

**EARLY ESTIMATION OF THE IMPACT OF DELAY  
DUE TO COUPLING CAPACITANCE IN VLSI  
CIRCUITS**

**A THESIS  
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF MINNESOTA  
BY**

**Vignesh Shriram**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
MASTER OF SCIENCE**

**Prof. Sachin Sapatnekar, Advisor**

**May, 2019**

© Vignesh Shriram 2019  
ALL RIGHTS RESERVED

# Acknowledgements

Firstly, I would like to wholeheartedly thank my adviser, Prof. Sachin Sapatnekar for his unwavering support and constant encouragement. I consider myself extremely fortunate for having him as my mentor and advisor. Without his expert guidance and feedback, none of this would have been possible.

I would like to thank the final examination committee, Prof. Kia Bazargan and Prof. Sara Algeri, for reviewing my thesis, and for their valuable feedback.

I would also like to thank the VLSI CAD group, Vidya Chhabria and Kishor Kunal, for all their support and guidance.

I am grateful to the mentors of my internship at Synopsys, Alireza Kasnavi and Jacob Thomas, for their guidance. Their tutelage has inspired me to pursue this research.

I would like to thank the mentor of my internship at NIAS(IISc), Dr. Nithin Nagaraj for helping me overcome my fear of programming, and Dr. Purushothaman A. for helping me overcome my fear of circuits. This thesis is my way of thanking them for all their invaluable lessons.

Finally, I would like to thank my friends and family for their unconditional love and support.

# Dedication

Dedicated to my parents and to my sister

## Abstract

Coupling capacitance is becoming increasingly problematic at the more advanced technology nodes and affects the timing and sign-off time line of integrated circuits (ICs). As the coupling capacitance information is only available after the detailed routing phase, it can be a difficult task to make any major changes post detailed routing towards fixing issues caused by coupling effects that were unaccounted for. The goal of the project is to come up with an estimate of coupling capacitance for a given net before the detailed routing phase with the help of congestion maps. This information can be fed back to the detailed router which can help avoid routes that are susceptible to heavy coupling effects.

The first part of this thesis (chapters 1,2) explains why beforehand knowledge of a net's coupling capacitance is crucial for a timely tape-out. This thesis revisits the Elmore delay model and extends the analysis to coupled RC structures. The notion of considering the coupling capacitance as a random variable is described to model the uncertainties that are introduced into the delay analysis which is performed ahead in time.

The second part of this thesis (chapters 3, 4) illustrates how congestion analysis can provide valuable information about the severity of coupling effects. A method for the expedited extraction of estimated parasitics using congestion maps and global router solutions is presented. A modification to existing driving-point analysis techniques is suggested to accommodate coupled RC structures with probabilistic coupling capacitance.

The last part of this thesis (chapter 5) compares the delay metrics obtained from an open-source timing analyzer with the delay metrics obtained through methods described in this thesis for a given net.

# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Dedication</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>List of Tables</b>	<b>vi</b>
<b>List of Figures</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Scaling Trends . . . . .	1
1.2 Research Objective and Contributions . . . . .	2
<b>2 Crosstalk Delay Analysis for Coupled RC Structures</b>	<b>4</b>
2.1 Crosstalk Delay Analysis in the Presence of Probabilistic Coupling Capacitance . . . . .	7
2.2 Support Size Reduction for Faster Analysis . . . . .	12
<b>3 Estimating Coupling Capacitance Using Congestion Maps</b>	<b>15</b>
3.1 Estimating Coupling Capacitance from Congestion Maps . . . . .	17
3.2 Constructing Approximate Parasitics from Congestion Maps and Global Router Solutions . . . . .	19
<b>4 Estimating Driving-Point Admittance</b>	<b>22</b>

<b>5</b>	<b>Validation of Results with OpenSTA</b>	<b>28</b>
5.1	Incorporating Crosstalk Delay . . . . .	28
5.2	Validation of Results . . . . .	29
<b>6</b>	<b>Conclusion</b>	<b>33</b>
	<b>References</b>	<b>34</b>
	<b>Appendix A. BoxRouter and DEF File Formats</b>	<b>38</b>
A.1	BoxRouter File Format . . . . .	39
A.1.1	Input File Format . . . . .	39
A.1.2	Output File Format . . . . .	41
A.2	DEF File Format . . . . .	42
	<b>Appendix B. Timing Reports</b>	<b>44</b>
B.1	Timing Report 1 . . . . .	45
B.2	Timing Report 2 . . . . .	48
B.3	Timing Report 3 . . . . .	50
B.4	Timing Report 4 . . . . .	53

