# Low Power Verification of Data Manipulation IP using UPF Methodology

Submitted in partial fulfillment of the

requirements for the degree of

**Master of Technology**

in

**Electronics and Communication Engineering**

**(VLSI Design)**


Submitted By

**Jatinkumar Koshiya**

**18MECV08**




**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**INSTITUTE OF TECHNOLOGY**

**NIRMA UNIVERSITY**

**AHMEDABAD-382481**


**May 2020**

# Certificate

This is to certify that the M.Tech thesis entitled **"Low Power Verification of Data Manipulation IP using UPF Methodology"** submitted by **Jatinkumar Koshiya (Roll No: 18MECV08)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Electronics and Communication (VLSI Design) of 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 M.Tech thesis, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.


Prof. Jayeshkumar Patel
Guide and Assistant Professor
Institute of Technology,
Nirma University, Ahmedabad

Dr. Usha S Mehta
PG Coordinator M.Tech - VLSI
Institute of Technology,
Nirma University, Ahmedabad

Dr. Dhaval Pujara
HOD and Professor
Institute of Technology,
Nirma University, Ahmedabad

Dr. Alka Mahajan
Director
Institute of Technology,
Nirma University, Ahmedabad

# External Certificate

This is to certify that the Major Project Report entitled **"Low Power Verification of Data Manipulation IP using UPF Methodology"** submitted by **Jatinkumar Koshiya** (Roll No. **18MECV08**) as the partial fulfillment of the requirements for the award of the degree of Master of Technology in VLSI Design, Electronics and 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:

Place: Bangalore

Kiran C

Engineering Manager

Intel Corporation

# Statement of Originality

I, **Jatinkumar Koshiya**, Roll No: **18MECV08**, give undertaking that the M.Tech thesis entitled "**Low Power Verification of Data Manipulation IP using UPF Methodology**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Electronics & Communication Engineering (VLSI Design)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student

Date:

Place: Ahmedabad

Endorsed by

Prof. Jayeshkumar Patel

(Signature of Guide)

# Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Jayeshkumar Patel**, Assistant Professor, EC Department, Institute of Technology, Nirma University for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me a great pleasure to thank **Dr. Usha Mehta**, PG Coordinator of VLSI Design, EC Department, Institute of Technology, Nirma university for his guidance throughout whole M.Tech and in this project thesis.

It gives me an immense pleasure to thank **Dr. Dhaval Pujara**, Hon'ble Head of EC Department, Institute of Technology, Nirma University for his kind support and providing basic infrastructure and healthy research environment.

I would also like to thank my Intel colleague **Amit Kumar Awasthy** (Pre-Si Verif Engineer), **Ashwini Khandekar** (IP Logic Design Engineer) and **Kiran C** (Engineering Manager) for their continuous support and guidance throughout my M.Tech thesis project at Intel Corporation.

I would also thank the Institution, all faculty members of Electronics and Communication Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

**Jatinkumar Koshiya**

**18MECV08**

# Abstract

A decade ago, the major challenges of semiconductor device design were performance and area. Designers used to aim at achieving the required target speed in as minimum chip area as possible. Power was not of major concern those days. With the increase in the demand for high performance and high-speed VLSI systems has shifted the focus from traditional performance parameters towards the analysis of power consumption. The power budget and management among the domains of a system is of real concern. Hence, the power aware design using clock gating, power gating, dynamic voltage scaling and frequency scaling are the most used design techniques. Employing such low power techniques at the RTL creates new design and verification challenges. The challenges are: how can one domain be power downed, how can it be put back to power up state and do they retain or restore the previously computed data and still function correctly. These can be answered and implemented using new methods of implementation and verification using the unified power format (UPF) standards for low power intent designs.

UPF is an IEEE 1801 standard format to describe the power architecture, also called as power intent, including power network connectivity and power reduction methods. It enables verification of power intent at early phases of the design cycle. The UPF developed should be consistent with the design at all stages of the design cycle and it should be updated according to the modifications made in the design.

This thesis covers power management concepts like power domains, isolation, retention, level shifter and its specification described using UPF. Normal RTL design is simulated using Synopsys VCS and synthesis is done using Synopsys Design Compiler. Low power verification is done using VCS NLP (Native Lower Power) and synthesis is done using Synopsys Design Compiler. I have also created library file (.lib) and compiled using Synopsys Library Compiler for synthesis of RTL design with UPF. At last we have compared the results of normal and low power synthesis and simulation. Based on low power verification, in actual IP all the functional test cases are enable with low power simulation mode and check the functionality.

# Contents

# List of Figures

# Chapter 1

# Introduction

A decade ago, the major challenges of semiconductor device design were performance and area. Designers used to aim at achieving the required target speed in as minimum chip area as possible. Power was not of major concern those days. With the increase in the demand for high performance and high speed VLSI systems such as network processors in networking or SOCs in communication and computing has shifted the focus from traditional performance parameters towards the analysis of power consumption. Power has become one of the major challenges of semiconductor design along with performance and area. The power budget and management among the domains of a system is of real concern. Hence, the power aware design using clock gating, power gating, dynamic voltage scaling and frequency scaling are the most used design techniques. Employing such low power techniques at the RTL creates new design and verification challenges. The challenges are: how can one domain be power downed, how can it be put back to power up state and do they retain or restore the previously computed data and still function correctly. These can be answered and implemented using new methods of implementation and verification using the Unified Power Format (UPF) standards for low power intent designs.

In this chapter, the need to low power design, power management concepts, the need for defining power intent through a standard format like Unified Power Format (UPF) and challenges faced in taking it through design cycle are discussed along with a brief introduction of this thesis organization.

## 1.1 The Need for Low Power Design

A few years back, major challenges in the semiconductor design were area and performance. Semiconductor industry were increasing speed as much as possible and reducing the chip area to sustain in market.With increasing in the demand for high speed with less area of portable electronics like Cell Phones, Smart Phones, Tablet, Laptops etc industry started move towards the power performance as major concern. Every new generation is expected to have more and more feature and longer battery life. To satisfy this requirements, we need active power management. As per the figure 1, active power management is becoming important at 90 nm and it is necessary at below 65 nm.

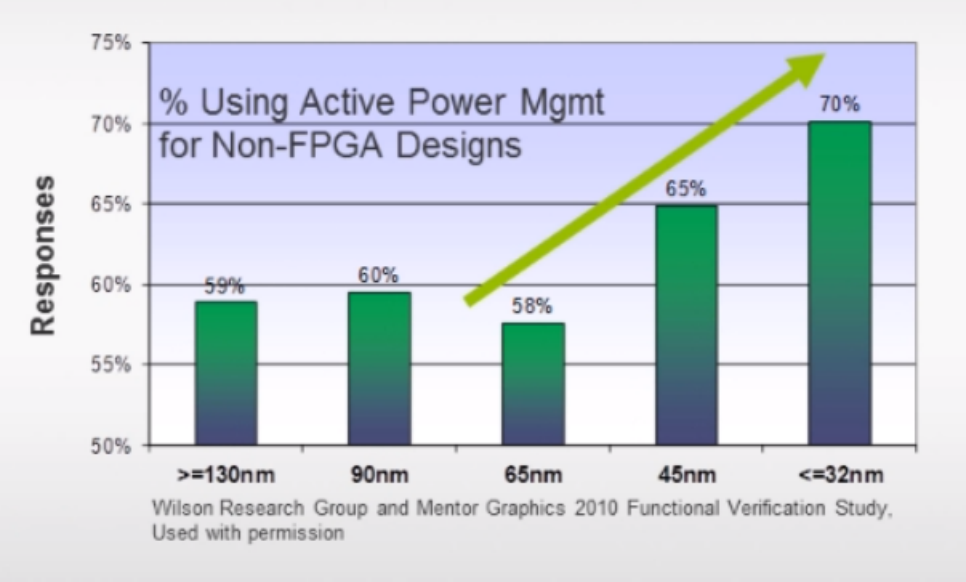


Figure 1.1: Active Power Management Requirement

As we all know that about the basics of power consumption and its types and we have also shown it in figure 2. There are majorly two types of power and that is

**1) Static Power**

**2) Dynamic Power**

$$P_{total} = P_{static} + P_{dynamic}$$

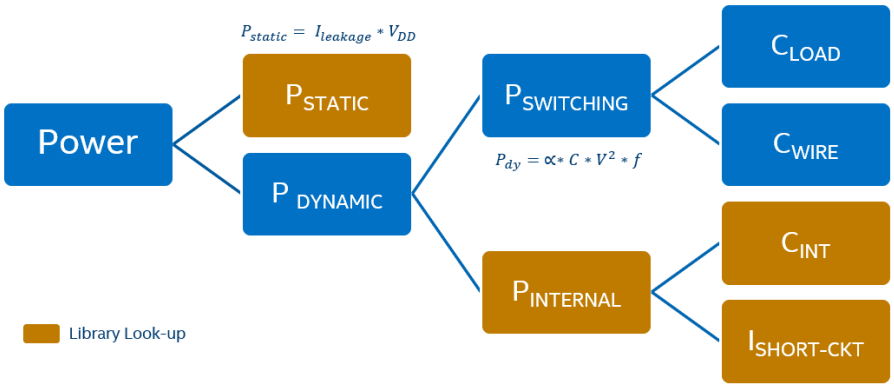


Figure 1.2: Power Calculation

The static power is because of leakage of current when the circuit is at steady state and it depends on device technology node. The potential sources of leakage current is majorly because of sub threshold current, gate leakage current and reversed biased junction leakage. It is look up from the energy table in the library file. The basic equation is shown below:

$$P_{static} = I_{leakage} * V_{DD}$$

The dynamic power comprised of two component: one is internal power and it is because of short circuit current of device and second component is because of switching of device. The switching power is the major part of power consumption. The basic switching power equation is shown below:

$$P_{switching} = \alpha * C_L * V_{DD}^2 * f$$

Here, in this the load capacitance is calculated from library file, Vdd is the supplied voltage, f is the frequency at which the device is running and Alpha is the activity factor which is calculated based on the activity of signals which will be less than 1.

