

TIMING AND POWER CO-OPTIMISATION BASED ON Z ADVISORY GENERATION

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

(Embedded Systems)

By

Rahul Kumar

(13MECE10)



Electronics & Communication Engineering Branch

Electrical Engineering Department

Institute of Technology

Nirma University

Ahmedabad-382 481

May 2015

TIMING AND POWER CO-OPTIMISATION BASED ON Z ADVISORY GENERATION

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

(Embedded Systems)

By

Rahul Kumar

(13MECE10)

Under the guidance of

External Project Guide:

Tiju Jacob

Engineering Manager,

Intel India Technology Pvt. Ltd.,

Bangalore.

Internal Project Guide:

Prof. Ruchi Gajjar

Assistant Professor (EC Dept.),

Institute of Technology,

Nirma University, Ahmedabad.



Electronics & Communication Engineering Branch

Electrical Engineering Department

Institute of Technology

Nirma University

Ahmedabad-382 481

May 2015

Declaration

This is to certify that

- a. The thesis comprises my original work towards the degree of Master of Technology in Embedded Systems at Nirma University and has not been submitted elsewhere for a degree.
- b. Due acknowledgment has been made in the text to all other material used.

- Rahul Lumar

13MECE10



Certificate

This is to certify that the Major Project entitled "TIMING AND POWER CO-OPTIMISATION BASED ON Z ADVISORY GENERATION" submitted by Rahul Kumar (13MECE10), towards the partial fulfillment of the requirements for the degree of Master of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of our knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Date:

Place: Ahmedabad

Prof. Ruchi Gajjar
(Internal Guide)

Dr.N.P.Gajjar
(Program Co-ordinator)

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

Prof. D. K. Kothari
(Section Head, EC)

Dr.K.Kotecha
(Director, IT-NU)

Intel Technology India Pvt. Ltd.

Certificate

This is to certify that the Project entitled "**TIMING AND POWER CO-OPTIMISATION BASED ON Z ADVISORY GENERATION**" submitted by **Rahul Kumar(13MECE10)**, towards the submission of the Project for requirements for the degree of Master of Technology in VLSI, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.

Mr. Tiju Jacob
Engineering Manager,
Big-Core India,
Intel India Pvt. Ltd.,
Bangalore.

Acknowledgements

I would like to express my gratitude and sincere thanks to **Dr.K.Kotecha**, Director, Institute of Technology,Nirma University, **Dr.P.N.Tekwani**, Head of Electrical Engineering Department, **Dr.Dilip Kothari**, Section Head,EC and **Dr.N.P.Gajjar**, Coordinator of M.Tech Embedded Systems program for allowing me to undertake this thesis work and for their guidelines during the review process.

I am deeply indebted to my thesis supervisor **Ms. Ruchi Gajjar**, Assistant Professor, E.C.Dept., Nirma University and **Mr. Tiju Jacob**, Engineering Manager at Intel India Technology Pvt. Ltd. for their constant guidance and motivation. I also wish to thank **Mr. Arpit Gandhi**, Manager, Intel India Technology Pvt. Ltd., **Mr. Praveen K. Gontla**, **Ms. Neelam C. Maniar** and all other team members at Intel for their constant help and support. Without their experience and insights, it would have been very difficult to do quality work.

I wish to thank my friends of my class for their delightful company which kept me in good humor throughout the journey. Last, but not the least, no words are enough to acknowledge constant support and sacrifices of my family members because of whom I am able to complete the degree program successfully.

- **Rahul Kumar**
13MECE10

Abstract

Power is gaining importance with every generation of any new Processor Architecture designs as products come up in different form factors and enter into new mobile segments. In fact, for many of the current and future products, power is the top priority. In design convergence flows, if margin on a timing path is more positive than a certain threshold, the logic cells in the path are either downsized or converted to low leakage cells for power reduction. Hence, better timing margin translates into more power optimization opportunities. In general during design cycle, focus of the designer is on meeting timing requirements. Very few designers spend additional effort to create extra positive margin for power reduction.

The work presented in this report is an advisory developed to find circuit nodes where timing improvements can lead to a larger number of cells moving to more positive margins subsequently leading to significant power savings. Main advantage of creating such advisory is to provide a list of handful nodes with high region of interest (ROI) to the designers. After doing timing improvement as suggested by advisory, designers will be able to increase low leakage cells count by 10% approx. Additionally, data from this advisory can be fed back to the synthesis flows which can be tuned to achieve better positive margins for high power ROI nodes. This advisory can also be used to allocate positive timing margin between different blocks for overall optimal power gain. This advisory can be part of any circuit design flows.

Contents

Declaration	iii
Certificate	iv
Intel Certificate	v
Acknowledgements	vi
Abstract	vii
List of Tables	xi
List of Figures	xii
1 Introduction	1
1.1 Hierarchy of Server/Client Core Design	2
1.1.1 Hierarchy of core	2
1.2 ASIC Design Approach	3
1.2.1 Full Custom Integrated Circuit Design	3
1.2.2 Semicustom Integrated Circuit Design	3
1.3 Datapath v/s RLS(RTL to Layout Synthesis)	4
1.4 Need for manual implementation	4
1.5 Report Organization	5
2 Literature Survey	6
2.1 Need for low power design	6
2.1.1 Design Flow with and without Power	7
2.2 Relationship Between Digital Design Abstraction Levels and Power Estimation	9
2.3 Summary	10
3 Basic Concepts of Power	11
3.1 Static Power	11
3.2 Dynamic Power	12

3.2.1	Switching power	12
3.3	Internal power	12
3.4	Short-Circuit Power	13
3.5	Leakage Power	13
3.5.1	Sub threshold leakage	13
3.6	Power Estimation Factors	14
3.6.1	ACTIVITY FACTOR (AF)	14
3.6.2	SIGNAL PROBABILITY (SP)	14
3.6.3	VCD (VALUE CHANGE DUMP) FILE	14
3.6.4	Diff Of AF	14
3.6.5	DRIVEN INPUTS	15
3.6.6	AVERAGE VALIDITY	15
3.6.7	CAP QUALITY	15
3.6.8	POWER ESTIMATION QUALITY METRIC	15
3.7	Dynamic Power Optimization	15
3.7.1	Reducing the supply voltage	16
3.7.2	Reducing the switching activity	16
3.8	Leakage power optimization	16
3.8.1	Leakage Reduction Methods	16
3.9	Summary	17
4	Delay Modeling	18
4.1	Introduction	18
4.2	Delay Model	18
4.3	Data Point	21
4.4	Capacitive Load Extraction	22
4.5	Circuit Delay	23
4.6	Summary	23
5	Preparing the Z Advisory	24
5.1	Concept	24
5.2	The Algorithm	25
5.3	The Output of advisory	28
5.4	The Onion Peeling Phenomenon	28
5.4.1	Onion Peeling	28
5.4.2	Workaround for Onion Peeling	29
5.5	Summary	30
6	Results	31
6.1	Test Results	31
6.2	Steps of Using the Advisory	32
6.2.1	Detailed Result	34

CONTENTS

x

7 Conclusion and Future Scope	35
7.1 Conclusion	35
7.2 Future Scope	35
References	37
Index	39

List of Tables

I	Example of data given to user by advisory	28
II	Typical Z profile of block	30

List of Figures

1.1	Hierarchy of Core	3
1.2	ASIC Circuit Implementation Approaches	4
2.1	VLSI DESIGN FLOW	8
2.2	Relationship between different abstraction level and Power estimation techniques	9
4.1	Nand gate with input load of two and output load of four	19
4.2	Fall delay of gate versus output load with different input transition times	20
4.3	Fall delay of gate versus input load with different output loads	20
4.4	Capacitive load determination	22
4.5	Delay of Inverter versus width of transistors (p and n)	23
5.1	Examples of Fan-in, Fan-out cones found in circuit	25
5.2	Valid Cone Propagation	26
5.3	RequiredConePropagation	27
5.4	Onion Peeling Effect	29
6.1	Graph of % of total block-Z controlled by different blocks	32
6.2	Z distribution for FUB A, before and after timing improvement	33
6.3	Z distribution for FUB B, before and after timing improvement	33

Chapter 1

Introduction

With increasing focus on power reduction in every digital product, circuit designers need to invest effort in power optimization. One of the most widely used power optimization technique is to downsize gates on timing wise non-critical path. Downsizing of gates reduces both active and leakage power. For the leakage power reduction, we have low leakage standard cells in the library. Low leakage cells have higher delay than their nominal counterparts. Hence, to convert nominal cells to low leakage cells, the need is to have positive timing margin. Both of the above techniques need margin to be more positive than a certain threshold as decided by the project methodologies. Designers run LR based optimization tools [11] in different stages of design to reduce power using both of above mentioned techniques. LR tools make best use of available timing margin for power reduction. However, they cannot suggest a-priori any changes to the design so that more gates can move to more positive margins. This report presents work on developing an advisory to find circuit nodes where small timing improvement can turn a larger number of cells to more positive margins. Our work tries to find out signals which prevent the automatic power optimization tools from recovering more Z or converting more cells into Low Leakage cells. Once such a list is available, designers can work on improving timing margin on such selected high power ROI nodes. Such a list helps designers to focus only on a selected number of nodes to achieve maximum returns for effort.

