Low Power Verification using Power Management Cells.

Thesis Report

Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER Of TECHNOLOGY

IN

Very Large-Scale Integration Engineering

By

Saurav Anil Jha
(20MECV05)

Department of Electronics and Communication Engineering,
Institute of Technology,
Nirma University,
Ahmedabad 382481
May 2022

CERTIFICATE

This is to certify that the Thesis Project Report entitled "Low Power Verification using power Management Cells." submitted by Mr. Saurav Anil Jha(20MECV05) towards the partial fulfillment of the requirements for the award of a degree in Master of Technology(MTech) in the field of Very Large-Scale Integration(VLSI) Engineering in Nirma University is the record of work carried out by him under our supervision and guidance. The work presented has reached a level that is required for being accepted for a related examination. The results embodied in this thesis project work to the best of our knowledge have not been submitted to any other University or Institution for the award of any degree or diploma.

Date: 13th May 2022

Dr. Amisha Naik
Associate Professor
Institute of Technology,
Nirma University, Ahmedabad

Dr. Usha Mehta
Program Coordinator,
Professor,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Dhaval Pujara
Professor and Head
EC Department
Institute of Technology,
Nirma University, Ahmedabad

Dr. Rajesh Patel
Director,
Institute of Technology,
Nirma University, Ahmedabad

External Certificate

This is to certify that the Major Project Report entitled **"Low Power Verification using Power Management Cells"** submitted by **Saurav Anil Jha**(Roll No. 20MECV05) as the partial fulfillment of the requirements for the award of the degree of Master of Technology in VLSI Design, Electronics & Communication Engineering, Institute of Technology, Nirma University is the record of work carried out by him under my supervision and guidance. The work submitted in our opinion has reached a level required for being accepted for the examination.

Date: 13th May 2022

Sunil Gokhale

Mr. Sunil Gokhale
Engineering Manager
Intel, SRR, Bangalore, India

Undertaking for Originality of the Work

I, Saurav Anil Jha, Roll-No.20MECV05, give undertaking that the Thesis Project entitled"Low Power Verification using Power Management Cells" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in Very Large-Scale Integration(VLSI) Engineering of Nirma University, Ahmedabad 382481, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any other published work or any project report elsewhere; it will result in severe disciplinary action.

Saurav Jha

(Signature of the Student)

Date: 13th May 2022

Place: Ahmedabad

Endorsed by:

Sunil Gokahle

(Signature of External Guide)

Amisha Naik

(Signature of Internal Guide)

Acknowledgment

A journey is easier when you travel together. Interdependence is certainly more valuable than independence. The result of the work in this thesis where I must go along with and support by many people. I express my gratitude to the people who supported me in this work.

With immense pleasure, I express my sincere gratitude, regards, and thanks to my supervisor Prof Amisha Naik for his excellent guidance, invaluable suggestions, and continuous encouragement at all the stages of my research work. Her time and effort in this project me was the reason for all my success. I have been fortunate to have him as my guide as he has been a great influence on me, both as a person and as a professional.

It was a pleasure to be associated with the Very Large-Scale Integration Lab at Nirma University and I would like to thank the entire Laboratory Staff. To acknowledge help taken from friends is always a joy. I take this opportunity to convey sincere thanks to my class fellows for their smiles and friendship making life at Nirma University enjoyable and memorable.

The chain of my gratitude would be incomplete if I would forget to thank the first cause of this chain, using Aristotle's words, The Prime Mover for showering His blessings on me always.

Saurav Jha

Name of the Student

Abstract

In the last many year's power management has become the major problem when it comes to lower node technology, where static power dissipation is more than dynamic power dissipation. In nanometre technology, the power is an important issue as the technology node decreases the transistor number, the Speed of the performance, and leakage current increases so power dissipation will be more. Power can be optimized using different power-saving techniques like Power Gating, Multi-Voltage, Dynamic Voltage Frequency Scaling (DVFS).

The low power verification tool uses a multi-voltage design and low power static verification rule checker which will check the functionality and verify whether the low power architecture is designed in a perfect manner and checks whether any electrical violation is found or not. Low power standard which is also known as Unified Power Format (UPF) makes ICs reliable, robust, and standardized and it needs to verify their functionality and behaviourally correctness which ensures that it follows IEEE 1801 low power standards.

Unified Power Format (UPF) comes into action when power intent information is not specified in Hardware Description Language (HDL). UPF specifies all power intent information in the design flow. Using UPF commands power management can be controlled by specific cells like isolation, retention, power switches, and level-shifter cells which can be checked by a low power verification tool. The Low Power Verification tool normally checks and verifies any missing cells in the design in the Design Stage and their connections in the Power/Ground stage and finally, it will debug the issues.

Contents/Index

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Subscripts

o/p	Output
i/p	Input
L	Low
H	High

Abbreviations

UPF	Unified Power Format
VC LP™	Verification Compiler Low Power
VCS® NLP	VCS Native Low Power
IC	Integrated Circuit
GUI	Graphical User Interface
PG	Power/Ground
ELS	Enable Level Shifter
LS	Level Shifter
AON	Always-ON
PNR	Placement and Route

Chapter 1

1.1 Introduction

In this modern world, technology is developing at a very fast rate, and technology node is decreasing per the Gordon Moore the co-founder and Chairman Emeritus of Intel Corporation has observed that the number of the transistor on a computer chip was doubling every 18 to 24 months and the size of IC's are shrinking too. The small size of an IC has many advantages, but its power density increases which lead to many concerns. An IC with a high-power density not only heats up the product but also becomes unstable early. Because of this issue, the IEEE 1801 low-power standards are formed to make the ICs reliable, robust, and standardized which will mitigate these issues. Once IC is designed, we need to verify it by verifying its functionality and behaviorally correct and ensure that it follows IEEE 1801 low power standard which is also known as Unified Power Format (UPF). This IEEE standard provides portable, specifications of low power which can be used in electronic products, design flow, analysis, and verification implementation flow.

As the node technology is even better every year and demand is also very high so power consumption is also increased. (i.e. the technology node, which is above 90 nm, dynamic power dissipation also increases compared to static power consumption. To decrease the dynamic power consumption, and decreases average switching frequency, it is the amount of power consumed to transit a node from a 0 to 1 state or vice versa). With the technology node, which is below 90 nm, static power consumption is dominant compared to dynamic power consumption. Static power consumption is due to the result of leakage quiescent current (i.e. when there is no toggling of a node from HIGH to LOW and vice versa).

Low Power Verification tools are used for voltage-aware functional verification as a low power solution. Coverage for all advanced power management functions is done using an advanced Low Power Static-rule Checker. Low power techniques are used for advanced low power management as it is used in SOC designs. Low power Verification static lower checker can check and analyze all the error it performs 500 plus checks, gives maximum chip capacity, and process all the steps to perform overall low power static signoff. Low Power verification supports the IEEE standard format known as Unified Power format and absolute static analysis for capturing the power intent, power state shift, and multi-rail macros of the low power design.

1.2 Motivation

Low Power Verification static signoff tool provides the design report, intelligent process analysis, fast unified debug, support unified power format, fast and rapid outputs, and performs overall checks in less time. Accurate issues and violations can be found in the design and low Power verification can be done using GUI to debug the design's violation.

1.3 Objective

The main objective of the low power verification static signoff tool is to perform different kinds of checks on design like a functional check, power intent consistency check, structural check, power ground check, functional check, and signal corruption checks on UPF. There can be many power domains in the design which can be easily solved by a low power verification tool also it understands the power intent and can transform many power states to a lesser one, thus it reduces the effort which is involved in it and then verifies all the power states, transitions, and sequences.

1.4 Problem Statement

How Low Power Verification performs various checks in synthesis and placement and route with following Functional check, Signal Corruption check, Power Intent Consistency Check, Structural checks, Power and Ground check. How it is used to pipe clean the UPF according to the IEEE standard 1801 according to these checks.

1.5 Approach

Checking signal corruption and continuing and performing various checks by Low Power verification tool and by debugging these signal and cells issues. Checking strategies and missing cells which will determine the properness of the UPF and netlist file.

1.6 Scope of the Project

The Scope of the project is to reduce the issues/violations by reducing the run time and machine usage using the Low power verification Tool. By using a small number of extra input files, the

Low power verification tool can be used to verify the power intent and can identify power-supply mistakes by low power tools. The Low Power Verification tool performs static checks from RTL level to the sign-off state.

Chapter 2

2 Literature review

2.1 Low Power verification Checks Flow

Low Power Verification performs different kinds of checks, here in Fig 2.1(a) shows a low power check where the first step is to check the RTL block which can be done using UPF consistency and signal corruption check, Synthesis design block can be checked using signal corruption, structural and functional check and Power/Ground connected block can be checked using signal corruption, structural, functional and PG check. At RTL Block, Low power verification helps in identifying the power intent issues in UPF early in the design life cycle and it will give clean UPF before initializing the design flow[3].

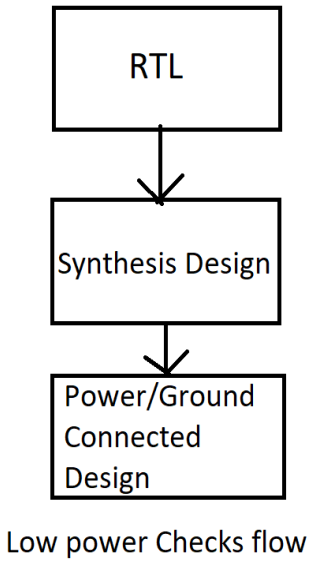


Fig 2.1(a) Low Power Checks Flow

Low power verification takes in RTL like Verilog, VHDL, SVD, PG Netlist, or post-layout netlist of the design. Low Power verification requires standard Liberty (Library) files (.db file), Low power verification read the Liberty DB file for timing information, annotates power connections, describes the design, and checks special cells for the Gate level netlist. It will take

power intent which is in UPF. In the output, Low power verification generates log files, errors, and warning reports for the violation specified in reports which are related to the low lower static rule check. low power verification provides TCL as a basic physical system that helps to debug these violations. Low Power verification also helps to debug using low power verification GUI (Graphical User Interface) as shown in Fig 2.1(b)

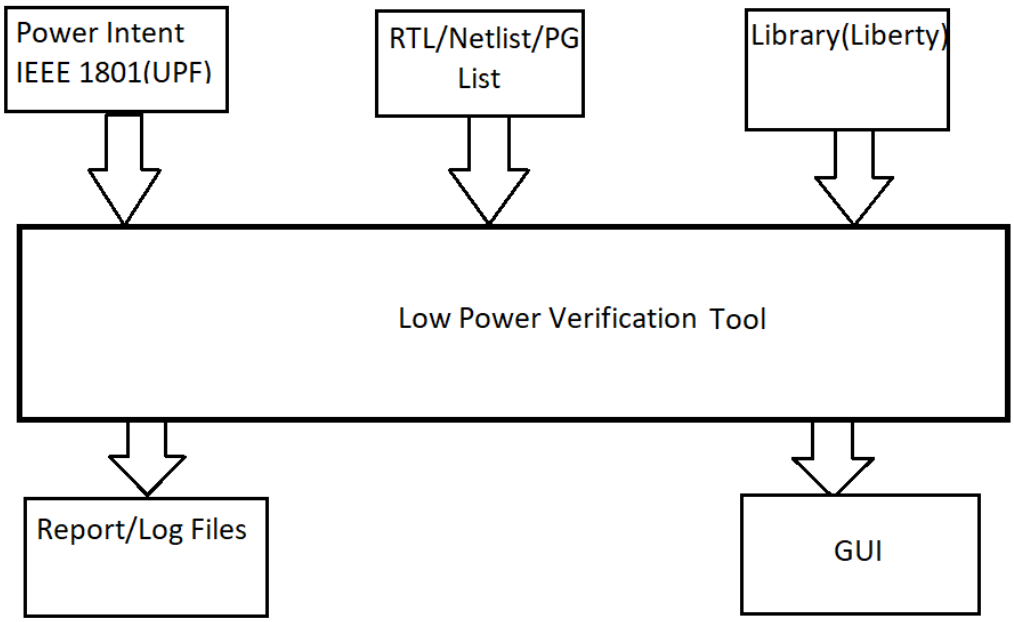


Fig 2.1(b) Low power verification Checks

The UPF is checked according to Fig 2.1 Low Power Check Flow, and these checks include:

2.1.1 Power Intent Consistency Checks

Performs syntax and semantic checks on the UPF that help to approve the consistency of the UPF before starting with the implementation.