As per the figure 3, when we move towards the lower technology node, leakage power is increased more and more compare to active power. It is become necessary below 65 nm to require active power management.

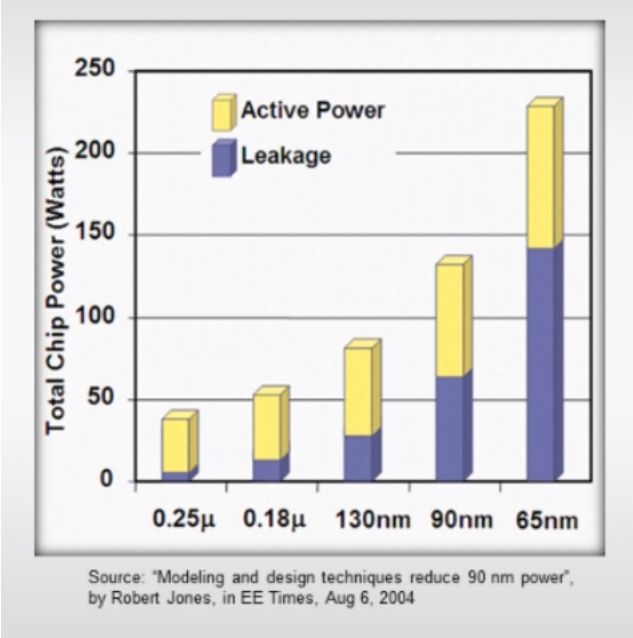


Figure 1.3: Active and Leakage Power at Lower Technology Node

We need active power management so that it can control the power reduction techniques to minimize the power consumption, especially leakage power. In the next section, we will look more on power management concepts like power domains, isolation, retention and level shifting in details.

## 1.2 Power Management Concepts

The power management scheme is built by the power architect of the SoC in order to reduce the current consumption of the chip. There are several common buildings blocks which are used by the power management architect like power domains, isolation, retention, level shifting will be described here.

### 1.2.1 Power Domains

In many low power designs, parts of the design have a switchable supply. Those parts are called switchable domains. The power-management module is the one that controls the power-up and power-down sequences. These are the independently powered regions. It toggles the control signals to each of the analog switches according to the power sequence and allows current to relevant power-domains by closing analog switches. The main idea of using power switches is to turn off massive unused parts of the design and, as a result,

gain low current consumption. Power domain enables the application of different power reduction techniques in each regions.

### 1.2.2 State Retention

Memories can be switched on and off in order to reduce their current consumption. Memories are shut-down when their IP or power-domain is in shut-down mode or when they are not in use. Some memories need to retain their values for fast wake-up. For these memories, only the memory array stays powered on using a low power mode, and the peripheral interfaces are powered off. The retention mode consumes a little more current than power-off mode but allows fast recovery of the memory content after waking from sleep mode. In figure 1.4, we have shown the retention cell that are used in power intent specifications.

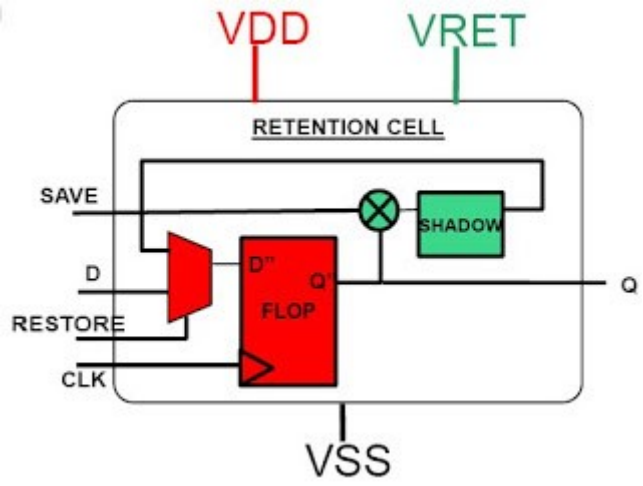


Figure 1.4: Retention Cell

A register's power is turned off when its power-domain or IP is turned off. Some registers also need to retain their values after being turned off. These are called retention registers, which restore their previous active value after being shut down. These registers are important for fast wake-up.

### 1.2.3 Isolation Cell

Isolation is typically used to isolate signals originating in a design element whose power is shut off from a part of the design which still remains powered on and able to read those signals.

An isolation cell is needed when a signal comes from a switchable power domain, to any other powered-on domain. This cell prevents a floating value from propagating. When its control signal is enabled, the isolation cell outputs a fixed logic value, 0 or 1. When its input and output sides are both powered on, the isolation cell behaves as a buffer. To define an isolation strategy, we need to specify the power domain where this cell is inserted, its source and sink, its output value, its power supply, its control signal and the elements impacted by this cell.

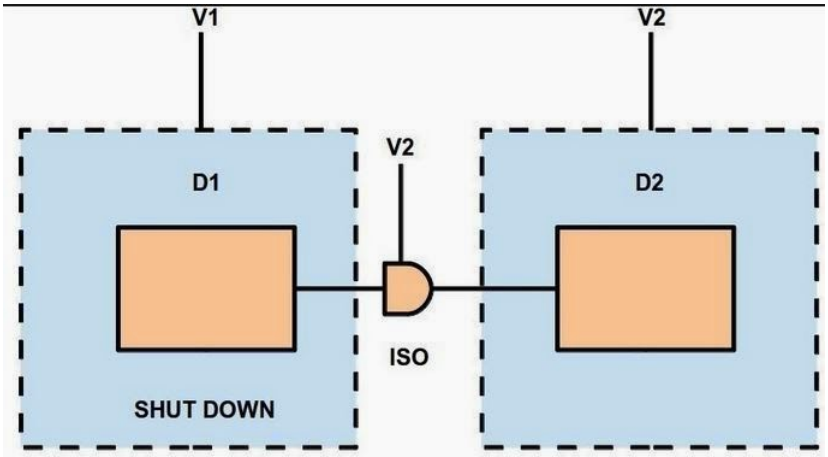


Figure 1.5: Isolation Cell

Isolation strategies means adding the isolation cells either at the input, output or both so that a particular design is isolated from the rest of the design. Isolation cells are logic gates that determine the values of a power domain's input or output port when the domain is powered down . Isolation cells are necessary because each power domain represents a design area comprised of particular features, and each feature corresponds to an area of physical silicon. Even though they may represent different power domains that can be independently powered ON and OFF, these areas of silicon remain physically connected. Therefore, when one domain is turned OFF, it is still connected electrically to other domains.

### 1.2.4 Level Shifter

A level shifter is needed when a signal crosses two power domains which do not have the same supply voltage. This component converts a signal from one voltage to another. Thereby, a logic value is seen as an unambiguous value by the domains the related signal

might cross . To define a level shifter rategy, we need to specify in which power domain the level shifter is inserted and which signals are impacted by the level shifter.

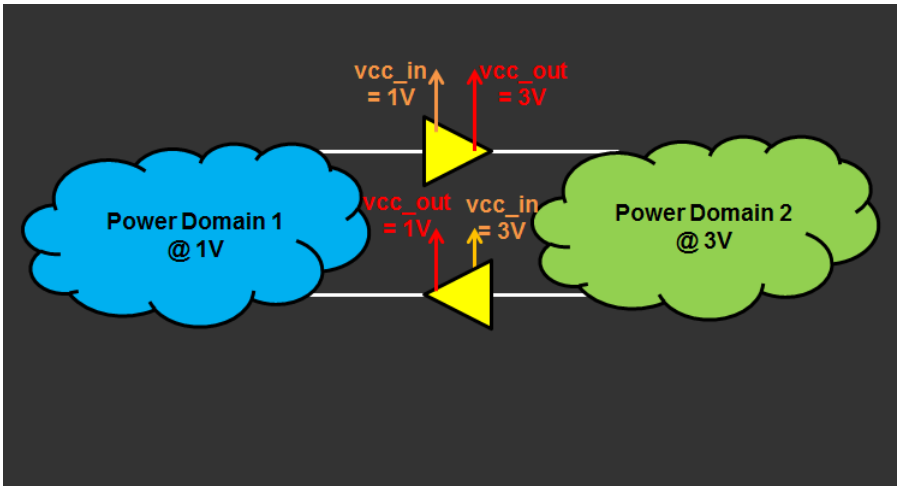


Figure 1.6: Level Shifter Cell

As per the shown in figure 1.6, there are two power domains and level shifter is added in between these domains. These all the power management concepts are included and shown in figure 1.7.