The approach to creating such an advisory is by tracing data arrival time through schematic nodes. The algorithm start with tracing from all the start points/end points of timing paths. Start points and end points of timing paths can either be clock pin of a sequential element or an interface signal in case hierarchical design. It trace

through logic cells between start and end point of the paths, and detect controlling signals whose timing if relaxed, can result in better timing margins for the entire cone. Based on the data from advisory, designers can work on improving timing margin. It is possible that designer may not find any possible way to improve timing for certain nodes in design. Our experiments on 14nm Core blocks suggest that there exist cases in design where margins can be improved with relatively less effort and gain more in power. Timing margin can be improved by clock tuning, logic optimization or interconnect delay improvement or any such conventional methods.

1.1 Hierarchy of Server/Client Core Design

Chip design commences with the conception of an idea dictated by the market. These ideas are then translated into architectural and electrical specifications. The architectural specifications define the functionality and partitioning of the chip into several manageable blocks, while the electrical specifications define the relationship between the blocks in terms of timing information. There are many methods commonly used to design a digital block but most of the industries use two popular methods named RLS flow and Structured data path flow. In this project, data path flow has been used.

In data path flow everything is implemented manually, templates are used to make use the same hierarchy multiple times. In this flow clock tree also build manually. All metal connections and via connections needs to do manually. It is challenging work to do manual implementation and meet all the constraints like timing, power, noise, reliability, layout quality rules, electrical specifications, circuit quality rules, utilization of metal resources, RTL to back end specifications and scan requirements for DFT.

1.1.1 Hierarchy of core

Full chip(server/client core) comprises clusters. Clusters are made up of different section. Each section incorporates bunch of functional blocks(FUB). A functional blocks(FUB) can be ude multiple times based on the requirements of the ovealall design.

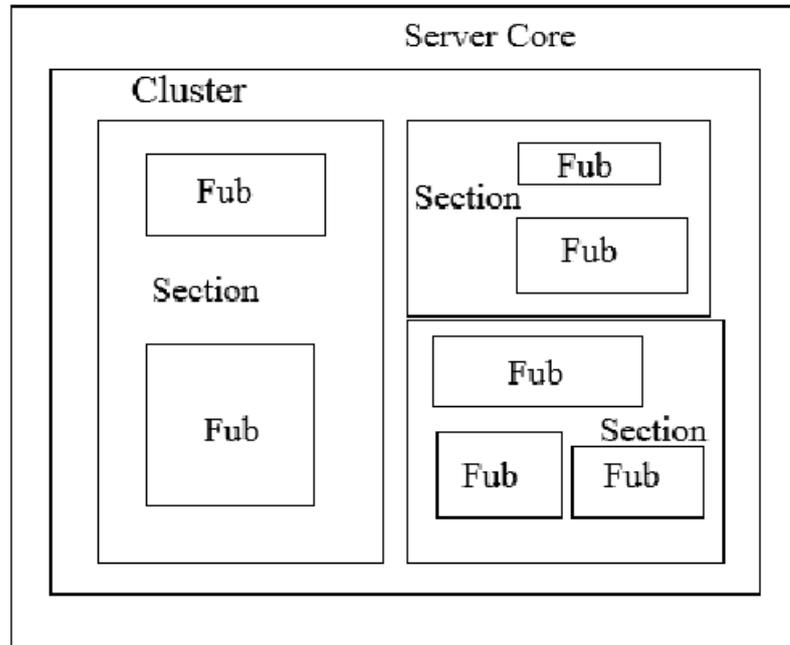


Figure 1.1: Hierarchy of Core

1.2 ASIC Design Approach

1.2.1 Full Custom Integrated Circuit Design

A full custom design defines all the photo lithographic layers of the device. It is a design at transistor level. The benefits of Full Custom Design include reduced area (reduces recurring component cost); performance improvements and also the ability to integrate analog components and other pre-designed components such as microprocessor cores that form a system on chip. The disadvantages of full custom design includes increased manufacturing and design time, increased non-recurring engineering costs, more complexity in the Computer Aided Design (CAD) system and a much higher skill requirement on the part of design team.

1.2.2 Semicustom Integrated Circuit Design

The design steps with standard cells are also common to full custom IC design. The difference is that standard cell design uses the manufacturers cell libraries that have been used in potentially hundreds of other design implementations and therefore are

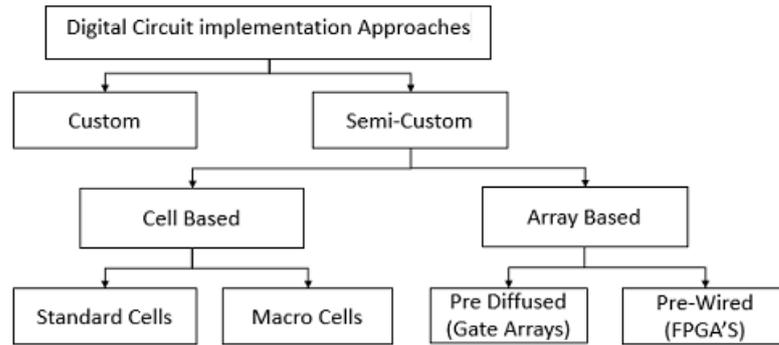


Figure 1.2: ASIC Circuit Implementation Approaches

of much lower risk than fully custom design. Standard cells produce a design density that is cost effective, and they can also integrate the processor cores effectively.

This is working methodology used for processor design mostly. Functional Block (FuB) are designed using characterized standard cell libraries and simulation tools, based on input RTL obtained from the front-end team, to meet various design constraints for 14nm technology.

1.3 Datapath v/s RLS(RTL to Layout Synthesis)

RLS and Data path are two ways of synthesizing the RTL code.

- a. DP(Data Path): Schematics (Netlist) - implemented manually Layout - implemented manually
- b. RLS(RTL to Layout Synthesis): Netlist - implemented by tool automatically Layout - implemented by tool automatically

1.4 Need for manual implementation

Custom IC designs are synthesized in two ways Random Logic Synthesis (RLS) and Data Path. There is a big methodological difference between the flows of both ways of design. RLS uses design automation tools for synthesis and layout of design whereas in Datapath synthesis and placement of schematic are done manually.

In custom IC designs Data path circuits are normally partitioned into dedicated blocks on the floor plan. The designers manually construct each data path circuit

and place it in a bit sliced style. This approach reduces the timing skews between different bits and accurately predicts the loading of individual nets. Synthesis tool has no knowledge on how the circuit might be placed, so the load and timing of interconnect cannot be estimated accurately. The placement tool has no knowledge on the regularity information from the data path circuit, so the layouts of RLS blocks are often sub optimal.

In case of hierarchical designs, knowledge of power cost on all the interface nodes can be useful in allocation of timing margin between blocks. Other possible usage of such advisory is to tune synthesis for power optimization. After seeing timing analysis based on first-cut synthesis, subsequent synthesis can be tuned to get better timing on high power ROI nodes and thus better power optimization.

1.5 Report Organization

The rest of the report is organized as follows.

Chapter 2, *Literature Survey*, describes the basic for need of Low Power Design.

Chapter 3, *Basic Concepts of Power*, describes the basic terminology used in low power VLSI design.

Chapter 4, *Delay Modeling*, presents the basic idea about delay models used to model the devices in the design.

In **chapter 5**, *Preparing the Z Advisory*, covers algorithm for preparing the advisory for power optimization based on timing margin.

In **chapter 6**, *Result*, covers the output of the algorithm for power optimization based on timing margin.

Finally, in **chapter 7** concluding remarks and scope for future work is presented.

Chapter 2

Literature Survey

2.1 Need for low power design

In the early 1970s the main design constraint for designing digital circuits were high speed and minimum area. Majority of the EDA tools were customized to meet these specific design criteria. Power was also one of the criteria for design but was not very visible. Today, the reduction of area is not a major issue in digital circuit design due to new Integrated Circuit(IC) production techniques and reduction in transistor sizes, millions of transistors can be fit in a very small area. This, shrinking in the size of the circuit has paved the way for reduction in power consumption by the circuit which in the end helps devices to have an extended battery life. Also, in the sub-micron Design technology, heat generated due larger power dissipation limits the proper functionality of the circuit.

Today the market demands low power devices not only for better battery life but also portability, reliability, performance, time to market and cost. This is true for all the devices that are being produced for personal use such as Smartphones, Music Players, personal computing devices, wireless communications systems, home entertainment systems. High-performance computing devices particularly need to dissipate less power for proper functionality and reliability for a long period of time [1]. Taking all these points in context, low power design has become one of the most important design criterion for VLSI(Very Large Scale Integration) systems design.