# List of Tables

1.1	Effect of scaling on technology parameters. . . . .	2
2.1	Switching factor values for different aggressor/victim switching configurations. . . . .	8
5.1	Sample set of nets chosen for analysis. . . . .	30
5.2	A comparison of delay values obtained through OpenSTA and expedited parasitic extraction. . . . .	32



# List of Figures

2.1	Example of an RC tree. . . . .	5
2.2	Example of an RC line. . . . .	6
2.3	Example of coupled RC lines. . . . .	9
2.4	RC line $b$ with coupling capacitance converted to equivalent ground capacitance. . . . .	10
2.5	Probability Mass Function of the switching factor $S$ . . . . .	11
2.6	Delay distributions for different switching probabilities. . . . .	11
2.7	Delay PMF for the range $[a, b]$ . . . . .	13
2.8	Reduced delay PMF for the range $[a, b]$ . . . . .	13
2.9	Demonstration of support reduction for a given distribution. . . . .	13
3.1	Example of a congestion map (horizontal layer). . . . .	16
3.2	Example of a congestion map (vertical layer). . . . .	16
3.3	Example of a global net in a grid. . . . .	17
3.4	Focusing on the highlighted grid cell in figure 3.3. . . . .	17
3.5	Example where horizontal layer congestion is lesser than vertical layer congestion. . . . .	18
3.6	Example where horizontal layer congestion is greater than vertical layer congestion. . . . .	19
3.7	Physical representation of a net segment. . . . .	20
3.8	Equivalent RC $\pi$ segment. . . . .	20
3.9	Flow chart for extracting estimated parasitics. . . . .	20
4.1	Illustration to identify different components of interconnect delay. . . . .	23

4.2	A general RC tree. . . . .	24
4.3	A reduced RC $\pi$ model. . . . .	24
4.4	Propagation of moments past a lumped capacitor. . . . .	25
4.5	Propagation of moments past a lumped resistor. . . . .	26
4.6	Propagation of moments past a branching point. . . . .	26
5.1	Illustration of crosstalk delay on a net. . . . .	29
5.2	Comparison of a net's total coupling capacitance. . . . .	31
5.3	Comparison of a worst-case maximum net delays. . . . .	32
A.1	Routing grid structure from BoxRouter input. . . . .	39
A.2	Net's route obtained from BoxRouter output. . . . .	41
A.3	Net's route obtained from DEF file. . . . .	42

# Chapter 1

## Introduction

### 1.1 Scaling Trends

Continual advancements in the complementary metal-oxide-semiconductor (CMOS) technology over the past few decades have resulted in an exponential increase in the performance and affordability of integrated circuits (ICs). The concept of CMOS scaling refers to the systematic shrinkage of the MOS devices, resulting in devices that are smaller, faster, and more power-efficient. Moore's Law [16] has been successfully predicting the technology scaling trends for past four decades. Modern processors are made of tens of billions of transistors, indicating that the momentum with which technology is scaling is not going to dampen anytime soon.

Table 1.1 shows the benefits of scaling trends in ICs (scaling factor  $\alpha > 1$  is applied for every new technology node). For the same area, IC designers are able to cram more devices that are faster and more power efficient.

While scaling trends have driven up the performance of transistors, they have had a contrasting effect on the performance of interconnects. This phenomenon, known as interconnect reverse scaling [2], refers to the fact that interconnects at more advanced technology nodes experience an increase in delay due to decreasing spaces between

Parameter	Scaling factor	
$L, W, T_{ox},$ and $V_{dd}$	$1/\alpha$	} Device parameters
Doping concentration $N$	$\alpha$	
Device area	$1/\alpha^2$	
Gates/area	$\alpha^2$	} Parameters with desirable scaling effects
Power consumption	$\alpha^2$	
Gate delay	$1/\alpha$	
Interconnect resistance	$\alpha$	} Parameters with undesirable scaling effects
Interconnect voltage drop	$\alpha$	
Interconnect delay	1	

