

ENCODING IN VLSI INTERCONNECTS

A DISSERTATION

*Submitted in partial fulfillment of the
requirements for the award of the degree*

of

MASTER OF TECHNOLOGY

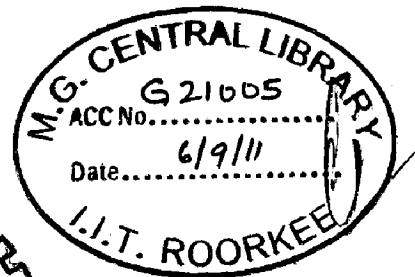
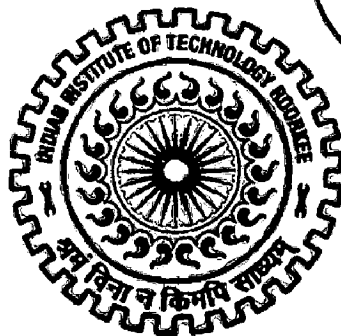
in

ELECTRONICS AND COMMUNICATION ENGINEERING

(With Specialization in Microelectronics and VLSI Technology)

By

DEEPIKA AGARWAL



DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

ROORKEE -247 667 (INDIA)

JUNE, 2011

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being reported in this dissertation report, entitled “**Encoding in VLSI Interconnects**”, is being submitted in partial fulfillment of the requirements for the award of the degree of **Master of Technology in Microelectronics and VLSI**, in the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee is an authentic record of my own work, carried out from June 2010 to June 2011, under guidance and supervision of **Dr. B. K. Kaushik** and **Dr. S. K. Manhas**, Assistant Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee.

The results embodied in this dissertation have not submitted for the award of any other Degree or Diploma.

Date : 30/06/11

Place : Roorkee



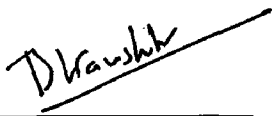
Deepika Agarwal

CERTIFICATE

This is to certify that the statement made by the candidate is correct to best of my knowledge and belief.

Date:

Place: Roorkee

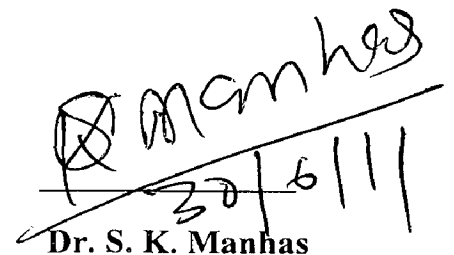


Dr. B. K. Kaushik

Assistant Professor

Department of E & CE

Indian Institute of Technology, Roorkee



Dr. S. K. Manhas

Assistant Professor

Department of E & CE

Indian Institute of Technology, Roorkee

ACKNOWLEDGEMENT

First and foremost, I would express my gratitude to Dr. Brajesh Kumar Kaushik, Department of Electronics and Computer Engineering at Indian Institute of Technology, Roorkee for his invaluable guidance, support and encouragement. Without him, I could not have realized my skill and potential. Interaction and discussion under his guidance helped in building my concepts.

I am deeply obliged to Dr. Sanjeev Kumar Manhas for his moral support. I owe a huge debt of thanks to the faculty of Microelectronics and VLSI Technology group, department of Electronics and Computer Engineering for their technical assistance and constant motivation to carry out my dissertation.

I especially thanks to my parents and my brothers Harish and Manish as they always have been any source of encouragement. Above all it is the Almighty who deserves the ultimate reverence for everything I have.

The life of campus is important in counter balancing life for that I would like to thank my friends. I have very fortunate to have so many good ones: Tripti, Baljit, Amresh, Shubham, Vikas, Santosh, Shubhranshu, Arif who always with me at the good and bad time.

I will always be grateful to my seniors and my junior Akhil, who were always there to support me to overcome the hurdles, I faced during my work. I would also like to mention the invaluable help received from my batch-mates through discussion and good practical suggestion.

Deepika Agarwal
M.Tech (MEV)

Dedicated to my loving parents

ABSTRACT

Interconnect play an important role in deep submicron technology. Rapidly decreasing minimum feature sizes lead to exponential growth of system-on-chip integration complexity. A novel circuit is introduced which eliminates the effects of interconnects on power dissipation, crosstalk, propagation delay and chip area by using bus encoding technique in *RC* modeled VLSI interconnect. Bus encoding techniques has been used to reduce inter-wire coupling which is primary source of power dissipation, crosstalk and delay in coupled interconnects. The proposed method focuses on simplified and improved circuit encoder for 4, 8 and 16 coupled lines. Previously used encoding schemes based on *RC* models had usually focused on only minimizing power dissipation and crosstalk while paying penalty in terms of chip area. However, our proposed encoder and decoder demonstrate an overall reduction in power dissipation by 68.7% through drastic reduction of switching activity. Furthermore, the propagation delay is reduced by 56.7% and other parameters like complexity, chip area and transistor count of the circuit is also minimized by more than 57%.

CONTENTS

Candidate's Declaration	i
Acknowledgement	ii
Abstract	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii
Chapter 1: Motivation of Thesis	1
1.1 Introduction	2
1.2 Motivation	4
1.3 Thesis Organization	4
Chapter 2: Basics of On-Chip Interconnects	5
2.1 Interconnect Characteristics	6
2.1.1 Resistance	6
2.1.2 Capacitance	7
2.2 Interconnect Design Criteria	8
2.2.1 Crosstalk Delay	8
2.2.2 Power Dissipation	10
2.2.3 Chip Area	11
Chapter 3: Design Methodologies for Interconnects	13
3.1 Introduction	14
3.1 Better Interconnect Materials	14
3.2 Shielding Line between Adjacent Signal Lines	14
3.3 Repeater Insertion	15
3.4 Bus Encoding Method	16
3.4.1 Bus Invert Method	16
3.4.2 Partial Bus-Invert Method	17
3.4.3 Odd-Even Bus Invert Method	17
3.4.4 Kim <i>et al.</i> Method	18
3.4.5 Victor and Keutzer Method	18
3.4.6 Lyuh and Kim Method	18
3.4.7 Khan Method	18

3.4.8	Fan <i>et al.</i> Method	19
Chapter 4:	Proposed RC Model	23
4.1	Introduction	24
4.2	Proposed Encoder	24
4.2.1	Transition Detector	25
4.2.2	Type-3 and Type-4 Detector	26
4.2.3	Multiplexer	27
4.2.4	Latch	28
4.3	Decoder	29
Chapter 5:	Results and Analysis	31
5.1	Simulation Result	32
5.2	Crosstalk Reduction	32
5.3	Chip Area Reduction	33
5.4	Total Power Reduction	33
5.5	Total Propagation Delay	37
Chapter 6:	Conclusions and Future Scope	41
6.1	Conclusion	42
6.2	Future Scope	42
References		44
Publications		46
Appendix A		47
Appendix B		49
Appendix C		55

List of Figures

Figure No.	Title of Figure	Page No.
1.1	Interconnect delay became dominant over gate delay	2
1.2	Cross-section of stacked interconnects	3
2.1	Cross-sectional dimension of a conductor	6
2.2	Capacitance in multi-layer representation	7
2.3	Impact of RC crosstalk	8
2.4	Components of power dissipation due to different capacitance sources	11
3.1	Shielding to reduce capacitive crosstalk	14
3.2	Breaking up a long transmission-gate chains by inserting buffers	15
3.3	Model of communication chain.	16
3.4	Block diagram of 5-bit bus encoder architecture	19
3.5	Internal block diagram of N4_count circuit	21
3.6	Internal block diagram of N3_count circuit	21
4.1	Main block diagram of proposed encoder	24
4.2	Block diagram representation of transition detector	25
4.3	Block diagram of Type-3 (($\uparrow\downarrow$), ($\downarrow\uparrow$)) and Type-4 detector	26
4.4	Block diagram of Type-3 (($\uparrow\downarrow$), ($\downarrow\uparrow$)) and Type-4 detector	27
4.5	A multiplexer	28
4.6	Latch with transmission gates	29
4.7	A decoder	29
5.1	Interconnect routing for an 8-bit bus	32
5.2	Variation of power dissipation of system with technology at 1 MHz	35
5.3	Power dissipation of system with technology at 1 GHz	37
5.4	Variation of total propagation delay of system with technology at 1 MHz	38
5.5	Total propagation delay of system with technology at 1 GHz	40

List of Tables

Table No.	Title of Table	Page No.
2.1	Resistivity of materials used for conductors in VLSI circuit	6
2.2	Dependence of effective capacitance of line 'A' on line 'B'	8
2.3	Dependence of switching configuration of line 'B' on line 'A' and 'C'	9
2.4	Classification of crosstalk	10
4.1	Truth table of multiplexer	28
5.1	Comparison of components of proposed method with Fan <i>et al.</i> method	33
5.2	Comparison of power dissipation of proposed method with Fan <i>et al.</i> at 1 MHz	34
5.3	Power dissipation of Fan <i>et al.</i> method at 125 GHz	35
5.4	Power dissipation of proposed method at 800 GHz	36
5.5	Power dissipation of proposed method at 1 GHz	37
5.6	Comparison of propagation delay of proposed method with Fan <i>et al.</i> at 1 MHz	38
5.7	Propagation delay of Fan <i>et al.</i> at 125 MHz	39
5.8	Propagation delay of proposed method at 800 MHz	39
5.9	Propagation delay of proposed method at 1 GHz	40

CHAPTER 1

MOTIVATION OF THESIS

1.1 Introduction

With advancement of VLSI technology, more and more number of devices or modules would be added on a single chip. This has been possible because of continuous reduction in feature size. In deep submicron technology, device geometries shrink, chip size increases and clock speeds get faster which increased the effect of interconnect delay significantly [1]. Figure 1.1 shows the interconnect delay becomes dominant over gate delay as the feature size decreases. This is due to the fact that interconnect no longer behave as resistors but may have associated parameter such as capacitance and inductance.

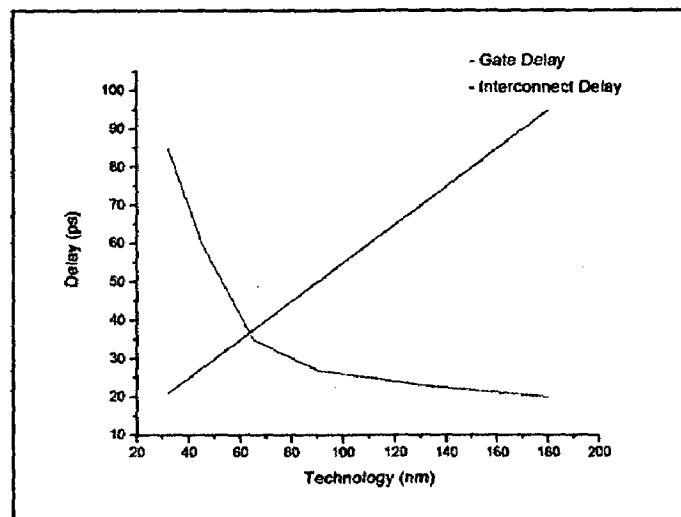


Figure 1.1: Interconnect delay became dominant over gate delay.

Interconnect delay becomes a significant bottleneck in system performance. Shrinking of feature sizes imply shorter gate length, decreased interconnects pitch, more closely packed interconnect level, lower device threshold voltages and on chip integration complexity which increases exponentially [2].

Interconnects are categorized into three groups according to their length are local, semi-global and global interconnects. The first and second layers of interconnects from the top are global ones, the third and fourth layers are semi-global/ intermediate ones, and the lowest layers are local ones. Vias are the metal fillings enabling inter-level wire connections. Interconnects are stacked with dielectric layer between two layers or between one layer to transistor, as shown in Figure 1.2. The wires are separated by interlayer dielectrics (ILD) from level to level and isolated by inter-metal dielectrics (IMD) within the same level.

- 1) **Local:** It consists of very thin lines connect gates and transistors within an execution unit or a functional block on the chip. Local wires usually span a few gates and occupy

first and sometimes second metal layers in a multi-level system. The lengths of these wires tend to scale down with technology.

- 2) **Semi-Global / Intermediate:** It provides clock and signal distribution within a functional block with typical lengths up to 3- 4 mm. Intermediate wires are wider and taller than local wires to provide lower resistance signal/clock paths.
- 3) **Global:** It provides clock and signal distribution between the functional blocks, and it delivers power/ground to all functions on a chip. Global wires, which occupy the top one or two layers, are longer than 4 mm and can be as long as half of the chip perimeter.

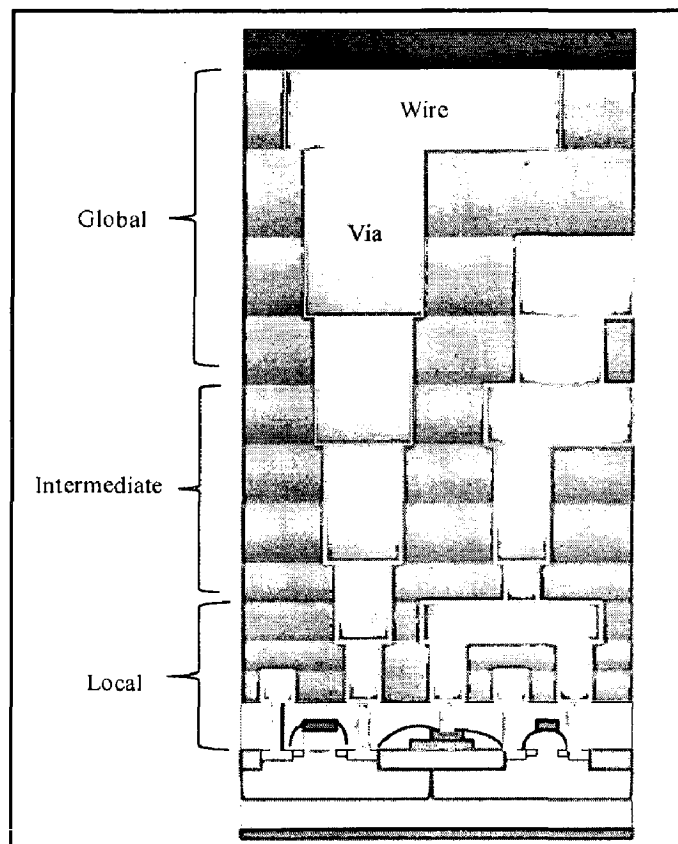


Figure 1.2: Cross-section of stacked interconnects.

Here we focus on global interconnect as feature size goes on shrunk, owing to increase in complexity by which number of metal layer has gone up [3]. As the result, the length of global interconnect has increased. Due to which both the interconnect capacitance and interconnect resistance increased linearly, making the RC delay (i.e. total propagation delay) increase quadratically.

There are other two factors which are increased due to interconnect effect i.e. power dissipation and crosstalk delay [4], [5]. Firstly, power dissipation divided into two i.e. static leakage power and dynamic power. Static leakage power becomes increased significantly due

to decreased device threshold voltage whereas the dynamic power dissipation has two major sources that are loading capacitance and coupling capacitance. In DSM technology, coupling capacitance is comparable to or exceeds the loading capacitance, which in turn causes the delay of a transition in a wire to be twice or more that of a wire transitioning next to a steady signal. This delay penalty is called crosstalk delay [6].