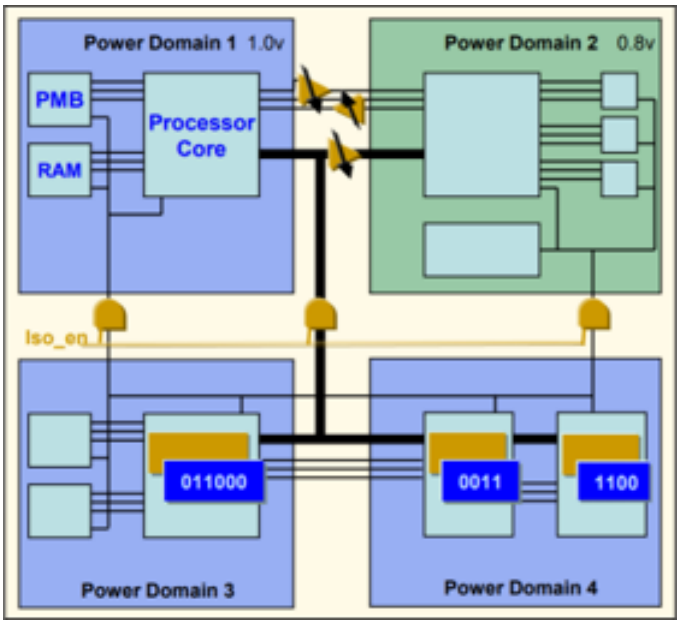


Figure 1.7: Power Management Concepts

## 1.3 Normal and Low Power Design-Verification Flow

In this section, we will discuss about normal ASIC design flow and where we are considering power implementation in this flow. We will also see the low power design and verification flow by adding power intent specification using Unified Power Format (UPF).

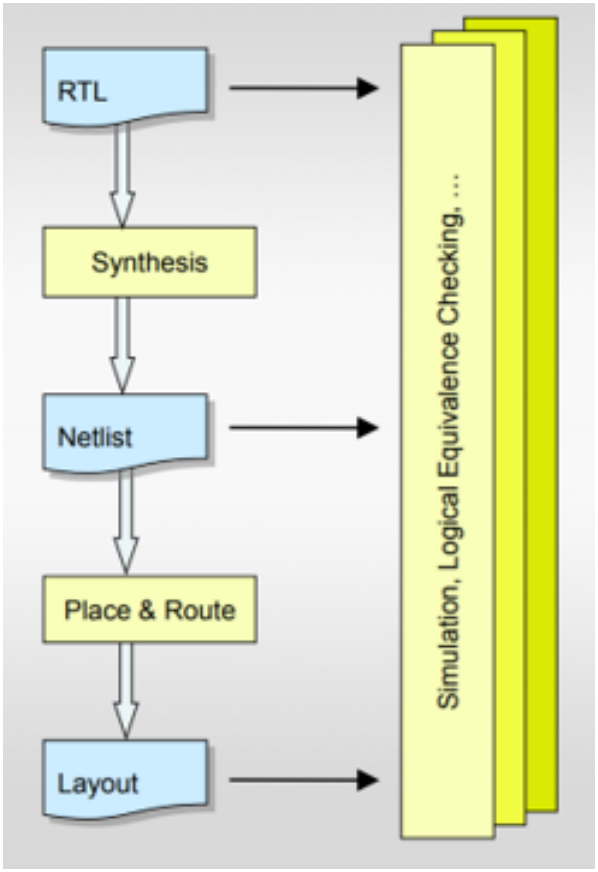


Figure 1.8: Normal Design and Verification Flow

As per the figure 1.8, we have shown normal design flow. Here, RTL design captures the design intent which can be described using hardware description language like Verilog, System Verilog, etc. RTL design also drives the functional verification and it is the main input for driving synthesis.

Logic Implementation is done using standard cell and macros. RTL design and standard cell library is used for logic implementation and after synthesis we get the netlist. We can also do logic equivalence checking of RTL design and generated netlist.

After the logic implementation, place and route completes the physical implementation of the design. This step produces the manufacturing data. In Normal design flow, power management is considered at this physical implementation stage but for low power design flow it should be considered at earlier stage of the design flow.

Low power intent specification specified using Unified Power Format (UPF). UPF is IEEE standard format use to define power management and to minimize power consumption, especially leakage power. It is based upon TCL language and can be mixed with Non - UPF TCL. UPF can be used for verification and implementation purpose like synthesis, DFT, P&R etc.

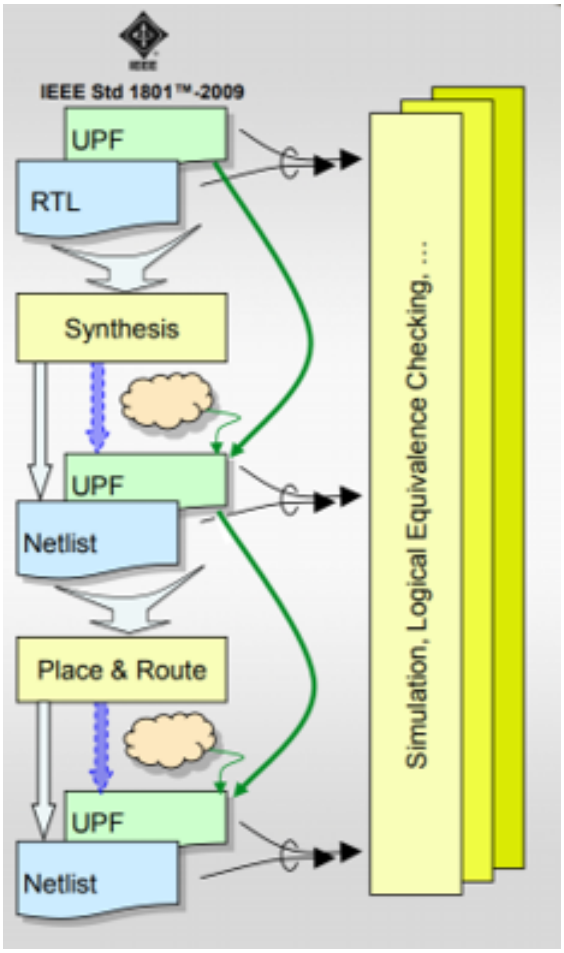


Figure 1.9: Low Power Design and Verification Flow

As per the shown in figure 1.9, RTL is augmented with UPF to define power architecture for a given implementation. By doing RTL + UPF verification, we can ensure that Power Architecture is complete and consistent with expected power states of the design.

We can also ensures that design will work correctly under power management with this power architecture. At every stage we can update the UPF and we can implement it for synthesis purpose.

## 1.4 Organization of the Thesis

**Chapter 2** consists of normal RTL with the power architecture which is defined using Unified Power Format (UPF). In this chapter, we have also added all the PM cells like Isolation, Level Shifter and Power Switches.

**Chapter 3** consists of explanation of the tools requires for simulation and for synthesis like Design Compiler Synopsys tool. We have also created library and compiled using Library Compiler.

**Chapter 4** consists of results of Normal design simulation results and synthesis of the design. We have also compared the results with the NLP simulation results and synthesis netlist with power architecture.

At last we have also shown the conclusion of thesis and future scope of this project. In the Last section, we have added references which are used for literature survey and thesis creation.

# Chapter 2

# Power Management Architecture with Example

In this chapter, We have taken one simple design with main module as design and inside this we have three different module P1, P2 and P3. This design figure is shown in figure 2.1. We have applied the power management architecture on this design described using Unified Power Format (UPF). The detailed PM block details are described in next sections.

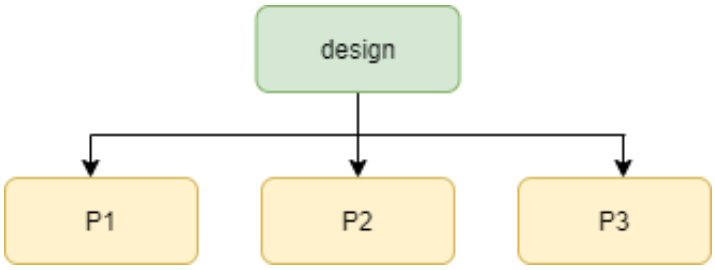


Figure 2.1: Normal HDL Design

## 2.1 Power Domains

A power domain is a collection of design elements that share a primary power and ground supply net. The logic hierarchy level where a power domain is created is called the scope of the power domain. Any design elements that belong to a power domain are said to be in the extent of that power domain. A design element can be in the scope of a number of power domains, it can only be in the extent of exactly one power domain.