2.1.2 Signal Corruption Checks

It detects the violation of power architecture at the gate-level netlist.

2.1.3 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.

2.1.4 Power and Ground (PG) Checks

Check the PG consistency against the UPF specification for power network routing on physical netlists.

2.1.5 Functional Checks

Check and verify the correct functionality of isolation cells and power switches.

After low power verification GUI debug, violation reports are generated which include special cells, checking the connections, strategy, and missing of these special cells include.

2.2 Low Power Techniques

Low power technology is used to reduce the total power consumption in a system by means of reducing capacitive load and activity factor which will further reduce the switching component of the dynamic power they are classified as:

1)**clock gating:** It is used to reduce the dynamic power in the design. It is simply to turn off the design when it is not needed without affecting the functionality of the design.

2)**Multi-Voltage:** It is used to save the static power and dynamic power of the design. The chip is implemented with different supply voltages. Various functional blocks run at different supply voltages, and we can save the power losses by reducing the supply voltage.

3)**Dynamic voltage and frequency scaling:** By changing the voltage with the frequency we can reduce the power consumption. When a high operating is required the supply voltage is increased to attain a higher frequency.

4)Multi Vth: In the Multi threshold voltage technique use both Low Vth and High Vth cells. Use lower threshold gates or critical paths with higher threshold gates of the critical path. Using the Multi Vth technique performance can be improved without increasing power but fabrication complexity using multi-Vth cell can be increased. It also lengthens the design time. Improper optimization of the design may utilize lower Vt cells and hence could end up with increased power.

5)**Power Gating:** Power gating is used to reduce the consumption of the power by shutting off the current which is not used in the blocks. Thus it saves overall power in the chip.

2.3 Special Power management cells:

Power management cells are the special cells that are used to manage the power distribution in the design through the Unified Power Format (UPF) file. These techniques use special cells which are power management cells. Power management command defines the characteristics of the instances of the power management cells used to implement and verify the power intent of the design. These power management cells solve the Voltage error, Voltage crossover, and power domain error. Power management cells are also known as Dual power rail Supply Cells as they have two sets of rails. Dual Power Rail has primary as well as secondary power rail to give power to these cells. These cells are:

1)Isolation Cell
2)Level Shifter Cell
3)Retention Cell
4)Power Switch Cell
5)Always-On Cell
6)Enable Level Shifter
7)Bi-directional Level Shifter

2.3.1 Isolation Cell:

Isolation cells are also known as clamp cells. These are the special cells used in low power design. It is inserted by a synthesis tool for the isolation of the wires and buses crossing from the power gated domain to the always-on domain. Isolation cells are used to reduce the power dissipation in the design. The Power domain, which is off will not drive any output, so these outputs will act as floating nodes and problems can occur for active power domains which will get floating nodes as input. It would lead to a crowbar current that exceeds the defined limit which would affect the functionality of the power-up domain. ‘AND’ gate and ‘OR’ gates are the types of clamp cells that are used when isolation cells pass Unknown signal ‘X’, which will lead to leakage of power. Hence to pass a Signal ‘HIGH’ or ‘1’, it needs an OR gate and if it passes a Signal ‘LOW’ or ‘0’, it needs AND gate as an isolation cell.

Consider that design has two power domains PD_1 and PD_2 and signal S1 d from PD_1 to PD_2. When both the power domain is powered-up, the signal has either 0 or 1 value. But when the power domain PD_1 is powered down and PD_2 is powered up. S1 value will become unknown. This corrupts the power-up logic in the PD_2 domain. To avoid this, the design engineer placed isolation cells between the outputs of the powered-down domain and the input of the powered-up domain.

During normal working conditions, an isolation cell works as a buffer. When PD_1 is powered down, the isolation cells clamp their output value to either 0 or 1 value. The isolation cell has an isolation enable signal that determines whether the isolation cell should provide isolation or work as a buffer.

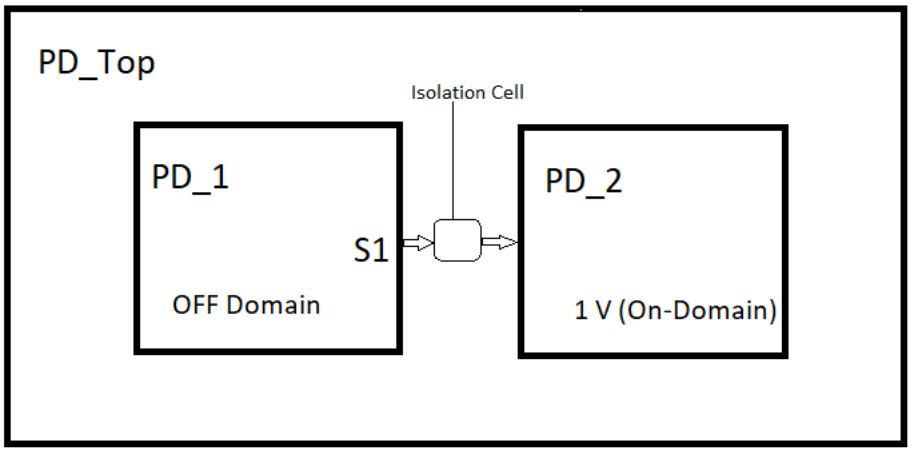


Fig 2.3.1(a) Block diagram of Isolation cell Function.

Without an isolation cell, there will be leakage of power when the MOSFET gate receives a don't care signal/Unknown signal (X). When to pass '0', AND gate is used as shown in fig 2.3(b), and to pass '1', Or gate is used as shown in fig 2.3(c).

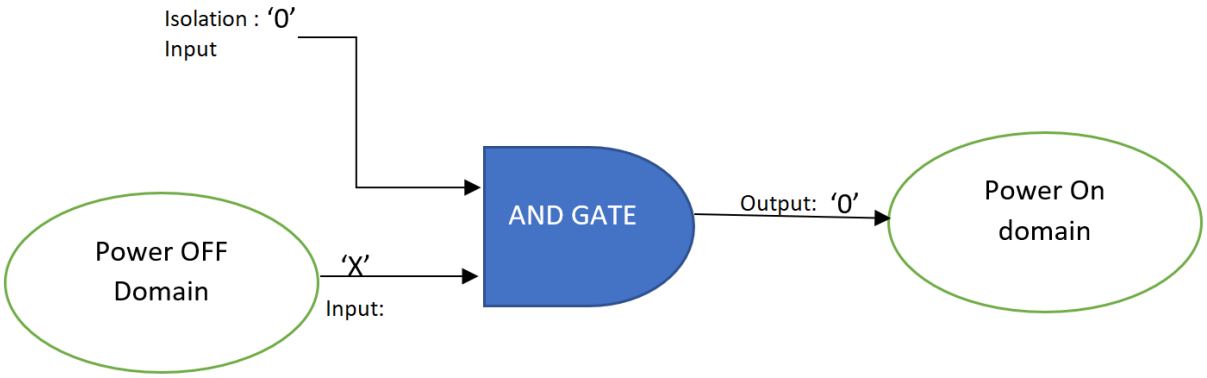


Fig 2.3.1(b) Type 'LOW' Signal transmission using 'AND' gate as Isolation cell.

Truth Table of ISOLATION Cell		
Isolation Input	Input	Output
0	X	0
1	X	X
X	1	X
X	0	0

Table no 2.3(a) Truth Table for isolation cell using AND GATE retaining or clamp '0'

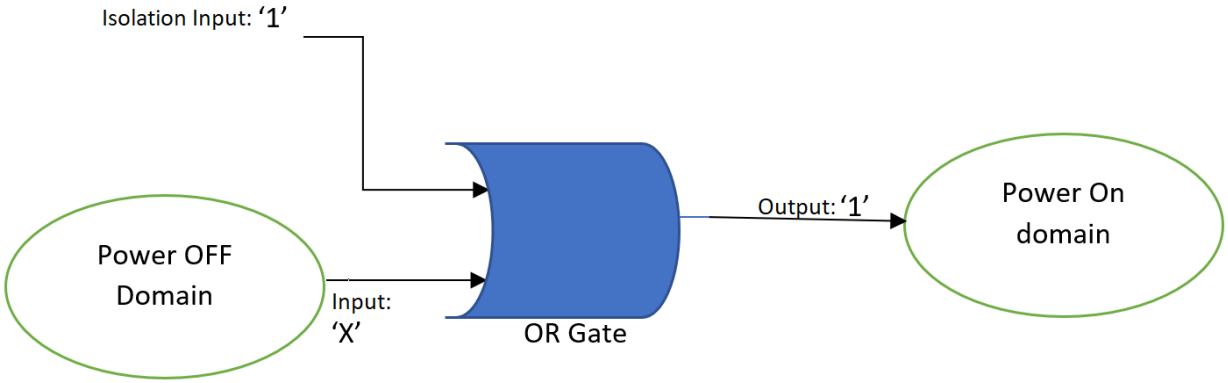


Fig 2.3.1(c) Type 'HIGH' Signal transmission using 'OR' gate as Isolation cell.