Several methods are employing to eliminate the power dissipation, crosstalk delay and propagation delay such as shield insertion between adjacent wires [7], repeater insertion [8], optimal spacing between signal lines [9] and lastly the most superior bus encoding method [10]-[17].

1.2 Motivation

On scaling down to deep submicron technology, reduction in interconnect spacing causes the dominance of coupling capacitance between adjacent lines. Increasing coupling capacitance has in turn led to increase the power dissipation and crosstalk delay on bus.

Bus encoding method is the method which converts or encode data bit stream in such a manner so that the transitions of the bit streams are minimized. By using this methods the switching activity and coupling activity reduces. On reduction these parameter both power dissipation and crosstalk reduces. There are different types of bus encoding scheme for data transmission. Here we focus on bus invert encoding scheme in which architectures of encoder and decoder should occupied lesser area, so that power and delay overheads due to codec circuitry can be compensated by the significant reduction of bus delay.

1.3 Thesis Organization

Chapter 2 presents a brief overview about the interconnect characteristics i.e. resistance and capacitance and then discuss briefly about the existing problem in interconnects.

Chapter 3 present the different design methodologies of interconnects.

Chapter 4 present a new proposed encoder to overcome the existing problem of interconnect.

Chapter 5 shows the simulation results and analysis at 1MHz to 1 GHz frequency.

Finally, conclusions and recommendations for future work are outlined in chapter 6.

CHAPTER 2

BASICS OF ON-CHIP INTERCONNECTS

2.1 Interconnect Characteristics

The impedance characteristics of on-chip interconnect includes the resistance and capacitance. These parameters can be extracted from the geometry of the interconnect structures, as illustrated in the following subsections.

2.1.1 Resistance

With further scaling down of technology, the cross sectional area of the lines are scaled down to provide more lines per unit area, while the length of the lines has increased. As a result the resistance of long signal lines increased significantly.

A rectangular cross-section is a fairly good approximation for an on-chip wire, as shown in Figure 2.1. The resistance of such a uniform strip of material is given by

$$R = \rho \frac{l}{A} \quad (2.1)$$

where ρ is the resistivity of the material, l and A are the length and area of on-chip wire.

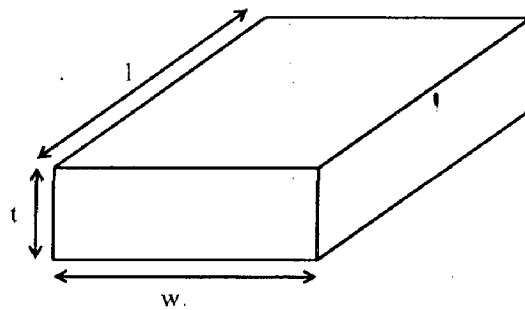


Figure 2.1: Cross-sectional dimensions of a conductor.

Table 2.1 shows the resistivity of several materials commonly used for conductors. Silver is the best in terms of conductivity, its high cost indicates that it is used only for special applications. The most commonly used material is aluminum, which is the more economical than silver and the most useful material is copper which is more expensive but has much better conductivity.

Table 2.1 Resistivity of materials used for conductors in VLSI circuit [19].

Material	Resistivity(ρ) Ω -m
Tungsten(W)	5.5×10^{-8}
Aluminum(Al)	2.7×10^{-8}
Gold(Au)	2.2×10^{-8}
Copper(Cu)	1.7×10^{-8}
Silver(Ag)	1.6×10^{-8}

2.1.2 Capacitance

As digital technology entered into the era of multi-layer interconnects lines and extremely dense integrated circuits, the line-to-reference capacitance alone has become insufficient for signal behavior analysis. The total capacitance of a line in multi-layer deep submicron digital integrated circuits is shown in Figure 2.2, now the summation of several components such as

- (i) Line-to-ground capacitance (C_G).
- (ii) Lateral or coupling capacitance (C_C) (formed between parallel edges of neighboring lines in the same plane).
- (iii) Parallel or crossover capacitance (C_P) (due to overlap area of two nets on different layers).
- (iv) Fringing capacitance (C_F) (formed between the edge of one conductor and the surface of another conductor on different layers).

The crossover (C_P) and fringing capacitances (C_F) increase as the number of neighboring metal layers increases around a particular conductor. But, the magnitudes of these two component capacitances are not that significant as compared to other two components of the total capacitance of a line in a multi-layer environment. However, the coupling capacitance to neighboring lines in the same layer is increased due to decrease of spacing between conducting lines, increase of interconnect aspect ratios and increase of length of lines running in parallel in the same layer. Therefore, coupling capacitance has become integral part of interconnect modeling along with line-to-ground capacitance.

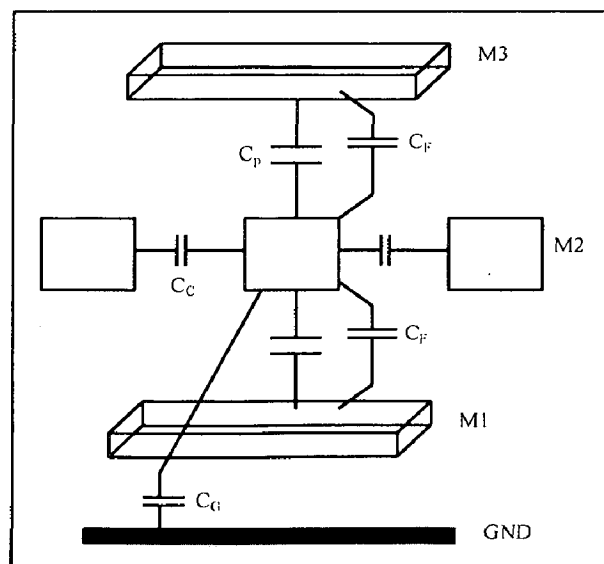


Figure 2.2: Capacitance in multi-layer representation.

2.2 Interconnect Design Criteria

Since interconnect has become a dominant issue in high performance ICs, the focus of the circuit design process has shifted from logic optimization to interconnect optimization. Multiple criteria should be considered during the interconnect design process, such as crosstalk delay, power dissipation and chip area. These criteria are individually discussed in the following subsections.

2.2.1 Crosstalk Delay

An unwanted coupling from a neighboring signal wire to a network node introduces an interference that is generally called crosstalk [6]. Due to crosstalk, the performance of the circuit becomes affected. Assume two lines namely A and B and their associate capacitances can be shown in Figure 2.3. According to behavior of neighboring wire, the effective coupling capacitance (C_{eff}) is defined. Table 2.2 shows the dependence of effective capacitance of line 'A' ($C_{eff(A)}$) on the line 'B' (assume line 'A' is switching).

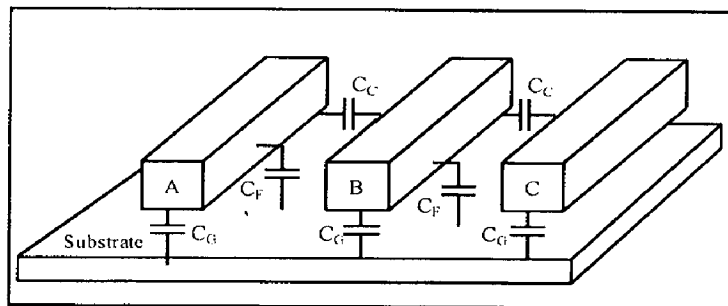


Figure 2.3: Impact of RC crosstalk.

Table 2.2 Dependence of effective capacitance of line 'A' on line 'B'.

Line 'B'	ΔV	Effective Coupling Capacitance of line 'A' ($C_{eff(A)}$)	Miller Coupling Factor (MCF)
Switching with 'A'	0	C_G	0
Constant	V_{dd}	$C_C + C_G$	1
Switching opposite 'A'	$2V_{dd}$	$2C_C + C_G$	2

Thus it can be concluded as follows. Firstly, when both the adjacent lines are switching in the same direction then the voltage over the coupling capacitances remains constant and miller coupling factor(MCF) is '0' which indicates that there is no coupling

capacitance. Secondly, if one line is switching and the other is quiet then MCF is '1' whose coupling capacitance is greater as compared to above case. Finally, when two adjacent lines are switching in opposite direction then coupling capacitance experience a voltage swing that is double the signal swing and thus MCF is '2' due to which crosstalk effect becomes dominant.

In a data bus there will be adjacent lines to the left and right side of the line which is of interest as shown in Figure 2.3. Therefore various coupling capacitances associated with the 3-bit configuration must be considered. Line 'B' is the line of interest and Line 'A', Line 'C' are the adjacent lines to it. Coupling factors associated with the Line 'B' depending on the switching configurations of Line 'A' and Line 'C' [19] are shown in Table 2.3.

Table 2.3 Dependence on switching configuration of line 'B' on line 'A' and 'C'.

Types	Line A	Line B	Line C	MCF
Type '0'	Quiet	Quiet	Quiet	0
	Switching in same direction of 'B'	Switching	Switching in same direction of 'B'	0
Type '1'	Quiet	Quiet	Switching	1
	Quiet	Switching	Switching in same direction of 'B'	1
	Switching	Quiet	Quiet	1
	Switching in same direction of 'B'	Switching	Quiet	1
Type '2'	Quiet	Switching	Quiet	2
	Switching	Quiet	Switching	2
	Switching in same direction of 'B'	Switching	Switching in opposite direction of 'B'	2
	Switching in opposite direction of 'B'	Switching	Switching in same direction of B	2
Type '3'	Quiet	Switching	Switching in opposite direction of 'B'	3
	Switching in opposite direction of 'B'	Switching	Quiet	3
Type '4'	Switching in opposite direction of 'B'	Switching	Switching in opposite direction of 'B'	4

Finally, all the possible switching configurations can be classified as Type-0, Type-1, Type-2, Type-3 and Type-4 and are summarized in the Table 2.4. ‘↑’ shows the switching from 0 to 1, ‘↓’ shows the switching from 1 to 0, and ‘-’ shows no transition. There are numbers of conditions which causes the crosstalk in interconnects i.e. Type-0, Type-1, Type-2, Type-3 and Type-4 causes three, eight, ten, four and two conditions respectively. The coupling capacitance becomes dominant in the case of Type-4 coupling when all the wires simultaneous switching in opposite direction and thus MCF is ‘4’. This is a worst case condition of *RC* model.

Table 2.4 Classification of crosstalk.

Type-0	Type-1	Type-2	Type-3	Type-4
---	--↑	-↑-	-↑↓	↑↓↑
↑↑↑	-↑↑	↑-↑	-↓↑	↓↑↓
↓↓↓	↑--	↑-↓	↑↓-	
	↑↑-	↑↑↓	↓↑-	
	--↓	↑↓↓		
	-↓↓	-↓-		
	↓--	↓-↓		
	↓↓-	↓-↑		
		↓↓↑		
		↓↑↑		

2.2.2 Power Dissipation

Due to higher clock frequencies and on-chip integration levels, power dissipation has significantly increased. The on-chip power dissipation of current state-of-the-art microprocessors is on the order of hundreds of watts. In Figure 2.4 shown the components of power dissipation due to different capacitance sources i.e. gate capacitance, diffusion capacitance and interconnect capacitance. The interconnect capacitance is the dominant one

among different capacitances which contains 51% out 100% total power dissipation. But the other components like gate capacitance and diffusion capacitance having 34 and 15 percentage power dissipation respectively. The dynamic power due to the interconnect capacitance can be greater than 50% of the total dynamic power. High power dissipation increases the packaging cost due to heating problems and shortens the battery life in portable applications. Power dissipation, therefore, is another important criterion in interconnect design.

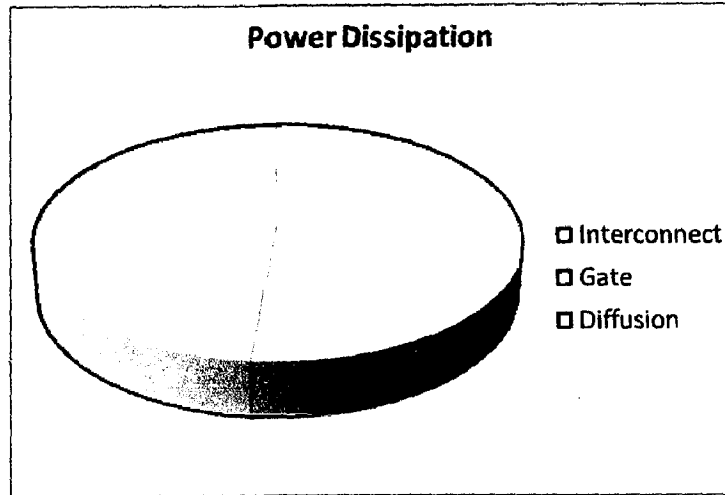


Figure 2.4: Components of power dissipation due to different capacitances sources.

The power dissipation in VLSI interconnects can be expressed as

$$P = (\alpha_{c_l} \times C_L + \alpha_{c_c} \times C_C) V_{dd}^2 \times f \quad (2.2)$$

$$P = (\alpha_{c_l} + \lambda \times \alpha_{c_c}) C_L \times V_{dd}^2 \times f \quad (2.3)$$

where C_L is the load capacitance, V_{dd} is supply voltage, f is the clock frequency, λ is the ratio of (C_C/C_L) , α_{c_l} is the average switching activities for load capacitances whose value lies between 0 and 1. The un-coded α_{c_l} value is equivalent to 1. The α_{c_c} indicates average coupling activities for coupling capacitances whose value lies between 0 and 1. Similarly, the un-coded α_{c_c} value also equals 1. As all other parameters are already optimized, the power dissipation which also depends on the switching activities is to be reduced (i.e. proportional to the number of signal transition).

2.2.3 Chip Area

With technology scaling, billions of transistors can now be integrated onto a single monolithic die. The number of interconnects has therefore also significantly increased. Therefore, the number of metal layers needs to be increased, to provide sufficient metal

resources for interconnect routing. Increasing the number of metal layers, however, increases the fabrication cost also. Furthermore, buffers and pipeline registers inserted along interconnects make the constraint on silicon area more stringent. The area criterion, therefore, should be considered during the interconnect design processes such as wire sizing and repeater insertion.

CHAPTER 3

DESIGN METHODOLOGIES FOR INTERCONNECTS

3.1 Introduction

Interconnect design methodologies have been developed at different levels to satisfy specific performance requirements i.e. total propagation delay, power dissipation, crosstalk delay and physical area. They are classified as:

- 1) Better interconnect materials.
- 2) Shielding line between two adjacent signal lines.
- 3) Repeater Insertion.
- 4) Bus Encoding Method.

3.1 Better Interconnect Materials

A first option for reducing RC delay is to use better interconnect materials. The material like **copper and silicides** has helped to reduce the resistance of polysilicon and metal wires respectively, while the adoption of dielectric materials with a **lower permittivity** lowers the capacitance. The disadvantage of these materials is that it only provides a temporary respite of one or two generations, and they do not solve the fundamental problems of the delay of long wires.

3.2 Shielding Line (V_{dd}/GND) between Adjacent Signal Lines [7]

Shielding in high speed digital circuits is an effective and common way to reduce crosstalk noise and signal delay. A common method of shielding is the placing of ground or power lines (V_{dd}/GND) between two adjacent signal lines to reduce noise and delay as shown in Figure 3.1.