2.1.1 Design Flow with and without Power

A top-down VLSI digital design methodology has been illustrated in Figure 2.1. The Figure shows a system level specification to the physical level implementation steps involved in VLSI design. The approach on left has two constraints, performance optimization and area minimization. However, a third parameter of power dissipation has made the designers to change the design flow to the one shown in the right-hand side of the Figure 2.1.

In each of the VLSI design flow levels with power as a parameter, two new important power factors, namely power optimization and power estimation has been added into each design step. Power optimization is the process to obtain the best design meeting all the design specifications and without violating any design constraints. To meet all the design specification and constraints, a power optimization technique unique to every design flow level has to be employed. Power estimation is the process of calculating power and energy dissipated at different stages of the design process with a certain percentage of accuracy. Power estimation techniques estimates the effect of various design optimizations and design modifications on power at different abstraction levels in the design flow.

Generally during the design a power optimization step is run first followed by a power estimation step, but in certain design flow level there is no specific design procedure followed for power minimisation. Each design flow stage can includes a collection of low power design techniques. Each step in the flow may result in a significant reduction of power dissipation with respect to the overall power dissipation reduction. However, different combination of low power techniques can lead to better results than another combination of techniques.

Generally, majority of power is consumed due to the switching activities of the capacitors i.e., charging and discharging of the capacitors. So, at a higher abstraction level in the design flow power dissipation can be reduced by reducing the switching activities in the design, which can be done by switching off the portions of the system when they are not needed. Large VLSI design contain different units like a processor consists of a functional units and controllers. The idea of power reduction is to switch off any of the units of the processor when they are not required so that less power will be dissipated when the processor is operating [2].

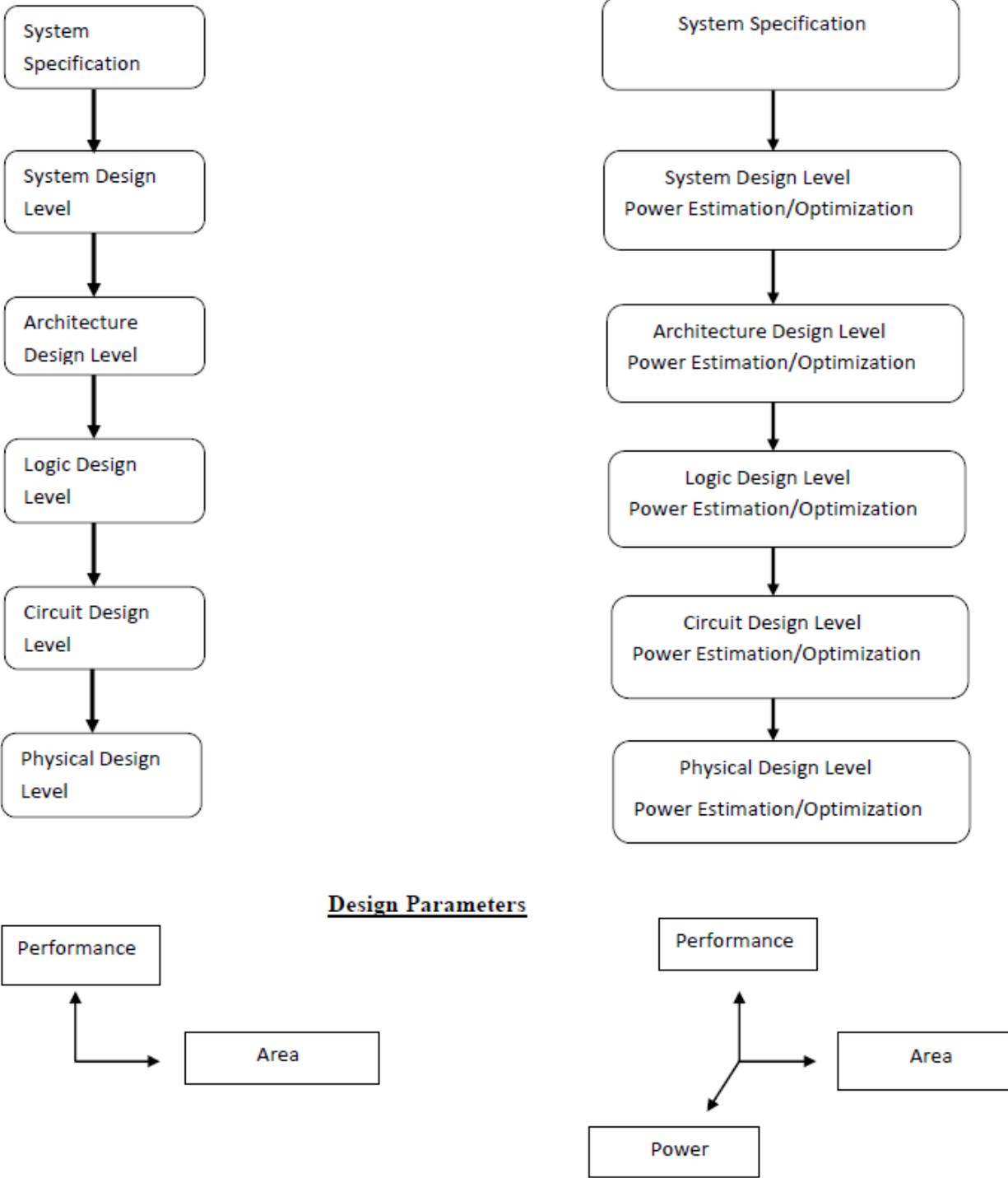


Figure 2.1: VLSI DESIGN FLOW

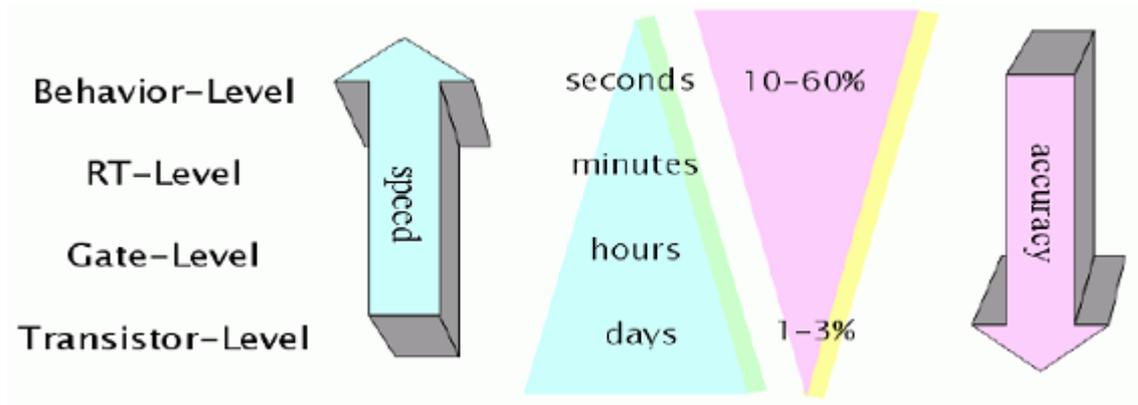


Figure 2.2: Relationship between different abstraction level and Power estimation techniques

2.2 Relationship Between Digital Design Abstraction Levels and Power Estimation

The relationship between digital design abstraction level and power estimation techniques is shown as Figure 2.2.

The power estimation techniques at higher abstraction level of design is much faster, but the accuracy is low due to the limited design information. A large number of CAD techniques for power estimation at lower abstraction levels of digital design, such as transistor-level [2-4] or gate-level [5], have been proposed.

Generally speaking, the power estimation techniques can provide more accurate results. However, due to too much computation resources requirement in low abstract levels they may become impractical for complex designs due to the whole system simulation. In addition, when the design has been specified down to gate level or lower, it may be too expensive to go back to fix high-power problems. Most importantly, IP vendors may not provide such low-level description for an IP to protect their knowledge.

2.3 Summary

In this chapter, it is explained why there is need for a low power design and compares the design models with and without power and what impact it will have on the modern VLSI design.

Chapter 3

Basic Concepts of Power

The power dissipation of digital CMOS circuits can be described by

$$P_{avg} = P_{dynamic} + P_{shortcircuit} + P_{static} \quad (3.1)$$

where,

Pavg is the average power dissipation,

Pdynamic is the dynamic power dissipation due to switching of transistors switching,

Pshort circuit is the short-circuit current power dissipation when there is a direct current path from power supply down to ground,

Pleakage is the power dissipation due to leakage currents,

Pstatic and is the static power dissipation[2][4].

3.1 Static Power

Static power is the power dissipated by a CMOS (Complementary Metal Oxide Semiconductor) gate when it is not switching that is, when it is inactive or static. Ideally, CMOS (Complementary Metal Oxide Semiconductor) circuits are expected not to dissipate any static (DC) power since in the steady state there is no direct path from Vdd to ground. Since in reality the MOS transistor is not a perfect switch, this scenario can never be realized in practice. There is always a leakage currents, sub threshold currents, and substrate injection currents, which give rise to the static (DC) component of power dissipation. The largest component of static power comes from source-to-drain sub threshold voltage, which is caused by reduced threshold voltages

that prevent the gate from completely turning off [2][4].