Table 1.1: Effect of scaling on technology parameters.

adjacent wires and increasing wire aspect ratio. The consequence of interconnect reverse scaling is that interconnects suffer from an increase in coupling capacitance as the technology node advances further into the deep sub-micron (DMS) region. High coupling capacitance results in degradation of the signal integrity (crosstalk) and increase in interconnect delay. Since coupling capacitance has become an increasingly dominating component of the total interconnect capacitance as a consequence of scaling trends, it is crucial to predict and analyze the effect of coupling capacitance at advanced technology nodes beforehand for a successful and timely tape-out.

## 1.2 Research Objective and Contributions

Information needed for accurate analysis of coupling capacitance and its impact on circuit performance is obtained only after the parasitic extraction of the fully routed design, performed at sign-off accuracy. At this stage, if the net delays do not conform to the timing restrictions imposed on them, it can be a tedious task to make any large-scale changes on the design, thereby impacting the time to tape-out. After the initial placement of the standard cells, a quick congestion analysis can give valuable information about the severity of the effect of coupling capacitance on the nets. This information can be fed to the place and route (PnR) tool, that would facilitate it to make a slight change to the net's route such that the overall coupling capacitance on that net is

reduced. The goal of this thesis is to make use of the information available from congestion analysis and physical information of nets after the global routing stage (where we estimate how the routing might take place with the current floorplan and placement of the cells) to obtain an estimate of net delays that also take coupling effects into account.

The major contributions of this thesis are:

- We have proposed a method to obtain a quick estimate of interconnect parasitics from congestion maps. This method can be considered as a speedy procedure to obtain a rough estimate of net parasitics that are not as accurate as the parasitics obtained from commercial parasitic extraction tools, but accurate enough to weed out the nets that are vulnerable to coupling effects in a given setting.
- We offer an alternate visualization of a net's timing window. Instead of considering a net's timing window as just a collection of extreme delay values (minimum and maximum possible interconnect delays), we introduce the notion of envisioning a net's timing window as a distribution of multiple possible delay values with their respective probabilities. By doing so, we minimize the pessimism introduced by conventional crosstalk analysis methodologies.
- We provide a comparison of the net's delay metrics obtained from the methods explained above with the delay values obtained from an open-source static timing analysis tool, OpenSTA [4]. We also propose a simple crosstalk analysis framework for the tool.

## Chapter 2

# Crosstalk Delay Analysis for Coupled RC Structures

Interconnects are modelled with a varying degree of accuracy at different levels of abstraction, ranging from quick and simple models with limited accuracy to slow and complex models that require usage of 3D field solvers. The work in [13] explores commonly used techniques for parasitic extraction.

One of the most commonly encountered structures when analyzing interconnect circuit models is an RC tree. Figure 2.1 shows an example of an RC tree. The input node of the RC network is  $n_0$ , and it is driven by the voltage source  $V_s$ . Figure 2.2 shows an example of an RC line, a special case of the RC tree. An RC line can be considered as an RC tree without any branches.

The RC tree, by reason of its tree-like structure, has the following properties:

- All capacitances that are connected to the tree nodes must be grounded.
- There may exist a resistor  $R_{ij}$  connecting nodes  $n_i$  and  $n_j$ , such that it does not form a cycle (a resistive loop).
- There exists a unique path from node  $n_i$  to  $n_j$ .