Truth Table of ISOLATION Cell		
Isolation Input	Input	Output
0	X	X
1	X	1
X	1	1
X	0	X

Table no 2.3(b) Truth Table for isolation cell using OR GATE retaining or clamp '1'

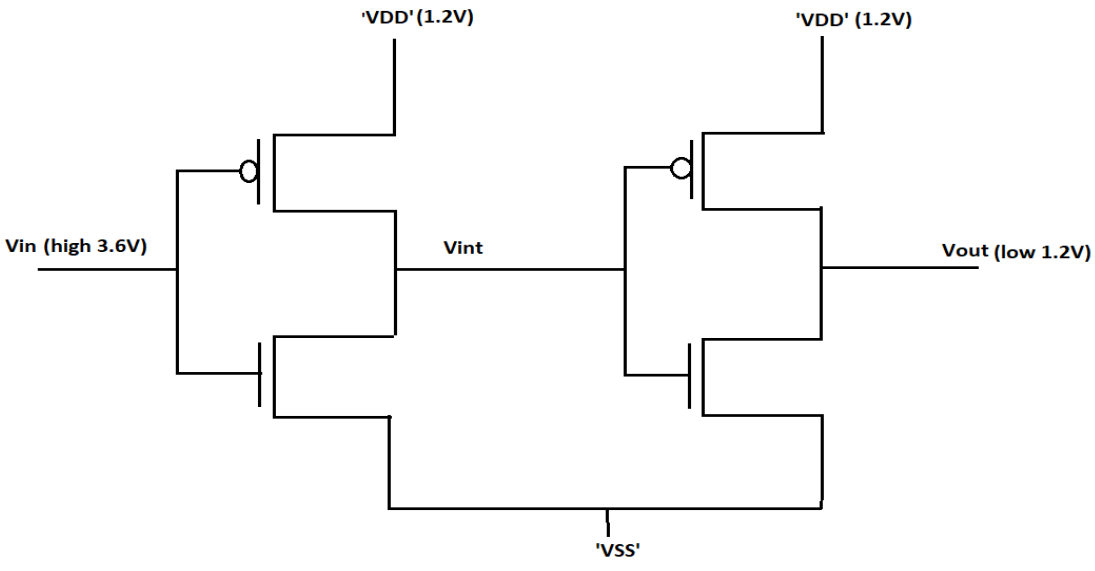
2.3.2 Level Shifter Cell:

Level Shifter Cell converts one voltage level to another voltage level. These are the special standard cells used in the multi-voltage design. Level shifters are inserted by the synthesis tool.

Level Shifter converts voltage in three modes:

1) High to Low-Level Shifter-It converts voltage from High to Low.
2) Low to High-Level Shifter-It converts voltage from Low to High.
3) Bidirectional Level Shifter-It converts both from L to H and H to L.

Level Shifter Specification should have a high slew rate for the signal transmission as input to logic gate is changing slowly which takes more time for transition (i.e. high-rise time and fall time high leakage of current) which will result in high crowbar current. Normally Low to High-Level Shifter amplifies the signal.



High to Low Level Shifter

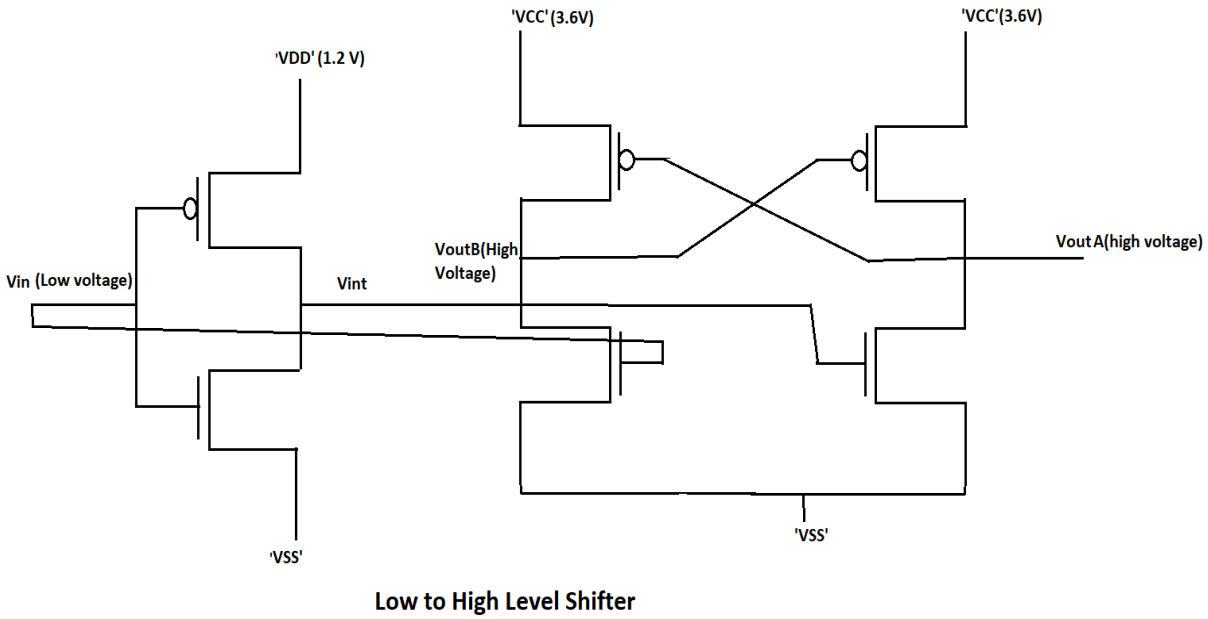


Fig 2.3.2(a) Requirement of Low to High & High to Low-level shifter

In fig 2.3.2(a) Let the signal is passing from analog(3.6V) to digital block(1.2V) we need a high to low-level shifter using a high voltage transistor to low voltage transistor.

Consider a design with two power domains PD_3 and PD_4 and signal S2 propagate from PD_3 to PD_4. When both power Domains are powered up and the voltage difference is lesser than the threshold value (which can be seen in the simulator), there are no issues in the design. However, if the voltage difference is greater than the threshold value. Assume PD_4 is operating at a higher voltage level than PD_3 then a logic 1 at PD_3 can be assumed as a logic 0 at PD_4. This led to incorrect data transmission. To avoid this issue the design engineer places level shifter cells between the power domains.

The level shifter cells convert high voltage levels to low voltage levels and vice versa. This enables correct data transmission between two different power domains. The level Shifter cell has a level sifter enable signal that determines whether the level shifter cell should convert the voltage level, or it work as a buffer.

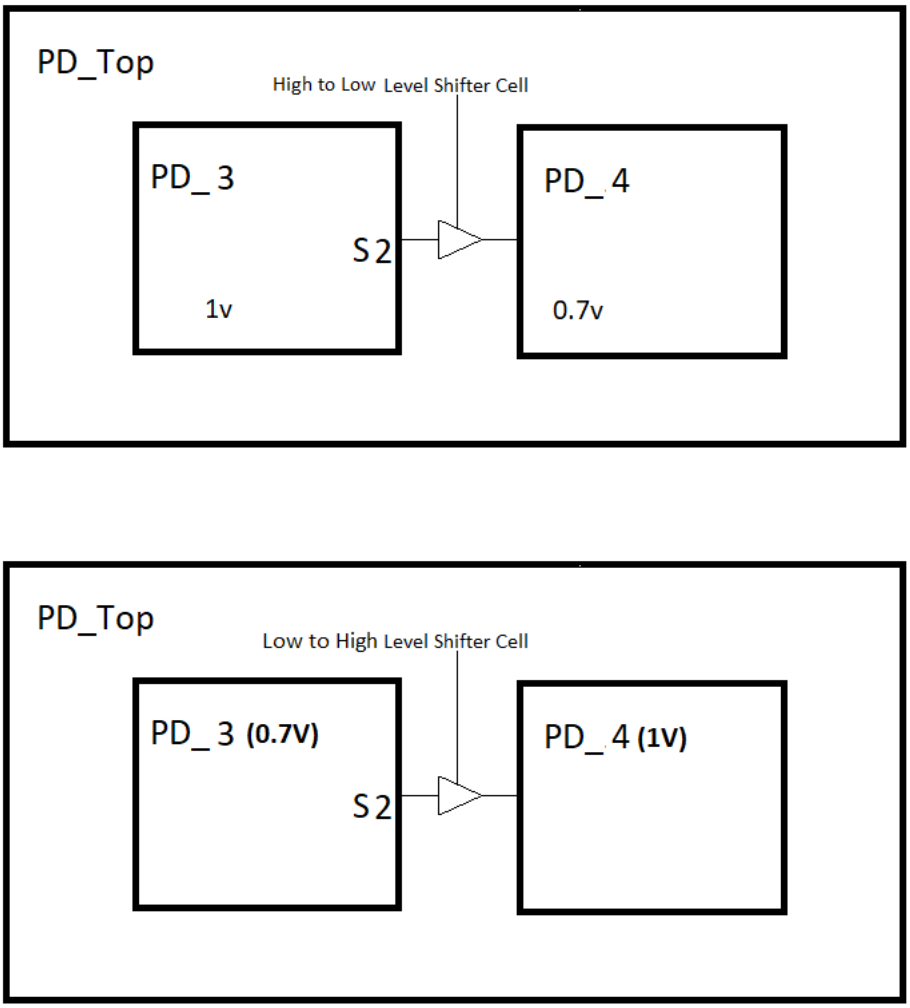


Fig 2.3.2(b) Level Shifter Cell Function

2.3.3 Retention Cell:

Retention cells are the special cells used to store information which is in the form of binary format. It can hold its internal state/data in the register which is also known as the shadow register before powering down. When the primary power supply is cut off/Shut down it has the ability to retain its state when power is switched on again. The best example for retention cell is Master-Slave alive Flip flop. It consists of a Master latch and a Slave latch whose output depends on a Negative/positive clock. The Slave latch stores

the data when primary power is switched off in the retention operation. The main power of the latch comes from the main power rail of the “ALWAYS-ON” cell. State saving latch is implemented with a high threshold transistor to decrease the power dissipation and this power is active even in power-down mode. The control of the retention cell operation is done by the “SLEEP” signal. States in the register in a power domain should be retained when their power supply is removed. Retention cells are always on so they always absorb power. It is generally a low leakage cell. Low Threshold cells are high-performance cells whose power comes from a normal supply it can be latch/regular Flip Flop.