3.2 Dynamic Power

The dynamic power consumption of a Circuit results due to the Charging and discharging of Load Capacitor. Each time the capacitor CL gets charged through the PMOS transistor, its voltage rises from 0 to Vdd and a certain amount of energy is drawn from the power supply. Part of this energy is dissipated in the PMOS device, while the remainder is stored on the load capacitor. During the high-to-low transition, this capacitor is discharged, and the stored energy is dissipated in the NMOS transistor. $P_{\text{dyn}} = CLV_{\text{dd}}^2 f$, represents the frequency of energy-consuming transitions or switching activity; this is transitions for static CMOS. CL is load capacitance and Vdd is supply voltage. Dynamic power of a circuit is composed of:

- a. Switching power
- b. Internal power

3.2.1 Switching power

The switching power of a driving cell is the power dissipated by the charging and discharging of the load capacitance at the output of the cell. The total load capacitance at the output of a driving cell is the sum of the net and gate capacitances on the driving output. The charging and discharging are result of logic transitions. Switching power increases as logic transitions increase. Therefore, the switching power of a cell is a function of both the total load capacitance at the cell output and the rate of logic transitions. Switching power comprises 70-90 percent of the power dissipation of an active CMOS circuit [2][4].

3.3 Internal power

Internal power is any power dissipated within the boundary of a cell. During switching, a circuit dissipates internal power by the charging or discharging of any existing capacitances internal to the cell. Internal power includes power dissipated by a momentary short circuit between the P and N transistors of a gate, called short-circuit power.

3.4 Short-Circuit Power

Short circuit power results due to the non-zero rise and fall times of the input waveforms. In actual designs, the assumption of the zero rise and fall times of the input wave forms is not correct. The finite slope of the input signal causes a direct current path between and GND for a short period of time during switching, while the NMOS and the PMOS transistors are conducting simultaneously. The average short circuit power consumption is,

$$P_{sc} = T_{sc} * V_{dd} * I_{peak} \quad (3.2)$$

where,

T_{sc} represents the short circuit duration,

V_{dd} is supply voltage and

I_{peak} is peakcurrent during short circuit period.

3.5 Leakage Power

The PMOS and NMOS transistors used in a CMOS logic circuit commonly have non-zero reverse leakage and sub-threshold currents. These currents can contribute to the total power dissipation even when the transistors are not performing any switching action. The leakage power dissipation, $P_{leakage}$ is caused by two types of leakage currents. The leakage power dissipation, $P_{leakage}$ is caused by two types of leakage currents

- a. Reverse-bias diode leakage current
- b. Sub threshold current through a turned-off transistor channel

3.5.1 Sub threshold leakage

The major component of leakage current is the sub-threshold current of the transistors. A MOS transistor can experience a drain-source current, even when is smaller than the threshold voltage. The closer the threshold voltage is to zero volts, the larger the leakage current at V and the larger the static power consumption. To offset this

effect, the threshold voltage of the device has generally been kept high. Standard processes feature values that are never smaller than 0.5-0.6V and that in some cases are even substantially higher (0.75V).

3.6 Power Estimation Factors

3.6.1 ACTIVITY FACTOR (AF)

Activity factor of a signal is defined as the average number of transitions per clock cycle. An activity factor of 0.1 implies that the signal changes once every 10 clock cycles. Activity factor at any node gives the dynamic power consumed at that node.

AF FILE

AF file is the file which contains activity factor and signal probability for all the inputs of a block.

3.6.2 SIGNAL PROBABILITY (SP)

Signal probability of a signal is defined as the probability of its ON state. A signal probability of 0.5 implies that the probability of HIGH state for that signal is 0.5.

3.6.3 VCD (VALUE CHANGE DUMP) FILE

VCD file contains all the time stamps and change in values of all inputs and outputs at those time stamps and also changes in values of some of the internal nodes (which can be specified explicitly by the designers) when test are run on a given block.

3.6.4 Diff Of AF

It is defined as the Difference of activity factors between RTL simulated values and values calculated by power estimation tool. For accurate power estimation this difference has to be small.

3.6.5 DRIVEN INPUTS

Driven inputs are defined as the percentage of inputs driven by the power estimation tool. Before estimating power one must ensure that all inputs are present in the AF file/VCD File.

3.6.6 AVERAGE VALIDITY

Average Validity is defined as the percentage of nodes driven by the power estimation tool. If there are large numbers of nodes which are not driven by the tool, then average validity will be less.

3.6.7 CAP QUALITY

Cap quality is defined as the percentage of nodes whose capacitance are taken from parasitic file. If there are any mismatches between schematics netlist and the parasitic file, then the tool tries to estimate the capacitance values. This leads to poor cap quality Metric.

3.6.8 POWER ESTIMATION QUALITY METRIC

Power estimation Quality metric depends on four factors namely Diff of AF, Driven Inputs, Average Validity, and Cap Quality.

$$\begin{aligned} \text{Quality metric of power estimation} = & 0.4 * \text{Diff of AF} + 0.3 * \text{Driven Inputs} + \\ & 0.2 * \text{Average Validity} + 0.1 * \text{Cap Quality} \end{aligned} \quad (3.3)$$

3.7 Dynamic Power Optimization

Dynamic power,

$$P_{dyn} = C_L * V_{dd}^2 \quad (3.4)$$

From the above Equation, dynamic power can be reduced by reducing either the Supply voltage or the switching activity of the transistor.

3.7.1 Reducing the supply voltage

Reducing V_{dd} has a quadratic effect on P_{Dyn} . Reducing the supply voltage impacts the performance. As there is reduction in supply voltage driving current I_d of the device decreases which leads to an increase in the propagation delay. This can be observed from the propagation delay equation. This results into reduction in the frequency of operation. As the Supply voltage reduction impacts performance, the other method of reducing dynamic power is to reduce the number of active transitions.

3.7.2 Reducing the switching activity

CLOCK GATING

In Clock Gating Technique, clock is gated whenever the output of the flip-flops is not changing, which effectively decreases the dynamic power consumed by flip-flops. Most clock gating is done at the Register Transfer Level (RTL). RTL clock-gating algorithms can be grouped into three categories: system-level, sequential and combinational. System-level clock-gating stops the clock for an entire block, effectively disabling all functionality. On the contrary, combinational and sequential clock-gating selectively suspend clocking while the block continues to produce output.

3.8 Leakage power optimization

Supply voltage must be reduced with the Technology scaling to restrain power density. To maintain circuit performance while scaling, the threshold voltage of the device must also be reduced. However, this causes the sub threshold current to increase exponentially since it is exponentially dependent on Threshold Voltage. And also the dynamic power is constant for a given switching activity and supply voltage. It is very difficult to optimize dynamic power without performance degradation. Therefore the leakage power which is due to transistor leakage currents must be optimized.

3.8.1 Leakage Reduction Methods

LL/UV devices reduce leakage by 4X. Z of circuit linearly depends on power. If somehow Z can be reduced for the Circuit, it can reduce power. Z can be reduced by avoiding duplications, Split templates and oversized standard cells. Automatic

ows that reduce leakage. One can improve the results significantly by improving the margin of big cones

3.9 Summary

In this chapter, the basics of Power, the different types of power dssipation that occur in design are explained and an Overview of power dissipation technique and optimisation iis explained.

Chapter 4

Delay Modeling

4.1 Introduction

An accurate delay model is a necessity for speed optimization. Often, in optimization routines, gate delays are treated as a fixed quantity, regardless of input slope and output load. There is also a tendency to associate a gate with only one delay, ignoring the difference between output rise (pull-up) and fall (pull-down) times. These issues, however, have been shown to affect the speed of a gate [12]. and hence need to be taken into account in delay estimation.

4.2 Delay Model

In this thesis, a "pin delay" model for delay is used. The output delay is modeled for each pin of a gate when it is the last one to change, or the one that will cause a switching event. This delay model introduces block and drive values for each input pin into the cell library characterization. Separate block and drive values are derived for rise and fall delay. Delay is estimated by:

$$\text{Gatedelay} = \text{blockdelay} + \text{Outputload}/\text{Outputdrive} \quad (4.1)$$

However, this delay model does not take input transition time into account. To verify this delay model and to investigate the effects of input transition time on output delay, cells from a standard cell library were simulated. Cells were laid out with different output load* to examine the dependence of delay on output load, as well as with different input loads (or the load the fanin node sees), to vary the input transition time Figure 4.1.

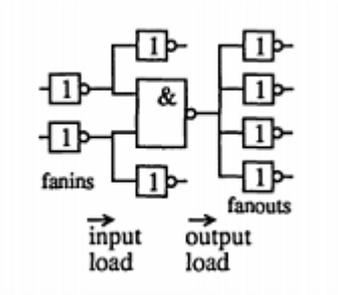


Figure 4.1: Nand gate with input load of two and output load of four

All cell* used in the design are static CMOS. The output load seen at a node is the sum of gate capacitances of all the p and n transistors it is connected to. The input load of a gate pin U the output load seen by the fanin node connected to this pin, and it includes the gate capacitances of the p and n transistors in the library cell that this pin leads to, as well as the gate capacitances of the transistors in other cells that are connected to the fanin node. All capacitive loads are measured in terms of the load of an inverter, or the capacitive load of a p-transistor and an n-transistor. The output of the gate was also connected to varying numbers of inverters to vary the output load.

