Repeater Optimization Methodologies for Custom CPU Designs

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Electronics & Communication Engineering (Embedded Systems)

By

Anup Singh Parihar (13MECE01)



Electronics & Communication Engineering Branch Electrical Engineering Department Institute Of Technology Nirma University Ahmedabad-382481 May 2015

Repeater Optimization Methodologies for Custom CPU Designs

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

 \mathbf{in}

Electronics & Communication Engineering (Embedded Systems)

By

Anup Singh Parihar (13MECE01)

Under the Guidance of

Mr. Saurabh Sharma Engineering Manager Intel India Pvt. Ltd. Dr. N. P. Gajjar Professor, EC Nirma University



Electronics & Communication Engineering Branch Electrical Engineering Department Institute Of Technology Nirma University Ahmedabad-382481 May 2015

Declaration

This is to certify that

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

- Anup Singh Parihar 13MECE01

Disclaimer

"The content of this paper does not represent the technology, opinions, beliefs, or positions of Intel Technology India Pvt. Ltd. Company, its employees, vendors, customers, or associates."



Certificate

This is to certify that the Major Project entitled " Repeater Optimization Methodologies for Custom CPU Designs " submitted by Anup Singh Parihar (13MECE01), towards the partial fulfillment of the requirements for the degree of Master of Technology in Embedded System (Electronics & Communication Engineering) of 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 the 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

Internal Guide & Course Co-ordinator	Section Head, EC		
Dr. N.P. Gajjar	Dr. D.K. Kothari		
(Professor,EC)	(Professor,EC)		
HOD	Director		
Dr. P. N. Tekwani	Dr. K. Kotecha		
(Professor, EE)	(Director, IT-NU)		



Intel Technology India Pvt. Ltd.

Certificate

This is to certify that the Project entitled "**Repeater Optimization Methodologies for Custom CPU Designs** " submitted by **Anup Singh Parihar** (13MECE01), towards the submission of the Project for 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.

> - Mr. Saurabh Sharma Engineering Manager, Backend Integration, Big-Core India, Intel India Pvt. Ltd., Bangalore.

Acknowledgements

With immense pleasure, I would like to present this thesis on the work related to "Repeater Optimization Methodologies for Custom CPU Designs". I am very thankful to all those who helped me for the successful completion of the first phase of the dissertation and for providing valuable guidance throughout the project work.

I would first of all like to offer thanks to **Dr. N. P. Gajjar**, PG Coordinator M.Tech. Embeddded System, Institute of Technology, Nirma University whose keen interest and excellent knowledge base helped me to finalize the Thesis. I would offer thanks to **Dr. P.N. Tekwani**, Head of Electrical Engineering Department and our director **Dr. K. Kotecha** for allowing me to undertake this thesis work.

I would offer thanks to my manager **Mr.Saurabh Sharma**, Engineering Manager, Big-Core India, Intel India. His constant support and interest in the subject equipped me with great understanding of different aspects of the required architecture for the project work. He has shown keen interest in this dissertation work right from beginning and has been a great motivating factor in outlining the flow of my work. My sincere thanks and gratitude to **Mr. Rajat Gupta**, Engineering Manager, Intel India and **Mr. Tejinder Singh Syan**, Engineering Manager, Intel India for their continual kind words of encouragement and motivation throughout the internship.

I thank the Almighty, my family, for supporting and encouraging me in all possible ways. I would also thank the **Big Core-India** Team and all my friends who have directly or indirectly helped me making this work successful.

> - Anup Singh Parihar 13MECE01

Abstract

In VLSI circuits, delay and power dissipation are the two major design constraints. The millions of devices in active state and interconnects connecting a large number of devices on chip are responsible for these problems. The repeaters are inserted in long interconnects in the VLSI circuits to reduce delay. Repeater insertion is done for the timing optimization of interconnects. But the use of repeater implies a significant cost in power and area. Repeaters consume a large part of the chip resources (area and power). Thus, there is a need of area and power optimization of the interconnect with repeaters in high speed VLSI circuits. The work carried out discusses area and power optimization of repeaters inserted in interconnects in the CPU Designs without violating the timing constraints.

The proposed methodology aims to come with a flow which does the Repeaters Optimization which is aware of the Timing Margins as well as it gives area and leakage reduction, thus saving both area and power in the design without violating the timing requirements. This repeaters optimization methodology has been implemented for different PVT (Process-Voltage-Temperature) corners : single corner as well as multi-corner (nominal and highv) to converge the timing for the two design corners after optimizing the repeaters.

Repeaters are used to repeat long interconnects to sustain the slopes according to project targets. Extractions as well as interconnect delay calculation tools support this well-established methodology. Simulator tool has traditionally been used to optimize these repeaters. However, running Repeater optimization flow using Simulator to downsize the repeater or swap to Low-leakage cells may cause timing miscorrelation between different PVT corners that is Nominal and High-Voltage corners. This is due to the fact that the Simulator optimization flow works on user provided timing specs that are generally generated using Nominal corner data alone. This approach would have worked well in the previous generation of projects where HighV convergence was not much of an issue, but in the current technologies (very low scaled down technologies) where interconnect RCs play a key role, the timing paths between the two corners can be different, depending on whether they are Gate-Capacitance dominated or whether they are RC-dominated. Due to this uniqueness of timing critical paths across corners, it is imperative that any optimization tool should consider the worst case timing margins across corners.

This technique suggest one of the workarounds that can be used to generate Repeater Optimization solution by considering the multi-corner worst case timing margins. The comparison of large-sized repeaters vs small-sized repeaters is done and the optimum (small-sized) repeater is used in the design. The comparison of low-leakage repeaters vs high-leakage repeaters for interconnects in the CPU design is analyzed, and the optimum repeater is used. All these optimizations are done without violating the timing constraints. The complete tool flow is automated and GUI is created for the selection of optimum repeaters used in the design. After optimizing the repeaters, there has been a significant saving in the area for the repeaters, on an average by about **15%-20%** depending upon the section, margin threshold value and slope threshold value. This approach causes about **three times** increase in the low-leakage repeater cell's count.

Abbreviation Notation and Nomenclature

VLSI Very Large Scale Integrated Circuit	its
SoCSystem-on-Ch	ip
SVTStaggered-V	Vt
DTD	no
LVTLow-V	Vt
CPU	nit
ITRS International Technology Roadmap for Semiconducto	\mathbf{rs}
MVRC	ck
UPF	at
DRC	ck
RLS	sis
RTL	gic
LVSLayout Versus Schemat	tic
CPPRCommon Path Pessimism Remova	al

Contents

Dec	clar	ation	iii
\mathbf{Dis}	clai	ner	iv
Cer	rtifi	ate	\mathbf{v}
Inte	el C	ertificate	vi
Ack	knov	vledgements	vii
Abs	stra	ct	viii
Abl	brev	iation Notation and Nomenclature	x
List	t of	Tables	xiii
List	t of	Figures	1
1	Intr 1.1 1.2 1.3	Dduction Motivation Problem Definition Thesis Organization	2 2 4 6
2	Lite 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8	rature SurveyIntroduction	7 8 11 11 11 14 15 16 16 16 17 18