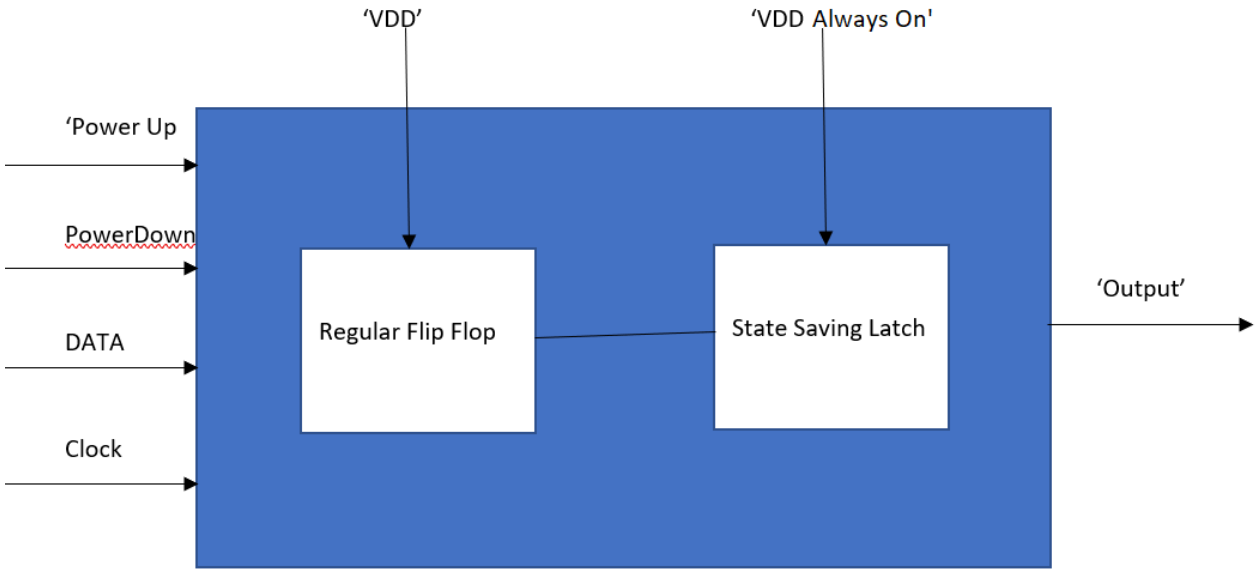


Fig 2.3.3(a) Simple block diagram of Retention Cell

The State saving latch is not active in ‘normal mode’.Before power down of the regular flop but the value of the regular flip flop will transfer its value to the state latch before power down. The power of regular flip flop is shut down but the state saving register is powered up for retention of states.

2.3.4 Power Switch

Power Switch cells are used for Power Gating in the design by powering off the portion of the design. When any of the subblocks are not needed or not in use it can be shut down using a power switch cell, scope command is used to switch off the voltage domain. Header (PMOS) and Footer (NMOS) transistor of fixed size are used as Power Switch. When Power Switch is needed to work then the sleep signal is given the signal 'LOW' to activate the power switch of the PMOS transistor. The power switch is inserted in the column as well as in a ring manner.

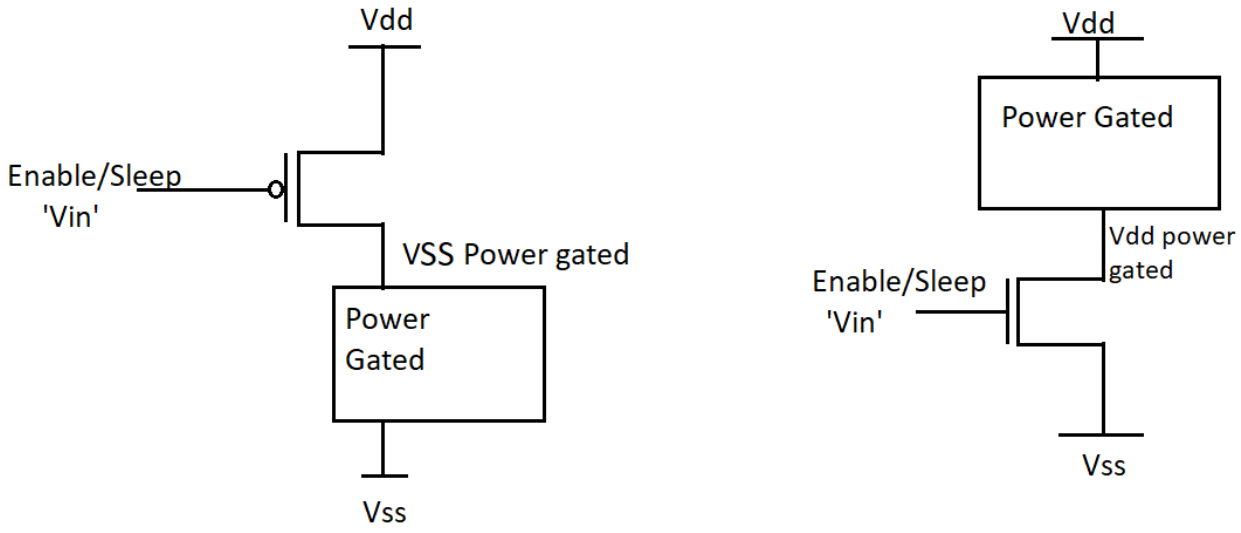


Fig 2.3.4(a) MTCMOS Power Switch used for Power Gating.

Consider PMOS with an MTCMOS power switch at drain which will control the power gating by powering off the MOSFET using power gated logic. This Multi Threshold CMOS are of two types 1)Low Voltage Threshold(LVT) 2)High Voltage Threshold (HVT)

Short circuit power can be reduced by a Low Voltage Threshold during normal mode and leakage power can be reduced by a High Voltage Threshold during off mode.

2.3.5 Always-On Cell

Always on the cell cannot be switched off. Generally, buffer and inverters are used as Always-On cells. Logic cells should be powered on irrespective of power domain is switched off. The power domain which can be switched off is known as the power down domain. The logic inside the cell is used to transfer signal in the off domain while Source and Sink domains are in the 'ON' state.

Always on cell are of two types 1) Single power rail supply 2)Dual Power rail supply. Single Rail Power cells: They contain only single power in inverters and buffers from the standard cell libraries.Dual Rail Power cells: It contains a dual power supply for inverters and buffers and it is used for always-on cells.

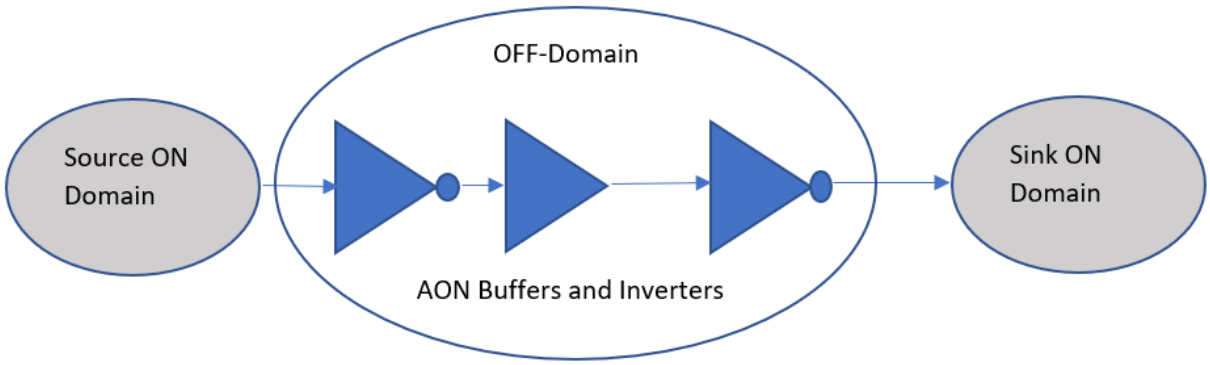


Fig 2.3.5(a) AON buffer & Inverter are in off domain for signal transfer between on domain

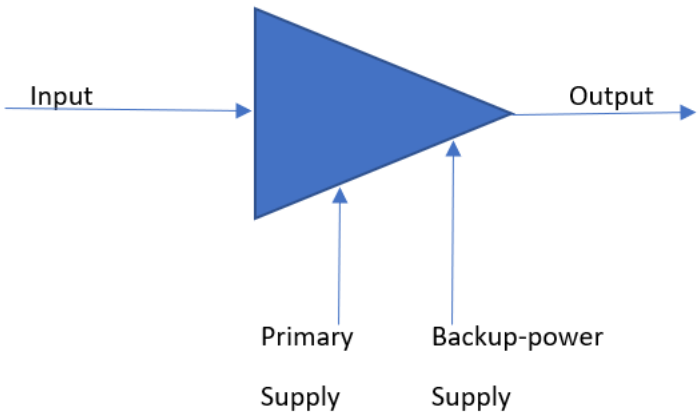


Fig 2.3.5(b) Dual Power Rail Supply AON Cell

The dual power rail supply consists of two power: primary power and secondary power. Primary power is always on power while secondary power can be switched off and turned on according to RTL Engineer. While Single Power rail supply only consists of primary power.

2.3.6 Enable Level shifter:

Enable Level Shifter is also known as ELS cells. It can work and function as both a Level Shifter cell as well an Isolation cell. Its advantage is to reduce the area in the chip. It can change the voltage from low to high and high to low and also perform isolation from the Off domain to the On domain**.** If there is a crossing between two power domains from the OFF domain to the On domain we need an isolation cell with a perfect control signal and clamp cell and if there is any voltage mismatch between the two domains a level shifter is required it can be LH, HL, or bi-dir LS. It can be possible that both situations are present so in this case, Enable Level Shifter is needed.

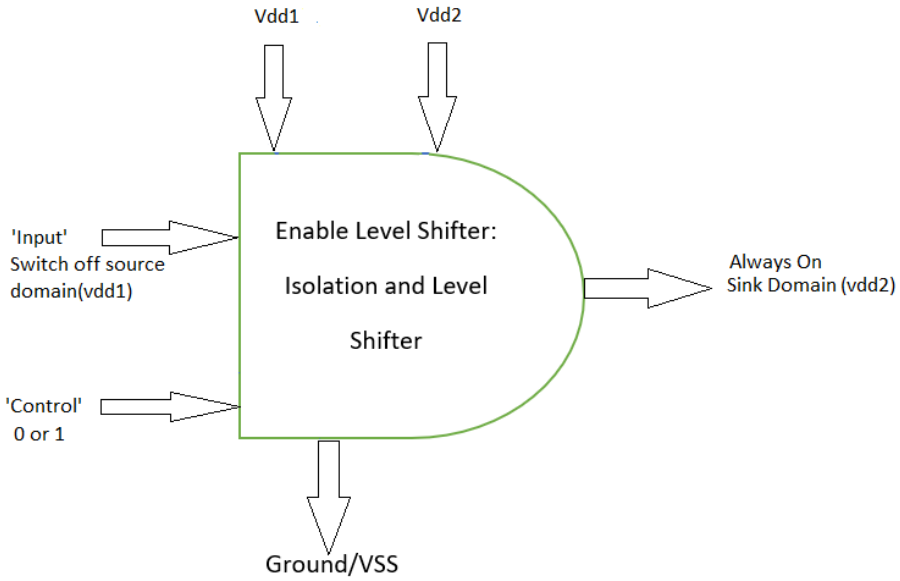


Fig 2.3.6(a) Enable Level Shifter

Here ‘Input’ represents data, for switch off-source domain isolation will work and ‘control’ enable input will go for logic 0 for clamping in enable level shifter and for level shifting from vdd1 voltage domain to vdd2 voltage domain, control pin will be at logic 1 for voltage level shifting.