The delay values were obtained for each pin in the gate by simulating this gate with the pin input as the one that will cause a transition at the output, other inputs remaining constant during this transition. These cells were simulated with HSPICE and their delay plotted against output load Figure 4.2.

For a given input transition time, delay is found to have a linear relation to output load. However, as input transition time is varied by varying the input load, delay is seen to shift upward substantially. This shows that input transition time cannot be ignored in the delay model.

To investigate the dependency of output delay on input transition time, delay is plotted against the input load Figure 4.3.

For a given output load, delay has a linear relation with input load as well. Recognizing the effect of input slope on output delay, a new parameter, the input drive, U added into the delay model. Delay is now estimated by:

$$\text{Gatedelay} = \text{Inputload}/\text{Inputdrive} + \text{blockdelay} + \text{Outputload}/\text{Outputdrive}(4.2)$$

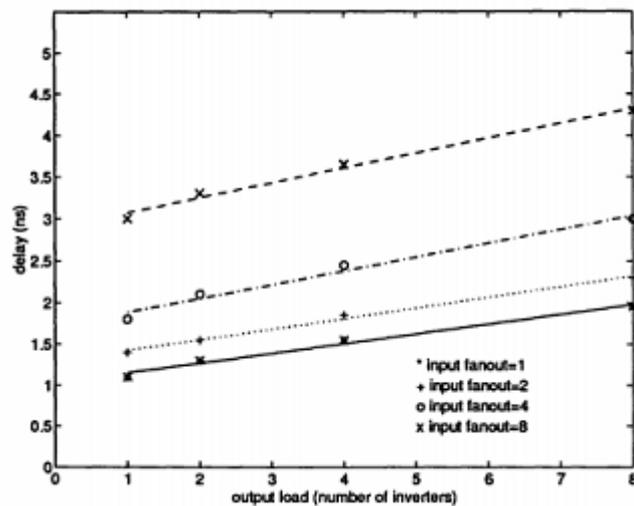


Figure 4.2: Fall delay of gate versus output load with different input transition times

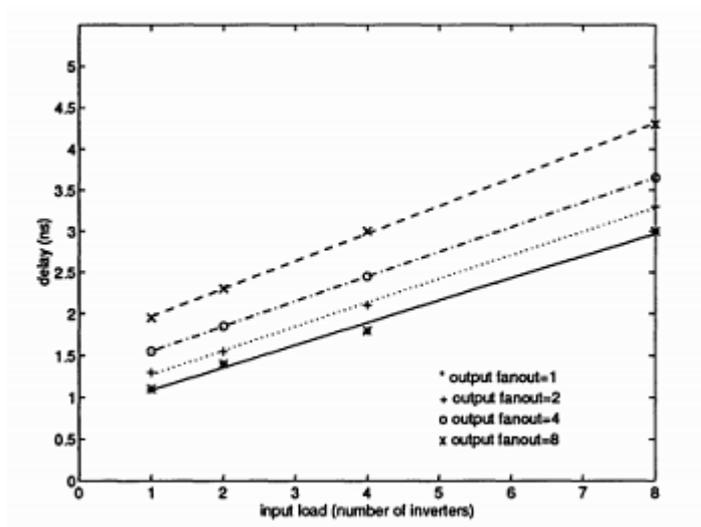


Figure 4.3: Fall delay of gate versus input load with different output loads

4.3 Data Point

To ensure that no overlap of delay calculations occurs between successive gates in a circuit, delay values are taken from a point in the input transition (e.g. when it reaches 50% of its final value) to a corresponding time in the output transition (e.g. when that output reaches 50% of its final value).

This method has been used in previous work to find delay. Brocco modelled gate delay as the time to reach 20% of final value after the input from the previous stage has reached 20% of its final value [12]. Kayssi used a similar method, using V_u as the data point. Weste and Eshraghian modelled the time taken for a logic transition to pass from input to output as the time difference between the 50.

To investigate the best data point with which to obtain delay values, voltage transition graphs of several gates were analyzed. The input drive, block delay and output drive were obtained with different data points. The voltage transition graph of a node is found to be initially very dependent on the rate of change of the input. Taking delay values from 25% (close to threshold voltage) of input transition to 25% of output transition results in a small input drive value, comparable with that of the output drive. As the data point is moved upwards from 25% to 50% to 70%. the input drive increases, while the output drive decreases. This indicates that the last portion of the transition graph is heavily dependent on the output load.

Analyzing the delay among several gates, the 50% point was found to be most consistent. Taking fall delay from the 25% data point yields a much larger delay than when using the 70% data point for certain gates, such as the 2-input nand gate, whereas for rise delay the 70% data point yielded a much larger delay. Delay values taken with the 50% point are in between the extremes and hence more reliable. The various data points were also tested on an adder circuit and a small nine-gate circuit. These delays were compared to that obtained by adding the gate delays, obtained from the corresponding data point, of all gates in the critical path. The best agreement between the two methods of calculating circuit delay was from the delays obtained with the 50% data point.

A problem that exists for such a data point as well as that for bigger levels, (e.g. 70%) is that for certain gates, the slope of the output transition is much steeper than that of the input transition, which means that output may reach 50% of its final value before input does, resulting in negative gate delay. Such negative gate delay* will not occur if the data point is close to the trigger value (20% or V_a).

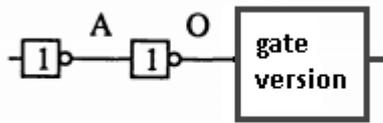


Figure 4.4: Capacitive load determination

For the 50% value point, negative delay were found to occur only for gates with very large input loads. Since large fanout loads cause large delays and are undesirable.

Many mapping algorithms disallow fanout to exceed a certain value or have options that utilize fanout optimization to improve delay. When large loads are rare, negative delays are rare when using the 50% data point.

The 50% data point is found to be the most consistent and accurate, hence delay values for the library cells are obtained using the 50% data point. Delay values are taken from the time input transition reaches 50% of its final value to the time when output transition reaches 50% of its final value.

4.4 Capacitive Load Extraction

Capacitive load has been shown by various work to be dependent on the width of transistors. However, parasitic and wiring capacitances contribute to the capacitive load as well. To ensure accurate delay estimation, capacitive load values need to be accurate, in addition to accurate block and drive values.

The dependence of capacitive load on the width of a transistor were investigated by laying out a chain of two inverters with a sized cell at the output Figure 4.4. The layouts were extracted into SPICE decks and simulated. Delay values were taken from point A to O.

The dependence of capacitive load on transistor width was found not to be directly proportional but linear Figure 4.5.

The constant offset is due to wiring capacitances, parasitic capacitances and other capacitive loads not associated with the gate, drain or source capacitances of a transistor. Capacitive input loads of each library cell are hence obtained by simulation of each sized gate with a chain of inverters as described above, to obtain accurate load values. This process, and the process of delay parameter extraction, are further

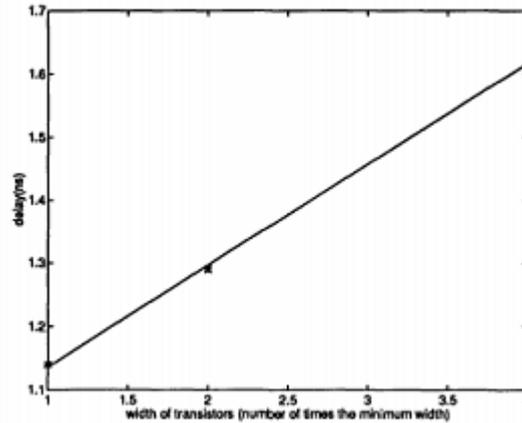


Figure 4.5: Delay of Inverter versus width of transistors (p and n)

described in section 4.5.

4.5 Circuit Delay

Circuit delay is estimated by calculating all gate arrival times (rise and fall calculated separately) using the pin delay model. The final circuit delay is obtained by the latest arriving primary output. Hence the delay of a circuit is defined by the delay of its largest delay path from input to output. This is the critical path. Decreasing the delay of the critical path will thus also decrease the delay of the circuit. Hence our approach to circuit delay reduction will be based on reducing the delay of the critical paths.

4.6 Summary

In this chapter, a detailed delay model that takes into account rise and fall times, varying pin delays and input transition times is used. Accurate capacitive load measurements are also done to ensure that delay calculations are accurate. Having a reliable delay model, we are now ready to proceed with establishing a good cell library and an optimization strategy to get the best power performance in a circuit for a given delay.