	2.9	Summary	18
3	Typ	bes Of Repeaters	19
	3.1	Inverter or Buffer	19
		3.1.1 Low Vt Repeater(LVT) \ldots \ldots \ldots \ldots \ldots \ldots \ldots	19
		3.1.2 High Vt Repeater(HVT)	20
	3.2	SVT: Staggered Vt Repeaters	20
	3.3	Dual - Vt Domino (DTD) Repeaters	21
	3.4	Summary	22
4	Imp	plementation Methodology	23
	4.1	Introduction	23
	4.2	Implementation Flow for Repeaters Optimization	24
		4.2.1 Explanation: \ldots	24
		4.2.2 Margin Parser Script:	25
		4.2.3 Step 1: Preparation of Specification file	26
		4.2.4 Step 2: Creation of Repeater file	29
		4.2.5 Step 3: Annotating repeater file to the layout (design)	30
		4.2.6 Step 4: Full Chip Interconnect's RC extraction and Running	
		FC Timing	30
	4.3	RPT OPT Flow GUI	31
	4.4	Repeaters Used in the Design	32
	4.5	Results, Analysis and Profiling	34
		4.5.1 Repeater Profiling	34
	4.6	Analysis	36
	4.7	Results :	43
	4.8	Summary	44
5	Cor	nclusion	46
	5.1	Conclusion	46

List of Tables

Ι	Format of Margin File	25
II	Format of Specification File	28
III	Format of Repeater File	29
IV	Repeaters used in the Design	34
V	Number of repeaters present in the central/reference repeater file (in	
	percentage)	35
VI	Format of Histogram : output of repeater profiler	36
VII	Table showing Types of Repeaters and the No. of Repeaters (in per-	
	centage) for different Design Cases	38
VIII	Table showing Number of Repeaters (in %age) for a particular repeater	
	type Before Repeater Optimization and After Repeater Optimization	40
IX	Table Showing comparison of RPT downsize and LL %age on one of	
	the section	43

List of Figures

2.1	Interconnects and scaling: local and global interconnects [7]	10
2.2	Schematic cross-section of backend structure, showing interconnects,	
	contacts and vias, separated by dielectric layers [10]	10
2.3	Scaling of a chip and interconnections [10]	12
2.4	CMOS buffer driving a wire load and its equivalent representation [5].	12
2.5	RC and RLC lumped model representations of an interconnect line [5].	13
2.6	Delay of wire without repeater growing exponentially [5]	13
2.7	Wire with repeaters	14
2.8	Delay grows Linear after repeater insertion [9]	14
2.9	Power Delay Tradeoff	17
3.1	SVT repeater in 0-state low leakage [6]	20
3.2	DTD repeaters $[8]$	21
4.1	Implementation flow chart for Repeaters Optimization Methodology .	27
4.2	Flow of Margin Parser script	27
4.3	Logic of creating the Cross-corner Margin File	27
4.4	RPT OPT Flow GUI	32
4.5	Repeater Profiler script	39
4.6	Graph showing Repeater Types Vs No. of Repeaters(in percentage)	
	for central/reference design	39
4.7	Graph showing Repeater Types Vs No. of Repeaters(in percentage)	
	for Nominal case	39
4.8	Graph showing Repeater Types VS No. of Repeaters(in percentage)	
	for HighV design case	41
4.9	Graph showing Repeater Types Vs No. of Repeaters(in percentage)	
	for Cross-Corner design case	41
4.10	Graph showing comparison Before Repeater Optimization and After	
	Repeater Optimization	42
4.11	Graph Showing Comparison between all the design cases	42

Chapter 1

Introduction

1.1 Motivation

The VLSI circuits are aggressively scaled and the performance of these ICs is being increasingly dominated by the global interconnects. As the technology scales, more functionalities are being integrated on to the chip which results in an increase in the die size. Because of this the number of long global lines increases with the scaling of technology.

The delay of a long unbuffered (without repeater) line is quadratic in its length, so long interconnects are divided into a number of segments with repeaters or buffers. For an optimally buffered (with repeater) line the delay is linear in its length. For a large high-performance designs, the number of such repeaters can be prohibitively high and can take up significant fraction of active silicon and routing area. *Thus, optimizing the repeater's size can play a significant role in saving the area in the design.*

Generally, the repeaters are optimally separated and sized to minimize the interconnect delay. However, since these optimally sized repeaters are quite large and also

CHAPTER 1. INTRODUCTION

dissipate a significant amount of power, the total power dissipation by such repeaters in large high-performance designs can be prohibitively high.

Since, all global interconnects are not on the critical path, a small delay penalty can be tolerated on these non-critical interconnects and there exists a potential for large power savings and area saving by using smaller repeaters and larger inter-repeater interconnect lengths.

The literature survey shows that the leakage power dissipation becomes the dominating component of the total power dissipation, and thus reducing the repeater size and the number of repeaters results in large power savings, but at the same the timing constraints should be met, the timing should not be violated. [3].

The short-circuit and leakage power are important components of the total power dissipation and ignoring them in power optimization can lead to errors. Short-circuit power becomes important as the allowed delay penalty increases. Since rise time of the signal increases. Similarly, leakage power increases exponentially with device scaling and is the dominant component of power dissipation for 50-nm technology node. It has been surveyed that for larger technology nodes like for 180-nm and 130-nm technology nodes the leakage power is not significant, the relative power saving is almost the same for a given delay penalty. But for smaller nodes, beyond 130-nm, the leakage power becomes significant and therefore the relative power savings increase with technology scaling for a given delay penalty [3].

The purpose of this project work is two fold. First, to do a bibliographic literature survey to study the interconnects, repeaters, different types of repeaters, challenges in repeater insertion, how the area and power can be saved for repeaters by optimizing them with respect to their size and threshold voltage. Second, to automate the tool flow for the cross-corner design case by converging the two designs (HighV and Nominal), and implement the repeater optimization methodology, to select the optimum repeater for the interconnects in the CPU design. Thus, saving significant area and power after the repeater optimization.

1.2 Problem Definition

Problem: Generally, a design is made for the following three design cases, each using different voltage and frequency:

- 1. Nominal case
- 2. HighV case
- 3. LowV case.

There is a flow for the creation of design model. The flow can be run for any one of the above three cases to create a model, due to this there are corner cases where the HighV, Nominal and LowV design cases do not converge for some paths, for these paths area wastage takes place, at the same time more power is being dissipated as the optimum repeater is not being used. For example: for a HighV Design case: some interconnect may have used a particular repeater, but in the Nominal or LowV Design cases the same interconnect may be using different repeater having different size and type. So, during the convergence of the designs the problem arises. The repeaters in the design are not optimized with respect to area and power.

Thus, there is problem in converging the two designs (called multi-corner case) and there is wastage of area and power as optimum repeater is not being used for the interconnect. In an interconnect a large-sized repeater can be replaced by a smallsized repeater. An optimum (small size and low power) repeater can be used for the interconnect if we have knowledge about both the design cases i.e. highV and Nominal case (creating a converged design). This project aims at using the optimum repeaters w.r.t. area and power for the converged design (multi-corner case where HighV design and LowV design are converged) and thus optimizing the CPU designs.