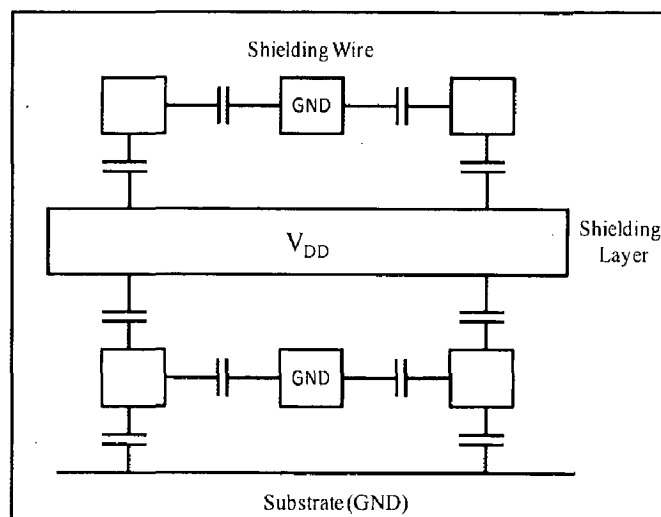


Figure 3.1: Shielding to reduce capacitive crosstalk.

This effectively turns the coupling capacitance into a capacitance to ground and eliminates interference. It eliminates the worst-case switching condition when two adjacent lines oppositely switch, resulting in a better worst-case delay. An adverse effect of shielding is that firstly it increased capacitance load and secondly it can only be applied to individual global interconnects and not to buses. This results in larger chip area.

3.3 Repeater Insertion [8]

Repeater insertion is used for long global interconnects. For driving long interconnects, a single buffer is not good solution as they present very large resistance and capacitive load at the terminal of the gate connected to it. Number of buffer need to insert at regular interval of distance which is termed as repeater which is shown in Figure 3.2.

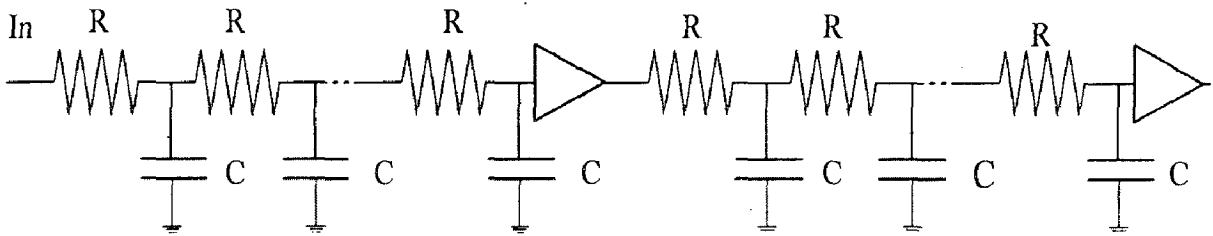


Figure 3.2: Breaking up a long transmission-gate chains by inserting buffers.

Assuming that the repeaters have a fixed delay t_{pbuf} , then the optimum number of repeater m_{opt} are

$$m_{opt} = L \sqrt{\frac{0.38 rc}{t_{pbuf}}} \quad (3.1)$$

For a given technology and a given interconnect layers, there exists an optimal length of the wire segments between repeaters. This critical length is given by the following expression:

$$L_{crit} = \frac{L}{m_{opt}} = \sqrt{\frac{t_{pbuf}}{0.38 rc}} \quad (3.2)$$

The advantage of repeater is that it reduces delay significantly when repeater is placed at regular interval. But the disadvantages of repeaters are

- 1) Repeater can take up significant fraction of active Si and routing area which degrade the performance.
- 2) Large number of buffer along an interconnect contribute overall delay to signal propagation.

- 3) Buffer themselves have certain switching time that contribute to delay and power dissipation.
- 4) It can only be applied to individual global interconnects and not to buses.

3.4 Bus Encoding Method

Bus encoding method is the method to convert or encode data bit stream. The purpose of this encoding is to prevent adjacent wires from transition in opposite direction, and this particular encoding achieves by forcing every other wire to a steady value.

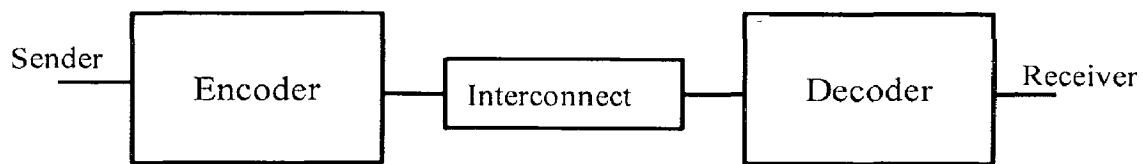


Figure 3.3: Model of communication chain.

We can model the chain of communication as shown in Figure 3.3. Adopting some terminology from coding theory, we say that the data words to be encoded are represented by symbols. The values placed on the channel by the encoder are called codewords and the mapping between symbols and codewords is called a codebook. The fundamental rule is achieved by giving a particular value currently on the channel, the next value cannot cause any adjacent wires to transition in opposite directions.

Various bus encoding methods are

- 1) Bus Invert Method (BI).
- 2) Partial Bus Invert Method (PBI).
- 3) Odd Even Bus Invert Method (OEBI).
- 4) Kim *et al.* Method.
- 5) Victor and Keutzer Method.
- 6) Lyuh and Kim Method.
- 7) Khan Method.
- 8) Fan *et al.* Method.

3.4.1 Bus Invert Method (BI) [11]

Bus invert method used to decrease the bus activity by reduction in number of transition. Due to which power dissipation reduces. It consist the random distribution of sequence of data in which one extra bus line (i.e. redundancy) is used. It considers all possible

next values on the bus. When the number of transmitting transitions is more than half of the bus width, the original data are inverted and the control line is set to ‘High’ , otherwise the original data are transmitted and control line is set to ‘Low’ or no transition when next value can be the same as the present one. The next value can differ in only one transition and there will be $C_{n+1}^1 = n+ 1$ such possible next values (there will be $C_n^1 = n$ values with *invert* = 0 and another one (all 1s data value) with *invert* = 1). Similarly there will be C_{n+1}^2 values that will generate two transitions up to values that will generate $n/2$ transitions.

$$C_{n+1}^0 + C_{n+1}^1 + C_{n+1}^2 + \dots + C_{n+1}^{n/2} = 2^n \quad (3.3)$$

Bus- invert method is preferred because any other code with 2^n codewords can use a permutation of these same patterns and then exhibit the same average bus activity or use patterns with more than $n/2$ transitions and generate a larger bus activity.

3.4.2 Partial Bus- Invert Method (PBI) [12]-[13]

The partial bus-invert method is an extension of the bus-invert method. It partitions bus lines into two parts, and then codes each part with the bus-invert method. Thus, the PBI method offers greater advantages than the BI method but it requires additional control lines. For instance, the PBI method increases two extra control lines in an 8-bit bus. In other words, the original data bus width changes from 8-bit to 10-bit, from 16-bit to 20-bit, and from 32-bit to 40-bit after PBI bus encoding. However, bus encoding techniques ignore crosstalk effects and mainly aim at reducing switching activities.

3.4.3 Odd-Even Bus Invert Method (OEBI) [14]

Coupling capacitances are always charged and discharged by activity on neighboring bus lines, where one line has an odd number and the other has an even number (if bus lines are numbered “in-order”). So it is intuitive that if we can handle the odd and even lines separately we may be able to reduce the coupling transitions. We propose the Odd-Even Bus-Invert Method (OEBI) to tackle the coupling problem this way. Somewhat similar to the original BI scheme, Odd-Even Bus Invert Method will use two extra lines to indicate the inversion of the odd lines, or of the even lines, respectively. There are four possible cases with two invert-lines: no bus lines are inverted (00), only odd lines are inverted (10), only even lines are inverted (01), or all lines are inverted (11). Unlike the regular BI case, determining the optimal encoding for the two invert-lines of OEBI is more difficult. In the first version of the OEBI scheme, which we call the *Calculated Odd Even Bus-Invert*, we

explicitly compute the coupling transitions for all four possible cases. We can then choose the case with the minimum number of coupling transitions as the encoding pattern to transmit over the bus.

3.4.4 Kim *et al.* Method [15]

Kim *et al.* method reduce the inter-wire coupling capacitance without eliminating any type of worst-case crosstalk (Type-4, Type-3 and Type-2) will result in low power, but will not reduce the maximum bound on delay penalty that limits the performance and reliability of high speed on-chip buses. Therefore, this method is not advantageous over un-encoded data.

3.4.5 Victor and Keutzer Method

Victor and Keutzer Method is well suited for reducing crosstalk as the coding eliminates all worst crosstalk types. However, there is no guarantee that the method will also be power efficient.

3.4.6 Lyuh and Kim Method [16]

Lyuh and Kim method is well suited for both power efficiency and elimination of all types of worst crosstalk (Type-4, Type-3 and Type-2). However, this method exploits the probabilistic information of the data stream, it cannot be applied to a data, the statistical properties of which cannot be known a priori, and therefore cannot be applied to generic SoC systems.

3.4.7 Khan Method [17]

Khan Method targets the crosstalk problem from both power and delay perspectives. It transforms the incoming data in such a way as to eliminate two worst crosstalk types (Type-4 and Type-2). By doing so, the worst-case delay in signal transition will be eliminated and the delay will now depend on the crosstalk which is less severe. At the same time, the scheme provides power reduction by minimizing self and coupled switched capacitance.

Two aspect of this encoding scheme are firstly the elimination/ minimization of worst crosstalk and second the energy efficiency. The energy expression for a 3-bit bus can be expressed as

$$E_1 = C_L \{ (1 + \lambda)(V_1^f - V_1^i) - \lambda(V_2^f - V_2^i) \} \cdot V_1^f \quad (3.4)$$

$$E_2 = C_L \{ (-\lambda)(V_1^f - V_1^i) + (1 + 2\lambda) \cdot (V_2^f - V_2^i) - \lambda(V_3^f - V_3^i) \} \cdot V_2^f \quad (3.5)$$

$$E_3 = C_L \{ (-\lambda)(V_2^f - V_2^i) + (1 + \beta) \cdot (V_3^f - V_3^i) \} \cdot V_3^f \quad (3.6)$$

$$E = E_1 + E_2 + E_3 \quad (3.7)$$

Here V_1^f, V_2^f, V_3^f are final and V_1^i, V_2^i, V_3^i are the initial states of the three wires, respectively. They can be either V_{DD} or 0. E_1, E_2 and E_3 represent energy for wires 1, 2 and 3 respectively.

The energy saving can be calculated by using the expression:

$$\text{Energy saving} = \left(\frac{1-N_c}{N_u} \right) \times 100 \quad (3.8)$$

where N_u and N_c are respectively, the net switching activities in the unencoded and corresponding encoded data.

The disadvantage of this method is that it is not eliminate Type-3 coupling, which is worst case crosstalk in RC modeled. Internal circuitry uses two N4_count and two N2_count due to which size of the chip increases, therefore complexity of chip increases. Other factors like power dissipation and total propagation delay also increases.

3.4.8 Fan *et al.* Method [18]

Fan *et al.* method is based on bus invert method. It focuses on crosstalk effects, and is specially applied to three adjacent wires. They reduce not only the dynamic power dissipation but also total propagation delay. Therefore, they improve overall system reliability.

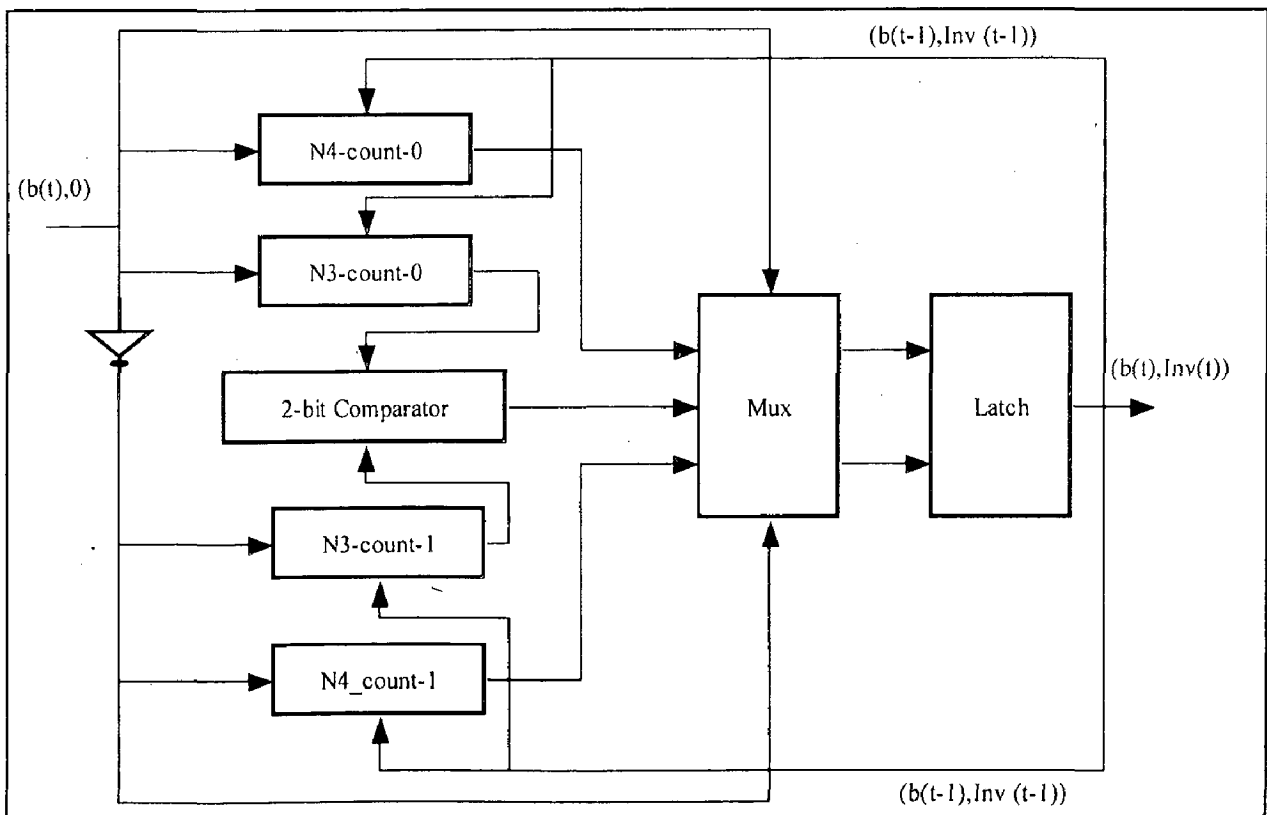


Figure 3.4: Block diagram of 5-bit bus encoder architecture.

The Fan *et al.* method divides the bus width into several clusters. Each cluster has a 4-bit width with an extra control bit. After the bus encoding is applied, the n -bit bus extends to $(n + n/4)$ bits. The bus encoder outputs the invert the input data when the original input data and the previous bus state, i.e. $((b(t), 0)$ and $((B(t - 1), Inv(t - 1)))$ cause the Type-3, or Type-4 crosstalk effect. On the contrary, it outputs the original input data when the reversed original data and the previous bus state, i.e. $((\overline{b(t)}, 1)$ and $((B(t - 1), Inv(t - 1)))$ cause the Type-3, or Type-4 crosstalk effect.