Chapter 5

Preparing the Z Advisory

5.1 Concept

Figure 5.1 is an example of the common structure found in any circuit block. FF A is driving a big fan-out cone and after passing through several combinational gates, it hits the fan-in cone ending into FF B. FF A drives gate G1 which is followed by big fan-out cone. All the inputs shown in green dotted line in the figure 1 are non-critical inputs from timing perspective for the corresponding gate. FF-A drives critical input to gate G1 which propagates to fan out cone through nets depicted in red. All the gates in fan-out cone of gate G1 remain in lower margin because of one critical input and cant get optimized for power when automatic tools are used. Once it is known for the fact that FF-A drives critical input for big fan-out cone, several timing optimization techniques like early clock for FF-A, Interconnect delay optimization for nets n0, n1, logic optimization or retiming may be used to improve margin. Once the margin through gate G1 is improved, traditional optimization tools, can use the margin to achieve optimal sizing for overall power reduction.

Logic on right side of Figure 5.1, is a fan-in cone ending in FF-B. Lets assume that output shown in green dotted lines is very relaxed in timing and setup time requirement at FF-B is creating tight required for gates in fan-in cone traced through nets shown in brown. In such a case, clock push for FF-B may improve margin for gates in fan-in cone and more number of gates will get optimized for power.

This advisory aims to find such cases from the schematic and timing data of a given block. In general if the timing path is positive, it doesnt attract attention of designer for further optimization. But as decided by the project methodologies, a

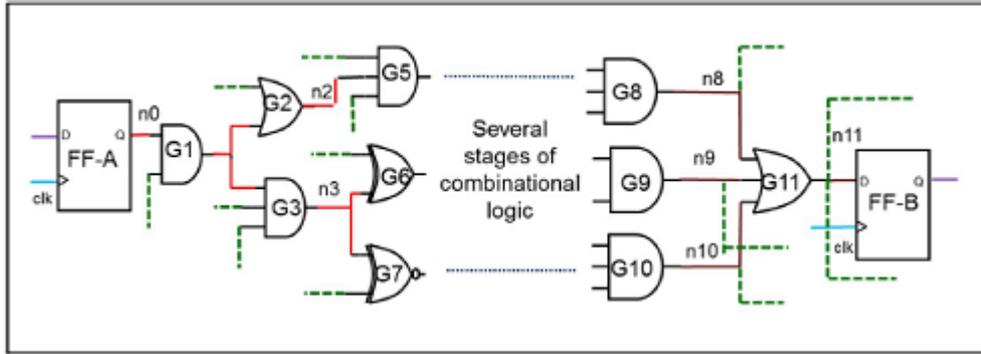


Figure 5.1: Examples of Fan-in, Fan-out cones found in circuit

small or barely positive margin may not be good enough to optimize all the gates in timing path for power reduction. This advisory presented here, aims to direct designers effort to the high ROI nodes where a relatively small timing improvement may yield reduction in power.

5.2 The Algorithm

The Advisory is created by tracing through the block schematic and timing data. Schematic tracing can be done either from start points of path or end points of timing path. Start points of path can be clock pin of sequential elements or Interface inputs for the block in case of hierarchical designs. Similarly a timing path can end at either setup check due to clock of a sequential element or at an output interface. Timing path always starts or end at Flip-Flop. For latches, if they are transparent, then they act as delay element in the timing path. For non-transparent latches in design they are two possibilities. Data can arrive even before latch open, timing at output of latch is decided by the opening edge of clock and new timing path starts from output of the latch. If data is not reaching to the latch before it closes, then timing path ends there due to setup time requirement at latch.

The cone found while tracing forward from the start point of the path as valid cone. Similarly, cone found while tracing backward form the output side is called required cone. In this report, the arrival time of data at particular pin has been termed as as valid 'V' for the pin. Similarly, time at which data is needed at particular pin is termed as required 'R'.

While tracing valid cone, first all the gates connected to the input are found

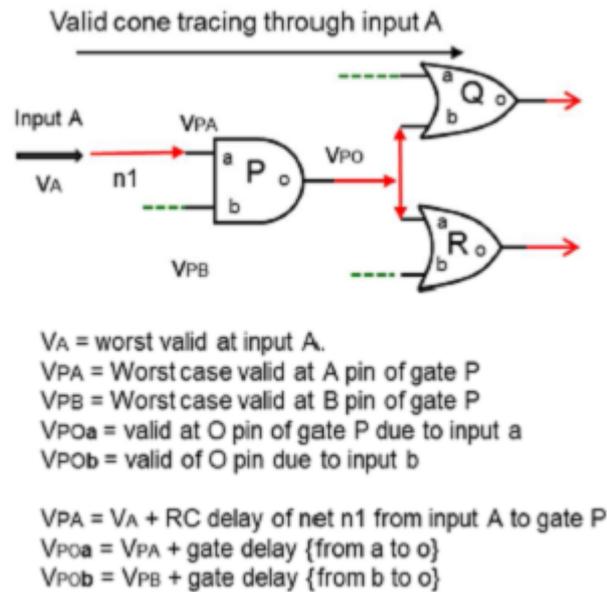


Figure 5.2: Valid Cone Propagation

using schematic data. Further all the valid's are found at the output of each gate propagated from the start point of interest. If the valid due to a particular start point is the worst valid of all, then the gate is considered to be controlled by that start point and forward tracing is continued. If the gate output gets worst valid due to other signal than the particular start point, then it is concluded that gate is not being controlled by start point and no forward trace is done in that fan-out. While tracing forward, if the end point of path is reached, then tracing is stopped further. Similarly, to determine the required cone, required propagation in backward direction is carried out. The gates which get worst required due to end point of interest, get added to the required cone list and cone tracing continues till either start point of path is reached or all the gates in fan-in cone gets worst required due to signal other than end point of interest. Figure 5.2 and Figure 5.3 below explains how to find valid and required cone.

a. Valid Cone propagation Criterion

If { V_{poa} is worst than V_{pob} }

{

Gate P's margin is controlled by input A;

Continue forward tracing;

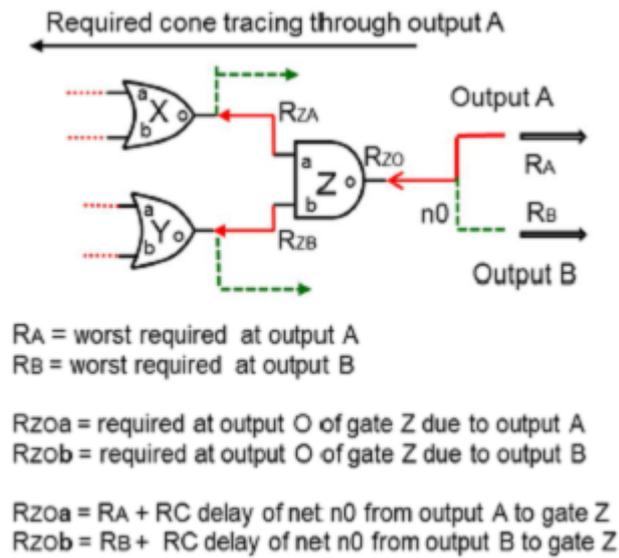


Figure 5.3: RequiredConePropagation

```

Check output of next gates Q and R for cone propagation
}
Else
{
Gate P's margin is not controlled by input A;
Stop forward tracing beyond gate P
}

```

b. Required cone propagation Criterion

```

If { Rzoa is worst than Rzob }
{
Output A controls the margin for Gate Z;
Continue backward tracing;
Check output of gate X,Y for propagation
}
Else
{
Output A doesn't control margin for Gate Z;
stop backward tracing beyond gate Z
}

```

}

It may be also noted that advisory is not just about finding big fan-in and fan-out cones. There exists fan-in, fan-out cone reports in the design tools based on just schematic tracing. With timing based filtering as mentioned above, advisory highlights real high ROI nodes in design. Also for this current study, only leakage has been assumed as power cost. But this can extend this concept with addition of dynamic power using available activity factor data. In such case, circuit nodes can be found which doesn't connect to big fan-in or fan-out logic but still controls margin for high power consuming gates.

5.3 The Output of advisory

Table I below shows an example of data given to user as a result of advisory. Term Z denotes width of transistor. UV denotes a high V_t (low leakage) transistor. As per following table, it is clear that the designer should focus on improving timing margin from input a rather than input b and c, since input a controls margin for a bigger chunk of logic (600 microns) as indicated by Non-UV Z and total Z of the cone. Similarly between b and c, designer should prefer to improve input b because it controls more (10) number of parallel timing paths. It may be noted that, one can also add the dynamic power of the cone in table by making use of activity factor data for each node. The designer may then choose the signal that may help create margin to downsizing opportunities in high dynamic power cones.