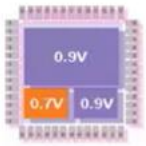
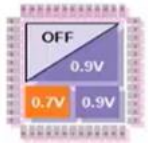
Multi-Voltage Special Cell Requirement						
Figure	**Working & Function**	**Isolation cell**	**Level Shifter Cell**	**Retention Cell**	**Power Switch Cell (MTCMOS)**	**Always-On Cell**
	Multi-Volatge Domain	No	Yes	No	No	No
	Multi-Supply with shutdown no state retension	Yes	No	No	Yes	No
	Multi-Voltage with Shutdown	Yes	Yes	No	Yes	No
	Multi-Voltage with Shutdown and State Retention	Yes	Yes	Yes	Yes	Yes

Table NO 2.3(c) Multi-Voltage Special Cell Requirement

Chapter 3

3 Low Power Verification

3.1 Low Power Verification Flow Design:

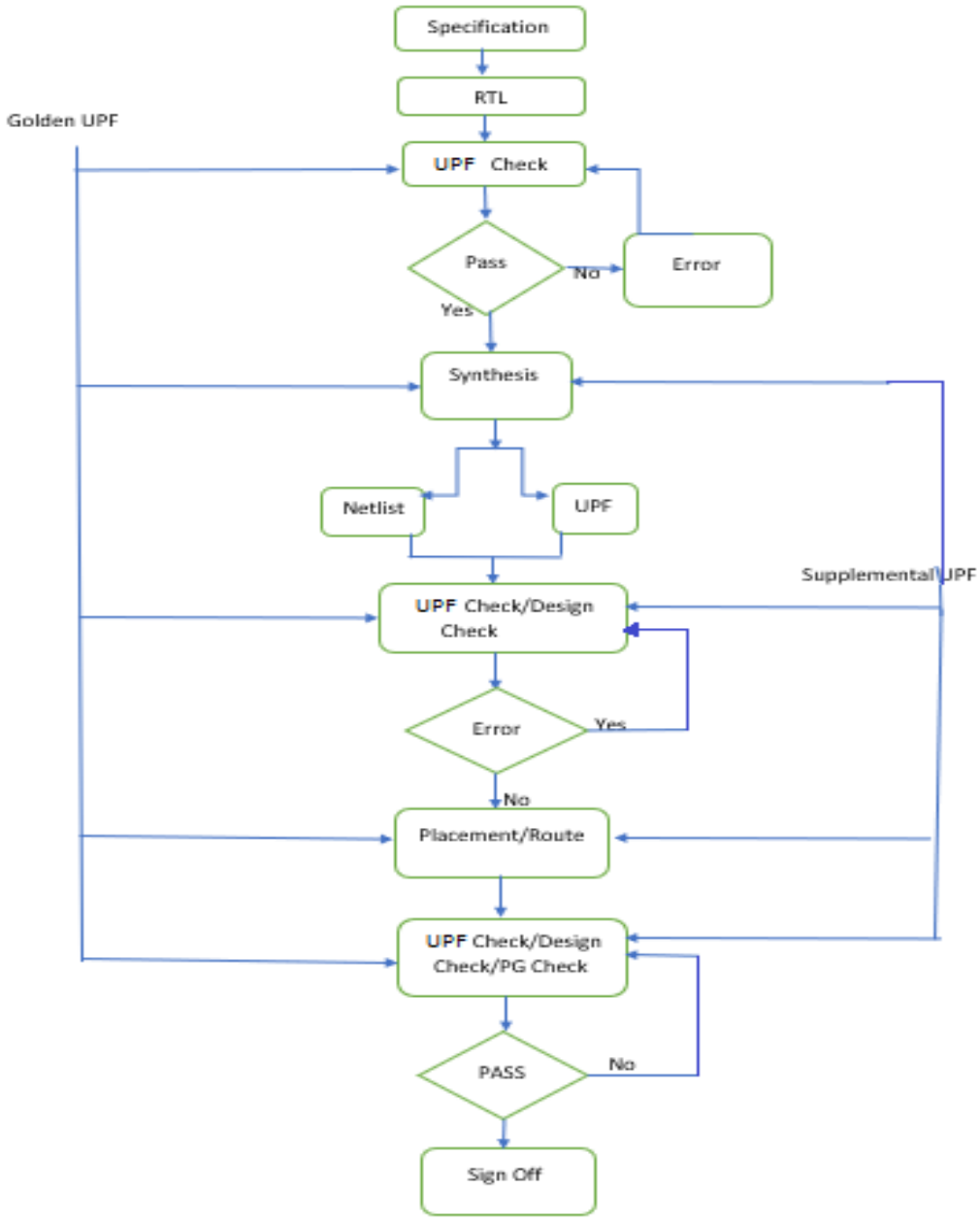


Fig 3(a) Low Power Verification Design Flow block diagram

In Low Power verification design Flow, It starts with the specification of the design specification including area, power, and performance. Register Transfer Level(RTL) block tells about how the data is transferred from one register to another register. This data is operated using boolean logic expression when transferred to different registers with each clock cycle. This process is done using high-level codes such as Verilog and VHDL and these files are accepted by the synthesis tool. RTL is added with UPF. It defines the power architecture. It ensures that the design will work properly under a power management unit(PMU) with defined power architecture.RTL then is converted to gate-level netlist. Here Golden-UPF comes with RTL. UPF file is used in various blocks in Low power verification design flow such as synthesis, placement, and route with various low power verification designs and PG checks. UPF is checked before the synthesis step, to check the quality of the UPF and its errors of it. Low Power Verification checks are performed at this stage. It is then transferred to the synthesis stage where low power synthesis is performed by the compiler. Synthesis normally converts the RTL to the gate-level netlist. All other low power information is written in Supplemental UPF by the design compiler. This combined Supplemental UPF and Golden UPF together are used for “UPF check” and “Design Check” using low power verification tool to check insertion of the cell is correct or not during synthesis. Then both are passed to PNR(Placement and Route) together with a synthesized netlist where “UPF/low power checks”, “Design Checks” and “PG Checks” are performed using the UPF file and physical netlist. The last stage is Sign-Off it is also known as tape out where our design is ready and can go for physical implementation.

3.2 UPF File Structure:

Unified Power Format file is saved in the tool to verify its functionality in the .upf file format.It follows some common impoertant upf_intent_command.Its file structure is as follows:

Create_power_domain

Set_design_attribute

Create_supply_port

Create_supply_net

Connect_supply_net

Add_power_state

Create_pst

Add_pst_state

Create_power_switch

Map_power_switch

Set_isolation

Set_isolation_control

Map_isolation

Set_level_shifter

Map_level_shifter

Set_retention

Map_retention

UPF File Structure

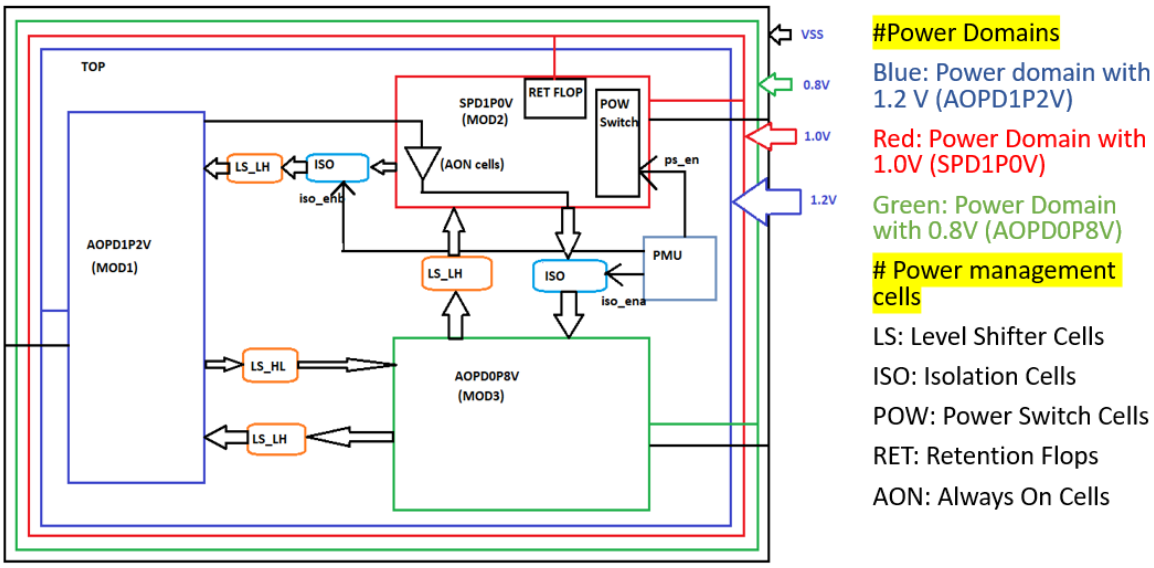


Fig 3.2.1 UPF File Structure

The example UPF file structure works using UPF commands, as shown above, some of the command descriptions is as follows:

```
//—Creating Power Domains —//
create_power_domain top -include_scope
create_power_domain AOPD1P2V -elements { top/mod1 }
create_power_domain SPD1P0V -elements { top/mod2 }
create_power_domain AOPD0P8V -elements { top/mod3 }
#——— Supply Ports & Net Connections ————#
create_supply_port VDD1P2
create_supply_net VDD1P2 -domain top
create_supply_net VDD1P2 -domain AOPD1P2V -reuse
connect_supply_net VDD1P2 -ports VDD1P2
create_supply_port VDD1P0
create_supply_net VDD1P0 -domain top
create_supply_net VDD1P0 -domain SPD1P0V -reuse
create_supply_net VDD1P0_SW -domain SPD1P0V   //switching path
connect_supply_net VDD1P0 -ports VDD1P0
create_supply_port VDD0P8
```