In Figure 3.4, the 5-bit bus encoder architecture is composed of a NOT gate, two $N4_count$ s, two $N3_count$ s, a 2-bit comparator, a multiplexer and a register. The $N4_count$ and $N3_count$ modules are crosstalk detection modules. The purpose of the $N4_count$ detects the Type-4 coupling, and the purpose of the $N3_count$ detects the Type-3 coupling. The $N4_count$, the $N3_count$ and the 2-bit comparator provide 1-bit, 2-bit, and 1-bit outputs, respectively. The $N4_count_0$ module calculates whether the $(b(t), 0)$ data will cause the Type-4 coupling with respect to the previous bus state $(B(t - 1), Inv(t - 1))$, and the $N4_count_1$ module calculates whether the $(\overline{b(t)}, 1)$ data will cause the Type-4 coupling with respect to the previous bus state $(B(t - 1), Inv(t - 1))$. If Type-4 coupling is present then $Inv(t)$ value is set to 'High', otherwise it becomes 'Low'. The $N3_count_0$ module calculates and outputs the amount of the Type-3 coupling between $((b(t), 0)$ and $((B(t - 1), Inv(t - 1)))$, and the $N3_count_1$ module calculates and outputs the amount of the Type-3 coupling between $((\overline{b(t)}, 1)$ and $((B(t - 1), Inv(t - 1)))$. If the output of the $N3_count_0$ module is larger than that of the $N3_count_1$ module, the $B(t)$ value is set to the $\overline{B(t)}$ value and the $Inv(t)$ value is set to 'High'. Otherwise, the $B(t)$ value is set to the $b(t)$ value and the $Inv(t)$ value is set to 'Low', therefore, it greatly decreases the Type-3 coupling. The multiplexer uses the data from the $N4_count$ and the $N3_count$ modules to choose output values (i.e. 4-bit data and the $Inv(t)$ bit) at the next positive clock edge.

The output signal of the $N4_count$ is either 1 or 0 per 5-bit data transmission. Figure 3.5 detects signals among $b0(t)$, $b1(t)$, and $b2(t)$ bits, or among $b1(t)$, $b2(t)$ and $b3(t)$ bits, or among $b2(t)$, $b3(t)$ bits, and 0. Internal circuits of the $N4_count$ are composed of basic logic gates, which are seven two-input XOR gates, three four-input AND gates and a three-input OR gate. In the $N4_count$ module, when the 3-bit input signal is switched oppositely with respect to their previous state, it is impossible that two adjacent bits in the 3-bit signal are transited with the same transitional direction. Therefore, the output from the four-input AND gate is set to 'High' signal.

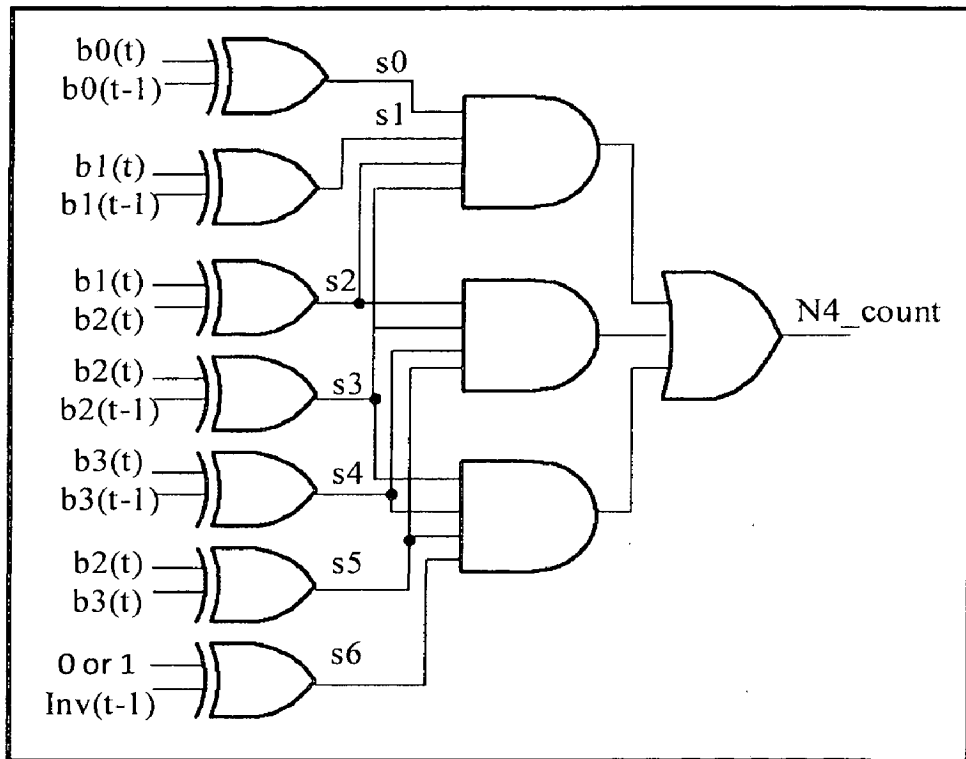


Figure 3.5: Internal block diagram of $N4_count$ circuit.

Internal circuits of the $N3_count$ module are composed of a 6-bit adder, six four-input AND gates, six NOT gates and two 2-input XOR gates, shown in Figure 3.6.

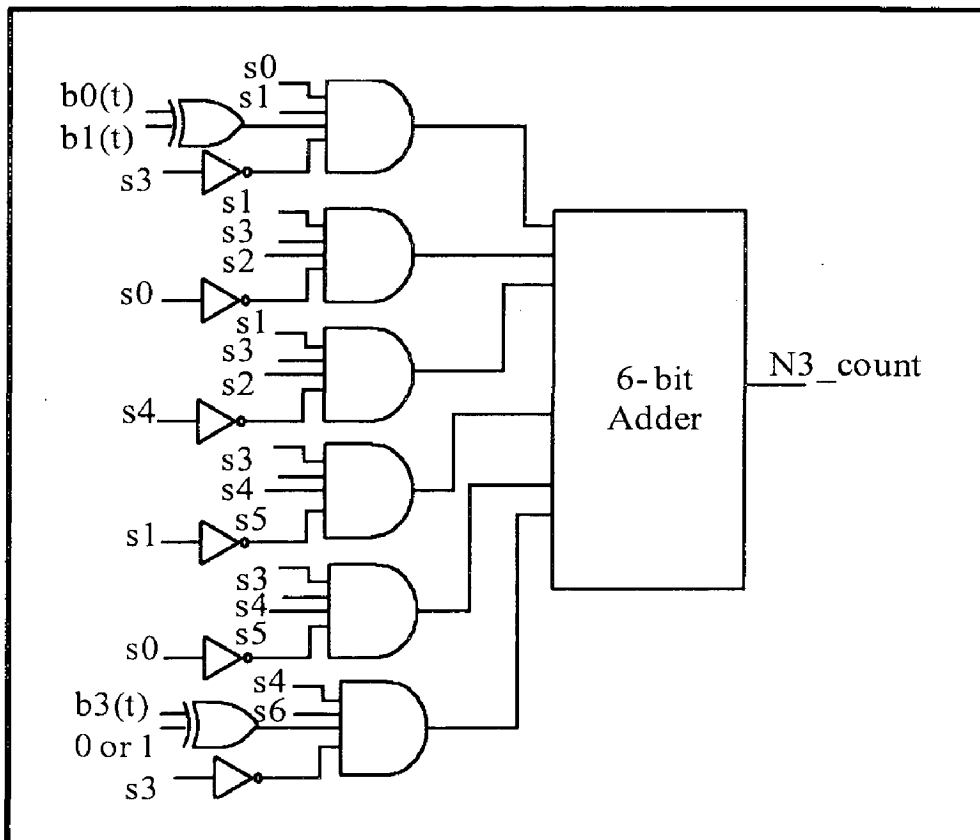


Figure 3.6: Internal block diagram of $N3_count$ circuit.

Although the 6-bit adder is used, it only operates a 2-bit addition because the maximum Type-3 coupling effect in a particular 5-bit data transfer is two. For example, the input signal is 10010 and the previous state is 01001. Therefore, the $N3_count$ module detects two Type-3 couplings. This study also uses the 2-bit comparator to compare the data from two $N3_count$ modules, and both the outputs from the $(b(t), 0)$ and $(\overline{b(t)}, 1)$ of the $N3_count$ modules are compared in the 2-bit comparator module. If the number of the Type-3 coupling in $b(t), 0$ is greater than that in $(\overline{b(t)}, 1)$, the comparator output is logic '1', otherwise, it is logic '0'.

This method turns the worst crosstalk effects into the Type-0 and Type-1 couplings; however, the Type-0 and Type 1 couplings will be increased after the bus encoding is applied. Most bus encoding methods primarily minimize the worst crosstalk effects, and do not consider the Type-1 coupling. The disadvantage of using this method is that due to the large circuitry which dissipates more power, consume more area and increase chip complexity.

CHAPTER 4

PROPOSED *RC* MODEL

4.1 Introduction

Different *RC* methods like Khan, Fan *et al.* method are used to reduce cross talk and power dissipation, but their circuitry became so complex due to which they acquire large area and dissipate more power. Therefore, a new method is proposed which reduces the crosstalk delay and power dissipation and acquires less area as compared to these *RC* models.

4.2 Proposed Encoder

In this proposed method, the data bus is divided into different clusters. Each cluster has 4-bit width with one extra control bit. Extra control bit is known as invert pin i.e. $INV(t)$. Bus invert method uses an extra line called invert pin to differentiate between the transmission of original data and inverted data. Decreasing the switching activity is one of the methods to decrease the power consumption in interconnects. This proposed encoder limits the number of transitions. If the number of transitions that are being transmitted are more than half of the bus width, then the original data is inverted and the control line $INV(t)$ is set to 'High', otherwise, the original data is transmitted and the control line $INV(t)$ is set to 'Low'.

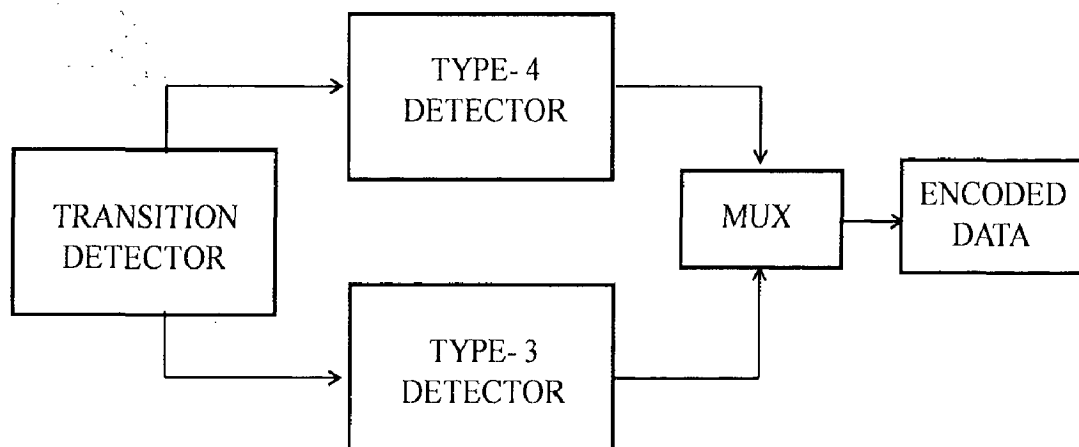


Figure 4.1: Main block diagram of proposed encoder.

Similarly in this paper, if the original input data causes crosstalk then the inverted data is transmitted and control line is set to 'High'. For this purpose the proposed design must detect the input data which causes a crosstalk effect. The proposed method detects the crosstalk condition by comparing the present data $(b(t), 0)$ with the previous data $(b(t - 1), 1)$ and depending on transition of the data bits a decision is made i.e. whether the input data causes a crosstalk or not. However, the architectures of the encoder and decoder should

be of low complexity so that the power and delay overheads due to the codec circuitry can be compensated by the significant reduction of bus delay.

The proposed scheme is shown in Figure 4.1. It shows the block diagram of encoder. The proposed encoder reduces the crosstalk by complementing the original data $(b(t), 0)$ which causes the crosstalk. This method consists of four major blocks which are Transition Detector, Type-4 Detector, Type-3 Detector, Multiplexer and Latch.

The first block is the transition detector which detects the transition by comparing the present data with the previous data. The next step after detecting transition is to examine whether these transitions causes crosstalk or not. The proposed method employs two detectors i.e. Type-4 detector to detect the Type-4 couplings and Type-3 detector check the Type-3 couplings. This method is also capable to reduced Type-2 coupling. As Type-4 and Type-2 coupling is fully eliminated but in the case of Type-2 coupling only fewer cases are diminished. If either of couplings (i.e. Type-4, Type-3 and Type-2) is present, through multiplexer the $INV(t)$ pin becomes 'High'

4.2.1 Transition Detector

Transition detector checks the occurrence of transition by using AND gates.

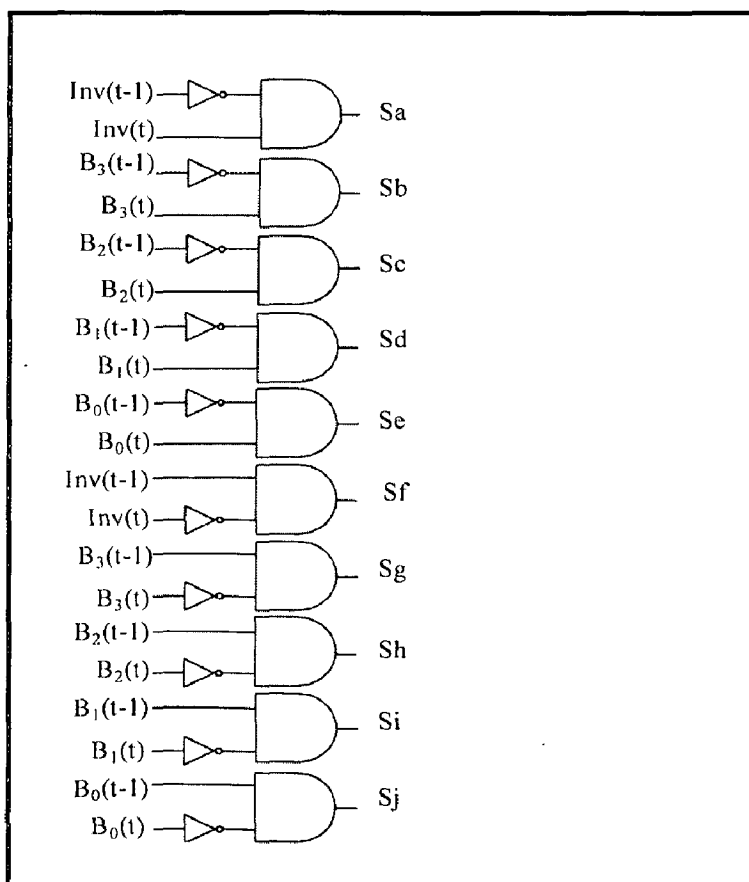
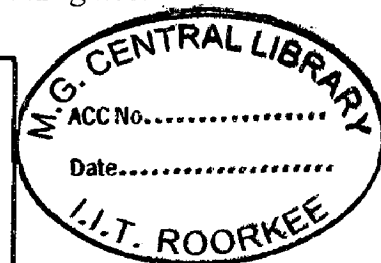


Figure 4.2: Block diagram representation of transition detector.