To help manage the complexity of the supply network specification, power domains are defined to group elements from the logic hierarchy that share common supply needs. By default, all logic elements in a power domain use the same primary supply. Additional supplies may be defined to serve different uses in a power domain.

```
set_scope design
   create_power_domain PD_TOP -include_scope
   create_power_domain PD_COMP -elements {P2}
   create_power_domain PD_MOD -elements {P3}
```

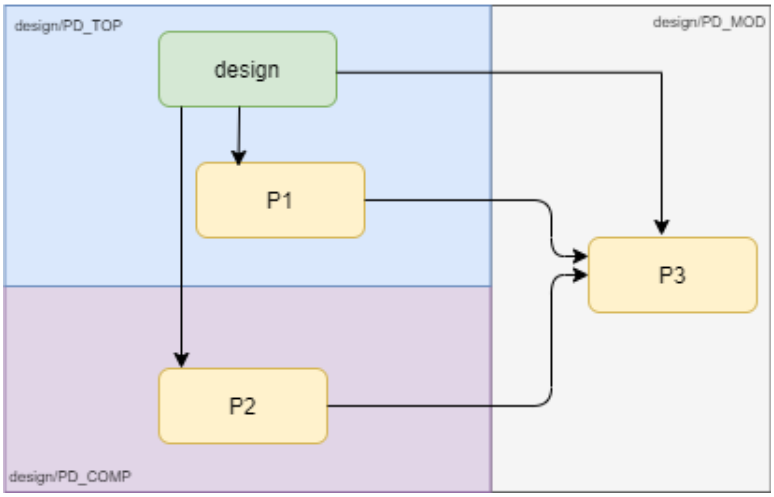


Figure 2.2: Different Power Domains in HDL Design

In this code snippet, we have created three power domain PD_TOP, PD_COMP and PD_MOD using create_power_domain UPF command. The visualization of code in design is shown in figure 2.2.

## 2.2 Supply Networks and Switching

A Power Network consists of supply ports, supply nets, supply sets and power switches. It represents the power supply of the design at a high abstraction level. A supply port is a power supply connection. A supply net can be either a power or ground net connected to a supply port. It crosses different levels of the design hierarchy. Each switchable domain needs a power switch to be turned on and off. The output supply net of a power switch must also be created.

- **create_supply_port**

```
create_supply_port VDD1 -direction in
create_supply_port VSS  -direction in
```

- **create_supply_net**

```
create_supply_net Pwr1 -domain PD_TOP
create_supply_net Gnd  -domain PD_TOP
```

- **connect_supply_net**

```
connect_supply_net Pwr1 -ports {VDD1}
connect_supply_net Gnd  -ports {VSS}
```

- **set_domain_supply_net**

```
set_domain_supply_net PD_TOP \
    -primary_power_net Pwr1 \
    -primary_ground_net Gnd
```

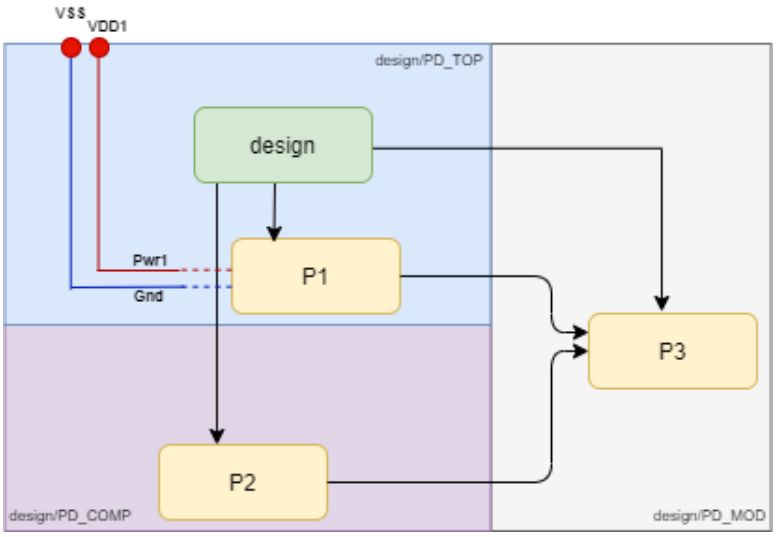


Figure 2.3: Supply Network in Different Power Domain

In power supply network, upper shown snippets will create the power supply port, supply net and connect this ports and nets. Using set_domain_supply_net command, we can set the domain supply net to different modules. Here in module P1 we have set Pwr1 and Gnd net supply. These power supply connections we can visualize in figure 2.3.

- **create_power_switch**

```
create_power_switch SW \
       -domain PD_MOD \
       -output_supply_port {swout VDD1_sw} \
       -input_supply_port {swin Pwr1} \
       -control_port {swctrl swCtl} \
       -on_state {SWon swin swctrl} -off_state {SWoff !swctrl}
```

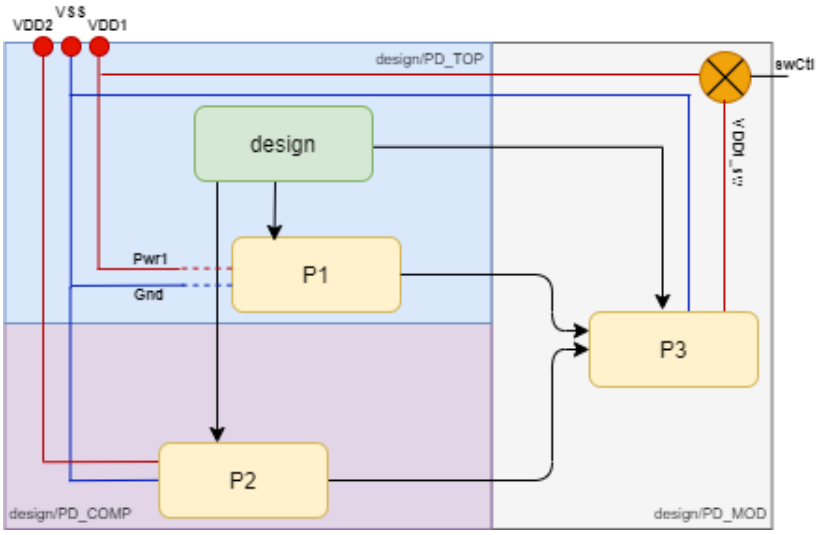


Figure 2.4: Power Switch in Domain P3

Power switch is like a controller, it provide the voltage and shut down power according to the control signal.During simulation, a power switch corresponds to a process that is sensitive to changes in its input port (net state and voltage value), as well as its control ports. Every time the signals on the control ports change, their value is compared with the corresponding on-state Boolean functions. If the value of the control signals match one of the on-state functions, the switch is closed, which causes the on/off state, full/partial state, and voltage value at its input port to propagate to the output port. If the control signals do not match any of the on-state functions, the switch is opened turning off the output port. If any of the control signals is X or Z, or the control signals match one of the error-state Boolean functions, the behavior of the power switch is undefined, in this case implementations may issue a warning or an error. Power switch code is shown in upper snippet and its visualization can be seen in figure 2.4 domain P3.

## 2.3 Supply Port States and Power State Table (PST)

A power state table is used for implementation specifically for synthesis, analysis, and optimization. It defines the legal combinations of states, i.e., those combinations of states that can exist at the same time during operation of the design.

The power state table has no simulation semantics. It is tool dependent whether simulation tools report an error if an illegal (unspecified) combination of states occurs. Each port in the design can have one or more states but only one state at any given time. The port-state and the related voltage value are specified in a Power State Table. The PST also lists all the combinations of states of power switches in the design.

```
add_port_state VDD1 \
      -state {ON_18 1.8}
create_pst DESIGN_PST \
      -supplies {VDD1 VDD2 Vdd1_sw VSS}
add_pst_state -pst DESIGN_PST \
     -state {ON_18 ON_10 ON_18 ON_00}
```

| State/Supply | VDD1 (Port) | VDD2 | Vdd1_sw | VSS |
|---|---|---|---|---|
| State 1 | ON_18 | ON_10 | ON_18 | ON_00 |
| State 2 | ON_18 | OFF | ON_18 | ON_00 |
| State 3 | ON_18 | ON_10 | OFF | ON_00 |
| State 4 | ON_18 | OFF | OFF | ON_00 |

Figure 2.5: Power State Table

As per the code snippet above, we can create the port state using add_port_state command. We have only shown one port state and like this we can create as many port state we want. Now we have different port state for different ports. Using create_pst command we can create the power state table of all the ports and all the combinations are shown in figure 2.5. Using PST, we can able to analyze where to add isolation strategies and where to add level shifting strategies.

## 2.4 Isolation Strategies

Isolation strategies means adding the isolation cells either at the input, output or both so that a particular design is isolated from the rest of the design. Isolation cells are logic gates that determine the values of a power domain's input or output port when the domain is powered down .