Solution: Repeaters are used to repeat long interconnects to sustain the slopes according to project targets. Extractions as well as interconnect delay calculation tools support this well-established methodology. Simulator tool has traditionally been used to insert and optimize these repeaters. However, running Repeater optimization flow using Simulator to downsize the repeater or swap to Low-leakage cells may cause timing mis-correlation between different PVT corners that is Nominal and High-Voltage corners. This is due to the fact that the Simulator optimization flow works on user provided timing specs that are generally generated using Nominal corner data alone. This approach would have worked well in the previous generation of projects where HighV convergence was not much of an issue, but in the current technologies (very low scaled down technologies) where interconnect RC's play a key role, the timing paths between the two corners can be different, depending on whether they are Gate-Cap dominated or whether they are RC-dominated. Due to this uniqueness of timing critical paths across corners, it is imperative that any optimization tool should consider the worst case timing margins across corners. This dissertation work suggest one of the workarounds that can be used to generate Repeater Optimization solution by considering the multi-corner worst case timing margins.

The comparison of large-sized repeaters vs small-sized repeaters is being done and the optimum (small-sized) repeater is used in the design. Also, the comparison of low-leakage repeaters vs high-leakage repeaters for interconnects in the CPU design is analysed, and the optimum repeater is used. All these optimizations are done without violating the timing constraints. The whole tool flow is automated and GUI is created for the selection of optimum repeaters used in the design.

1.3 Thesis Organization

The rest of the thesis is organized as follows.

- Chapter 2, Literature Survey, gives overview of the basics of wires, global interconnects, wires without repeaters, basics of repeaters, types of repeaters, different challenges which are being faced during the repeater insertion.
- Chapter 3, Types of Repeaters, highlights the different types of repeater, it can be a simple buffer or an inverter. Other repeaters can be LVT, HVT, SVT, SR,etc
- Chapter 4, Implementation Methodology, enumerates the implementation method adopted for the repeaters optimization in the custom CPU design, work done till now, creation of histogram for analysis, tabular and graphical analysis, the results, etc.

Finally, in the **chapter 5** the concluding remarks are presented.

Chapter 2

Literature Survey

2.1 Introduction

Over the years, the VLSI technology is advancing and the minimum feature (technology nodes) size is decreasing. Because of this, both the size of die and the density of devices of the VLSI circuits are increasing. In the VLSI chips with the increase in die size, there is increase in the long interconnect lines. These long interconnects in the VLSI chip lead to high propagation delays [5]. In order to keep pace with the required speed, the Buffers are required. The buffers or inverters drive the capacitive load. A single inverter or buffer is not used for driving long interconnects, because they present very large RC load to the gate(s) connected to the interconnect.

Thus, a number of buffers (or inverters) are inserted after regular intervals of distance in interconnect, and these are called as repeaters [5]. Therefore, in long interconnects insertion of repeaters at optimum distance along the line reduces delay significantly. The repeaters are inserted to decrease the interconnect response time by mollifying the effect of RC. Long interconnects lead to more RC and thus more increased load, thus this leads to more excessive power dissipation, and this power dissipation is a very important issue in VLSI circuits. For portable battery operated devices and modern systems, there is Low-power requirement. This Low-power requirement has always been demanding the need for voltage scaling method for VLSI design. The reason for this is that the power dissipation is directly proportional to the square of power supply voltage. There has been requirement of designing of optimum repeaters chain insertion in the long interconnects [5].

2.2 Interconnects

Interconnects are the wires linking the transistors together. Today, the CMOS technology has reduced to nanometer. As a result of this continuous scaling, high speed circuits, lower power and larger packing densities of transistors are achieved. The technology scaling also affects the interconnects. Both the thickness of the metal layers and the thickness of the oxide layer between the metal layers decrease with scaling. The minimum width of an interconnect and the minimum spacing between the two interconnects also decreases [7].

Interconnects can be categorised as global interconnects or local interconnects [10]. The local interconnects are the first, or lowest, level of interconnects. The Local interconnects usually connect gates, sources and drains in MOS technology, and emitters, bases, and collectors in bipolar technology. In MOS technology a local interconnect, polycrystalline silicon, also serves as the gate electrode material. Silicided gates and silicided source/drain regions also act as local interconnects. In addition, TiN, a byproduct of a silicided gate process, can be used as a local interconnect, and W is sometimes used as well. Local interconnects can have higher resistivities as compared to the global interconnects because they do not travel very long distances. But they must also be able to withstand higher processing temperatures.

Global interconnects are mostly made of Al. These are generally all of the interconnect

levels above the local interconnect level. The global interconnects often travel over large distances, between different devices and different parts of the circuit. Therefore the global interconnects are always low resistant metals, since they have to travel larger distance [10].

In modern ICs, with the increase in the complexity of the interconnect an additional level of innterconnects have been introduced between local and global interconnects called as Semiglobal interconnects. The hierarchy of interconnects is given as: [10] :

- a. Local interconnect : Local interconnects are used for very short distance at the device level. Very short wires where delay is primarily device governed are placed in the local level. Under scaling, a circuit with the same functionality will be smaller in a new technology [10]. The interconnects of this circuit will also become shorter and these scaled interconnects are called *local interconnects* [7].
- b. Semiglobal interconnect : Used to connect devices within a block. Mid length wires for communication within a block are placed in the semiglobal level [7].
- c. Global interconnect : Global interconnects are used to connect long interconnects(distances) between the blocks, including power, ground and clocks. These are the longer wires for clocks, power, ground and long distance communication [10]. As more functionality is packed on a chip, the total size of the chip remains roughly the same under scaling. The global interconnects span the entire chip. These interconnects do not scale in length and are called *global interconnects* [7].

Figure: 2.1 shows the concept of on-chip interconnects and scaling. Local and global interconnects are shown in the multilevel schematic diagram in Figure 2.2 and 2.3.



Figure 2.1: Interconnects and scaling: local and global interconnects [7].



Figure 2.2: Schematic cross-section of backend structure, showing interconnects, contacts and vias, separated by dielectric layers [10].

2.3 CMOS inverter/repeater

The simplest repeater in the VLSI interconnects is the CMOS inverter. Figure: 2.4 presents a CMOS buffer driving a wire(interconnect) load and its equivalent representation [5]. The figure 2.5 gives equivalent RLC and RC lumped model representations of a long interconnect line [5].

2.4 Wires without Repeaters

The longer interconnects lead to higher propagation delays. Repeaters are needed to drive the high capacitive load in order to meet the slope and timing requirements. A single buffer is not a good solution for driving long interconnect, because long interconnects represent a very large RC load at the terminals of the gate(s) connected to it. Instead of inserting a single buffer, a number of buffers (or repeaters) are inserted after regular intervals in the interconnect, and these are called repeaters [1]. Therefore, in long interconnects insertion of repeaters at optimum distance reduce delay significantly [1]. Long interconnects lead to more RC and thus more increased load, thus this leads to more excessive power dissipation, and this power dissipation is a very important issue in VLSI circuits. [5]. The figure 2.6 shows the delay of the wire without repeaters [5]. This shows that the delay of the wire without repeaters increases exponentially with the increase in the length of the wire [5]. The delay increases quadratically with the increase in the length of the wire.