Startpoint Name	Timing Margin at start/end point	Non-UV Z in the cone	Total Z of the cone	Number of timing paths in the cone
X%a	10ps	600u	900u	20
X%b	10ps	300u	450u	10
X%c	10ps	300u	450u	5

Table I: Example of data given to user by advisory

5.4 The Onion Peeling Phenomenon

5.4.1 Onion Peeling

Figure 5.4 below describes the problem of Onion Peeling. As shown in Figure 5.4, a large fan-out cone starts from output of gate G1. Now let's assume a case in which

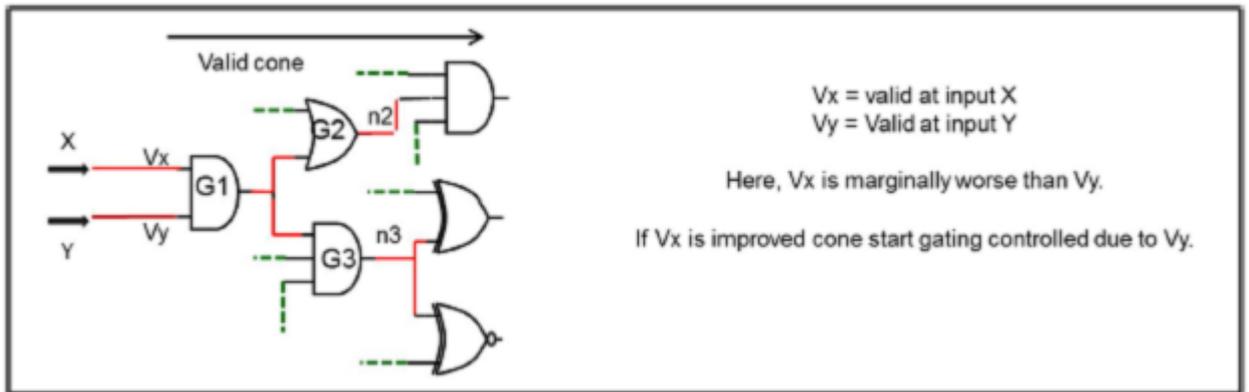


Figure 5.4: Onion Peeling Effect

valid values at the input 'X' and Y are closely matching but valid at 'X' is worse. In such a case, advisory will indicate that, input 'X' is controlling margin for all gates in fan-out of G1. However, even after improving valid at X input, margin for all gates in fan-out will be controlled by Y and they can't be optimized for power. This onion peeling effect happens due to the fact that valid at Y was only marginally better than valid of 'X'.

5.4.2 Workaround for Onion Peeling

To counter the Onion Peeling effect, it was decided to add timing guard band in cone propagation criterion. For example in figure 5.4 while tracing the cone from input 'X' through gate G1, if we found V_x to be marginally worse than V_y , then the forward tracing may be terminated. This approach will guard against the onion peeling and optimistic reporting of start-points. However, there may be a possibility of improving both V_x and V_y to achieve the intended power reduction. Also, it is desirable to give the designer more credible information about possible power gain. To satisfy above two requirements, the Z profile of the block is introduced and a what-if capability as explained next.

Table II is one such example of Z profile of block. Each column indicates the amount of Z residing in a particular timing margin range.

Designer creates Z-profile of the block before using the advisory results. Once advisory is run, designer selects a few high ROI start/end points for improving timing margin. Based on design knowledge, the designer may prune the list based on feasi-

MarginRange(ps)	-inf to -20	-20 to -10	-10 to 0	0 to 10	10 to 20	20 to 30	30 to 40	50 to inf
Z(u)	100	200	300	400	1000	1500	1000	

Table II: Typical Z profile of block

bility of timing margin improvement. Then, improvement will be modeled through override commands on timing database and create new incremental timing database. Next, the designer can run Z profile once again to check if there is indeed improvement in Z-profile of the block i.e. more amount of Z reported in higher margin range. Once designer implements the changes, he will find expected amount of Z moving to a higher margin and achieve power reduction by running traditional LR-sizing tool on the new database.

Onion Peeling Problem can be addressed partially with running Z-profile and advisory iteratively. For the case mentioned in figure 5.4, after first iteration designer runs Z-profile by modeling timing improvement only for input X. Designer doesn't see expected change in Z profile since now input Y creates low margin for the cone. If designer runs advisory one more time on timing database where improvement at 'X' is modeled, designer will find input Y controlling next big cone. Later in next iteration, modeling improvement for both 'X' and 'Y', designer will be able to verify expected change in Z profile.

5.5 Summary

In this chapter, the detailed approach to the proposed work has been explained along with the results of the implementation. Also, the problem with preparing the cone has been explained and a work around algorithm for that has been proposed.

Chapter 6

Results

6.1 Test Results

The prepared advisory was run on all the blocks of Execution cluster of 14 nm Core and then sorted results of the advisory for each block in the descending order of Non-UV Z of the cone. Figure 6.1 shows graph of percentage of total Z controlled by top five start/end points for different blocks. Such top five start/end points of valid /required cone may be checked for timing improvement via RC delay improvement, logic optimization or any other conventional optimization techniques. It may be noted that for handful of nets, RC delay can be improved even in routing congested design by shielding or widening the routes. It is quite possible that for certain cases, designer would have already explored all the timing improvement options and no more further optimization options may be possible. Intention of the advisory is to bring all such high ROI cases to the attention of designer. Also from Figure 6.1, it is evident that for most of the blocks, there exists handful of start/end points of path which controls timing margin for significant Z portion of the blocks logic. Hence, there is merit in making this advisory to be part of design flow. Next, a detailed discussion about experiments on two blocks FUB A and FUB B is presented.

At the time of working, most of the FUB blocks have already made use of all available timing-power optimization tools. The starting database for the experiment had most detailed power optimization already done with available timing margins.

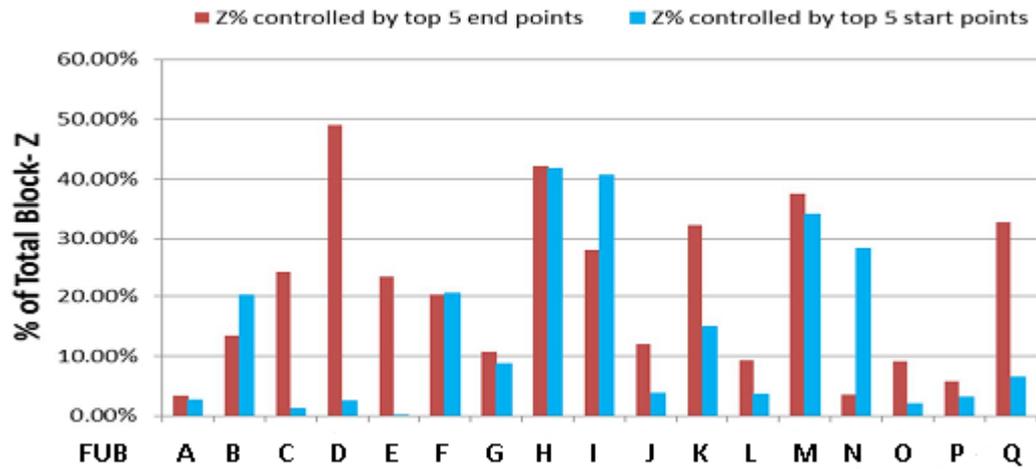


Figure 6.1: Graph of % of total block-Z controlled by different blocks

6.2 Steps of Using the Advisory

Following steps were followed on the FUB to find high ROI start/end points of path using advisory:

- Run advisory on latest timing data-base of the block
- Sort the advisory result in descending order of Non-UV Z in the valid or required cone.
- Short-list a few start/end points of path where timing can be improved and ROI is high.
- Designer models the timing improvement through timing override commands and compares Z profile of the block before and after modeling the timing improvement.
- If designer finds expected amount of logic moving to more positive margin, timing improvement is actually implemented on the database.
- On the new database where timing is improved for selected high ROI start/end points, designer runs optimization tools to achieve desired power reduction by downsizing gates or by adding more low-leakage cells to the design.