- **set_isolation**

```
set_isolation ISO_COMP -domain PD_COMP \
     -applies_to_outputs -clamp_value 0 \
     -isolation_power_net   Pwr1 \
     -isolation_ground_net  Gnd
```

- **set_isolation_control**

```
set_isolation_control ISO_COMP \
     -domain PD_COMP \
     -isolation_signal  iso_en \
     -isolation_sense   high \
     -location self
```

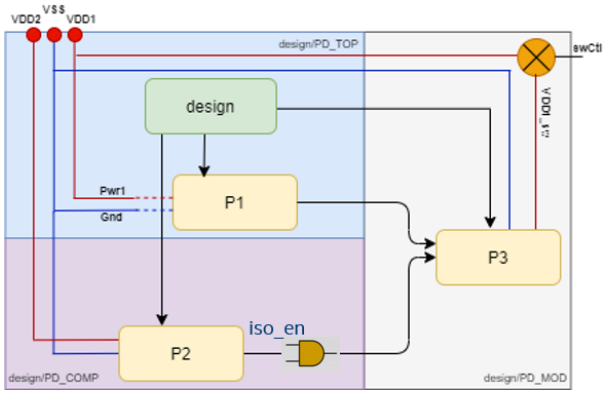


Figure 2.6: Isolation Strategies

As per the code snippet above set_isolation command will set the isolation strategies in PD_COMP domain. We have set the clamp value to 0. The control signal iso_en is added by set_isolation_control command.

## 2.5 Level Shifter Strategies

A level shifter is needed when a signal crosses two power domains which do not have the same supply voltage. This component converts a signal from one voltage to another. Thereby, a logic value is seen as an unambiguous value by the domains the related signal might cross.

```
set_level_shifter LS_COMP \
       -domain PD_COMP \
       -applies_to outputs \
       -rule low_to_high \
       -location self
```

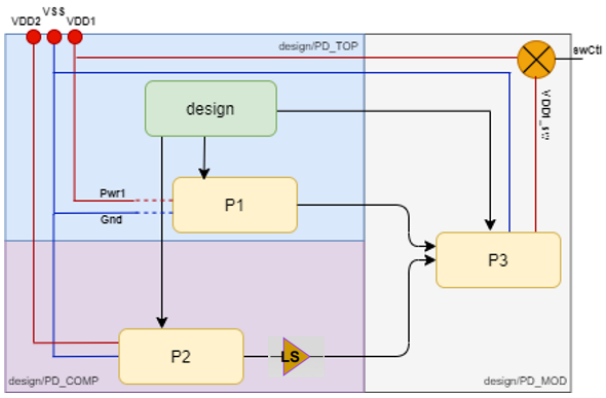


Figure 2.7: Level Shifter Strategies

Level Shifter strategies can be analyzed by looking at the Power State Table in figure 2.5. When we look at the State 1, both the VDD2 and Vdd1_sw are n ON condition. Both these domains are working at different power supply. When output signals from goes from VDD2 domain to Vdd1_sw domain, we need Level Shifter to be placed and it can be places using set_level_shifter command as per code snippet shown above. We can also set which type of LS we want either Low to High or High to Low or Both. The visualization of above code in power domain is shown in figure 2.7.

These all the power domain commands, supply network - switching commands, PST, isolation and level shifting commands are grouped together and made the power intent specification in UPF file of our normal RTL design. The full UPF code of our design is mentioned in the next section.

## 2.6 UPF Code

```
set_design_top design
create_power_domain PD_TOP  -include_scope
create_power_domain PD_COMP -elements {comp}
create_power_domain PD_MOD  -elements {m3}

create_supply_port VDD1 -direction in
create_supply_port VSS  -direction in
create_supply_port VDD2 -direction in -domain PD_COMP

create_supply_net Pwr1 -domain PD_TOP
create_supply_net Gnd  -domain PD_TOP
create_supply_net Pwr2 -domain PD_COMP
create_supply_net Gnd  -reuse -domain PD_COMP
create_supply_net Pwr1 -reuse -domain PD_COMP

connect_supply_net Pwr1 -ports {VDD1}
connect_supply_net Pwr2 -ports {VDD2}
connect_supply_net Gnd  -ports {VSS}

set_domain_supply_net PD_TOP \
   -primary_power_net Pwr1 \
   -primary_ground_net Gnd

set_domain_supply_net PD_COMP \
   -primary_power_net Pwr2 \
   -primary_ground_net Gnd

create_supply_net VDD1_sw     -domain PD_MOD
create_supply_net Pwr1 -reuse -domain PD_MOD
create_supply_net Gnd  -reuse -domain PD_MOD

create_logic_port swCtl
create_logic_net  swCtl
connect_logic_net swCtl -ports swCtl

create_power_switch SW \
```

```
  -domain PD_MOD \
  -output_supply_port {swout VDD1_sw} \
  -input_supply_port {swin Pwr1} \
  -control_port {swctrl swCtl} \
  -on_state {SWon swin swctrl} \
  -off_state {SWoff !swctrl}

set_domain_supply_net PD_MOD \
   -primary_power_net VDD1_sw \
   -primary_ground_net Gnd

add_port_state VDD1 -state {ON_18 1.8}
add_port_state SW/swout \
   -state {ON_18 1.8} \
   -state {OFF off}
add_port_state VDD2 \
   -state {ON_10 1.0} \
   -state {OFF off}
add_port_state VSS -state {ON_00 0.0}

create_pst DESIGN_PST -supplies { VDD1 VDD2 VDD1_sw VSS }

add_pst_state state_1 -pst DESIGN_PST  \
   -state { ON_18 ON_10 ON_18 ON_00 }
add_pst_state state_2 -pst DESIGN_PST  \
   -state { ON_18 OFF   ON_18 ON_00 }
add_pst_state state_3 -pst DESIGN_PST  \
   -state { ON_18 ON_10 OFF   ON_00 }
add_pst_state state_4 -pst DESIGN_PST  \
   -state { ON_18 OFF   OFF   ON_00 }

create_logic_port iso_en
create_logic_net  iso_en
connect_logic_net iso_en -ports iso_en

set_isolation ISO_COMP \
   -domain PD_COMP \
   -applies_to outputs \
   -clamp_value 0 \
```

```
76    -isolation_power_net Pwr1 \
77    -isolation_ground_net Gnd \
78
79 set_isolation_control ISO_COMP \
80    -domain PD_COMP \
81    -isolation_signal iso_en \
82    -isolation_sense high \
83    -location self
84
85 set_level_shifter LS_COMP -domain PD_COMP -applies_to outputs
86    -rule both -location self
```

This required power intent specification is described using UPF and the full UPF code is shown above. We have added Isolation strategies, Level Shifting Strategies, Power Switches in these specifications based on the Power State Table and different power ports in power domain.

In the next chapter we will look at the Simulation and Synthesis of our design RTL and the UPF file. We will see the tools required for simulation and synthesis of our design. We will also look at the library creation and compilation of this technology library file using Synopsys Library Compiler tool.

# Chapter 3

# Simulation and Synthesis of RTL and UPF Design

In previous chapter, we have specified to power intent specification using UPF and we discussed in detail all the strategies. In this chapter, we will look at the simulation of our normal RTL design and combination of RTL + UPF design. When we consider UPF with the RTL design then we need the technology library file. We will also look at creation of the liberty IEEE standard library file and compilation of this file.

After simulation, we will also look at the synthesis of normal RTL design and RTL + UPF design. We will also look at the tool requirement and creation of TCL command file to run the synthesis process using tool.

## 3.1 Simulation of Normal RTL Design

Simulation is the process of using a simulation software (simulator) to verify the functional correctness of a digital design that is modeled using a HDL (hardware description language) like Verilog, System Verilog, etc. We have described our RTL design using System Verilog Hardware Description Language and we have also written test bench on top of it to verify the design.

In this project, for simulation, we have used Synopsys VCS simulation tool and the input and output figure is shown in figure 3.1.

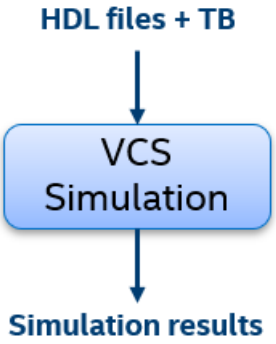


Figure 3.1: VCS Simulation

As per the above figure as you can see that, we have RTL files that is described using System Verilog and with on top of it we have also created test bench to get the simulation result based on given inputs. Based on the simulation results, we can check the functionality of the design. The simulation results of normal RTL design is shown in next chapter.

## 3.2 Simulation of RTL + UPF Design

In previous section as you can see, we have shown the normal RTL design simulation tool requirements and what tool we have used. In this section, we will see which tool we need when we add UPF file with RTL design.

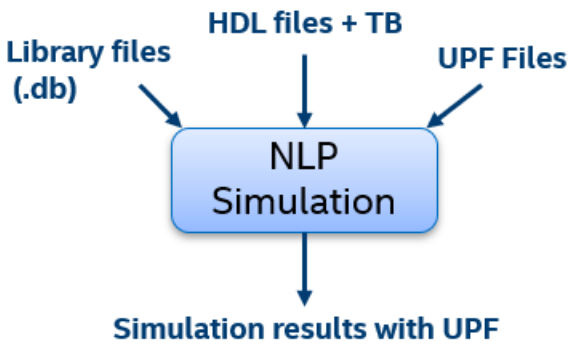


Figure 3.2: NLP (Native Low Power) Simulation

When we add low power specification with the original functional design at that time we can not use the same simulation tool as we used earlier. When we add UPF with the RTL design at time we used NLP (Native Low Power) simulator tool. This is also the same tool from Synopsys but simulator is in low power mode.