The top 5 AND gates detects the low to high transition (\uparrow) and the bottom 5 AND gates detects the high to low transition (\downarrow) as shown in Figure 4.2. For this purpose it uses the data which is transmitted previously, it compares the present ($b(t), 0$) data with the previous ($b(t - 1), 1$) data. If there is any transition i.e. from high to low (\downarrow) or from low to high (\uparrow) the output becomes 'High'. Otherwise, if any transition is not present then it became 'Low'.

4.2.2 Type-3 and Type-4 Detector

As discussed in the chapter 2 there are two cases which cause Type-4 couplings and four cases which causes Type-3 coupling as tabulated in Table 2.3. Figure 4.3 shows the two cases of Type-3 coupling and all the cases of Type-4 coupling. The top three NAND gates demonstrate first case ($- \uparrow \downarrow$) of Type-3 and ($\downarrow \uparrow \downarrow$) of Type-4 coupling and makes the three combinations of lines i.e. (Sa, Sb, Sh), (Sb, Sc, Si) and (Sc, Sd, Sj).

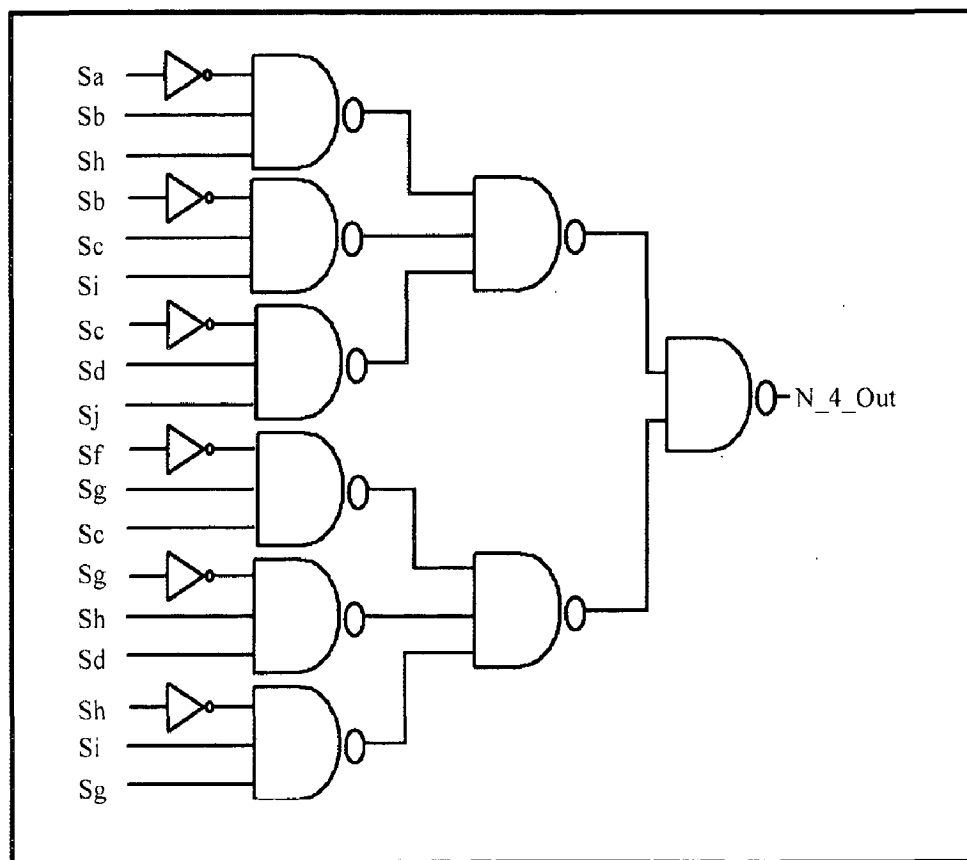


Figure 4.3: Block diagram of Type-3 ($(-\uparrow\downarrow), (-\downarrow\uparrow)$) and Type-4 detector.

Similarly, bottom three NAND gates demonstrate second case ($- \downarrow \uparrow$) of Type-3 and ($\uparrow \downarrow \uparrow$) of Type-4 coupling and makes a three combinations of lines i.e. (Sf, Sg, Sc), (Sg, Sh, Sd) and (Sh, Si, Sg). The outputs from the transition detector are used to detect the crosstalk by connecting them logically using NAND gates and if any Type-3 and Type-4 coupling are present then the output of first stage NAND gate become 'High', otherwise it is 'Low'. The

output from the final stage NAND gate i.e. N_4_out goes 'High' if there is any $\{(- \uparrow \downarrow), (- \downarrow \uparrow), (\downarrow \uparrow \downarrow), (\uparrow \downarrow \uparrow)\}$ crosstalk are present.

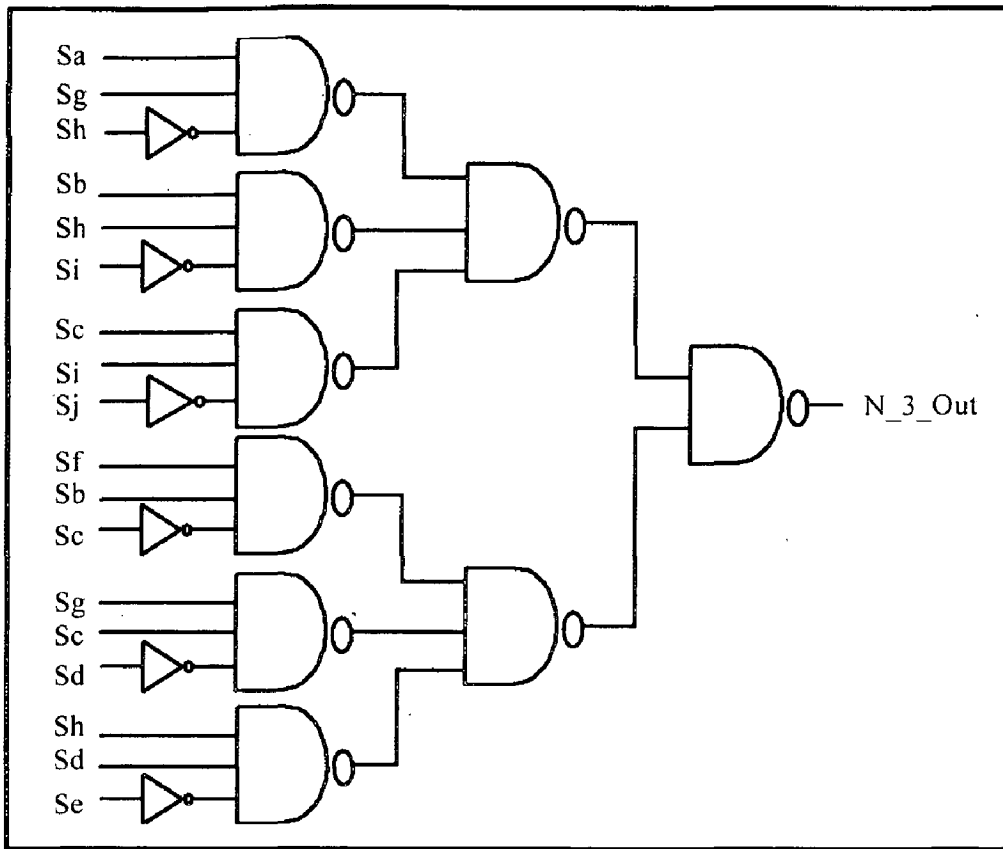


Figure 4.4: Block diagram of Type-3 $((-\uparrow\downarrow), (-\downarrow\uparrow))$ and Type-4 detector.

Figure 4.4 shows another two cases of Type-3 coupling and all the cases of Type-4 coupling. The top three NAND gates demonstrate third case $(\uparrow \downarrow -)$ of Type-3 and $(\uparrow \downarrow \uparrow)$ of Type-4 coupling and makes the three combinations of lines i.e. (Sa, Sg, Sh) , (Sb, Sh, Si) and (Sc, Si, Sj) . Similarly, bottom three NAND gates demonstrate fourth case $(\downarrow \uparrow -)$ of Type-3 and $(\downarrow \uparrow \downarrow)$ of Type-4 coupling and makes a three combinations of lines i.e. (Sf, Sb, Sc) , (Sg, Sc, Sd) and (Sh, Sd, Se) . The outputs from the transition detector are used to detect the crosstalk by connecting them logically using NAND gates and if any Type-3 and Type-4 coupling are present then the output of first stage NAND gate become 'High', otherwise it is 'Low'. The output from the final stage NAND gate i.e. N_3_out goes 'High' if there is any $\{(\uparrow \downarrow -), (\downarrow \uparrow -), (\downarrow \uparrow \downarrow), (\uparrow \downarrow \uparrow)\}$ crosstalk are present.

4.2.3 Multiplexer

The truth table of multiplexer is shown in Table 4.1. When either N_4_out or N_3_out is 'High', the inverted data must be transmitted otherwise original data bits are transmitted. In this case, the data bit is fed as one of the input for the XOR gate and control line $(INV(t))$ is

Table 4.1 Truth table of multiplexer.

N_4_out	N_3_out	MUX OUTPUT
0	0	$(B(t), 0)$
1	0	$(\overline{B(t)}, 1)$
0	1	$(\overline{B(t)}, 1)$
1	1	$(\overline{B(t)}, 1)$

fed as the second input for the 2-input XOR gate as shown in Figure 4.5. If the $(INV(t))$ line is 'High' then it indicates that the inverted data must be transmitted and $B(t)$ must be inverted to avoid the crosstalk and if it is 'Low' which indicates that the original data is to be transmitted.

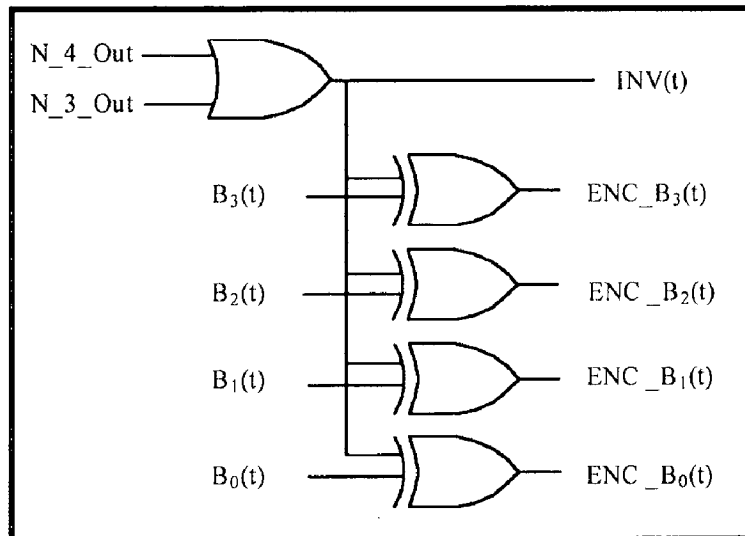


Figure 4.5: A multiplexer.

4.2.4 Latch

As proposed design is comparing the present data $(b(t), 0)$ with the previously transmitted data $(b(t - 1), 1)$ which means that there is a necessity for storing the previously transmitted data. For this purpose latches [19] are necessary that are implemented using transmission gates as shown in Figure 4.6.

Latch consists of two transmission gates (T_1 & T_2) and two inverters (one inverter is connecting in forward path and another one is in feedback path). The transmission gate

consists of both PMOS and NMOS transistors whose drains & sources are shorted together. When CLK is 'High', then T_1 goes into 'ON' and T_2 goes into 'OFF' condition and thus the input data is transmitted to output of T_1 (i.e. $B(t)$). Similarly, when CLK goes 'Low' then T_2 goes into ON and T_1 goes into OFF condition and thus the data is retained using the inverters which are connected in feedback.

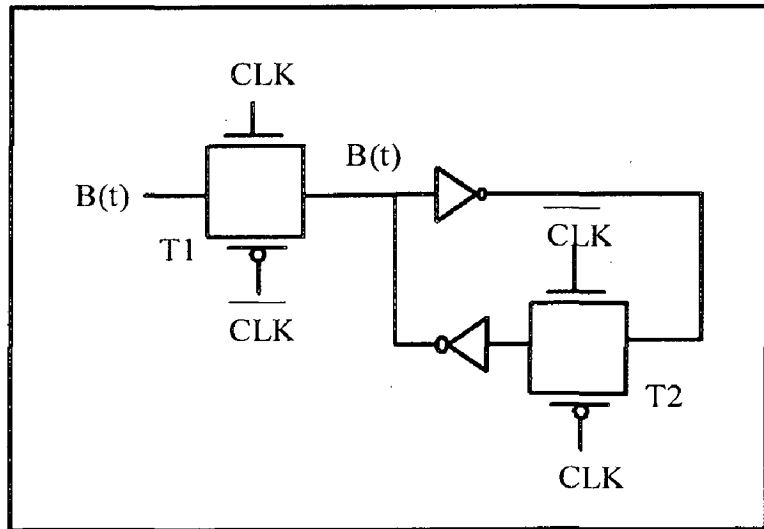


Figure 4.6: Latch with transmission gates.

4.3 Decoder

The function of decoder is to decode the encoded data. The internal circuit of decoder is shown in Figure 4.7.

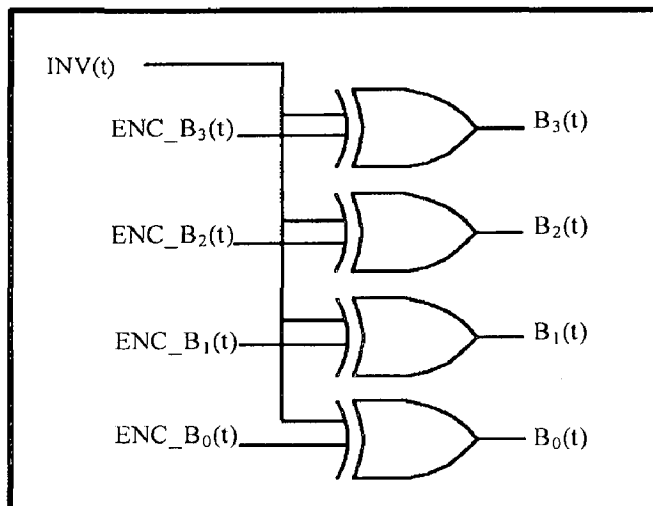


Figure 4.7: A decoder.

The encoded data is fed as one of the input for the XOR gate and control line

$(INV(t))$ is given as the other input for the 2-input XOR gate. If the $(INV(t))$ line is 'High' then it indicates that the inverted data has been transmitted and $(ENC_B(t))$ must be inverted to get the original data and if it is 'Low', indicates that the original data has been transmitted.

CHAPTER 5

RESULTS AND ANALYSIS

5.1 Simulation Results

The proposed method has been simulated at CMOS 180, 130, 90, 70, 45 nm technology nodes by using H-SPICE with pulse stimuli of frequency 1 MHz to 1 GHz whose rise time and fall times are 4ps. The length, width, thickness and spacing of the signal wire are 1300, 0.99, 0.53 and 1.37- μm respectively. Both Fan *et al.* [18] RC model and the proposed model are simulated in the above HSPICE environment. The proposed model is showing significant reduction of the power consumed by the encoder, propagation delay, crosstalk and chip area of the encoder than Fan *et al.* [18] RC model. We adopt 4-bit, 8-bit, and 16-bit bus data to simulate different bus encoding techniques.