create_supply_net VDD0P8 -domain top

create_supply_net VDD0P8 -domain AOPD0P8V -reuse

connect_supply_net VDD0P8 -ports VDD0P8

create_supply_port VSS

create_supply_net VSS -domain top

create_supply_net VSS -domain AOPD1P2V -reuse

create_supply_net VSS -domain SPD1P0V -reuse

create_supply_net VSS -domain AOPD0P8V -reuse

connect_supply_net VSS -ports VSS

//— Reciever shutdown logic ——//

create_power_switch power_switch -domain SPD1P0V \

-input_supply_port {VDD1P0 VDD1P0}\

-output_supply_port { VDD1P0 VDD1P0_SW} \

-control_port {PMU/ps_en } \

-on_state {state_name VDD1P0 {!ps_en}}

//— Isolation Cell Insertion —//

set_isolation iso_out -domain SPD1P0V \

-applies_to outputs \

-isolation_power_net VDD1P0 -isolation_ground_net VSS \

-clamp_value 1 \

-isolation_signal PMU/iso_ena \

-location default

```
//— Retention Logic for Switchable domain —//
set_retention RTF -domain SPD1P0V \
-retention_power_net VDD1P0 \
-retention_ground_net VSS \
-save_signal {PMU/rtf_en high} \
-restore_signal {PMU/rtf_en low} \
//—Level Shifter for multi-Voltage Domain —//
set_level_shifter LS_0P8_1P0 -domain SPD1P0V \
-applies_to inputs \
-location self \
-source AOPD0P8V.primary \
-input_supply_set AOPD0P8V.primary -output_supply_set SPD1P0V.primary
set_level_shifter LS_1P0_1P2 -domain AOPD1P2V \
-applies_to inputs \
-location self \
-source SPD1P0V.primary \
-input_supply_set SPD1P0V.primary -output_supply_set AOPD1P2V.primary
set_level_shifter LS_1P2_0P8 -domain AOPD0P8V \
-applies_to inputs \
-location self \
-source AOPD1P2V.primary \
-input_supply_set AOPD1P2V.primary -output_supply_set APD0P8V.primary
set_level_shifter LS_0P8_1P2 -domain AOPD0P8V \
-applies_to inputs \
-location self \
-source AOPD0P8V.primary \
-input_supply_set APD0P8V.primary -output_supply_set APD1P2V.primary
//— Defining  AON Cell —//
define_always_on_cell -cells AON_BUF \
-power_switchable VDD1P0_SW -ground_switchable VSS \
-power VDD1P0 -ground VSS
//— Creating Power State Table(PST) —//
```

```
//— Defining  AON Cell —//
define_always_on_cell -cells AON_BUF \
-power_switchable VDD1P0_SW -ground_switchable VSS \
-power VDD1P0 -ground VSS
//— Creating Power State Table(PST) —//
add_power_state TOP.primary \
-state ON { -supply_expr {power == ‘ {FULL_ON, 1.0} && ground == ‘ {FULL_ON, 0.0 }} \ -
simstate NORMAL }
add_power_state AOPD1P2V.primary \
-state ON { -supply_expr {power == ‘ {FULL_ON, 1.2} && ground == ‘ {FULL_ON, 0.0 }} \ -
simstate NORMAL }

add_power_state SPD1P0V.primary \
-state ON { -supply_expr {power == ‘ {FULL_ON, 1.0} && ground == ‘ {FULL_ON, 0.0 }} \ -

 add_power_state AOPD0P8V.primary  \
 -state ON { -supply_expr {power == ‘ {FULL_ON, 0.8} && ground == ‘ {FULL_ON, 0.0 }} \
 -simstate NORMAL }
```

Some of the command descriptions are as follows:

- **power domain generation:** This command is used when a common set of power supplies is shared by a group of elements. At a specified scope power domain is created it supplies power distribution in the entire network.

  create_power_domain domain_name **[-subdomains** domain_list**]** [-**atomic**] **[-available_supplies** supply_set_ref_list**]** **[-elements** element_list**] [-exclude_elements** exclude_list**] [-supply** {supply_set_handle **[**supply_set_ref**]}]***

- **Set_design_attribute:** This command specifies properties for instance and model.

  set_design_attributes [**-models** *model_list*] [**-is_soft_macro** [<**TRUE** | **FALSE**>]] [**-is_hard_macro** [<**TRUE** | **FALSE**>]] [**-attribute** {*name value*}]* [**-switch_cell_type** <**coarse_grain** | **fine_grain**>]
  [**-elements** *element_list*] [**-exclude_elements** *exclude_list*]

- **Create_supply_port:** This command creates a port in the scope of the power domain or outside of it using the scope command. If the scope is not included, the port is created in the current scope.
  create_supply_port *port-name* [**-domain** *domain-name*] [**-direction** <**in** | **out** | **inout**>]

- **Create_supply_net:** This command creates a net in the domain or outside of it using direction.

  create_supply_net *net-name* [**-domain** *domain-name*] [**-resolve** <**unresolved** | **one_hot** | **parallel** | **parallel_one_hot** |*resolution- function-name* >][**-reuse**]

- **Connect_supply_net:** This command connects the supply port and supply net together.

  create_supply_net *net-name* [**-domain** *domain-name*] [**-resolve** <**unresolved** | **one_hot** | **parallel** | **parallel_one_hot** |*resolution- function-name* >][**-reuse**]

- **Connect_supply_net:** This command connects the supply port and supply net together.

  connect_supply_net *net_name* [**-elements** *element_list*] [**-ports** *port_list*] [**-pg_type** *pg_type_list*]* [**-vct** *vct_name*] [**-cells** *cell_list*] [**-domain** *domain_name*]

- **Add_power_state:** This command is used to define the power state of the object.i.e It is Full on or OFF or it

  add_power_state [**-supply** | **-domain** | **-group** | **-model** | **-instance**] *object_name* [**-update**] [**-state** {*state_name* [**-logic_expr** {*boolean_expression*}] [**-supply_expr** {*boolean_expression*}] [**-power_expr** {*power_expression*}] [**-simstate** *simstate*] [**-legal** | **-illegal**]}]* [**-complete**]

- **Create_pst**: It will create a Power state table and define its PST name and supply nets for use in add pst state.

  create_pst *table_name* **-supplies** *supply_list*

- **Add_pst_state:** For one possible state of the design it defines the state of every supply net.
  add_pst_state *state_name* **-pst** *table_name* **-state** *supply_states*

- **Create_power_switch:** This command creates the power switch.

  create_power_switch power *switch name* [**-switch_type** <**fine_grain** | **coarse_grain** | **both**>] [**-output_supply_port** {*name of port* [*supply net name*]}] {**-input_supply_port** { *name of port* [*supply net name*]}* {**-control_port** { *name of port* [*net name*]}* {**-on_state** {*name of state input_supply_port* {*boolean_expression*}}* [**-off_state** { *name of state* {*boolean_expression*}}]* [**-supply_set** *supply set ref*] [**-on_partial_state** { *name of state input_supply_port* {*boolean_expression*}}]*

- **Set_isolation:** When the primary power supply is removed how it protects the power domains is set isolated.
- **Set_isolation_control:** The isolation strategy is controlled by a control signal coming from PMU.

- **Map_isolation:** It will choose and specify an isolation strategy.

  set_isolation *name of strategy* **-domain** *name of domain* [**-elements** list of elements] [**-exclude_elements** *excluded list*] [**-source** <*source_domain_name* | *source_supply_ref*>] [**-sink** <*sink_domain_name* | *sink_supply_ref*>] [**-diff_supply_only** [<**TRUE** | **FALSE**>]] [**-use_equivalence** [<**TRUE** | **FALSE**>]] [**-applies_to** <**inputs** | **outputs** | **both**>] [**-applies_to_boundary** <**lower** | **upper** | **both**>] [**-applies_to_clamp** <**0** | **1** | **any** | **Z** | **latch** | *value*>] [**-applies_to_sink_off_clamp** <**0** | **1** | **any** | **Z** | **latch** | *value*>] [**-applies_to_source_off_clamp** <**0** | **1** | **any** | **Z** | **latch** | *value*>] [**-no_isolation**] [**-force_isolation**] [**-location** <**self** | **other** | **parent** | **fanout**>] [**-clamp_value** <**0** | **1** | **Z** | **latch** | *value* | {<**0** | **1** | **Z** | **latch** | *value*>*}>] [**-isolation_signal** *list of signals]* [**-isolation_sense** <**high** | **low** | {<**high** | **low**>*}>]] [**-isolation_supply** *supply_set_list*] [**-name_prefix** *pattern*] [**-name_suffix** *pattern*] [**-instance** {{*instance name port_name*}*}] [**-update**]

- **Set_level_shifter:** It will set the level from domain crossing of different voltages.
- **Map_level_shifter:** This command is used at a different voltage which is operated at crossing domain should be shifted.

  set_level_shifter *strategy_name* **-domain** *domain_name* [**-elements** *element_list*] [**-exclude_elements** *exclude_list*] [**-source** <*source_domain_name* | *source_supply_ref*>] [**-sink** <*sink_domain_name* | *sink_supply_ref*>] [**-use_equivalence** [<**TRUE** | **FALSE**>]] [**-applies_to** <**inputs** | **outputs** | **both**>] [**-applies_to_boundary** <**lower** | **upper** | **both**>] [**-rule** <**low_to_high** | **high_to_low** | **both**>] [**-threshold** <*value*>] [**-no_shift**] [**-force_shift**] [**-location** <**self** | **other** | **parent** | **fanout**>] [**-input_supply** *supply_set_ref*] [**-output_supply** *supply_set_ref*] [**-internal_supply** *supply_set_ref*] [**-name_prefix** *pattern*] [**-name_suffix** *pattern*]
  [**-instance** {{*instance_name port_name*}*}] [**-update**]

- **Set_retention:** It is used when the primary power supply is off, a registered state in the power domain is getting back.

  set_retention *retention_name* **-domain** *domain_name* [**-elements** *element_list*] [**-exclude_elements** *exclude_list*] [**-retention_supply** *ret_supply_set*] [**-no_retention**][**-save_signal** {*logic_net* <**high** | **low** | **posedge** | **negedge**>} **-restore_signal** {*logic_net* <**high** | **low** | **posedge** | **negedge**>}] [**-save_condition** {*boolean_expression*}] [**-restore_condition** {*boolean_expression*}]
  [**-retention_condition** {*boolean_expression*}] [**-use_retention_as_primary**] [**-parameters** {<**RET_SUP_COR** | **NO_RET_SUP_COR** | **SAV_RES_COR** | **NO_SAV_RES_COR**> *}] [**-instance** {{*instance_name* [*signal_name*]}*}] [**-update**] [**-retention_power_net** *net_name*] [**-retention_ground_net** *net_name*]

- **Map_retention_cell:** It will choose which retention strategy should be implemented.

  map_retention_cell *retention_name_list* **-domain** *domain_name* [**-elements** *element_list*] [**-exclude_elements** *exclude_list*] [**-lib_cells** *lib_cell_list*] [**-lib_cell_type** *lib_cell_type*] [**-lib_model_name** *name* **-port_map** {{*port_name net_ref*} *}]

- **set_port_attributes:** It defines the information to the port of the model and instance.

set_port_attributes [**-model** *name*] [**-elements** *element_list*] [**-exclude_elements** *element_exclude_list*] [**-ports** *port_list*] [**-exclude_ports** *port_exclude_list*] [**-applies_to** <**inputs** | **outputs** | **inouts** | {<**inputs** | **outputs** | **inouts** >*****}>] [**-attribute** {*name value*}]* [**-clamp_value** <**0** | **1** | **any** | **Z** | **latch** | *value*>] [**-sink_off_clamp** <**0** | **1** | **any** | **Z** | **latch** | *value*>] [**-source_off_clamp** <**0** | **1** | **any** | **Z** | **latch** | *value*>][**-driver_supply** *supply_set_ref*] [**-receiver_supply** *supply_set_ref*] [**-literal_supply** *supply_set_ref*] [**-pg_type** *pg_type_value*] [**-related_power_port** *supply_port_name*] [**-related_ground_port** *supply_port_name*] [**-related_bias_ports** *supply_port_name_list*] [**-feedthrough**] [**-unconnected**] [**-is_analog**] [**-is_isolated**]

Chapter 4

4 Steps in Low Power Verification Run Flow and Debugging.

4.1 Stages performed in low power verification

There are three stages in Low Power verification that should be checked and performed.