2.5 Wires with Repeaters

With the increase in the wire length, the wire delay also increases. This can be mitigated by introducing the repeaters along the wire as shown in the figure 2.7 [9]. So, the wire is divided into "m" sections of length Ls = L/m each and a repeater (an inverter or a buffer) is attached at each section. Figure: 2.8 shows the delay of the



Figure 2.3: Scaling of a chip and interconnections [10].



Figure 2.4: CMOS buffer driving a wire load and its equivalent representation [5].



Figure 2.5: RC and RLC lumped model representations of an interconnect line [5].



Figure 2.6: Delay of wire without repeater growing exponentially [5].

wire with repeaters. This shows that after the repeaters insertion in the wire, the delay of the wire increases linearly instead of increasing exponentially. The formula 2.1 [9] gives the minimum delay that occurs for a certain section length.

$$L_{sopt} = \sqrt{\frac{t_{pl}}{0.38rc}} \tag{2.1}$$

where t_{pl} is the delay of an inverter driving another similar inverter.



Figure 2.7: Wire with repeaters [9]



Figure 2.8: Delay grows Linear after repeater insertion [9]

2.6 Power consumption related to Interconnects

Long nets are described by RC or RLC lumped model where as short nets are described by their total capacitance C_w . In all the cases the power is consumed due to charging and discharging of the interconnect capacitance. Thus, the power consumption related to the interconnect given by equation 2.2 [2].

$$P_w = \frac{1}{2}\alpha f_c C_w \Delta V^2 \tag{2.2}$$

where α is the signal activity (the probability that the signal will change per clock cycle), f_c is the clock frequency, and the ΔV is the signal voltage swing.

The power consumption depends on the signal activity leads to severe difficulties in power prediction and hence power optimization. The signal activity should be considered for any prediction and optimization. This signal activity depends on the actual data statistics and therefore actual architectures and applications run on these. One consequence is that the total length of interconnect is not sufficient for power estimation. Instead we need individual wires and individual signal activities [2].

The worst case power consumption for the wires with crosstalk, may be larger than predicted by the equation 2.2, due to miller effect [2].

If the neighboring interconnect has a signal with opposite polarity transition, then capacitance C_c effective value may get doubled, thus larger C_w in equation 2.2 and hence larger power consumption. This effect is called crosstalk delta delay. Becaus of this crosstalk delta delay there is more power consumption for more (delta) period of time. The cross-talk effect depends on the correlation between neighboring signals and the exact timing relation between the two transitions [2]. This can be decreased with the clock net shielding.

2.7 Drivers and Repeaters Power Consumption

The driver has to be upsized (that is increase the drive strength of the driver), When the capacitance of wire is large (large RC load), to have short delay and fast-rise time. For driving large load we normally use tapered inverter chain in order to minimize delay. This means that we have an upsized inverter to drive the wire and then a multistage predriver to drive the upsized inverter [2]. When there is large load, the drivers become vulnerable to consume additional power, this happen due to shortcircuit power.

Repeaters are often used on chip to optimize wire delays that is to meet the slope(transition time) requirements. Repeaters are also used to mitigate crosstalk in the long wires. Dynamic Power is the dominating power consumption in well-designed repeaters, thus in the equation 2.2 (assuming electrically short wires) the C_w is replaced by the sum of C_w and the total switching capacitance of the repeaters [2] :

$$C_{tot} = \frac{C_s}{L_s} L \tag{2.3}$$

2.8 Strategies for Power Savings in Interconnect

2.8.1 Introduction

To reduce interconnect-related power consumption as much as possible is very profitable for us. One method to do this is to reduce the long interconnections. This is an important method of power reduction, but this needs to change the architecture.

The wire capacitance can be minimized through changes in fabrication process. This can be accomplished by using the dielectrics with lower dielectric constants. A very efficient method of power saving is to reduce the signal voltage swing. This method leads to some delay penality, which may be harder to accept in high performance systems. In the wires with repeaters, reduced voltage swing is tricky and leads to delay penalities. If repeaters are used then their delay must be traded for lower power consumption. Another method to save power is to reduce the product of data activity and capacitance. This can be accomplished by considering the data activity/length product. Another method is to reduce data activity on buses through coding, and thus minimize the amount of crosstalk-related power consumption. For power saving in data buses, several coding methods have been evaluated.

2.8.2 Optimal Power Delay Tradeoff

Figure 2.9 shows the optimal power-delay trade-off. It has been reported that for larger technology nodes like for 180-nm and 130-nm technology nodes the leakage power is not significant, the relative power saving is almost the same for a given delay penalty [3]. However, for smaller nodes that is beyond 130-nm, the leakage power becomes significant and therefore the relative power savings increase with technology scaling for a given delay penalty [3]. The Figure 2.9 shows that the relative power savings increase with technology scaling for a given delay penalty.



Figure 2.9: Power Delay Tradeoff

2.8.3 Power Savings in Drivers and Repeaters

In high performance system, the wire driver's power overhead may be quite large (80 % or more). By increasing the tapering factor, f [4], this power overhead can be reduced with a delay penality. If we increase f from 3.5 to 9, then it reduces the power overhead from 80 to 25 % at a delay penality of 20 % [9]. There is always trade-off between speed and power consumption. we can reduce speed (add delay penality) and thus save more power. We can optimize power consumption for a given delay penality, that is the power dissipation of the repeater can be decreased by 50% [4]. Allowing a larger delay penality facilitates larger power savings.

2.9 Summary

This chapter discussed the brief literature survey on interconnects and repeaters. The types of interconnects, need of repeaters and CMOS inverter used as a repeater. It described the behaviour of wires without repeaters and wires with repeaters. Also, the power consumption related to interconnects, drivers and repeaters has been discussed along with the strategies to save the power. Therefore it is beneficial to look for power reduction and area reduction of an interconnect.

Chapter 3

Types Of Repeaters

3.1 Inverter or Buffer

Repeaters are the Buffers or inverters which are inserted at regular intervals of distance in the interconnect [5]. Depending upon the logic need , the repeater can be either a buffer or an inverter. In case of inverters being used as repeaters the size further increases and the logic also reverses. So, one should select and insert repeaters carefully.

3.1.1 Low Vt Repeater(LVT)

The repeaters having low threshold voltage are called Low Vt repeaters, also called (LVT) [8]. Low-Vt (LVT) repeaters are the generally used high-performance repeaters which have low threshold voltage in order to increase the speed. But, this Low Vt causes high leakage current and due to this they have higher power consumption [8]. Thus, where delay penality can be tolerated but power is of most important concern, then in that case these LVT repeaters are not used, but these are used when power in not our concern but the speed of operation is most important concern.

3.1.2 High Vt Repeater(HVT)

The repeaters which have high threshold voltage, are called High Vt or Low Leakage(LL) repeaters. Significant amount of leakage power can be saved by replacing Low threshold Voltage (LVT) repeaters with high threshold voltage (HVT) repeaters, but it leads to timing degradation because speed of these HVT repeaters decreases. Advantage is that leakage power is reduced due to High Threshold voltage. Hence, these HVT repeaters are used when more delay penality can be afforded but the power saving is of primary concern.