5.2 Crosstalk Reduction

The proposed method eliminates the worst case crosstalk effect which is introduced in between redundant bit and clusters. Figure 5.1 shows the use of redundant shielding line to eliminate Type-4 and Type-3 couplings between inter-cluster regions.

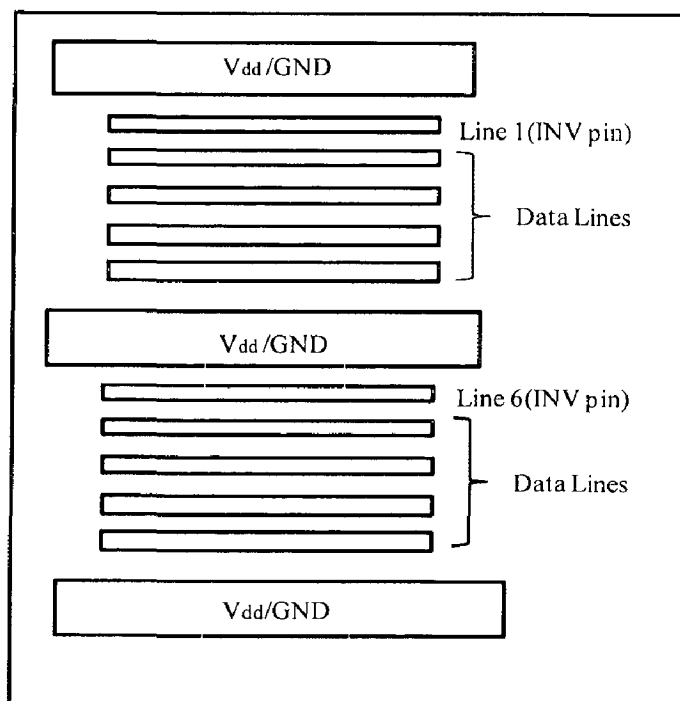


Figure 5.1: Interconnect routing for an 8-bit bus.

Reduction of crosstalk can be estimated by considering the number of switching configurations which cause worst case crosstalk (Type-3 and Type-4). In this bus encoding scheme the data which causes crosstalk is inverted and upon inversion it gets converted to a switching configuration whose coupling factor is less than the previous. Thus a Type-4 is

converted to either Type-3 or Type-2 or Type-1 or Type-0 depending on the switching configuration and Type-3 is converted to Type-2 or Type-1 or Type-0.

Thus it is concluded that all the Type-4 couplings are reduced (100%) but some of the Type-4 are being converted to Type-3 and so the reduction in Type-3 couplings is 76.8% and some of the Type-2 couplings are automatically reduced (48.46%). Reducing the crosstalk in interconnects also reduces the propagation delay which is introduced on the victim due to an aggressor switching in the opposite direction.

5.3 Chip Area Reduction

The proposed method has greatly reduced the chip area by reducing (57%) the number of transistors as compared to Fan *et al.* [18] RC model as shown in Table 5.1. In proposed scheme, the numbers of components are going to be reduced, so that the complexity of circuit gets diminished. Both area and complexity are reduced by using proposed method.

Table 5.1 Comparison of components of proposed method with Fan *et al.* method.

Components	Fan <i>et al.</i> [18]	Proposed Method	% of saving
AND gate	4-input	2-input	50%
6-bit adder	2	0	-----
XOR gate	18	8	55%
Number of transistors	664	284	57%

5.4 Total Power Reduction

Total power dissipation of the system includes the power dissipated by encoder, interconnects and the decoder (*i. e.* $P_{enc} + P_{interconnects} + P_{dec}$). The V_{dd} for 180, 130, 90, 70, 45 nm technologies are taken as 1.8, 1.5, 1.2, 1.0 and 0.9 V respectively. Encoders for 4, 8 and 16 bit interconnects are considered here.

An increasing frequency of 1MHz to 1GHz causes the signal to be distorted. Therefore, two cases are to be considered here. For first case, the signals passes through transistors without sizing and for second case, the signal passes through transistors with

sizing. The sizing of transistors has been done by using logical effort [20]. Logical effort offers a systematic approach to topology selection and gate sizing. It is a technique which is used to estimate delay in CMOS circuits. Minimum delay is possible for a circuit by including the proper sizing of the gates which are mainly concerned about to design the fast chip.

Firstly, at low frequency i.e. 1MHz, the signals are easily passed through transistors without sizing. Table 5.2 shows the comparison of power dissipation of proposed method with Fan *et al.* method. The Fan *et al.* method dissipated more power as compared to proposed scheme as 57% more components are required for Fan *et al.* method as compared to proposed method. For 180, 130, 90, 70 and 45 nm technologies, the percentage saving of power for proposed method as compared to Fan *et al.* method are 67.33%, 70.40%, 68.12%, 65.33%, and 51.45% respectively.

Table 5.2 Comparison of power dissipation of proposed method with Fan *et al.* at 1 MHz.

Technology (nm)	Coding Methods	Power Dissipation (μ W)			% of power saved
		4 bit	8 bit	16 bit	
180	Fan <i>et al.</i>	28.32	77.40	99.14	67.33%
	Proposed	9.26	21.24	32.69	
130	Fan <i>et al.</i>	9.17	24.24	32.12	70.40%
	Proposed	2.55	7.26	10.19	
90	Fan <i>et al.</i>	5.39	13.81	19.25	68.12%
	Proposed	1.81	4.26	6.31	
70	Fan <i>et al.</i>	4.21	10.55	14.57	65.33%
	Proposed	1.46	3.31	5.12	
45	Fan <i>et al.</i>	1.14	9.74	10.26	51.45%
	Proposed	0.95	2.43	4.24	

As the feature size or technology goes on decreasing then the power dissipation corresponding to them also reduces. It is due to fact that on reduction in feature sizes, power dissipation is going to be minimized as V_{dd} reduces. Figure 5.2 shows the variation of power dissipation of system with technology. The graph shows that at the same technology node, increasing bit size (i.e. 4, 8 and 16) increases the power dissipation, which is less than the Fan *et al.* method.

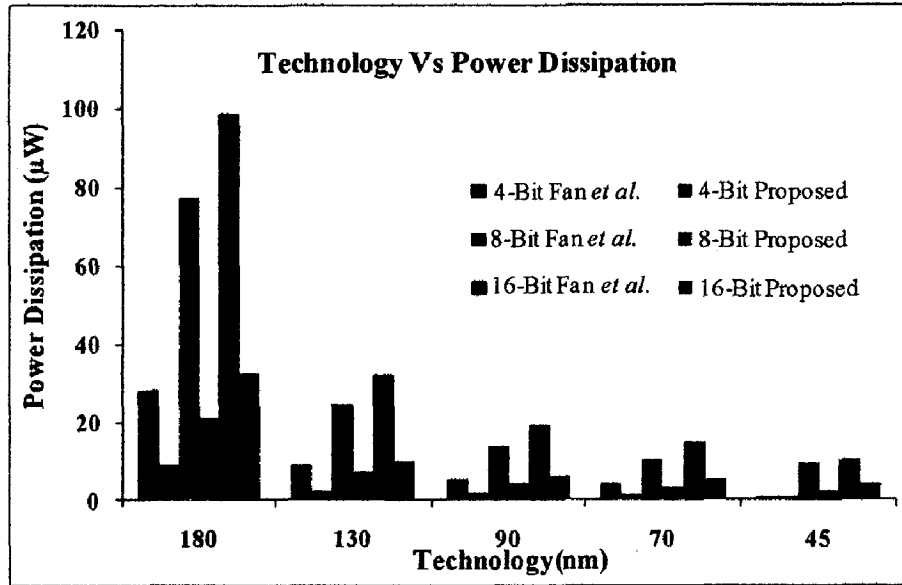


Figure 5.2: Variation of power dissipation of system with technology at 1 MHz.

Now increasing frequency at 125MHz, the Fan *et al.* method is working without using the logical effort. Table 5.3 shows the power dissipation of Fan *et al.* method at 125MHz. Since increasing frequency causes to increase the power dissipation because frequency and power dissipation are directly proportional to each other as discussed in section 2.2.2.

Table 5.3 Power dissipation of Fan *et al.* method at 125 MHz.

Technology (nm)	Power dissipation (µW)
180	1300.23
130	1063.77
90	835.75
70	509.35
45	330.47

But above 125MHz frequency, the Fan *et al.* method is not capable to work properly. This is due to the fact that signal passes through the transistors are going to be distorted. But the proposed scheme is capable to function properly at this frequency whereas it has a limitation to work upto 800 MHz without using logical effort.

Table 5.4 shows the power dissipation of proposed method at 800 MHz. Similarly, the proposed method is not able to work at above 800MHz. But, the basic requirement of the digital circuits is to design the fast chip. So above 800 MHz, sizing of the transistors becomes an important issue.

Table 5.4 Power dissipation of proposed method at 800 MHz.

Technology (nm)	Power dissipation (μW)
180	778.8
130	498.44
90	284.23
70	169.57
45	96.93

By using logical effort, the sizing of transistors is performed. The first step is to calculate the electrical effort along the path. The electrical effort (H) along the path through a network is simply the ratio of the load capacitance of the last logic gate in the path to the input capacitance of the first logic gate in the path.

$$H = \frac{C_{out}}{C_{in}} \quad (5.1)$$

The value of C_{out} and C_{in} are calculated by using H-Spice tool. The values of C_{out} and C_{in} are 82.966 fF and 0.683 fF respectively. Therefore the electrical effort is 121.47. The second step is to calculate the logical effort (G) which captures the effect of the logic gate's topology on its ability to produce output current. The third step is to calculate the branching effort (B) which is the ratio of total load capacitance along the path. The fourth step is to calculate the path effort which is given as

$$F = GBH \quad (5.2)$$

The fifth step is employed for calculation of the effort delay (f) which is given by

$$f = \sqrt[N]{F} \quad (5.3)$$

where N is number of the stages. Finally the C_{in} of each stage is calculate by using the formula,

$$C_{in_i} = \frac{gC_{out_i}}{f} \quad (5.4)$$

By the value of C_{in_i} , the value of W/L ratio of each transistor is calculated. Table 5.5 shows the power dissipation of proposed method at 1GHz frequency. Power dissipation at 1 GHz is quite large because on sizing, the transistor width is going to be increased which in turn increases the area of the chip.

Table 5.5 Power dissipation of proposed method at 1 GHz.

Technology (nm)	Power dissipation (μW)
180	956.34
130	742.87
90	455.82
70	336.76
45	156.35

Figure 5.3 shows the power dissipation of proposed method with technology at 1 GHz frequency.

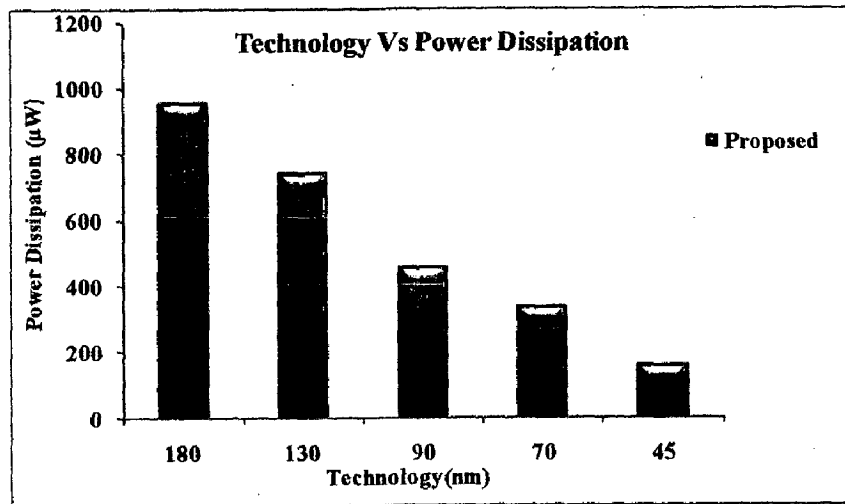


Figure 5.3: Power dissipation of system with technology at 1 GHz.

5.5 Total Propagation Delay

Propagation Delay on the victim line increases due to crosstalk. For proposed method, this crosstalk is going to be reduced by introducing the encoder but some sort of overhead delay may be introduced there. Although there is a reduction in the propagation delay with the reduction of crosstalk, the overhead delay should be considered also. Therefore, an encoder with low propagation delay is needed and this proposed model introduces less overhead delay as compared to Fan *et al.* [18] RC model.

Here, also two cases are needed to be considered for calculating the total propagation delay. Firstly, the signals pass through transistors without sizing and secondly with sizing. At low frequency of 1MHz, the signals easily pass through transistors without sizing. Table 5.6 shows the reduction of delay for proposed model as compared to Fan *et al.* [18] at 180,

130, 90, 70, 45 nm technologies are 63.29%, 57.54%, 56.47%, 54.76% and 57.33% respectively.

Table 5.6 Comparison of propagation delay of proposed method with Fan *et al.* at 1 MHz.

Technology (nm)	Coding Methods	Propagation Delay (ps)	% Reduction in Delay
180	Fan <i>et al.</i>	287.63	63.29%
	Proposed	105.58	
130	Fan <i>et al.</i>	315.42	57.54%
	Proposed	133.91	
90	Fan <i>et al.</i>	384.43	56.47%
	Proposed	167.32	
70	Fan <i>et al.</i>	433.2	54.76%
	Proposed	195.94	
45	Fan <i>et al.</i>	619.37	57.33%
	Proposed	264.26	

Propagation delay introduced at 1 MHz for the proposed method and Fan *et al.* [18] model for various technologies are shown in Figure 5.4. The total propagation delay increases with the scaling of technology which is discussed in the section 1.1.

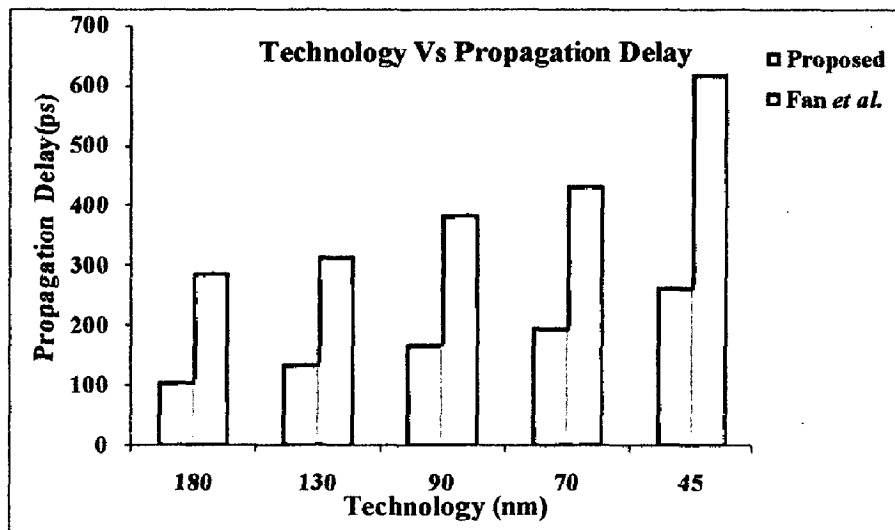


Figure 5.4: Variation of total propagation of system with technology at 1 MHz.