1)Design/UPF creation

2)Post-Synthesis

3)Post-Route

1) **Design/ UPF creation:** At this stage, UPF is created and written but it hasn't inserted low power management special cells such as isolation cell gates, level shifter cell gates, or PG connection checks so that in later stages this type of issue will not occur. For example, if the object is referred to UPF but absent in design and if the driver is off and the receiver is on such type of issue is noted like Isolation Strategy is missing.

2) **Post-Synthesis:** At this stage, all before checks are performed and UPF is also stable at this point but any change in design will cause an error in the new UPF. So, to solve the problem change in the UPF is acceptable. At this stage, all cells might be inserted but the connection of power and ground are remaining and yet to perform. Check in the electrical correctness can be done at this post-synthesis stage. For example, the level shifter is needed in the crossing between the two domains, but the level shifter is missing there and the need for the isolation gate at the required position is there but the isolation control which is assigned to the isolation cell is not specified in the UPF.

3) **Post-Route:** At this stage, all previous checks are performed and ensure all low power objects, cells, and Power/Ground connection, the problem might occur, and they can be fixed by a change in UPF file and Synthesis result. All electrical correctness is checked on

the Power/Ground connection to be sure that it is consistent with UPF. For example, at the crossing level shifter is present but the input pin is connected to the supply, but it's not mentioned as the same value in the UPF file. Suppose the power pin of the macro is connected to the supply power net, but the supply connection is not matched with the supply in the UPF file.

4.2 Error violations using GUI

4.2.1 UPF Stage Messages

1. Isolation Strategy is Missing
2. Isolation Strategy Supply

4.2.2 Design Stage Messages

1. Analog Net is Incorrect
2. Supply of Design is Shorted

4.2.3 Power/Ground Stage Messages

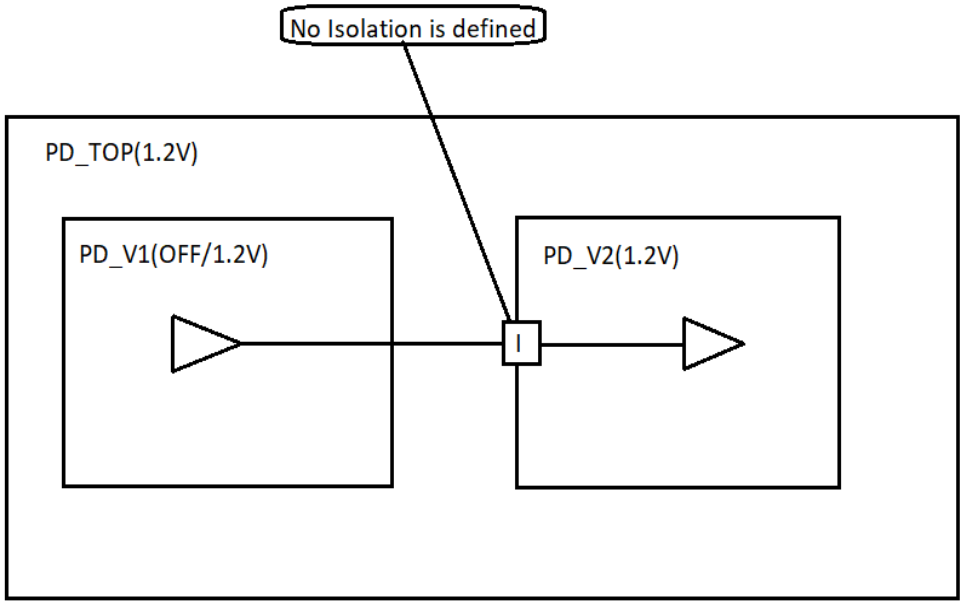
1. Power/Ground Data Supply

4.2.1 UPF Stage Messages

1. Isolation Strategy is Missing (UPF Stage)

Error: This error occurs when isolation is required at the crossing between the domains according to Power State Table specified in the UPF.

How to Resolve: It can be resolved by verifying the power states which are defined for the power supplies associated with the source and sink. If it is correct, then must protect that cross-over by writing the strategy.



	PD_TOP	PD_V1	PD_V2
S1	1.2V	1.2V	1.2V
S2	1.2V	OFF	1.2V

Fig 4.2.1.1 Isolation Strategy is Missing (UPF Stage)

2. Isolation Strategy Supply (UPF Stage)

Error: This error occurs according to Power State Table, isolation cell is inserted at the crossover but is 'ON' for less time compared to source and sink supply.

How to Resolve: It can be resolved, by powering 'ON' the Isolation as long as the power of Source and Sink.

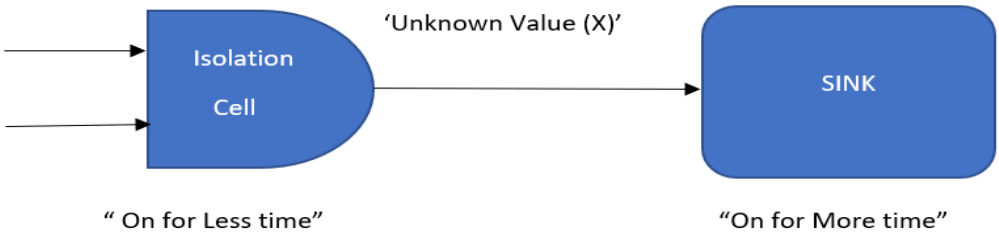


Fig 4.2.1.2 Isolation Strategy Supply (UPF Stage)

4.2.2 Design Stage Messages

1. Analog net is Incorrect (Design Stage)

Error: This error indicates, the net of the design is connected to both an analog pin as well as a digital pin.

How to Resolve: It can be resolved by connecting the design net to either an analog pin or a digital pin.

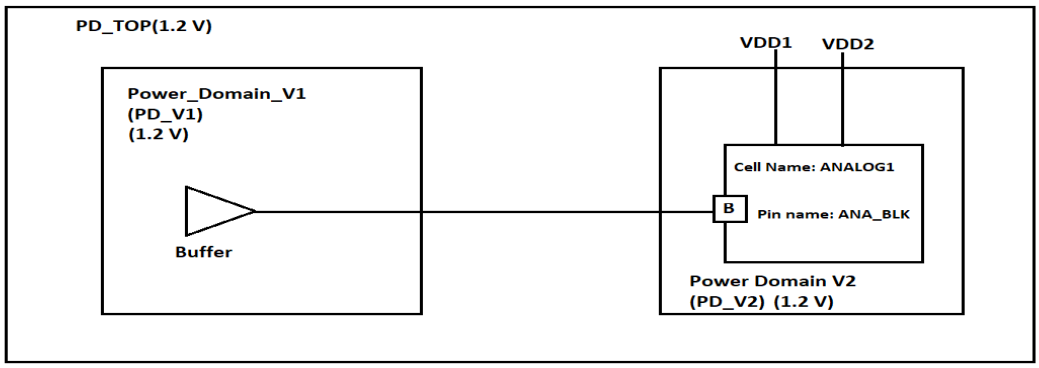


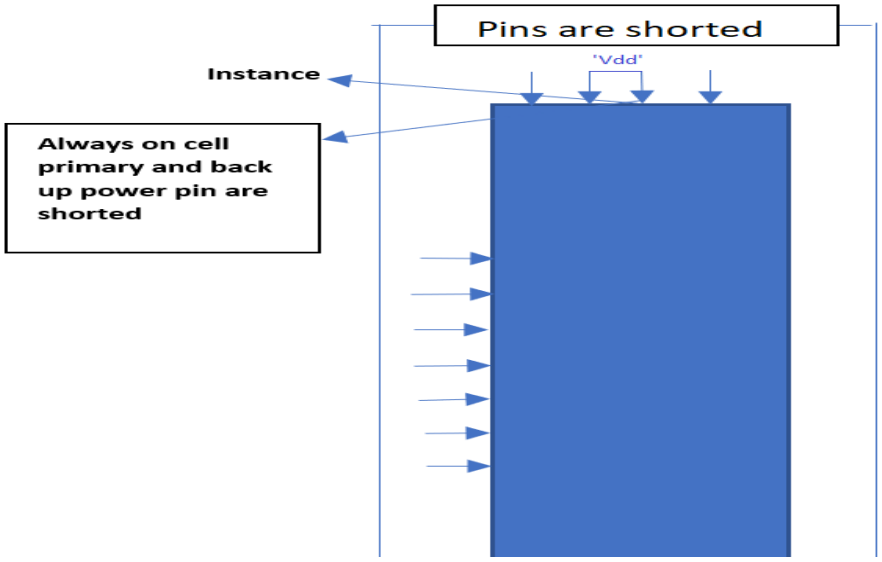
Fig 4.2.2.1 Analog net is Incorrect (Design Stage)

2. Supply of Design is Shorted (Design Stage)

Error: This error occurs when power and ground nets are shorted.(i.e primary pins and internal pins are shorted in power switch cell; primary and back-up power are shorted in AON cells)

How to Resolve:

This error can be fixed by connecting the shorts to different appropriate supplies.



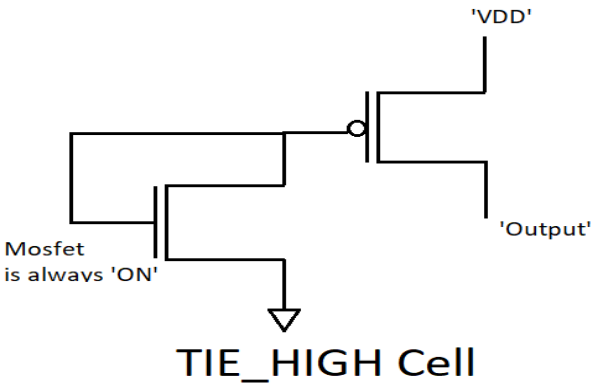
4.2.2.2 Supply of Design is Shorted (Design Stage)

4.2.3 Power/Ground Stage Messages

1. PG Data Supply (PG Stage)

Error: This error shows that the logic pin is connected to the design supply.