3.2 SVT: Staggered Vt Repeaters

Staggered Vt Repeater is a new design approach of staggered threshold voltage (SVT) buffers with selective use of HVT transistors for power reduction [6]. SVT buffers are based on inverters combining low-Vt and high-Vt transistors as shown in the figure 3.1. During the operation in standby state the active devices are all low-Vt while during the off-state devices the active devices are high-Vt with lower leakage. This SVT technique is effective in both modes active mode and standby mode.



Figure 3.1: SVT repeater in 0-state low leakage [6]

The figure 3.1 [6] shows the basic structure of the interconnect having SVT repeaters. The links using SVT buffers have alternating inverters with low-Vt and high-Vt transistors. During the operation of SVT buffers, the low-Vt devices are active for most of the tme while the high-Vt devices are in the off-state with lower leakage. This is accomplished by encoding input signals to states that result in lowest leakage. Though the power is reduced using SVT buffers but the basic inverter-based structure remains unchanged. **Disadvantage:** The area of SVT repeaters is increased because of sizing of high-threshold devices, in order to meet the delay target. Since, these SVT repeaters include both the Low-Vt and High-Vt, thus the area is increased.

3.3 Dual - Vt Domino (DTD) Repeaters

This section presents the Dual-Vt Domino (DTD) [8] repeaters technique which offers a low-area alternative to SVT. The figure 3.2 shows the basic structure of the DTD repeaters. The DTD repeaters structure is similar to SVT repeaters. The DTD repeaters also encompass both the transistors having high threshold vlotage and low threshold voltage. But the difference is that DTD repeaters operate in domino structure instead of using regular inverters.

Data is applied to high-Vt transistors dedicated for evaluation. Clock is used for setting the standby state on the line by driving the Low-Vt transistors during the pre-charge phase.



Figure 3.2: DTD repeaters [8]

3.4 Summary

The chapter discussed the different types of repeaters. The repeaters can be categorised depending upon their threshold voltage level, logic (inverter or buffer), area required by them, etc. Different types of repeaters have been discussed after doing the literature survey.

Chapter 4

Implementation Methodology

4.1 Introduction

Generally a design is made for the following three cases:

- a. HighV : High voltage
- b. Nominal : Normal voltage
- c. LowV : Low voltage

In all these cases different voltages are being used. There is a flow for the creation of design Model. Generally we run the whole flow for any one of the above three cases to create a model. At a time the flow can be run only for one of the above design cases. Due to this, we cannot see all the other corner cases. Such as for a HighV design case: some paths are critical and these paths may have used different repeaters for that particular interconnect. But in the Nominal or LowV Design cases the same path may be using different repeater having size and type. We can see only one of them for which the flow has been running that is either HighV, Nominal or LowV. So, during the convergence of the design the problem arises. The design is not optimized w.r.t. area and power.

Problem: In this way, there are corner cases where the HighV, Nominal and LowV design cases do not converge for some paths , for these paths area wastage takes place, at the same time more power is being dissipated as the optimum repeater is not being used. This optimum repeater could have been used for that path if we have knowledge about both the design cases i.e. highV and Nominal case and we can create a converged design.

Solution: We want to see both the corner cases : HighV and Nominal at the same time. By doing this we can optimize the repeaters, by seeing which repeater is the best w.r.t. area and power provided that this repeater should not violate the timing requirements for that particular interconnect that is the timing of that particular interconnect should not be violated.

4.2 Implementation Flow for Repeaters Optimization

Figure 4.1 is showing the flow chart for automating the whole flow by writing the scripts and then integrating them in the Place & route tool and Timing tool.

4.2.1 Explanation:

First of all we copy all the files in the present working directory. Created a GUI (Graphical User Interface) by writing scripts in Tcl/Tk. Through this GUI we change all the files, the particular variables of the files such as reference area, timing paths, slope, threshold timings, etc. Written a perl parser to extract out all the timing values(Margin values) from the file generated by the timing tool. Integrated this perl parser with the main Tcl file which is used to prepare the specification file. This changed Tcl file is then sourced in the Place & Route Tool which generates the main Specification file. This Specification file contains all the specifications for each Pin

and Net that is all the delay's information and the information about the selection of repeaters. The format of the specification file is shown in the Table II.Giving this specification file as input to the shell script, we run the shell script and then a repeater file containing information about all the repeaters in the particular design case is generated. This repeater file is then annotated on the reference design using the Place & Route tool. Then the timing model is run and the comparison data is being produced. The implementation has been explained step-by-step in detail in the section 4.2.3, 4.2.4, and 4.2.5. 4.2.6

4.2.2 Margin Parser Script:

The input to the margin parser script is a file containing the timing information. This file is called margin file as it contains all the margin values for setup and hold time for the particular pin and interconnect. This file is being created after running the timing model for a design case. If the timing model is being run for nominal case then nominal margin file is being created. Similarly for other design cases. Table I shows the format of the margin file.

Pin Name	Setup Up	Setup Dn	Hold Up	Hold Dn
core/module/pin_xyz304l[7][15]	0.0014	0.0037	-0.0380	-0.0480
$core/module/pin_abc803h[3]$	0.0037	0.0051	0.0251	0.0251
$core/module/pin_pqr304l_rpt\#[3][14]$	-0.0088	-0.0074	0.0050	0.0110
$core/module/pin_uvx304l_rpt#[3][15]$	-0.0051	-0.0074	0.0140	0.0170

Table I: Format of Margin File

When the flow run for the HighV design case it creates HighV margin file having timing information for the HighV design case and when the flow when run for the Nominal Design case, it will create the Nominal margin file containing the timing information for the nominal design case. There is no way to converge the two design cases, as the flow can run only on one of the cases. In order to converge the two design cases (Nominal and the HighV case) and remove the corner cases, thus optimizing the critical paths and optimizing the design. As we cannot change the flow ,so we create a new margin file having information of both the Design cases. This new margin file is created by writing a margin parser script using perl. Figure. 4.2 shows the flow of Margin Parser script.

Logic for creating Cross-Corner Margin file:

For the same interconnect, the margin values of HighV design case is divided by the scale factor to create a new value, we call this new value as Virtual nominal value. Then we compare this virtual nominal value with the original nominal value from the nominal margin file for the same innterconnect and the minimum value out of the two values is selected. Thus, For the same interconnect, by selecting minimum of the following two values new margin file is created.

a.
$$\left(HighV/ScaleFactor\right) = VirtualNominal$$

b. Value from the nominal margin file

Thus new margin file is created , this new margin file is called Cross-corner margin file as it is created from both the design cases: Nominal and HighV case , called the Cross-corner case. Figure 4.3 shows the Logic used for the creation of the Cross-corner margin file. It has already been explained in the section 4.2.2.

4.2.3 Step 1: Preparation of Specification file

There is a TCL/Tk script which is present in the flow which generates the Specification file. The format of specification file is shown in the table II. The Specification file basically contains all the information related to a path. its driver , receiver, capacitance delays, RC delays, and many other delays. The margin parser script is being integrated with the Tcl/Tk script which is already present in the flow, to integrate the cross corner case.