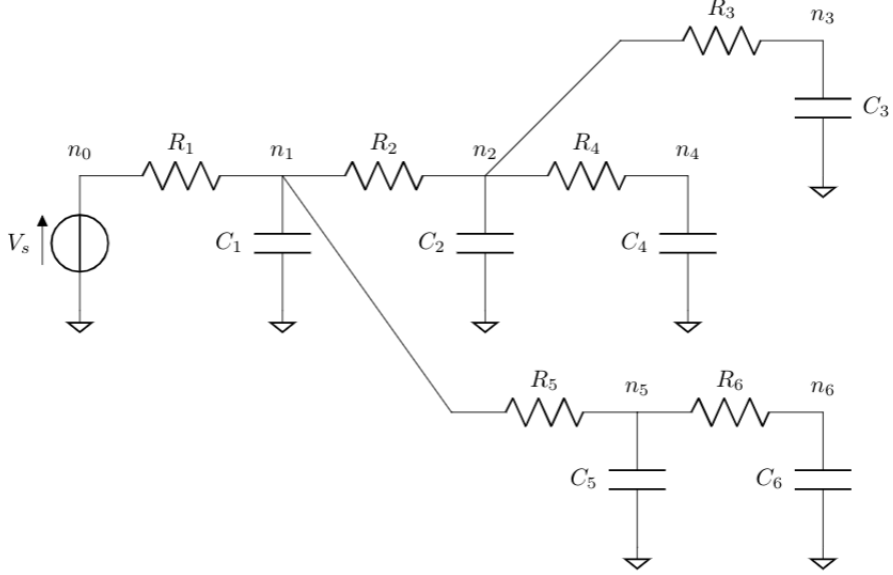


Figure 2.1: Example of an RC tree.

The Elmore delay metric [8] is a first-order approximation of delay through an RC network. It is computationally inexpensive to obtain and is often reasonably accurate. It usually has a strong correlation with the true delay through the RC network. The Elmore delay  $T_{D_i}$  to a node  $i$  is given by:

$$T_{D_i} = \sum_{j=0}^n C_j \sum_{k \in \text{upstream}(j)} R_k \quad (2.1)$$

where the upstream resistance at  $j$  is the sum of all resistances encountered along the path starting from node  $k$  to the root, that are also present along the path from node  $i$  to the root. For example, consider the circuit shown in figure 2.1. The Elmore delay to  $n_5$  in terms of upstream resistances is given by:

$$\begin{aligned} T_D(n_5) = & C_1 R_1 + C_2 R_1 + C_3 R_1 + C_4 R_1 + C_5 (R_1 + R_5) \\ & + C_6 (R_1 + R_5) \end{aligned} \quad (2.2)$$

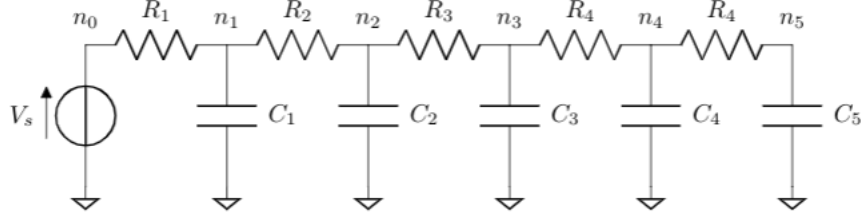


Figure 2.2: Example of an RC line.

Terms of equation (2.1) can be rearranged to obtain:

$$T_{D_i} = \sum_{j \in P_i}^n R_j \sum_{k \in \text{downstream}(j)} C_k \quad (2.3)$$

where node  $j$  is contained in the path from the root to the node  $i$ , and the downstream capacitance at node  $j$  is sum of all capacitances at a node  $k$  such that the path from the root to node  $j$  passes through node  $k$ . For the example in figure 2.1, the Elmore delay to  $n_5$  in terms of downstream capacitance is given by:

$$\begin{aligned} T_D(n_5) &= R_1(C_1 + C_2 + C_3 + C_4 + C_5 + C_6) \\ &\quad + R_5(C_5 + C_6) \end{aligned} \quad (2.4)$$



## 2.1 Crosstalk Delay Analysis in the Presence of Probabilistic Coupling Capacitance

Crosstalk can alter the behavior of circuits in the following ways:

- It can lead to a functionality error through introduction of noise in sensitive nodes.
- It can lead to an unpredictable increase (or decrease) in interconnect delays, causing a timing uncertainty.

Crosstalk analysis through circuit simulation is computationally expensive and infeasible for a large circuit. Since *static* timing analysis techniques take into account only the worst-case delays for timing verification, several performance enhancement strategies have been proposed in [9, 19, 23]. Standard crosstalk analysis methodology is to multiply the coupling capacitance with a “switching factor” ( $S$ ) to obtain an *equivalent* ground capacitance that can be used in delay calculations and noise analysis.

The work done in [10] show that delay analysis of a *victim* line in presence crosstalk due to a neighboring *aggressor* line yields a switching factor that ranges between 0 and 2 depending on the relative switching direction of the lines. A line that causes a coupling event is called the aggressor and the affected line is called the victim.