How to Resolve: Logic pins should be connected to the design supply via TIE HIGH or TIE Low



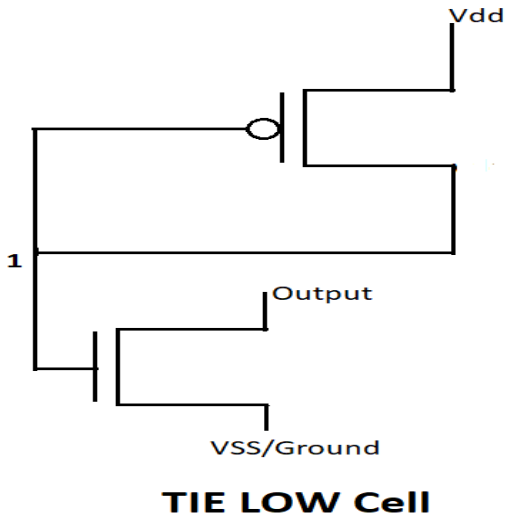


Fig 4.2.3.1 Tie High and Tie Low Cell

Chapter 5

5 Results and Discussion:

In our design, performance and process of many tasks are done using child partitions and top partitions, for the proper working of this partition we generally perform different kinds of checks as shown in fig 2.1(a) low power verification check on UPF file and netlist at different stages as shown in Fig 3(a). Child partitions are added to the top partition using the load_upf command. At this stage debugging process starts. The low Power verification tool will give a very simple report to understand the issue in the power distribution design and various files in it. Two input files are required in a tool for debugging the first one is the gate-level netlist file which shows the physicality of the functioning of power design connectivity and the UPF file for logical connections. Various cells perform various function as shown in section 2.3. In the report, Management summary and Tree summary violations help to resolve and reduce issues in low power design. The management_summary and tree_summary files are generated in the partition uncompressed file. One can find this violation summary using some commands which are shown below like read_upf, and report_violation. As the design and low power complexity are increased rapidly. More signoff functional check is performed on design to fulfill the requirement. Large report volume is generated by manual checking using TCL, GUI, and UPF/Design debugging will be very slow. The Low Power Verification tool helps to reduce the debugging time by a smart grouping of the violation and shows the exact main cause so that time can be saved. A similar violation is generated in an uncompressed file that can be compressed and it is shown in the partition compressed file. In management_summary it shows violation and error using checking stages while tree_summary shows its result using severity checks.

Management Summary				
Stage	**Family**	**Errors**	**Warnings**	**Infos**
UPF	Isolation	0	1330	0
UPF	Upfconsistency	0	2	0
Design	Analog	76	0	0
Design	DesignConsistency	9	0	0
Design	Isolation	0	3	0
PG	PowerGround	0	3	0
Total		**85**	**1338**	**0**

Table no 5(a) Management Summary of the uncompressed design in Top partition with various family stage violations.

Tree Summary			
Severity	**Stage**	**Tag**	**Count**
Error	Design	---------	
Error	Design	---------	
warning	UPF	---------	
warning	UPF	---------	
warning	UPF	--------	
warning	Design	---------	
warning	PG	---------	
Total			

Fig 5(b) Tree Summary of the uncompressed design in top Partition with error, warnings and info.

The command report_upf will show the design top name which is the main project. It generates the power intent information after it reads the UPF file. By report_upf command, one can find how many isolation instances, a level shifter, retention, power switch, and multi-rail macros instance are there in the UPF file similarly by using report_design one can get knowledge about the design and library file.

Read upf	
Design top:	Design top name
Isolation instances:	The total number of isolation instances used
Level Shifter Instances:	The total number of Level Shifter instances used
Retention instances:	The total number of retention instances used
Power Switch instances:	Total number of Power switch instances used
Multirail macro Instances:	The total number of Multi rail macro instances used
Total Instances:	Total number of all the instance
Crossovers:	Total number of crossovers
Merged power states:	The number of power states merged

Table 5(c) Report_upf generates power intent information

The **report_read_violation** command is used to get the report of all SDC, design read, and UPF violations same as GUI violations. The Low Power verification tool shows the source and Sink destination in GUI format which can be very flexible to debug. By using the command report_read_violation in the Low Power verification shell one can see the summary view of the message.

Management Summary				
Stage	**Family**	**Errors**	**Warnings**	**Infos**
DESIGN_READ	**DESIGN**	**0**	**142511**	**0**
DESIGN_READ	**UPF**	**0**	**0**	**10**
TCL	**SETUP**	**492**	**0**	**0**
Total			**142511**	**10**

Tree Summary			
Severity	**Stage**	**Tag**	**Count**
Error	**TCL**	**----------**	**2**
Error	**TCL**	**----------**	**490**
Warning	**DESIGN_READ**	**----------**	**1**
Warning	**DESIGN_READ**	**----------**	**105209**
Warning	**DESIGN_READ**	**----------**	**37301**
info	**DESIGN_READ**	**----------**	**1**
info	**DESIGN_READ**	**----------**	**5**
info	**DESIGN_READ**	**----------**	**3**
info	**DESIGN_READ**	**----------**	**1**
Total			**143013**

Fig 5(d) Report read violation read stages like SDC, UPF parsing, TCL, design read.

As shown in fig 4.2.2, Here is this GUI representation of the error supply short, the figure shows an arrow to the pin with specified ‘instance’, here it shows one pin but in our design, there are numerous pins of different instances which are shorted with each other. Supply short is a here serious electrical violation which is between power and ground nets. There are some cases where

different kinds of power pins (i.e internal power pins and power pins of power switch cell) are shorted, always-on cell pins like primary and backup power pins are shorted which is a kind of redundancy in the design.

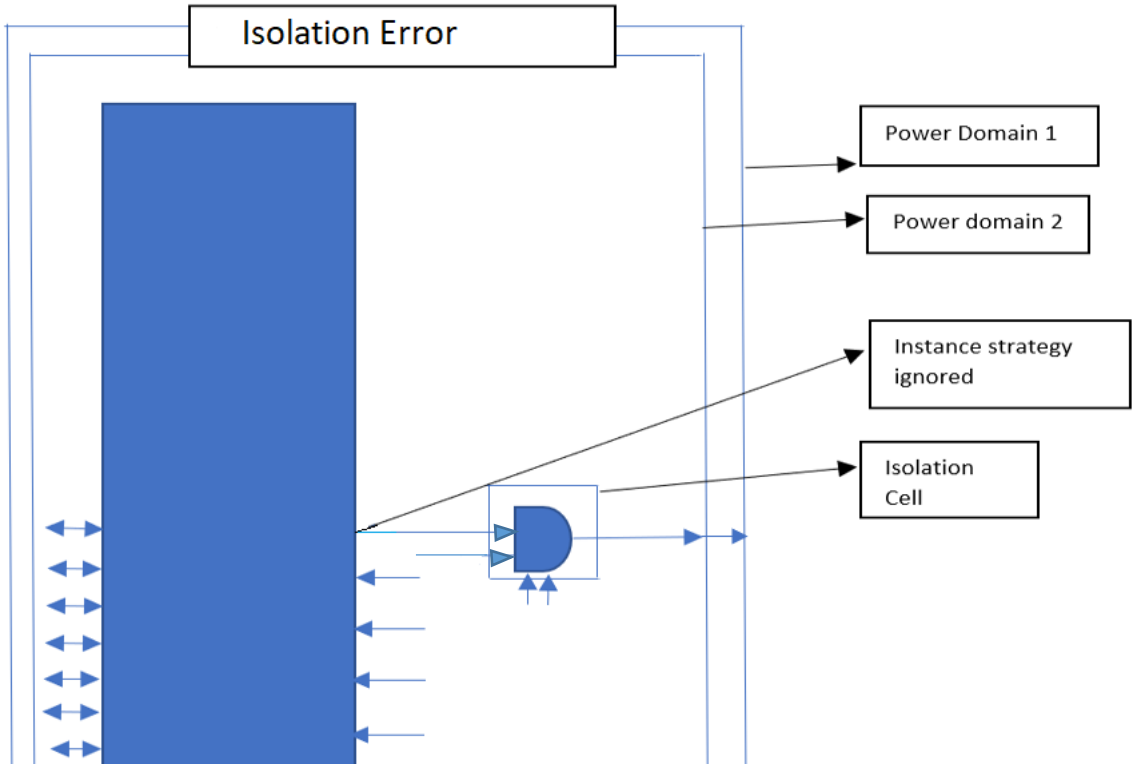


Fig 5(f) Shows GUI Representation of the error for the isolation strategy

As shown in fig 5(f), the Isolation Strategy is ignored in GUI representation by debugging through Low power verification shows an arrow where it shows the pin where the Isolation strategy at the crossing between two power domains will be ignored during checks.BY color in GUI represents two separate power domains.

Chapter 6

6 Conclusion and Future Scope

In this low-power era, it is very difficult to find manually each low-power design issue. So, Low Power verification static Low Power verification signoff includes more than 500 violation checks in less time and Low power verification tools will reduce UPF complexity, machine usage, and run time is very less. It can handle large databases and it also supports GUI. Low Power verification helps us to identify and report low-power issues and provide a bug-free design.

References

Books

[1] "IEEE Standard for Design and Verification of Low-Power, Energy Aware Electronic Systems," Revision of IEEE Std 1801-2013.

[2] Synopsys, "Verification Continuum VC Static Platform Command Reference Guide," Version R-2020.12, December 2020

[3] Synopsys, "Verification Continuum VC LP User Guide," Version R-2020.12, December 2020.

[4] Synopsys, "VC LP Message Reference Guide," Version M-2017.03, March 2017.

[5] Gary K. Yeap, "Practical Low Power Digital VLSI Design"

Journals

[1] Jinal Patel, Ekta Chotai, Naincy Desai, Hemang Patel, "Low Power Design Flow using power Format" Department of electronics GTU, Ahmedabad, Gujarat, India. pp 238-240.

[2] Lie Shaotao Debajani Majhi, "UPF-Based Static Low-Power Verification in Complex Power Structure SoC Design Using VCLP"Irvine, US. pp 4-19.

[3] NAVEEN KUMAR CHALLA, USHA RANI NELAKUDITI, "THE NEW ERA ON LOW POWER DESIGN AND VERIFICATION METHODOLOGY", Department of Electronics and Communications Engineering, Vignan's Foundation for Science, Technology and Research University, Guntur, Andhra Pradesh, India. pp 160-165.

[4] K.R.HARIPRIYA, AJAY SOMKUWAR, LAXMI KUMRE, "Low Power Checks in Multi Voltage Designs" , VLSI & Embedded system design MANIT, Bhopal, INDIA pp 105-111

Conference

[1] Venkatesh Gourisetty[(1)], Hamid Mahmoodi[(1)], Vazgen Melikyan[(2)(3)], Eduard Babayan[(2)], Rich Goldman[(4)], Katie Holcomb[(4)], Troy Wood[(4)], "Low Power Design Flow Based on Unified Power Format and Synopsys Tool Chain", (1) School of Engineering, San Francisco State University, San Francisco, CA, USA, (2) Synopsys Armenia CJSC, Yerevan, Republic of Armenia, (3) State Engineering University of Armenia, Yerevan, Republic of Armenia (4) Synopsys Inc., Mountain View, CA, USA pp. 28-31.

[2] Chao Liang, "Mixed-Signal Verification Methods for Multi-Power Mixed-SignalSystem-on-Chip (SoC) Design" 2013 IEEE 10th International Conference on ASIC.