Figure 4.1: Implementation flow chart for Repeaters Optimization Methodology



Figure 4.2: Flow of Margin Parser script



Figure 4.3: Logic of creating the Cross-corner Margin File

NET	DRV	\mathbf{RCV}	\mathbf{RC}	\mathbf{SLP}
$net_abc305h[13]$	$module1\%net_abc305h[13]$	$module2\%net_abc$	0.034	0.034
net_xyz305h[3][10]	$module1\%net_xyz305h[3][10]$	$module3\% net_abc$	0.032	0.019
$net_pqr305h[7][1]$	$module2\%net_pqr305h[7][1]$	$module3\% net_abc$	0.065	0.027

Table II: Format of Specification File

LOGIC: In the Tcl script which is used for the preparation of Specification file, a flag is declared for the cross-corner case and set to "0" for the default case as the Nominal design case. This script runs for the nominal design case by default. If the flag is set to "1", then the margin parser script runs and the new cross-corner margin file is being created and then this file is given as the input to this Tcl script for the generation of the specification file. Thus, when we want to run the flow for the cross-corner case, we just set the flag = "1" and then create a Specification file for the cross-corner case. Following logic shows when to run the Nominal design case and When to run the Cross-Corner case. This logic has been integrated in the Tcl file present for the generation of the Specification file.

```
set flag_both_margin "0"
if {$flag_both_margin == "0"} {
    puts "CROSS CORNER Flag is OFF"
    Read the nominal margin file (No cross-corner case)
}
if {$flag_both_margin == "1"} {
    calling parse_margin.pl script
    Execute the margin parser PERL Script to create corner-case margin file
    puts "CROSS CORNER Flag is ON"
    Read the cross-corner margin file being created above
}
```

The above logic tells that by default : the flow runs for nominal design case, and the specification file for nominal case is being prepared. But to run the flow for cross-corner case, we need to prepare the Specification file for the Cross-corner and that is done by just setting the Flag ="1" in the above code snippet. Then after changing the flag value in the Tcl script, the Tcl Script is run in the Place & Route tool, to create Specification file.

4.2.4 Step 2: Creation of Repeater file

This specification file is given as the input to the C-shell script which is already present in the flow to create the repeater file. This C-shell script is used to create the repeater file. The input to this shell script is the Specification file generated in the step1 and the output of this script is the repeater file. The repeater file format is shown in the Table III

Net	rpt_name	Χ	Y	cell
abc305h[13]	$abc305h[13]_bfrHL36x5.X13036Y8074$	13.03	80.74	bfrHL36x5
xyz804h[5]	$xyz804h[5]_bfrHL48x5.X21575Y8233$	215.75	82.33	bfrHL48x5
pqr9091	pqr909l_inrLL64x5.X13026Y280704	281.53	79.12	inrLL64x5

Table III: Format of Repeater File

Repeaters file description: It contains the name of the repeater, the Net on which this repeater is being present, type of the repeater, size of the repeater. The type tells whether the repeater is a inverter or a buffer. The size field tells whether the repeater is of size 16X, 18X,...,64X ,etc. This repeater file is the most important file as it contains all the information about all the repeaters. This file has to be used in the subsequent stages.

For cross-corner Case: To create repeater file for the cross-corner case, the specification file generated for the cross-corner case has to be given as input to the C-Shell script.

4.2.5 Step 3: Annotating repeater file to the layout (design)

Load the Full Chip design (layout) from the central database using the Place & Route Tool and annotate the generated repeater file on this design (layout) using the tool command. By annotating the generated repeater file, we mean that we add the repeaters in the current design according to the generated repeater file and thus change the design. Then save this design in the local area, thus create a local model.

For Cross-Corner case: By annotating the new repeater file for the cross-corner case into the design using the place & route tool, we change the repeaters, their counts and their types according to the repeaters file generated for the cross-corner case and thus change the design (layout) and then save this changed layout in the local area. After saving this design (layout) file in the local area, we run a timing model using this new layout file and then generate comparison data.

4.2.6 Step 4: Full Chip Interconnect's RC extraction and Running FC Timing

After back-annotating the repeater's file of all the sections on the reference Design (layout), the design is saved and using this design the RC extraction of the interconnects is done using the RC-extraction tool and RC-extraction engine. After the RC-extraction is completed the results of RC-extraction before optimization and after optimization are compared. The quality of extracted RC (Resistance-Capacitance) after the repeaters optimization should not be degraded. If it improves then well and good, but it should not degrade. There should not be any new bad-repeaters introduced. The bad-repeaters count should be same or low as compared to the reference design's RC-extraction bad repeater. All the P2P delays including the repeaters delays are simulated using Full Chip Interconnect RC-extraction.

The reference Full Chip RC-extraction has 98.94% age of good sources and after the repeaters optimization the Full Chip RC-extraction has 98.18%. The two figures are almost same and thus there is no degradation in the RC quality.

After the RC extraction stage is completed and we have the extracted RC values for the interconnects, the full chip timing model or section timing model is run. The timing runs on this design (layout) is then compared with the timing on reference design (layout) for any degradations.

4.3 RPT OPT Flow GUI

The figure 4.4 shows the basic GUI of the tool flow. Key features :

- Merged the two Margin Files using Margin Parsing Technique to take the WC margins from Nominal and Highv corners.
- User-friendly GUI
- No hard-coded settings
- All the variables now can be entered by user through GUI
- Two or more than two process can be run for different Sections in Parallel (Saves Time)
- Optimization for all the sections can be done by just launching in one go (Easy Operation)
- Allow users to easily change settings (slope, margin threshold) and do multiple analysis to finalize the Max Slope and Margin Threshold

- Pre-parser utility in Place & route tool to filter out Local nets only to enable High-confidence Section timing data.
- All pointers are coming from central area and accessible at all sites.



Figure 4.4: RPT OPT Flow GUI

4.4 Repeaters Used in the Design

The repeaters used in the design can be categorised broadly into different types, depending upon their Size, threshold voltage level, depending upon the logic whether inverter or buffer. The theoretical aspects are being covered in the chapter 3. The repeater used in the design can be a inverter or a buffer. The repeater used is selected from one of the cell types which are already defined in the library database. These cell types tell all the aspects whether the particular repeater is a buffer or inverter and also repeater's size. These repeater cells are being shown in the Table IV. As we move from top to bottom in the table that is from 8X repeater cell to the 64X repeater cell, the size of the repeaters increases. With the increase in the size of the repeater, there is significant increase in the area as well as increase in the power consumption by the repeater. Thus, we can say that as we move from smallest repeater cell (8X) to the largest repeater cell (64X), the area used by the repeater and the power consumed by the repeater both increases. This increase in power consumption is due to increase in the size of the repeater.

Further increase in the power consumption also depends on whether the repeater is of High-Vt or Low-Vt type that is the leakage power come into the picture. If the repeater is of Low-Vt type then the power consumption will further increase because of increase in the leakage power and if the repeater is of type High-Vt then the leakage power will be significantly low then in this case the total power consumption by High-Vt repeater will be low as compared to low-Vt repeaters. The theoretical aspects are covered in the chapter 3.