Table 2.1 shows the possible values a switching factor can take depending on the switching direction of the victim and aggressor lines. The investigations in [12] show that a switching factor that ranges between -1 and 3 yields the true worst-case delay, and that the switching factor can be greater than 3 in presence of exponential waveforms. The current framework assumes that the switching factor ranges between 0 and 2, but can easily be extended to a more permissive range.

Consider the example in figure 2.3, which illustrates the RC lines of two nets  $a$  and  $b$  that are coupled to each other. Delay analysis of RC line  $b$  in presence of coupling capacitance can be done by analyzing the equivalent RC line in figure 2.4. Notice that

Victim switching direction	Aggressor switching direction	Voltage difference across $C_g$ and $C_c$ before transition	Voltage difference across $C_g$ and $C_c$ after transition	$S$
rising	steady (HIGH)	$V(C_g) = 0,$ $V(C_c) = -V_{dd}$	$V(C_g) = V_{dd},$ $V(C_c) = 0$	1
rising	steady (LOW)	$V(C_g) = 0,$ $V(C_c) = 0$	$V(C_g) = V_{dd},$ $V(C_c) = V_{dd}$	1
rising	rising	$V(C_g) = 0,$ $V(C_c) = 0$	$V(C_g) = V_{dd},$ $V(C_c) = 0$	0
rising	falling	$V(C_g) = 0,$ $V(C_c) = -V_{dd}$	$V(C_g) = V_{dd},$ $V(C_c) = V_{dd}$	2

Table 2.1: Switching factor values for different aggressor/victim switching configurations.

the coupling capacitance are converted to an equivalent ground capacitance (coupling capacitance multiplied with the switching factor  $S$ ). The Elmore delay at node  $n_{b4}$  in figure 2.4 using the upstream resistance formula in equation (2.1) is given by:

$$\begin{aligned}
T_D(n_{b4}) &= (S.C_{c1} + C_{b1})(R_{b1}) + (S.C_{c2} + C_{b2})(R_{b1} + R_{b2}) \\
&\quad + (S.C_{c3} + C_{b3})(R_{b1} + R_{b2} + R_{b3}) \\
&\quad + (S.C_{c4} + C_{b4})(R_{b1} + R_{b1} + R_{b3} + R_{b4})
\end{aligned} \tag{2.5}$$

Terms of equation (2.5) can be rearranged to obtain:

$$\begin{aligned}
T_D(n_{b4}) &= (C_{b1})(R_{b1}) + (C_{b2})(R_{b1} + R_{b2}) + (C_{b3})(R_{b1} + R_{b2} + R_{b3}) \\
&\quad + (C_{b4})(R_{b1} + R_{b1} + R_{b3} + R_{b4}) \\
&\quad + \\
&\quad ((S.C_{c1})(R_{b1}) + ((S.C_{c2})(R_{b1} + R_{b2}) + ((S.C_{c3})(R_{b1} + R_{b2} + R_{b3}) \\
&\quad + ((S.C_{c4})(R_{b1} + R_{b1} + R_{b3} + R_{b4})
\end{aligned} \tag{2.6}$$

The final value of the Elmore delay at node  $n_{b4}$ ,  $T_D(n_{b4})$  can be considered as a sum of two components: (i) The Elmore delay of the line *without* considering the coupling

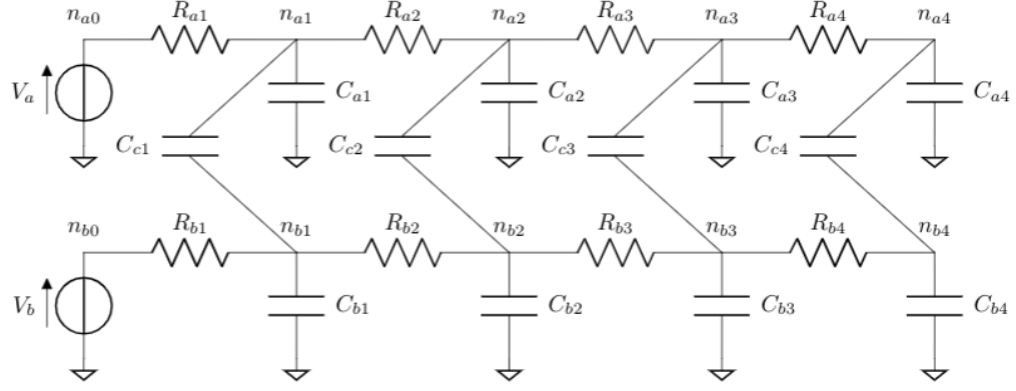


Figure 2.3: Example of coupled RC lines.