Now on increasing in frequency i.e. at 125 MHz, the Fan *et al.* method is working without using the logical effort. Table 5.7 shows the total propagation delay of Fan *et al.* method at 125 MHz. Since increasing frequency causes to increase the propagation delay

because of the propagation delay and time period are indirectly proportional to each other. Therefore, the propagation delay and frequency are directly proportional to each other.

Table 5.7 Propagation delay of Fan *et al.* method at 125 MHz.

Technology (nm)	Total Propagation Delay (ps)
180	365.43
130	480.13
90	539.65
70	628.92
45	780.67

But above 125MHz frequency, the Fan *et al.* method is not capable to work properly. But the proposed scheme is capable to function properly up to 800 MHz frequency. Table 5.8 shows the total propagation delay of proposed method at 800 MHz. Similarly, the proposed method is not able to work at above 800MHz. But, above 800 MHz, sizing of the transistors becomes an important issue.

Table 5.8 Propagation delay of proposed method at 800 MHz.

Technology (nm)	Total Propagation Delay (ps)
180	195.45
130	279.86
90	328.29
70	421.84
45	540.45

By using logical effort, the sizing of transistors is performed. Table 5.9 shows the total propagation delay of proposed method at 1GHz frequency. Sizing of the transistors increases the widths which in turns reply that current capability is going to be increased with minimization of resistances which further reduces the overall propagation delay.

Table 5.9 Propagation delay of proposed method at 1 GHz.

Technology (nm)	Total Propagation Delay (ps)
180	123.14
130	235.56
90	278.64
70	385.43
45	493.67

Figure 5.5 shows the total propagation delay of proposed method with technology at 1 GHz frequency.

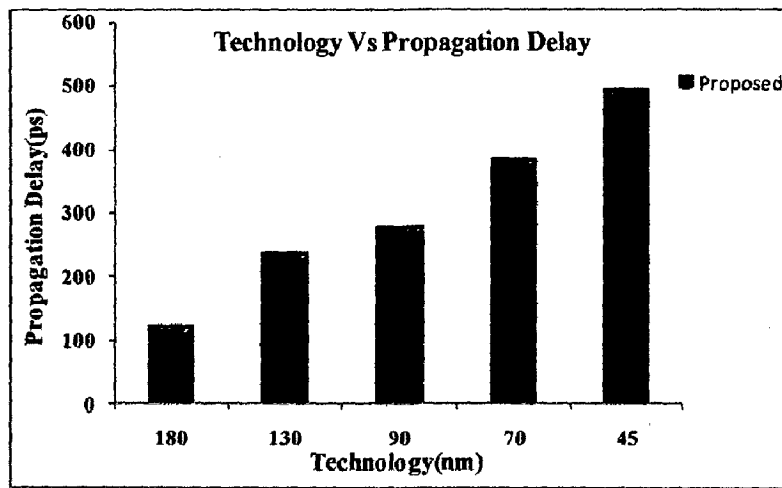


Figure 5.5: Total propagation of system with technology at 1GHz.

CHAPTER 6

CONCLUSIONS AND FUTURE SCOPE

6.1 Conclusions

In deep submicron technology, as feature size goes on shrinking then the two factors i.e. decreasing in interconnect pitch and increasing in interconnects length (i.e. global interconnect) becomes crucial. Due to presence of these effects certain factors become dominant i.e. increasing in power dissipation, crosstalk, total propagation delay and chip area.

To overcome from these effects, a new *RC* model is introduced that is called proposed encoder which is based on bus-invert method. The proposed model is having lesser number of transistors which diminish the chip area. As the proposed model has reduced the overall size of the circuit by logical simplification, there is no trade off parameter for the reduction in power and area. The results show a reduction in numbers of components, power dissipation and propagation delay by 57%, 68.7% and 56.7% respectively compared to previously available capacitive modeled interconnects at 1 MHz frequency and there is reduction in crosstalk also. The reduction in crosstalk for Type-4 and Type-3 coupling is 100% and 76.8% respectively. It also reduces some cases of Type-2 coupling. This proposed method only considers Type-3 and Type-4 couplings mainly because of their dominant nature in *RC* coupled interconnect. The efficiency of proposed model is improved by including the concept of logical effort. On introducing the logical effort, the circuit is capable to operate in 1 GHz frequency. The disadvantage of using logical effort is to increase the power dissipation of the circuit but at the same time it increases the current capability of the circuit. Due to which the propagation delay of the circuit reduces.

There are two limitation of the proposed model. Firstly, it works upto 1 GHz frequency because on increasing the frequency, the time period get reduces. Due to this it does not match the set-up and hold time of latch and signals pass through transistor get distorted. Secondly, it doesn't include the inductive coupling effect which becomes dominant at high frequency.

6.2 Future Scope

This work can be continued by implementation of a new *RLC* model circuit. With ever-growing length of interconnect and on chip clock frequency, the effects of interconnects cannot be restricted to *RC* models. The importance of on-chip inductance is continuously increasing with faster rise times, wider wires, and introduction of new materials for low resistance interconnects. This increases the importance of inductive effects in interconnects,

where the traditional lumped and distributed RC models of interconnects are no longer accurate as they result in substantial errors in predicting delay and crosstalk. Therefore, it has been important to include the impact of self-inductance during interconnect delay prediction.

References

- [1] J. Cong, L. He, K. Y. Khoo, C. K. Koh, and D. Z. Pan, "Interconnect design for deep submicron ICs," in Proc. Int. Conf. Computer-Aided Design, pp. 478-485, Nov. 1997.
- [2] *International Technology Roadmap for Semiconductors*, 2007.
- [3] Min Tang, and Jun-Fa Mao Animes, "Optimization of Global Interconnects in High Performance VLSI Circuits", IEEE Proceedings of the 19th International Conference on VLSI Design, 2006.
- [4] B. Victor, and K. Keutzer, "Bus encoding to prevent crosstalk delay," in Proc. Int. Conf. on Computer-Aided Design, pp. 57-63, 2001.
- [5] L. Benini, G. D. Micheli, E. Macii, D. Sciuto, and C. Silvano, "Asymptotic zero-transition activity encoding for address busses in low- power microprocessor-based system," 7th Great Lakes Symp. on VLSI, Urbana, IL, USA, pp. 77-82, March 1997.
- [6] Ashok Vittal, and Malgorzata Marek-Sadowska, "Crosstalk Reduction for VLSI", IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, vol. 16, no. 3, pp. 290-198, March 1997.
- [7] M. Ghoneima, Y. I. Ismail, M. M. Khellah, J. W. Tschanz, and V. De, "Formal derivation of optimal active shielding for low-power on-chip buses," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, no. 5, pp. 821-836, May 2006.
- [8] Rajeevan Chandel, S. Sarkar, and R. P. Agarwal, "Repeater insertion in global interconnects in VLSI circuits," *Microelectronics International*, pp. 43-50, 2005.
- [9] L. Macchiarulo, E. Macii, and M. Poncino, "Wire placement for crosstalk energy minimization in address buses", Proc. Design, Automation and test in Europe Conf., pp. 158-162, March 2002.
- [10] L. Avinash, M. K. Krishna, and M. B. Srinivas, "A novel encoding scheme for delay

- and energy minimization in VLSI interconnects with built-in error detection”, IEEE Computer Society Annual Symposium on VLSI, pp. 128-133, 2008.
- [11] M. R. Stan, and W. P. Burleson, “Bus-Invert coding for low-power I/O,” IEEE Trans. on Very Large Scale Integration System, vol. 3, no. 1, pp. 49-58, March 1995.
- [12] Y. Shin, S. I. Chae, and K. Choi, “Reduction of bus transitions with partial bus-invert coding,” Electronics Letters, vol. 34, no. 7, pp. 642-643, April 1998.
- [13] Y. Shin, S. I. Chae, and K. Choi, “Partial bus-invert coding for power optimization of application-specific systems,” IEEE Trans. on Very Large Scale Integration Systems, vol. 9, no. 2, pp. 377-383, April 2001.
- [14] Y. Zhang, J. Lach, K. Skadron, and M. R. Stan, “Odd/even bus invert with two-phase transfer for buses with coupling”, In: Proceedings of the 2002 International Symposium on Low Power Electronics and Design, pp. 80-83, 2002.
- [15] K. W. Kim, K. H. Baek, N. Shanbhag, C. L. Liu, and S. M. Kang, “Coupling-drive signal encoding scheme for low-power interface design”, Computer Aided Design, ICCAD, pp. 318-321, 2000.
- [16] C. G. Lyuh, and T. Kim, “Low power bus encoding with crosstalk delay elimination”, ASIC/SOC Conf., IEEE Int. Conf., pp. 389-393, 2002.
- [17] Z. Khan, T. Arslan, and A. T. Erdogan, “Low power system on chip bus encoding scheme with crosstalk noise reduction capability”, IEE Proceedings Computers and Digital Techniques, vol. 153, no. 2, pp. 101-108, 2006.
- [18] Chih-Peng Fan, and Chia-Hao Fang, “Efficient RC low-power bus encoding methods for crosstalk reduction,” Integration VLSI Journal, Elsevier, vol. 44, no. 1, pp. 75-86, Jan. 2011.
- [19] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic, *Digital Integrated Circuits*, 2nd Edition, Prentice Hall of India Pvt Ltd, New Delhi, 2006.
- [20] Ivan Sutherland, Bob Sproull, and David Harris, *Logical Effort: Designing Fast CMOS Circuits*, Morgan Kaufmann Publishers, Inc. San Francisco, California.

Publications

- [1] Deepika Agarwal, G. Nagendra Babu, B. K. Kaushik, and S. K. Manhas, "Reduction of crosstalk in *RC* modeled interconnects with low power encoder", *International Conference on Emerging Trends in Networks and Computer Communications-2011(ETNCC-2011), Udaipur (in press)*.
- [2] Deepika Agarwal, G. Nagendra Babu, B. K. Kaushik, and S. K. Manhas, "Highly simplified encoder for reduced crosstalk, power and area for multiline coupled VLSI interconnects", *Integration VLSI Journal, Elsevier (Communicated)*.

Appendix A

VHDL Coding of RC Model

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity rc_model is
port( y : in std_logic_vector(3 downto 0);

      x : in std_logic_vector(3 downto 0);
      invp :in std_logic ;
      z: in std_logic;
      D_enc : inout std_logic_vector(3 downto 0);
      inv_pre: inout std_logic;
      D_dec : out std_logic_vector(3 downto 0));

end rc_model;

architecture Behavioral of rc_model is

signal sa, sb, sc, sd, se, sf, sg, sh, si, sj : std_logic;
signal N4_count, N3_count : std_logic;
signal s01,s02,s03,s04,s05,s06,sm1,sm2 : std_logic;
signal s07,s08,s09,s10,s11,s12,sm3,sm4 : std_logic;

begin

sa <= ((not(invpre)) and z);
sb <= ((not(x(3))) and y(3));
sc <= ((not(x(2))) and y(2));
sd <= ((not(x(1))) and y(1));
se <= ((not(x(0))) and y(0));
```



```

sf <= ((not z) and invp);
sg <= ((not(y(3))) and x(3));
sh <= ((not(y(2))) and x(2));
si <= ((not(y(1))) and x(1));
sj <= ((not(y(0))) and x(0));
s01 <= (not (not (sa) and sb and sh));
s02 <= (not (not (sb) and sc and si));
s03 <= (not (not (sc) and sd and sj));
s04 <= (not (not (sf) and sg and sh));
s05 <= (not (not (sg) and sh and sd));
s06 <= (not (not (sh) and si and sg));
sm1 <= (not (s01 and s02 and s03));
sm2 <= (not (s04 and s05 and s06));
N4_count <= (not (sm1 and sm2));
s07 <= (not (sa and sg and not (sh)));
s08 <= (not (sb and sh and not (si)));
s09 <= (not (sc and si and not (sj)));
s10 <= (not (sf and sb and not (sc)));
s11 <= (not (sg and sc and not (sd)));
s12 <= (not (sh and sd and not (se)));
sm3 <= (not (s07 and s08 and s09));
sm4 <= (not (s10 and s11 and s12));
N3_count <= (not (sm3 and sm4));
inv_pre <= N4_count or N3_count;
D_enc(3) <= (y(3) xor inv_pre);
D_enc(2) <= (y(2) xor inv_pre);
D_enc(1) <= (y(1) xor inv_pre);
D_enc(0) <= (y(0) xor inv_pre);
D_dec(3) <= (inv_pre xor D_enc(3));
D_dec(2) <= (inv_pre xor D_enc(2));
D_dec(1) <= (inv_pre xor D_enc(1));
D_dec(0) <= (inv_pre xor D_enc(0));

end Behavioral

```

Appendix B

H-Spice Coding of RC Model without Sizing

```
*complete ckt_rc
.include '45nm_NMOS_bulk17963.md'
.include '45nm_PMOS_bulk29891.md'
.subckt nand a11 b11 out17
m1 out17 b11 1 gnd NMOS l=45n w=135n
m2 1 a11 gnd gnd NMOS l=45n w=135n
m3 out17 a11 vdd vdd PMOS l=45n w=135n
m4 out17 b11 vdd vdd PMOS l=45n w=135n
.ends
.subckt inverter i o
m5 o i gnd gnd NMOS l=45n w=67.5n
m6 o i vdd vdd PMOS l=45n w=135n
.ends
.subckt xor a b f
x1 b b_bar inverter
m7 f a b vdd PMOS l=45n w=135n
m8 f a b_bar gnd NMOS l=45n w=67.5n
m9 a b f vdd PMOS l=45n w=135n
m10 f b_bar a gnd NMOS l=45n w=67.5n
.ends
.subckt nor1 x y out4
m11 out4 x gnd gnd NMOS l=45n w=67.5n
m12 out4 y gnd gnd NMOS l=45n w=67.5n
m13 out4 y 6 vdd PMOS l=45n w=270n
m14 6 x vdd vdd PMOS l=45n w=270n
.ends
.subckt nand2 p1 q1 r1 out11
```

```

m15 out11 p1 21 gnd NMOS l=45n w=202.5n
m16 21 q1 31 gnd NMOS l=45n w=202.5n
- - - m17 31 r1 gnd gnd NMOS l=45n w=202.5n - - -
m18 out11 p1 vdd vdd PMOS l=45n w=135n
m19 out11 q1 vdd vdd PMOS l=45n w=135n
m20 out11 r1 vdd vdd PMOS l=45n w=135n
.ends
.subckt and t0 t1 t2 t3 out3
x2 t0 t1 t2 t3 11 nand1
x3 11 out3 inverter
.ends
.subckt nor n1 n2 n3 n4 out6
m21 out6 n1 gnd gnd NMOS l=45n w=67.5n
m22 out6 n2 gnd gnd NMOS l=45n w=67.5n
m23 out6 n3 gnd gnd NMOS l=45n w=67.5n
m24 out6 n4 gnd gnd NMOS l=45n w=67.5n
m25 out6 n4 25 vdd PMOS l=45n w=540n
m26 25 n3 26 vdd PMOS l=45n w=540n
m27 26 n2 27 vdd PMOS l=45n w=540n
m28 27 n1 vdd vdd PMOS l=45n w=540n
.ends
.subckt or o1 o2 o3 o4 out5
x4 o1 o2 o3 o4 28 nor
x5 28 out5 inverter
.ends
.subckt or1 ot1 ot2 out7
x6 ot1 ot2 35 nor1
x7 35 out7 inverter
.ends
.subckt and1 a1 a2 output1
x9 a1 a2 78 nand
x10 78 output1 inverter
.ends

```