So, in this repeater optimization methodology our *main aim* is to use more and more number of smaller repeater cells instead of using larger repeater cells to save area. Also trying to use more number of High-Vt repeater cells instead of low-Vt repeater cells to reduce leakage power consumption.But with meeting the above aim , timing should not be violated. Example: We would use 48X or 52X instead of using 64X but without violating the timing constraints that is the timing should not be violated otherwise this area and power optimization is of no use if the timing is being violated. Because in Our design timing is the top most priority and it should be satisfied, after that the power and area comes into picture. This methodology is being implemented converging the two Margin file of HighV design case and Nominal design cases and creating a cross-corner margin file and using the optimum repeater wherever possible.This implementation methodology has been explained in the section 4.2.

The Table V shows the number of repeaters present in the central/reference design.

Repeater Type	Repeater Size
8X	smallest
9X	
12X	
16X	
20X	
24X	
28X	
36X	
40X	
48X	
52X	
64X	largest

Table IV: Repeaters used in the Design

4.5 Results, Analysis and Profiling

4.5.1 Repeater Profiling

A script for profiling the repeaters called repeater profiler has been written to create histograms for the comparison of two repeater files. The language used is the Perl. The two inputs to the repeater profiler script are:

- a. The repeater file containing the name of repeaters , size, type of repeaters and all the information of the repeaters used in the design.
- b. Another input to the repeater profiler script is Repeater Cell information file, it contains the length (l) and width (w) of all the repeater cell types defined in the library database.

Logic behind Repeater Profiling: The script matches the repeater name from the repeater file and for that particular repeater it finds the length (l) and width (w) of that repeater. Thus, once we have "l" and "w" then we find the area of the repeater using the formula 4.1:

$$Area = l * w \tag{4.1}$$

Repeater Type	Count (in %age)
8X	0.27
9X	0
12X	0.54
16X	0.27
18X	0
20X	1.21
24X	0
28X	22.1
30X	0.54
32X	0.54
35X	0
36X	38.27
40X	12.39
48X	24.26
55X	0
64X	0.135

Table V: Number of repeaters present in the central/reference repeater file (in percentage)

The figure 4.5 explains the logic of the repeater profiler script. From the repeater file the script finds the total count of the repeaters for each repeater type and creates a histogram. From the *Cell information File*, the script extracts the "l" and "w" for each repeater cell type. After having "l" and "w" scripts find the area for each repeater cell using the formula 4.1.

After having count(number) for each repeater cell type and area for one such cell type, the script multiply the count of repeaters and area of one repeater cells to get the total area for each repeater cell type present in the repeater file. Then we add the total area of all these different individual repeater cells present in the repeater file to get the total area of all the repeaters present in the design.

For cross corner case : The repeater file generated in the section 4.2.4 for the

cross-corner case is given as input to the repeater profiler script. Then it multiplies the total count of a repeaters with the total area consumed for one repeater cell. Similarly, finds the total area consumed by the repeaters for the cross-corner design case.

Format of Histogram : Histogram is the output of the Repeater Profiler script. Its format is shown in the Table VI.

- a. Column *rpt_cell* shows the particular repeater type present in the repeater file.
- b. Column *rpt_count* shows the Count (total) number of repeaters of that type present in design.
- c. Column *one_cell_area* shows area for one repeater cell of each type.
- d. Column *total_area_of_each_cell* shows the total area for each repeater cell type present in the design. This is found by multiplying the *rpt_count* column and *one_cell_area* column.

rpt_cell	rpt_count	one_cell_area	$total_area_of_each_cell$
$rpt_bfrLL20x5$	6	1.496	8.967
$rpt_bfrHL24x5$	219	1.836	402.084
$rpt_bfrLL40x5$	134	3.06	410.04
$rpt_inrLL32x5$	30	2.448	73.44
rpt_inrLL36x5	891	2.72	2423.52
rpt_inrHL40x5	667	2.992	1995.664
rpt_inrLL48x5	201	3.536	710.736

Table VI: Format of Histogram : output of repeater profiler

4.6 Analysis

The Histogram which is the output of Repeater profiler script is used to do the analysis of how much area has been saved by comparing the total area of repeaters used in the design before(reference/central) the repeater optimization and design after the repeater optimization. Using this data different graphs are being plotted to show the significant change in the count of different types of repeaters. The large sized repeaters have decreased and small sized repeaters have increased in number.

The graph 4.10 shows the comparison between the results before the repeater optimization and the results after the repeater optimization. The graph is plotted between the repeater type and the number of repeaters (in percentage) for the reference design (before repeater optimization) and after the repeater optimization (cross-corner case). On the X- Axis repeaters types are given and on Y-Axis count (in %age) is given. It can be observed that after optimization (shown in red color) there is significant decrease in the count of large sized repeaters (shown in red color) and instead of using large size repeaters there is increase in the small size repeaters that is 16X and 18X. Thus, the observation is meeting the expectation and is according to the aim of this repeater optimization methodology.

The Table VII shows the types of repeaters and their counts (in percentage) for different design cases (Reference/Central design, Nominal, HighV and Cross-corner design).

Observation: It can be observed that after optimization that is for the cross-corner design case there is significant decrease in the count of large sized repeaters and instead of using large sized repeaters there is increase in the small sized repeaters that is 16X and 18X when compared with the reference design case. Thus, the observation is meeting the expectation and is according to the aim of this repeater optimization methodology.

The total area saved after the repeater optimization is found by the formula 4.2.

%Saved Area = (Area Before Optm. – Area After Optm.) $*\,100/{\rm Area}$ Before Optm.

(4.2)

The graph 4.6 shows the Repeater Types Vs their counts (in percentage) in the Cen-

Repeater Type	Reference	Nominal	HighV	Cross-corner
8X	0.27	0.272	0.272	0.272
9X	0	0.136	0.136	0
12X	0.54	0.679	0.543	0.679
16X	0.27	8.97	7.065	6.114
18X	0	7.88	7.744	8.016
20X	1.21	2.038	2.179	2.174
24X	0	8.83	5.299	3.67
28X	22.1	15.625	15.896	15.49
30X	0.54	2.174	2.038	1.767
32X	0.54	3.396	2.309	2.58
35X	0	0.679	0.408	0.407
36X	38.27	29.067	32.2	33.695
40X	12.39	8.695	9.24	9.375
48X	24.26	11.413	14.54	15.625
55X	0	0.136	0.136	0.136
64X	0.135	0	0	0

Table VII: Table showing Types of Repeaters and the No. of Repeaters (in percentage) for different Design Cases

tral/Reference design. This analysis data is being obtained from the repeater profiler script.

The graph 4.7 shows the Repeater Types Vs their count (in percentage) for the Nominal design case.

The graph 4.8 shows the Repeater Types Vs the Number of repeaters(in percentage) for the HighV design case.



Figure 4.5: Repeater Profiler script



Figure 4.6: Graph showing Repeater Types Vs No. of Repeaters(in percentage) for central/reference design



Figure 4.7: Graph showing Repeater Types Vs No. of Repeaters(in percentage) for Nominal case

The graph 4.9 shows the Type of Repeater Vs the Number of repeaters (in percentage) for the Cross-corner design case.