capacitance  $T_{D(C_g)}$ , and (ii) The increase in Elmore delay of the RC line due to addition of coupling capacitance  $T_{D(C_c)}$ . Equations for  $T_{D(C_g)}$ ,  $T_{D(C_c)}$ , and  $T_D$  at node  $n_{b4}$  are given below:

$$T_{D(C_g)}(n_{b4}) = (C_{b1})(R_{b1}) + (C_{b2})(R_{b1} + R_{b2}) + (C_{b3})(R_{b1} + R_{b2} + R_{b3}) + (C_{b4})(R_{b1} + R_{b2} + R_{b3} + R_{b4}) \quad (2.7)$$

$$T_{D(C_c)}(n_{b4}) = ((S.C_{c1})(R_{b1}) + ((S.C_{c2})(R_{b1} + R_{b2}) + ((S.C_{c3})(R_{b1} + R_{b2} + R_{b3}) + ((S.C_{c4})(R_{b1} + R_{b2} + R_{b3} + R_{b4})) \quad (2.8)$$

$$T_D(n_{b4}) = T_{D(C_g)}(n_{b4}) + T_{D(C_c)}(n_{b4}) \quad (2.9)$$

For the above equations, since the relative switching directions of the victims and aggressors are not available yet, the switching factor  $S$  is assumed to be a discrete random variable whose probability mass function is shown in figure 2.5, where  $P_1$ ,  $P_2$ , and  $P_3$  are probabilities of the switching factor being 0, 1, or 2 respectively.

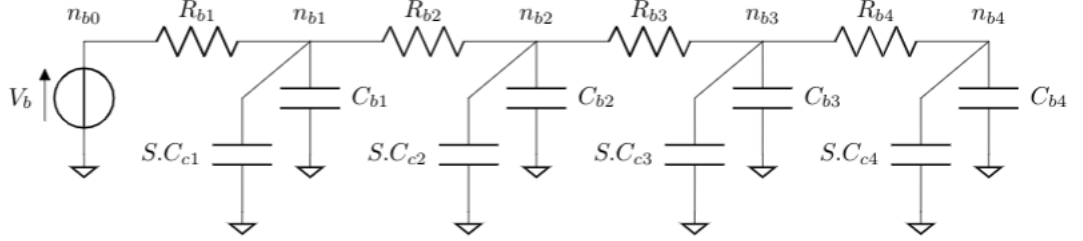


Figure 2.4: RC line  $b$  with coupling capacitance converted to equivalent ground capacitance.

With the above assumption, equation (2.8) can be seen as sum of  $n$  discrete random variables, where  $n$  is the number of nodes in the RC network. Sum of two discrete random variables  $X$  and  $Y$ ,  $Z = X + Y$  is given below:

$$P(Z = z) = \sum_{k=-\infty}^{\infty} (P(X = k)P(Y = z - k)) \quad (2.10)$$

where  $z$  is any value the resultant random variable  $Z$  can take. Equation (2.10) represents a convolution operation on  $X$  and  $Y$ . Therefore, we can define the distribution function of  $Z$ ,  $m_z(k)$  as the convolution of distribution functions of  $X$  and  $Y$ ,  $m_x(k)$  and  $m_y(k)$  respectively. As a consequence of modeling the switching factor  $S$  as a random variable,  $T_{D(C_c)}$  from equation (2.8) is now a distribution of delays. Figure 2.6 shows how the delay distribution is affected by the choice of switching factor probabilities.  $T_{min}$  and  $T_{max}$  denote the minimum and maximum possible delays for a given net. The figure highlights the difference in the spread of the final delay distribution for 3 different switching factors as follows:

$$\text{Case-1: } S_1 = \{0 : 33.33\%, 1 : 33.33\%, 2 : 33.33\%\}$$

$$\text{Case-2: } S_2 = \{0 : 20\%, 1 : 60\%, 2 : 20\%\}$$

$$\text{Case-3: } S_3 = \{0 : 10\%, 1 : 80\%, 2 : 10\%\}$$

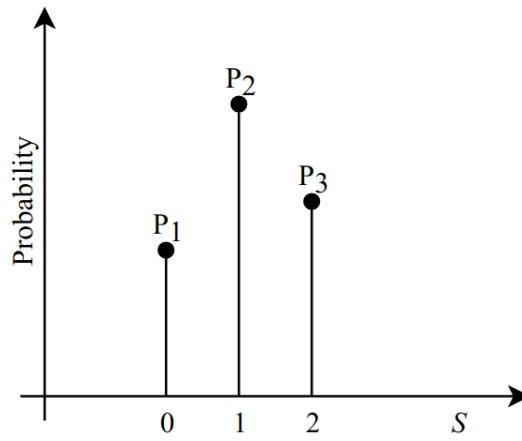


Figure 2.5: Probability Mass Function of the switching factor  $S$ .

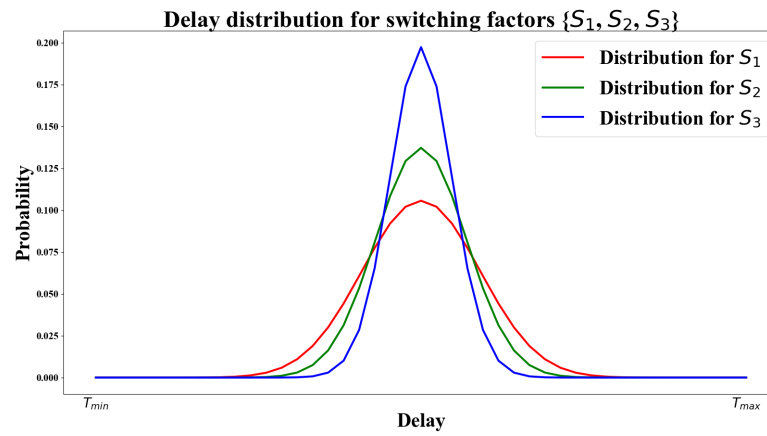


Figure 2.6: Delay distributions for different switching probabilities.

## 2.2 Support Size Reduction for Faster Analysis

The increase in Elmore delay of the RC network due to addition of coupling capacitances,  $T_{D(C_c)}$ , is expensive to compute for the following reasons:

- Time complexity to obtain the convolution sum for two random variables  $X$  and  $Y$ , defined as  $X * Y$ , is  $O(l_x \cdot l_y)$ , where  $l_x$  and  $l_y$  are the support lengths of random variables  $X$  and  $Y$  respectively.
- $O(n)$  convolutions are needed to obtain the final delay distribution ( $T_{D(C_c)}$ ), where  $n$  is the number of nodes in the RC network. Each convolution increases the support length of the resultant sum, thereby increasing the time complexity for the next stage of convolutions.

To overcome this issue, we attempt to reduce the granularity of the delay PMF after every convolution operation. We can afford to do so since the only metrics we are concerned with, in a static analysis setting, are the worst-case net delay values (maximum and minimum possible net delays) and an idea about the spread of the delay distribution. If we are able to assert that any point during the calculation of ( $T_{D(C_c)}$ ), if the support length (the size of the set containing all the possible values a distribution can take) of the intermediary delay distribution is bounded by a threshold  $T$ , we can achieve the final delay distribution with a time complexity that is  $O(T \cdot n) = O(n)$ , where  $n$  is the number of nodes in the RC network. The only consequence of this approach is that, in the final delay distribution, the probabilities indicate the chance of a net having a delay *around*  $t'_0$ , instead of indicating the chance of a net having a delay of exactly  $t_0$ , where  $t'_0$  and  $t_0$  are possible delay values of the net obtained from the final delay distribution with and without support length reduction respectively.

Figures 2.7 and 2.8 illustrate the process of reducing the granularity of a given PMF. In figure 2.7, instead of storing all the probabilities  $[P_i, P_j]$  for the range  $[a, b]$ , we are only storing the cumulative probability  $\sum_{c=i}^j P_c$  for the range  $[a, b]$ . The cumulative probability is assigned to the mean value of the range  $[a, b]$ .