As per the figure 3.2 shown above, for low power mode simulation with the RTL + UPF, we also need the library file which should contain standard cell and power management cell like isolation, level shifter, etc. Using UPF with the RTL, we can assure that our functionality will work correctly under different power domain. Library creation and compilation of library is mentioned in next section.

## 3.3 IEEE Library Creation and Compilation

Library is needed when we need to do Synthesis and Simulation of our RTL design in low power mode. Library contains the standard cell and power management cell with number of inputs, outputs, functions, power information, delay, capacitance and all other required information.

In this project, we have created basic library which has basic standard cell gates like AND, OR, NOT, NOR, NAND, XOR and XNOR. We have also added low power management cell like isolation and level shifter. These all the cell information we have covered in next subsections. In this project, we have used Synopsys Library compiler tool to compile the library. We have created IEEE based Liberty Standard Library (.lib) and compiled using Library Compiler tool so that there should be no error in this file and converted from .lib file to .db file. This is shown in figure 3.3.


Figure 3.3: Library Compiler Tool

### 3.3.1 IEEE Library Standard Cell

In this subsection, we will discuss what are the standard cells we have added in our library and compiled it. We have taken the reference of IEEE Liberty Standard Library format and made the different standard cell like AND, OR, NAND, NOR, NOT, XOR, XNOR, etc. We have added the voltage map based on our UPF file requirements.

As you can see the code snippet below, at the starting we have global veriable set for whole library file and after that we have standard AND gate description. Here, we have only shown one cell but in main file we have added all the required standard cells.

```
library (tech_lib) {

  technology (cmos) ;
  delay_model : table_lookup ;
  bus_naming_style : "%s[%d]" ;
  date : "March 10, 2020" ;
  revision : 2020.01;
  input_threshold_pct_fall : 50.0 ;
  input_threshold_pct_rise : 50.0 ;
  output_threshold_pct_fall : 50.0 ;
  output_threshold_pct_rise : 50.0 ;
  slew_lower_threshold_pct_fall : 20.0 ;
  slew_lower_threshold_pct_rise : 20.0 ;
  slew_upper_threshold_pct_fall : 80.0 ;
  slew_upper_threshold_pct_rise : 80.0 ;
  time_unit : "1ns" ;
  voltage_unit : "1V" ;
  current_unit : "1mA" ;
  pulling_resistance_unit : "1kohm" ;
  capacitive_load_unit (1,pf) ;
  leakage_power_unit : "1nW" ;
  default_cell_leakage_power : 0.5 ;
  default_inout_pin_cap : 1.0 ;
  default_input_pin_cap : 1.0 ;
  default_output_pin_cap : 0.0 ;
  default_fanout_load : 1.0 ;
  default_max_fanout : 10.0 ;
```

```
  default_max_transition : 15.0 ;
  lu_table_template (scaler) {
  variable_1 : input_net_transition ;
  index_1 ("0.0") ;
  }
  voltage_map (pwr, 1.8) ;
  voltage_map (pwr1, 1.0) ;
  voltage_map (gnd, 0.0) ;
  operating_conditions (OP_DEFAULT) {
    process : 1.0 ;
    temperature : 25.0 ;
    voltage : 1.8 ;
    tree_type : best_case_tree ;
  }
  default_operating_conditions : OP_DEFAULT ;

/*------------AND Gate Description----------------*/

  cell ( AND2 ) {
    area : 3.0 ;
    pg_pin (pwr) {
      voltage_name : "pwr" ;
      pg_type : "primary_power" ;
    }
    pg_pin (gnd) {
      voltage_name : "gnd" ;
      pg_type : "primary_ground" ;
    }
    pin (A) {
      related_power_pin : "pwr" ;
      related_ground_pin : "gnd" ;
      direction : input ;
      fall_capacitance : 1 ;
      rise_capacitance : 2 ;
      capacitance : 1 ;
    }
    pin (B) {
      related_power_pin : "pwr" ;
      related_ground_pin : "gnd" ;
```

```
67      direction : input ;
68      fall_capacitance : 1 ;
69      rise_capacitance : 2 ;
70      capacitance : 1 ;
71    }
72    pin (Y) {
73      related_power_pin : "pwr" ;
74      related_ground_pin : "gnd" ;
75      direction : output ;
76      power_down_function : "(!pwr + gnd)" ;
77      function : "( A & B )" ;
78      timing () {
79        related_pin : "A" ;
80        timing_sense : non_unate ;
81        cell_rise(scaler) {
82          values(" 1.4 ") ;
83        }
84        cell_fall(scaler) {
85          values(" 1.6 ") ;
86        }
87        rise_transition(scalar) {
88          values( " 0.0 ");
89        }
90        fall_transition(scalar) {
91          values( " 0.0 ");
92        }
93      }
94      timing () {
95        related_pin : "B" ;
96        timing_sense : non_unate ;
97        cell_rise(scaler) {
98          values(" 1.4 ") ;
99        }
100       cell_fall(scaler) {
101         values(" 1.6 ") ;
102       }
103       rise_transition(scalar) {
104         values( " 0.0 ");
105       }
```

```
        fall_transition(scalar) {
          values( " 0.0 ");
        }
      }
    }

  }   /*-------AND Gate Ended----------*/
```

### 3.3.2 IEEE Library Power Management Cell

When we work on the low power specification using UPF, at that time during Synthesis and Simulation, we required library file with power management cell added. Here in our project, we only required two type of power management cell and that is Isolation Cell and Level Shifter Cell.

- **Isolation Cell**

Isolation cell code snippet based on IEEE Liberty Standard format is shown below. It is compiled using Synopsys Library Compiler tool and converted to .db file from .lib file.

```
/*------------Isolation Cell Description----------------*/

  cell ( isolation_cell ) {
    is_isolation_cell : true ;
    pg_pin (pwr) {
      voltage_name : "pwr" ;
      pg_type : "primary_power" ;
    }
    pg_pin (gnd) {
      voltage_name : "gnd" ;
      pg_type : "primary_ground" ;
    }
    pin (data) {
      direction : input;
      related_power_pin : "pwr" ;
      related_ground_pin : "gnd" ;
      isolation_cell_data_pin : true ;
    }
```

```
    pin (EN) {
      direction : input;
      related_power_pin : "pwr" ;
      related_ground_pin : "gnd" ;
      isolation_cell_enable_pin : true ;
    }
    pin (output) {
      related_power_pin : "pwr" ;
      related_ground_pin : "gnd" ;
      direction : output ;
      power_down_function : "(!pwr + gnd)" ;
      function : "data * EN" ;
      timing () {
        related_pin : "data EN" ;
        timing_sense : non_unate ;
        cell_rise(scaler) {
          values(" 1.4 ") ;
        }
        cell_fall(scaler) {
          values(" 1.6 ") ;
        }
        rise_transition(scalar) {
          values( " 0.0 ");
        }
        fall_transition(scalar) {
          values( " 0.0 ");
        }
      }
    }

  }   /*------------Isolation Cell Ended----------*/
```

- **Level Shifter Cell**

Level Shifter cell code snippet based on IEEE Liberty Standard format is shown below. It is compiled using Synopsys Library Compiler tool and converted to .db file from .lib file.

```
/*------------Level Shifter Cell Description----------------*/
```

```
2
3  cell ( level_shifter ) {
4    is_level_shifter : true ;
5    level_shifter_type : HL_LH ;
6    pg_pin (pwr) {
7      voltage_name : "pwr" ;
8      pg_type : "primary_power" ;
9      std_cell_main_rail : true ;
10   }
11   pg_pin (pwr1) {
12     voltage_name : "pwr1" ;
13     pg_type : "primary_power" ;
14   }
15   pg_pin (gnd) {
16     voltage_name : "gnd" ;
17     pg_type : "primary_ground" ;
18   }
19   pin (data) {
20     direction : input;
21     related_power_pin : "pwr1" ;
22     related_ground_pin : "gnd" ;
23     input_voltage_range (0.9,1.1) ;
24     level_shifter_data_pin : true ;
25   }
26   pin (output) {
27     related_power_pin : "pwr" ;
28     related_ground_pin : "gnd" ;
29     direction : output ;
30     power_down_function : "(!pwr + !pwr1 + gnd)" ;
31     function : "data" ;
32     timing () {
33       related_pin : "data" ;
34       timing_sense : non_unate ;
35       cell_rise(scaler) {
36         values(" 1.4 ") ;
37       }
38       cell_fall(scaler) {
39         values(" 1.6 ") ;
40       }
```

```
        rise_transition(scalar) {
          values( " 0.0 ");
        }
        fall_transition(scalar) {
          values( " 0.0 ");
        }
      }
    }

  }   /*------------Level Shifter Cell Ended----------*/
```

## 3.4 Synthesis of RTL Design

Synthesis is a process in which a design behavior that is modeled using a HDL (Hardware Description Language) is translated into an implementation consisting of logic gates. This is done by a synthesis tool. One main difference in terms of modelling using a language like Verilog is that for synthesis , the design behavior should be modeled at an RTL (Register Transfer Level) abstraction. This means modelling in terms of the flow of digital signals between hardware registers (flip-flops) and the logical operations (combinational logic) performed on those signals.