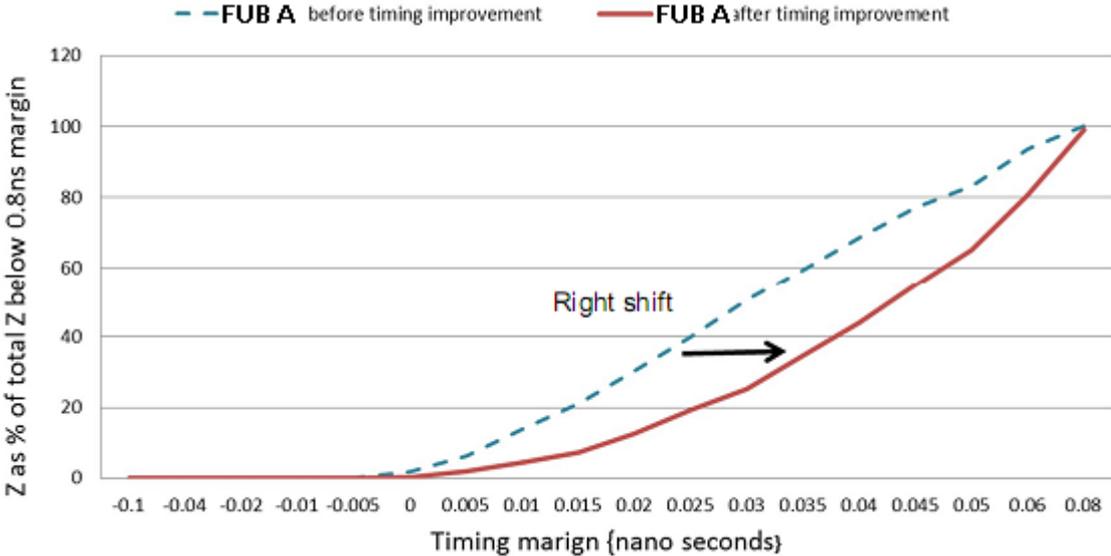


Figure 6.2: Z distribution for FUB A, before and after timing improvement

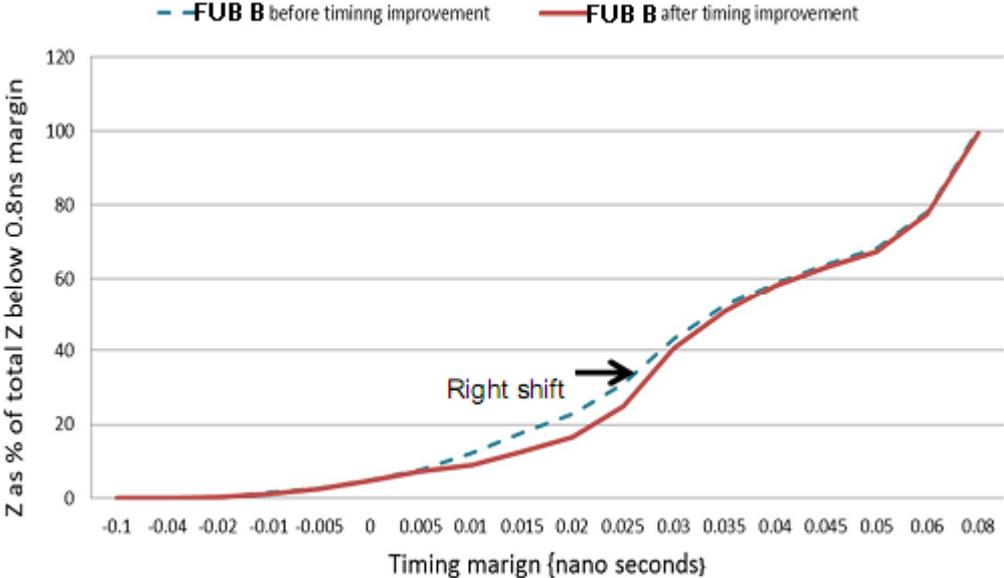


Figure 6.3: Z distribution for FUB B, before and after timing improvement

6.2.1 Detailed Result

Figure 6.2, Figure 6.3 shows, change in Z distribution, before and after timing improvement for blocks FUB A and FUB B respectively. Timing margin are on the horizontal axis in nano seconds. Vertical axis represents amount of Z with less margin than corresponding timing range as percentage total Z below 80 ps margin. For 14 nm Core blocks, threshold for downsizing is kept at 25 ps. For converting nominal cells to low leakage cells, threshold is chosen at 10 ps margin. It can be seen from the both the graph that with the right-shift of Z distribution after timing improvement good amount of logic cells will move to higher margin than the project defined power optimization thresholds. These cells now can be optimized by automatic optimization tools. With improved timing margins, the advisory was helpful to increase low leakage cell count by 30% FUB A and 8% for FUB B. Better gain in case of FUB A is also evident from the graphs in Figure 6.2, Figure6.3 as it is seen more to right-shift in Z distribution for FUB A.

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

LR based optimization tools achieve optimum power reduction making use of the available timing margin. To achieve further power reduction, designers need to create extra positive margin, which may be hidden possibly due to sub-optimal clocking or other such reasons. In this report, our work was presented on novel approach of creating additional positive timing margin at the right places to enable automatic tools to take the best advantage. Advisory presented in this report lists high ROI start/end points of timing paths, where relatively small timing improvements may lead to significant number logic cells moving into higher margin. During our experiments on Functional Unit blocks, it was found that for each block there exist relatively few signals which control the margins for large cones of logic. Designers were able to improve timing on a few high ROI nodes resulting into addition in low leakage cells count in the design.

7.2 Future Scope

In future, while defining power ROI per start/end points of path, dynamic power for the valid and required cone can also be added as a part of ROI calculation. Data generated by such advisory can find more usages in circuit design flows. In case of hierarchical design, advisory can be used to find the power cost for all the interface nodes. Knowledge of power cost on each interface nodes can help the designer to distribute positive timing margin between blocks. In synthesis based design RLS, data

from the advisory can be fed back to synthesis tools for further power optimization. Knowledge of valid and required cone from the input and output pin of sequential element can help in tuning clock arrival for optimum margin distribution. The authors intend to start work on developing above mentioned usages of the advisory in immediate future.

References

- [1] W.Goh, S.Rofail, and K.Yeo, *Low-Power Design: An Overview*, Prentice Hall.
- [2] D.Soudris, C.Piguet, and C.Goutis, *Designing CMOS Circuits for Low Power*, Kluwer Academic Publishers, 2002.
- [3] F.Najm, A Survey of Power Estimation Techniques in VLSI Circuits, *IEEE Transaction on VLSI Systems*, vol.2, pp.446-455, 1994.
- [4] K.Roy and S.Prasad, *Low Power CMOS VLSI: Circuit Design*, John Wiley and Sons, 1999.
- [5] P.Landman, High-level power estimation in Proc. Int. Symp. Low Power Electronics and Design, pp.29-35, Monterey, CA, Aug.12-14, 1996.
- [6] Subodh Gupta and Farid N. Najm., Power Modeling for High-Level Power Estimation, *IEEE Transaction on VLSI Systems*, pp.18-29, Feb.2000.
- [7] A.Chandrakasan and R.W.Brodersen, *Low Power Digital CMOS Design*, Kluwer Academic Publishers, 1995.
- [8] A.Raghunathan, N.K.Jha and S.Dey, *High-Level Power Analysis and Optimization*, Kluwer Academic Publishers, 1998.
- [9] A.Chandrakasan, S.Sheng and R.Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, Vol.27, No.4, April 1992.
- [10] M.Berkelaar, "Area-Power-Delay Trade-off in Logic Synthesis," Ph.D.dissertation, Technische Universiteit Eindhoven, September 1992.
- [11] V.Tiwari, P.Ashar, S.Malik, "Technology Mapping for Low Power," in *Proceedings of the 30th Design Automation Conference*, pg.74-79, 1993.

- [12] H.Veendrick, "Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits, "IEEE Journal of Solid-State Circuits, folsc-19, No.4, August1984.
- [13] A.Kayssi, K.Sakallah and T.Mudge, "The Impact of Signal Transition Time on Path Delay Computation, "IEEE Transactions on Circuits and Systems II : Analog and Digital Signal Processing, Vol.40, No.5, May1993.
- [14] Chen,C.P, Chu,C.C, Wong, D.F, Fast and Exact Simultaneous Gate and Wire Sizing by Lagrangian Relaxation, IEEE transactions on Computer-aided Design of Integrated Circuits and systems, July1999.
- [15] Intel Wikipedia Site, <https://intelpedia.intel.com>.

Index

- Summary, 23
- Abstract, vii
- Acknowledgements, vi
- Activity Factor, 14
- AF File, 14
- ASIC Design Approach, 3
- AVERAGE VALIDITY, 15
- Basic Concepts of Power, 11
- CAP QUALITY, 15
- Capacitive Load Extraction, 22
- Circuit Delay, 23
- CLOCK GATING, 16
- Concept, 24
- Data Point, 21
- Datapath v/s RLS(RTL to Layout Synthesis), 4
- Declaration, iii
- Delay Model, 18
- Delay Modeling, 18
- Design Flow with and without Power, 7
- Detailed Result, 34
- Diff Of AF, 14
- DRIVEN INPUTS, 15
- Dynamic Power, 12
- Dynamic Power Optimization, 15
- Full Custom Integrated Circuit Design, 3
- Hierarchy of core, 2
- Hierarchy of Server/Client Core Design, 2
- Intel Certificate, v
- Internal power, 12
- Introduction, 1, 18
- Leakage Power, 13
- Leakage power optimization, 16
- Leakage Reduction Methods, 16
- Literature Survey, 6
- Need for low power design, 6
- Need for manual implementation, 4
- Onion Peeling, 28
- Power Estimation Factors, 14
- POWER ESTIMATION QUALITY METRIC, 15
- Preparing the Z Advisory, 24
- Reducing the supply voltage, 16
- Reducing the switching activity, 16
- Relationship Between Digital Design Abstraction Levels and Power Estimation, 9
- Report Organization, 5
- Semicustom Integrated Circuit Design, 3
- Short-Circuit Power, 13
- Signal Probability, 14

Static Power, 11

Steps of Using the Advisory, 32

Sub threshold leakage, 13

Summary, 10, 17, 30

Switching power, 12

The Algorithm, 25

The Onion Peeling Phenomenon, 28

The Output of advisory, 28

VALUE CHANGE DUMP (VCD) FILE,
14

Workaround for Onion Peeling, 29