The graph 4.11 shows the Comparison between all the design cases with respect to the reference/central design case. The graph is drawn between the Repeater type and the Number of repeaters.

The Table VIII shows the comparison results between before the repeaters optimization and after the repeaters optimization in terms of number of repeaters (in percentage).

RPT Type	Before RPT. OPT.(in %age)	After RPT. OPT.(in %age)
8X	0.27	0.272
9X	0	0
12X	0.54	0.679
16X	0.27	6.114
18X	0	8.016
20X	1.21	2.174
24X	0	3.67
28X	22.1	15.49
30X	0.54	1.767
32X	0.54	2.58
35X	0	0.407
36X	38.27	33.695
40X	12.39	9.375
48X	24.26	15.625
55X	0	0.136
64X	0.135	0

Table VIII: Table showing Number of Repeaters (in %age) for a particular repeater type Before Repeater Optimization and After Repeater Optimization

The Table ?? shows the comparison results between reference design , single-corner



Figure 4.8: Graph showing Repeater Types VS No. of Repeaters(in percentage) for HighV design case



Figure 4.9: Graph showing Repeater Types Vs No. of Repeaters(in percentage) for Cross-Corner design case



Figure 4.10: Graph showing comparison Before Repeater Optimization and After Repeater Optimization



Figure 4.11: Graph Showing Comparison between all the design cases

design after repeaters optimization and multi-corner design after repeaters optimization in terms of number of repeaters (in percentage).

RPT Size	Reference (%age)	Single-corner(%age)	Multi-corner (%age)
8X	.0161	0.403	0.096
9X	0.032	1.33	0.499
12X	0	2.658	2.497
16X	0	9.344	8.152
18X	0	7.717	5.268
20X	0.064	4.478	1.901
24X	3.061	3.91	5.429
28X	8.619	8.699	9.569
32X	24.44	10.568	13.27
36X	16.46	9.827	10.83
40X	24.87	19.768	20.51
48X	22.42	21.28	21.97
LL CELLs $\%$	15.3	31.7	23.4

Table IX: Table Showing comparison of RPT downsize and LL % age on one of the section

4.7 Results :

This section describes the percentage of area saved after implementing the repeaters optimization methodology. The central/reference design is considered as the reference design for all the calculations. It is the design before repeaters optimization and the cross-corner design case is considered to be the design after the repeaters optimization. The repeater optimization methodology can be implemented for single corner that is only for Nominal design case also. In that case also, the reference design is taken from the central.

From the formula 4.2 the total area saved for the repeaters in the design can be found. It is shown below:

Thus, **12.732%** of area has been saved for the repeaters after implementing the repeaters optimization methodology on one of the section. Hence, there is significant reduction in the area.

Power Saving (LL insertion) :

The total percentage of LL (low-leakage) cells in the full chip before implementing repeaters optimization was 5.36%, and after the repeaters optimization (in the multi-corner case) the LL insertion percentage was increased to 15.01%. Thus, there is significant power saving.

Challenges :

- a. Some Functional Unit Blocks (FUBs) have repeater placeholders instantiated inside them. Since we run RPT OPT flow only after Section timing is matured, most of these functional FUBs are already converged. In such cases, these repeaters inside the FUB place-holders should not be optimized. These repeaters should be kept as it is.
- b. Also the repeaters present in the DFT interconnects should not be optimized.
- c. The repeaters optimization is done only for the local nets of that particular section.

4.8 Summary

This chapter describes the implementation methodology, different scripts written, their logic and their integration with the tool flow to automate the flow. The profiling, graphical analysis and comparisons have been done between the design before the repeater optimization and after the repeater optimization. After the analysis, the conclusion drawn is that there has been a significant saving in area by **12.732%** for the repeaters. This approach causes about three times increase in low-leakage repeater cell's count (to about 20%), that translates to about **1%** reduction in full chip leakage power.

Chapter 5

Conclusion

5.1 Conclusion

The trade off between delay and power dissipation in repeaters inserted in long interconnects has been reviewed with a bibliographic survey. Optimum size repeaters inserted in long interconnects, leads to area minimization. By changing threshold voltage, and thus using the High-Vt repeater cells lead to power saving. Analysis has been done using graphical comparison and the creation of histograms have been done to validate the findings.

The proposed multi-corner aware repeater optimization flow presents a methodology to optimize the existing repeaters inside the Core to low-leakage flavors, along with downsizing the repeater cells, without impacting the multi-corner timing Margins. The GUI based implementation makes the flow easily scalable across different process nodes, allows parallel execution on all sections in one-go, and enables user to experiment with slope and timing thresholds to achieve optimum repeater sizes. The output repeater file is easy to back-annotate on the design (layout) without any negative impact on existing design (layout) and without any surprises on the extracted RC (Resistance-Capacitance) quality. After optimizing the repeaters, there has been a significant saving in the area for the repeaters, on an average by about 15%-20% depending upon the section, margin threshold value and slope threshold value. This approach causes about **three times** increase in the low-leakage repeater cell's count.

References

- Victor Adler, Student Member, Eby G. Friedman, and Senior Member. Repeater design to reduce delay and power in resistive interconnect. In *IEEE Trans. Circuits Syst. II*, pages 607–616, 1998.
- [2] A. Alvandpour, P. Larsson-Edefors, and C. Svensson. Glmc: interconnect length estimation by growth-limited multifold clustering. In *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, volume 5, pages 465–468 vol.5, 2000.
- [3] K. Banerjee and A. Mehrotra. A power-optimal repeater insertion methodology for global interconnects in nanometer designs. *Electron Devices*, *IEEE Transactions on*, 49(11):2001–2007, Nov 2002.
- [4] P. Caputa and C. Svensson. Low-power, low-latency global interconnect. In ASIC/SOC Conference, 2002. 15th Annual IEEE International, pages 394–398, Sept 2002.
- [5] Rajeevan Chandel, S. Sarkar, and R.P. Agarwal. Repeater insertion in global interconnects in vlsi circuits. *Microelectronics International: An International Journal*, 22(1):43–50, 2005.
- [6] H.S. Deogun, R.R. Rao, D. Sylvester, and D. Blaauw. Leakage-and crosstalkaware bus encoding for total power reduction. In *Design Automation Conference*, 2004. Proceedings. 41st, pages 779–782, July 2004.

- [7] E. Mensink, D. Schinkel, E.A.M. Klumperink, E. van Tuijl, and B. Nauta. Optimal positions of twists in global on-chip differential interconnects. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 15(4):438–446, April 2007.
- [8] A. Morgenshtein, I. Cidon, A. Kolodny, and R. Ginosar. Low-leakage repeaters for noc interconnects. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 600–603 Vol. 1, May 2005.
- [9] Jan M. Rabaey. Digital Integrated Circuits: A Design Perspective. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.
- [10] K.C. Saraswat and F. Mohammadi. Effect of scaling of interconnections on the time delay of vlsi circuits. *Electron Devices, IEEE Transactions on*, 29(4):645– 650, Apr 1982.