Hence if a Verilog design model is intended for synthesis, then only synthesizable constructs should be used, while for simulation there are no such restrictions.

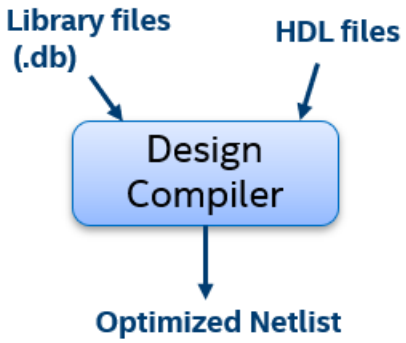


Figure 3.4: Design Compiler Tool

In this project, we have used Synopsys Design Compiler tool for synthesis of our RTL design. As per above figure 3.4, Design Compiler needs HDL files and Library files as a input and it will generate the optimized netlist based on RTL design. The schematic view of results is shown in the next chapter.

## 3.5 Synthesis of RTL + UPF Design

As per the above section, we only have the RTL files and we used Design Compiler tool for synthesis. When we add the power intent specification in the RTL design at that time we need library file which consist of power management cells added in it. Here we also use the same Synopsys Design Compiler tool but in topographical mode for UPF based design. This tool will generate the optimized netlist with the power specifications like isolation, level shifter, power domains, ports, etc.

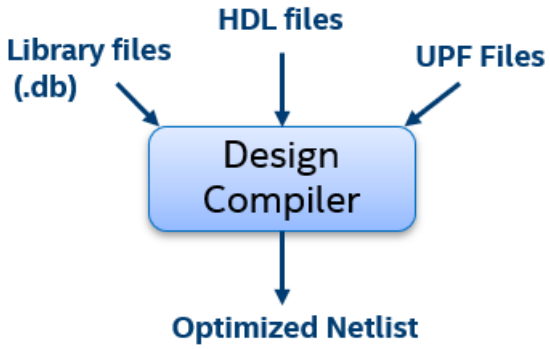


Figure 3.5: Design Compiler Tool for Low Power Design

As per the figure 3.5 shown above, Design Compiler tool with inputs and output are mentioned. In the next chapter, we will show the results of simulation and synthesis of Normal RTL and RTL + UPF design.

# Chapter 4

# Results and Comparison of RTL and Low Power RTL Design

In the previous chapter, we have seen low power architecture specified using UPF and tools use in the project with different inputs and outputs. In previous chapters we have mentioned tool for Simulation and Synthesis for normal RTL and RTL with low power specifications.

In this chapter, we will see the final results of Simulation of normal RTL and RTL with the UPF files. We will also see the results of Synthesis of normal design and design with low power specifications. At last based on the simulation results and Synthesis results, we can compare it and analyse the design and low power specification. Based on this comparison, we can verify the functionality of design that by adding power specification design is working correctly.

## 4.1 Simulation Results and Comparison

In this project, for our normal RTL design we have used Synopsys VCS simulator tool. In this project in our main design we have three module P1, P2, and P3. The outputs of module P1 and P2 goes to input of module P3. Module P3 has two outputs. We also have written System Verilog test bench on to off our design RTL. We simulated our design and checked the output of all the module and verified it. The simulation results we can see it in figure 4.1 shown below.

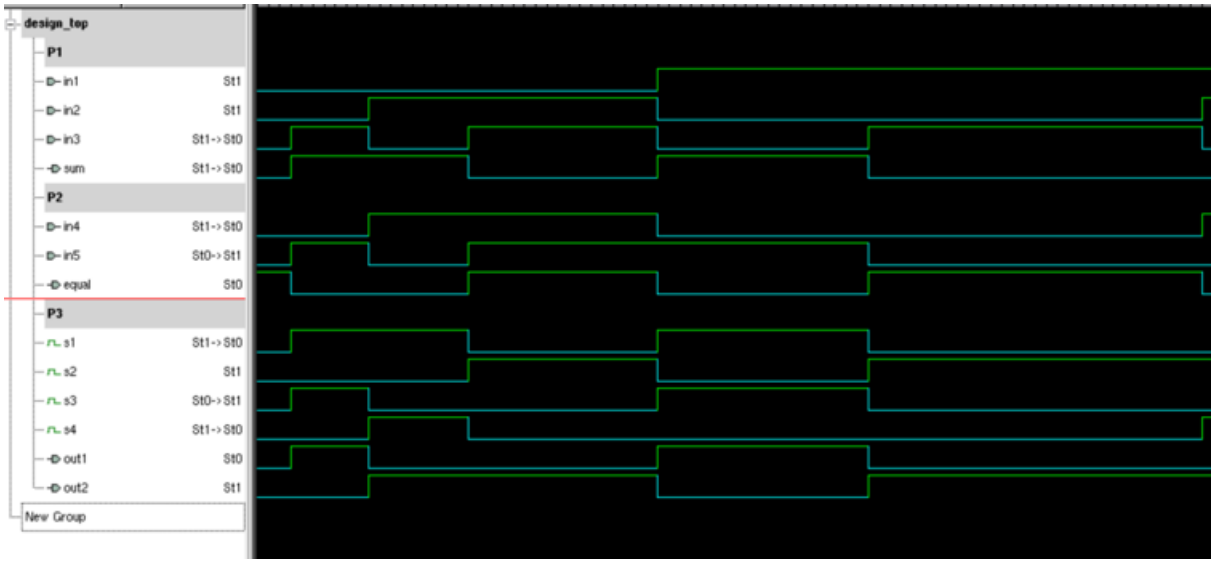


Figure 4.1: Simulation Results of Normal Design

As per the figure 4.1 shown above, we can see the three different section P1, P2 and P3 which are the main module in our design. Module P3 has two outputs that is out1 and out2. We have written test bench on top of RTL and gave the directed inputs and checked the outputs. Outputs of every module is correct and verified it for every combinations.

## 4.2 Synthesis Results and Comparison

In the previous section, we have shown the simulation result of our normal RTL design using directed input test bench and we have verified by using Synopsys VCS tool. In this section, we will see the Synthesis result of normal RTL design and RTL design with power specification that is through UPF.

Normal RTL design is Synthesized using Synopsys Design Compiler tool. We have wrote the tcl file with basic synthesis commands with the inputs like all RTL files, library database file and run those files using tcl in Design Compiler tool for synthesis. Using Design compiler tool, we got generated synthesized netlist. This netlist we can easily view in Design Compiler GUI. The abstract view of generated netlist is shown below in figure 4.2.

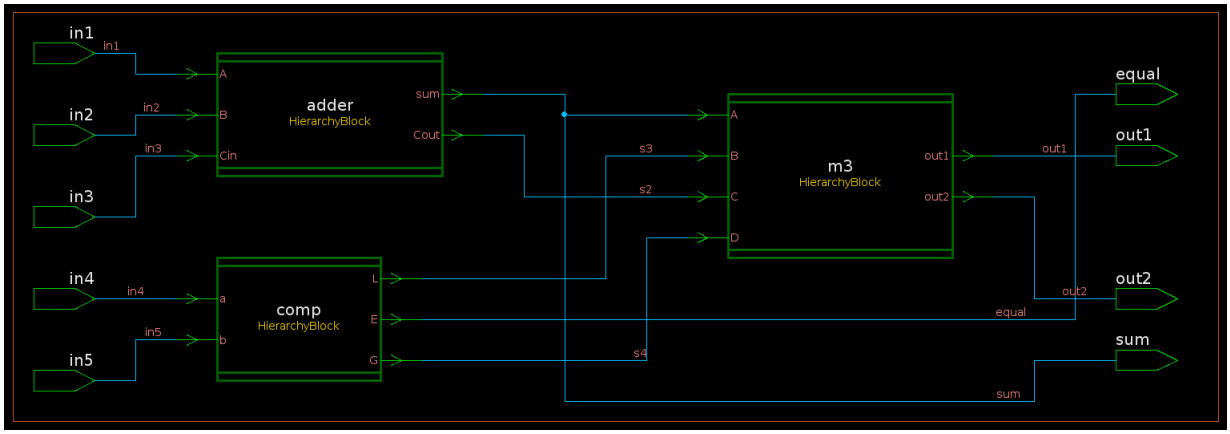


Figure 4.2: Synthesized RTL with Abstract View

As per the figure 4.2 shown above, in abstract view we have three module that is P1, P2 and P3. We can see that outputs module P1 and P2 goes to module P3 ad generated netlist is correct. The in detail gate level diagram we can see by going inside of these modules. We have also added detailed gate level diagram of normal RTL design in figure 4.3.

