RC

[6] CPF Methodology

[7] Power Planning

[8] Static Timing Analysis for Nanometer Designs A Practical Approach by J. Bhasker and Rakesh Chadha

[9] http://www.si2.org