```

.subckt tg 7 8
m29 8 clk 7 gnd NMOS l=45n w=67.5n
m30 8 clk 7 vdd PMOS l=45n w=135n
vclock clk gnd pulse(0 1 0n 4p 4p 0.5n 1n)
m33 clk clk gnd gnd NMOS l=45n w=67.5n
m34 clk clk vdd vdd PMOS l=45n w=135n
m35 out 8 gnd gnd NMOS l=45n w=67.5n
m36 out 8 vdd vdd PMOS l=45n w=135n
*feedback transmission gate
m39 9 clk out gnd NMOS l=45n w=67.5n
m40 9 clk out vdd PMOS l=45n w=135n
.ends tg

```

*******Proposed Model*******

```

x11 u0 bar_u0 inverter
x12 u1 bar_u1 inverter
x13 u2 bar_u2 inverter
x14 u3 bar_u3 inverter
x15 invp bar_invp inverter
x16 y0 bar_y0 inverter
x17 y1 bar_y1 inverter
x18 y2 bar_y2 inverter
x19 y3 bar_y3 inverter
x20 z bar_z inverter
x21 bar_u0 y0 s4 and1
x22 bar_u1 y1 s3 and1
x23 bar_u2 y2 s2 and1
x24 bar_u3 y3 s1 and1
x25 bar_invp z s0 and1
x26 u0 bar_y0 s9 and1
x27 u1 bar_y1 s8 and1
x28 u2 bar_y2 s7 and1
x29 u3 bar_y3 s6 and1
x30 invp bar_z s5 and1

```

```

x31 s0 bar_s0 inverter
x32 s1 bar_s1 inverter
x33 s2 bar_s2 inverter
x34 s3 bar_s3 inverter
x35 s4 bar_s4 inverter
x36 s5 bar_s5 inverter
x37 s6 bar_s6 inverter
x38 s7 bar_s7 inverter
x39 s8 bar_s8 inverter
x40 s9 bar_s9 inverter
x41 bar_s0 s1 s7 s01 nand2
x42 bar_s1 s2 s8 s02 nand2
x43 bar_s2 s3 s9 s03 nand2
x44 bar_s5 s6 s2 s04 nand2
x45 bar_s6 s7 s3 s05 nand2
x47 bar_s7 s8 s4 s06 nand2
x48 s01 s02 s03 sm1 nand2
x49 s04 s05 s06 sm2 nand2
x50 s0 s6 bar_s7 s07 nand2
x51 s1 s7 bar_s8 s08 nand2
x52 s2 s8 bar_s9 s09 nand2
x53 s5 s1 bar_s2 s10 nand2
x54 s6 s2 bar_s3 s11 nand2
x55 s7 s3 bar_s4 s12 nand2
x56 s10 s11 s12 sm4 nand2
x57 s07 s08 s09 sm3 nand2
x81 sm1 sm2 smm1 nand
x82 sm3 sm4 smm2 nand
x83 sm1 sm2 sd1 or1
*****Multiplexer*****
x63 sd1 y0 e0 xor
x64 sd1 y1 e1 xor
x65 sd1 y2 e2 xor

```

x66 sd1 y3 e3 xor

*******Latch*******

x71 e0 u0 tg

x72 e1 u1 tg

x73 e2 u2 tg

x74 e3 u3 tg

x75 sd1 invp tg

*******Interconnects*******

C1 e0 0 82.966f

R1 e0 ou0 39.722

C12 e0 e1 89.245f

C21 ou0 ou1 89.245f

C2 e1 0 82.966f

R2 e1 ou1 39.722

C22 ou1 0 82.966f

C23 e1 e2 89.245f

C32 ou1 ou2 89.245f

C3 e2 0 82.966f

R3 e2 ou2 39.722

C33 ou2 0 82.966f

C34 e2 e3 89.245f

C43 ou2 ou3 89.245f

C4 e3 0 82.966f

R4 e3 ou3 39.722

C44 ou3 0 82.966f

*******Decoder*******

x76 ou0 sd1 d0 xor

x77 ou1 sd1 d1 xor

x78 ou2 sd1 d2 xor

x79 ou3 sd1 d3 xor

vdc vdd gnd 1

vin1 y0 gnd pulse(1 0 0n 4p 4p 0.4n 1n)

vin2 y1 gnd pulse(0 1 0n 4p 4p 0.4n 1n)

vin3 y2 gnd pulse(1 0 0n 4p 4p 0.4n 1n)

vin4 y3 gnd pulse(0 1 0n 4p 4p 0.4n 1n)

vin5 z gnd 0

.TRAN 0.01n 1n

.plot TRAN v(u0) v(u1) v(u2) v(u3) v(invp) v(y0) v(y1) v(y2) v(y3) v(y4)

.meas tran avg_pow1 AVG p(vdc) FROM=0n TO=1n

.measure tran d1 trig v(y0) val=0.5 fall=1 targ v(d0) val=0.5 fall=1

.measure tran d2 trig v(y1) val=0.5 rise=1 targ v(d1) val=0.5 rise=1

.measure tran d3 trig v(y2) val=0.5 fall=1 targ v(d2) val=0.5 fall=1

.measure tran d4 trig v(y3) val=0.5 rise=1 targ v(d3) val=0.5 rise=1

.option list post node

.end

Appendix-C

H-Spice Coding of RC Model with Sizing

```
*complete ckt_rc_logical_effort
.include '45nm_NMOS_bulk17963.md'
.include '45nm_PMOS_bulk29891.md'
.subckt nand3 a11 b11 out17
m65 out17 b11 1 gnd NMOS l=45n w=135n
m66 1 a11 gnd gnd NMOS l=45n w=135n
m67 out17 a11 vdd vdd PMOS l=45n w=135n
m68 out17 b11 vdd vdd PMOS l=45n w=135n
.ends nand3
.subckt nand a11 b11 out17
m1 out17 b11 1 gnd NMOS l=45n w=1976.4n
m2 1 a11 gnd gnd NMOS l=45n w=1976.4n
m3 out17 a11 vdd vdd PMOS l=45n w=1976.4n
m4 out17 b11 vdd vdd PMOS l=45n w=1976.4n
.ends nand
.subckt inverter i o
m5 o i gnd gnd NMOS l=45n w=67.5n
m6 o i vdd vdd PMOS l=45n w=135n
.ends inverter
.subckt inverter7 i o
m107 o i gnd gnd NMOS l=45n w=243.945n
m108 o i vdd vdd PMOS l=45n w=487.35n
.ends inverter7
.subckt inverter8 i o
m67 o i gnd gnd NMOS l=45n w=2232.15n
m68 o i vdd vdd PMOS l=45n w=4464.315n
.ends inverter8
```



```

.subckt inverter9 i o
m69 o i gnd gnd NMOS l=45n w=5000n
m70 o i vdd vdd PMOS l=45n w=10000n
.ends inverter9

.subckt xor a b f
x1 b b_bar inverter9
m7 f a b vdd PMOS l=45n w=10000n
m8 f a b_bar gnd NMOS l=45n w=5000n
m9 a b f vdd PMOS l=45n w=10000n
m10 f b_bar a gnd NMOS l=45n w=5000n
.ends xor

.subckt nor1 x y out4
m11 out4 x gnd gnd NMOS l=45n w=1660
m12 out4 y gnd gnd NMOS l=45n w=1660
m13 out4 y 6 vdd PMOS l=45n w=13284n
m14 6 x vdd vdd PMOS l=45n w=13284n
.ends nor1

.subckt nand2 p1 q1 r1 out11
m15 out11 p1 21 gnd NMOS l=45n w=2025.63n
m16 21 q1 31 gnd NMOS l=45n w=2025.63n
m17 31 r1 gnd gnd NMOS l=45n w=2025.63n
m18 out11 p1 vdd vdd PMOS l=45n w=1470.42n
m19 out11 q1 vdd vdd PMOS l=45n w=1470.42n
m20 out11 r1 vdd vdd PMOS l=45n w=1470.42n
.ends nand2

.subckt nand1 p1 q1 r1 out11
m95 out11 p1 21 gnd NMOS l=45n w=1641.06n
m96 21 q1 31 gnd NMOS l=45n w=1641.06n
m97 31 r1 gnd gnd NMOS l=45n w=1641.06n
m98 out11 p1 vdd vdd PMOS l=45n w=1094.04n
m99 out11 q1 vdd vdd PMOS l=45n w=1094.04n
m100 out11 r1 vdd vdd PMOS l=45n w=1094.04n
.ends nand1

```

```

.subckt and t0 t1 t2 t3 out3
x2 t0 t1 t2 t3 11 nand1
x3 11 out3 inverter
.ends and

.subckt nor n1 n2 n3 n4 out6
m21 out6 n1 gnd gnd NMOS l=45n w=1660.5n
m22 out6 n2 gnd gnd NMOS l=45n w=1660.5n
m23 out6 n3 gnd gnd NMOS l=45n w=1660.5n
m24 out6 n4 gnd gnd NMOS l=45n w=1660.5n
m25 out6 n4 25 vdd PMOS l=45n w=13284n
m26 25 n3 26 vdd PMOS l=45n w=13284n
m27 26 n2 27 vdd PMOS l=45n w=13284n
m28 27 n1 vdd vdd PMOS l=45n w=13284n
.ends nor

.subckt or o1 o2 o3 o4 out5
x4 o1 o2 o3 o4 28 nor
x5 28 out5 inverter8
.ends or

.subckt or1 ot1 ot2 out7
x6 ot1 ot2 35 nor1
x7 35 out7 inverter8
.ends or1

.subckt and1 a1 a2 output1
x9 a1 a2 78 nand3
x10 78 output1 inverter
.ends

.subckt tg 7 8
m29 8 clk 7 gnd NMOS l=45n w=12054n
m30 8 clk 7 vdd PMOS l=45n w=24108n
vclock clk gnd pulse(0 1 0n 4p 4p 0.5n 1n)
m33 clk clk gnd gnd NMOS l=45n w=12054n
m34 clk clk vdd vdd PMOS l=45n w=24108n
m35 out 8 gnd gnd NMOS l=45n w=12054n

```

m36 out 8 vdd vdd PMOS l=45n w=24108n

*feedback transmission gate

~~m39 8 clk out gnd NMOS l=45n w=12054n~~

m40 8 clk out vdd PMOS l=45n w=12054n

.ends tg

*******Proposed RC Model*******

x11 u0 bar_u0 inverter

x12 u1 bar_u1 inverter

x13 u2 bar_u2 inverter

x14 u3 bar_u3 inverter

x15 invp bar_invp inverter

x16 y0 bar_y0 inverter

x17 y1 bar_y1 inverter

x18 y2 bar_y2 inverter

x19 y3 bar_y3 inverter

x20 z bar_z inverter

x21 bar_u0 y0 s4 and1

x22 bar_u1 y1 s3 and1

x23 bar_u2 y2 s2 and1

x24 bar_u3 y3 s1 and1

x25 bar_invp z s0 and1

x26 u0 bar_y0 s9 and1

x27 u1 bar_y1 s8 and1

x28 u2 bar_y2 s7 and1

x29 u3 bar_y3 s6 and1

x30 invp bar_z s5 and1

x31 s0 bar_s0 inverter7

x32 s1 bar_s1 inverter7

x33 s2 bar_s2 inverter7

x34 s3 bar_s3 inverter7

x35 s4 bar_s4 inverter7

x36 s5 bar_s5 inverter7

x37 s6 bar_s6 inverter7

x38 s7 bar_s7 inverter7
x39 s8 bar_s8 inverter7
x40 s9 bar_s9 inverter7
x41 bar_s0 s1 s7 s01 nand1
x42 bar_s1 s2 s8 s02 nand1
x43 bar_s2 s3 s9 s03 nand1
x44 bar_s5 s6 s2 s04 nand1
x45 bar_s6 s7 s3 s05 nand1
x47 bar_s7 s8 s4 s06 nand1
x48 s01 s02 s03 sm1 nand2
x49 s04 s05 s06 sm2 nand2
x50 s0 s6 bar_s7 s07 nand1
x51 s1 s7 bar_s8 s08 nand1
x52 s2 s8 bar_s9 s09 nand1
x53 s5 s1 bar_s2 s10 nand1
x54 s6 s2 bar_s3 s11 nand1
x55 s7 s3 bar_s4 s12 nand1
x56 s10 s11 s12 sm4 nand2
x57 s07 s08 s09 sm3 nand2
x81 sm1 sm2 smm1 nand
x82 sm3 sm4 smm2 nand
x83 sm1 sm2 sd1 or1

*****Multiplexer*****

x63 sd1 y0 e0 xor
x64 sd1 y1 e1 xor
x65 sd1 y2 e2 xor
x66 sd1 y3 e3 xor

*****Latch*****

x71 e0 u0 tg
x72 e1 u1 tg
x73 e2 u2 tg
x74 e3 u3 tg
x75 sd1 invp tg

*****Interconnects*****

C1 e0 0 82.966f
R1 e0 ou0 39.722
C11 ou0 0 82.966f
C12 e0 e1 89.245f
C21 ou0 ou1 89.245f

C2 e1 0 82.966f
R2 e1 ou1 39.722
C22 ou1 0 82.966f
C23 e1 e2 89.245f
C32 ou1 ou2 89.245f

C3 e2 0 82.966f
R3 e2 ou2 39.722
C33 ou2 0 82.966f
C34 e2 e3 89.245f
C43 ou2 ou3 89.245f

C4 e3 0 82.966f
R4 e3 ou3 39.722
C44 ou3 0 82.966f

*****Decoder*****

x76 ou0 sd1 d0 xor
x77 ou1 sd1 d1 xor
x78 ou2 sd1 d2 xor
x79 ou3 sd1 d3 xor

vdc vdd gnd 1
vin1 y0 gnd pulse (1 0 0n 4p 4p 0.4n 1n)
vin2 y1 gnd pulse (0 1 0n 4p 4p 0.4n 1n)
vin3 y2 gnd pulse (1 0 0n 4p 4p 0.4n 1n)
vin4 y3 gnd pulse (0 1 0n 4p 4p 0.4n 1n)

```
vin5 z gnd 0
.TRAN 0.01n 1n
.plot TRAN v(u0) v(u1) v(u2) v(u3) v(inv) v(y0) v(y1) v(y2) v(y3) v(d0) v(d1) v(d2) v(d3)
.meas tran avg_pow1 AVG p(vdc) FROM=0n To 1n
.measure tran d1 trig v(y0) val=0.5fall=1 targ v(d0) val=0.5 fall=1
.measure tran d2 trig v(y1) val=0.5rise=1 targ v(d1) val=0.5 rise=1
.measure tran d3 trig v(y2) val=0.5 fall=1 targ v(d2) val=0.5 fall=1
.measure tran d4 trig v(y3) val=0.5 rise=1 targ v(d3) val=0.5 rise=1
.end
```