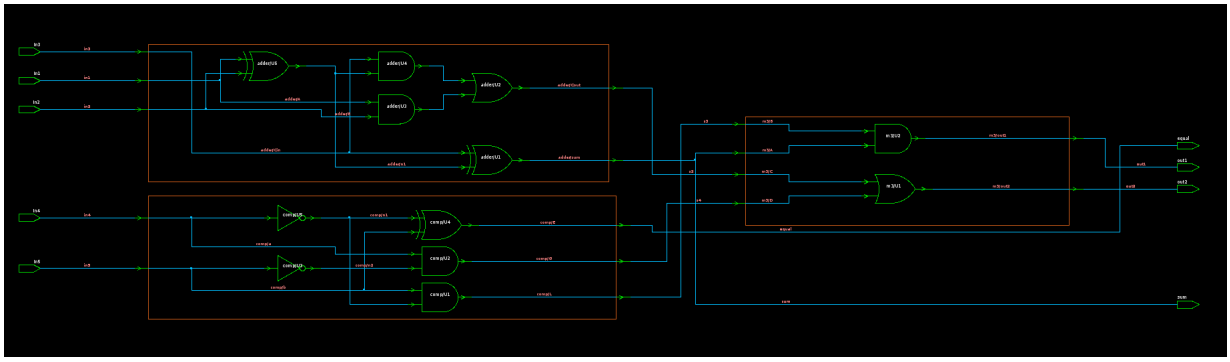

Figure 4.3: Synthesized RTL with Gate Level View

As per the figure 4.3 shown above, gate level view of all the three module P1, P2 and P3 shown. Here, as we can see that some of the output of gates in module P1 and P2 goes to input of module P3. When we add the low power specification on this RTL file we have to add isolation and level shifter on the output of gates on module P2.

RTL design files and added low power specification using UPF is synthesized using Synopsys Design Compiler tool in topographical mode. When we have to synthesize RTL + UPF file then at that time we have to use the topographical mode in Design Compiler to

actually see the isolation cell and level shifter in GUI view of the netlist. These all the RTL files, library database with power management cell added and UPF file is synthesize using tcl commands and after the compile process generates the gate level netlist with UPF view and we can see it in figure 4.4 shown below.

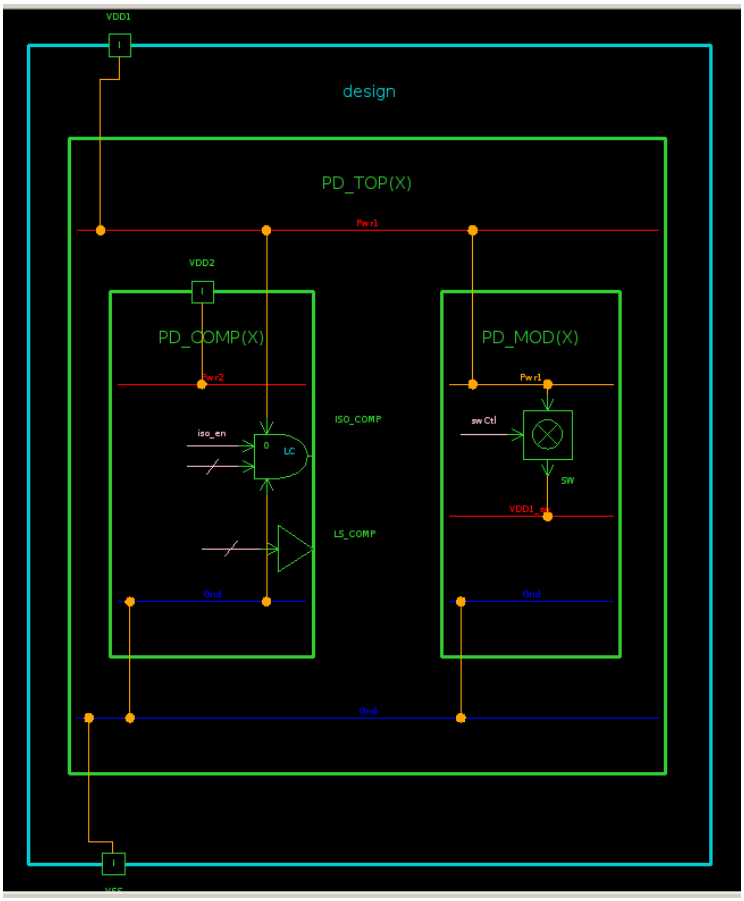


Figure 4.4: Synthesized of RTL with UPF View

As per the figure 4.4 shown above, We have three module PD_TOP, PD_COMP and PD_MOD. In this PD_TOP is the top module and output of module PD_COMP goes to input of module PD_MOD. Here module PD_COMP and PD_MOD works on different power supply and we added isolation cell and level shifter on outputs of module PD_COMP. As you can see in the figure PD_COMP has isolation and level shifter cell shown in GUI view. These all the results are verified and by using simulation and synthesis process we can verify the low power design.

## 4.3 Future scope of the project

As per the above section simulation and synthesis results, we can clearly state that when we move towards lower technology node at that time we require active power management and low power verification is required for device that can work on low power with same functionality as before.

In our actual Data Manipulation IP we have different kinds of partitions with different functionality. These all the different partitions will not works at the same time so when some partition is ON and other is OFF then we have to control that partition with power intent specification so that we can reduce the leakage power consumption.

In our Data Manipulation IP, we have power intent specification is written in UPF and we also have hundreds of functionality test cases written for dynamic simulation to check the functionality. So our main future scope of the project is to enable all these functional test cases with UPF and run these test cases in low power mode that is NLP (Native Low Power) simulation mode.

Ideally all the test cases should pass with low power mode but if some of the tests are failed then we need to debug the test case and find the power bug present in the design with power intent specification. By this approach, we can easily able to find the power bug in the design.

# Conclusion

- More and more consumer demands, features and longer battery life, requirement of low power design arises.
- Active power management is important at 90 nm and it becomes necessary below 65 nm technology.
- When we move towards the lower technology node, leakage power is more compare to active power and because of this active power management is required to reduce the power using power reduction techniques.
- To reduce the leakage power in a design, power management concepts like power domains, isolation, level shifter and retention registers is needed.
- Power intent specification is described using Unified Power Format (UPF) and we can add power management cells in that.
- All the functionality of design should work same as earlier when we use UPF with the design to describe power specification.

# Bibliography


[1] F. Bembaron, S. Kakkar, R. Mukherjee, and A. Srivastava, "Low Power Verification Methodology Using UPF," in Conference on Electronic Systems Design and Verification Solutions, DVCON, 2009, pp. 228-233

[2] Unified Power Format, IEEE Draft Standard for Design and Verification of Low Power Integrated Circuits, IEEE P1801/D18, 23 October, 2008.

[3] Trummer, C. ; Kirchsteiger, C.M. ; Steger, C. ; Weiss, R. ; Dalton, D. "Simulation-based verification of power aware System-on-Chip designs using UPF IEEE 1801",NORCHP.2009., pp 1 -4

[4] Roopa Kulkarni ; S Y Kulkarni. "Energy Efficient Implementation, Power Aware Simulation and Verification of 16-bit ALU using Unified Power Format Standards", in Conference on 2014 International Conference on Advances in Electronics, Computers and Communications (ICAECC)

[5] Renduchinthala Anusha; "An effective and efficient methodology for SoC power management through UPF", Thesis, 2016.

[6] Library CompilerTM User Guide, Synopsys, Version P-2019.03, March 2019

[7] Design CompilerR User Guide, Synopsys, Version P-2019.03, March 2019

[8] IEEE Standards Association. *IEEE Standard for SystemVerilog - Unified Hardware Design, Specification, and Verification Language (2013).* Website link

# M.Tech Thesis

ORIGINALITY REPORT

| 9% | 4% | 5% | 8% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | "Low Power Methodology Manual", Springer Science and Business Media LLC, 2007<br>Publication | 1% |
| 2 | Submitted to Visvesvaraya National Institute of Technology<br>Student Paper | 1% |
| 3 | www.micro-ip.com<br>Internet Source | 1% |
| 4 | Submitted to National Institute of Technology Warangal<br>Student Paper | 1% |
| 5 | Submitted to Oregon State University<br>Student Paper | 1% |
| 6 | repository.iiitd.edu.in<br>Internet Source | 1% |
| 7 | vscripts.ru<br>Internet Source | 1% |
| 8 | Submitted to Vel Tech University<br>Student Paper | 1% |

| | | |
|---|---|---|
| 9 | Luís Gomes. "Power reduction of a CMOS high-speed interface using Power Gating", Repositório Aberto da Universidade do Porto, 2014.<br>Publication | 1% |
| 10 | Submitted to University of Southampton<br>Student Paper | 1% |
| 11 | "Power-Aware Testing and Test Strategies for Low Power Devices", Springer Science and Business Media LLC, 2010<br>Publication | 1% |

| | | | |
|---|---|---|---|
| Exclude quotes | Off | Exclude matches | < 1% |
| Exclude bibliography | On | | |