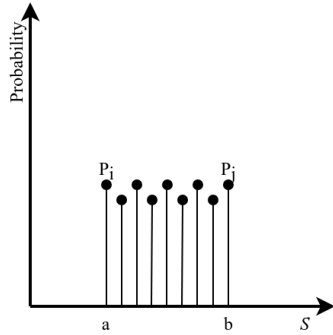


Figure 2.7: Delay PMF for the range  $[a, b]$ .

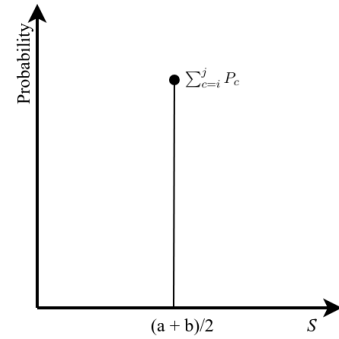


Figure 2.8: Reduced delay PMF for the range  $[a, b]$ .

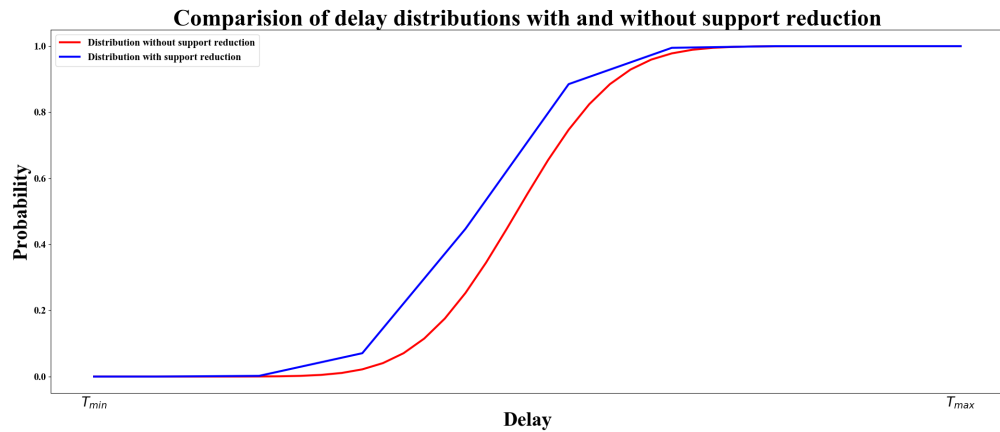


Figure 2.9: Demonstration of support reduction for a given distribution.

Figure 2.9 shows the actual cumulative delay distribution (support length = 43) and the cumulative delay distribution with reduced support length (support length threshold = 12). The probabilities of the occurrence of  $T_{min}$  and  $T_{max}$  (the minimum and maximum possible delays for the net) are kept intact in both the cases, the distribution with support reduction and without support reduction, respectively.

To ensure that the support length of the delay distribution is always less than the threshold, the range  $[a, b]$  on which support reduction is to be performed should depend

on the support length of the original distribution. For a PMF with fine granularity, the range  $[a, b]$  should be large, and for a PMF with coarse granularity, the range  $[a, b]$  should be small.



## Chapter 3

# Estimating Coupling Capacitance Using Congestion Maps

Once the placement stage determines the exact locations of the cells, wires are needed to be created to connect all the cells, while obeying all the design rules of the process. This stage is called routing.

Routing is performed in two stages: (i) In the global routing stage, the rough routes for the nets are estimated by mapping the components onto a coarse grid of larger cells called global routing cells or *gcells*, (ii) In the detailed routing stage, precise net geometries are found based on global routes created. The representations of gcells can have a varying degree of complexity ranging from simple squares with capacities to complex representations with capacities assigned to each layer and each direction. For the purpose of this thesis, we assume all the grid cells to be uniformly sized squares with horizontal and vertical edge crossing thresholds.

The global routing stage is fast because many fewer cells are involved. It also gives an idea about the difficulty of the detailed routing problem through congestion analysis. Congestion maps and reports obtained from congestion analysis compares the

routes assigned to a grid to its capacity. One of the most widely used congestion analysis technique is to perform a fast global routing and use that solution to generate a congestion map [3, 6].

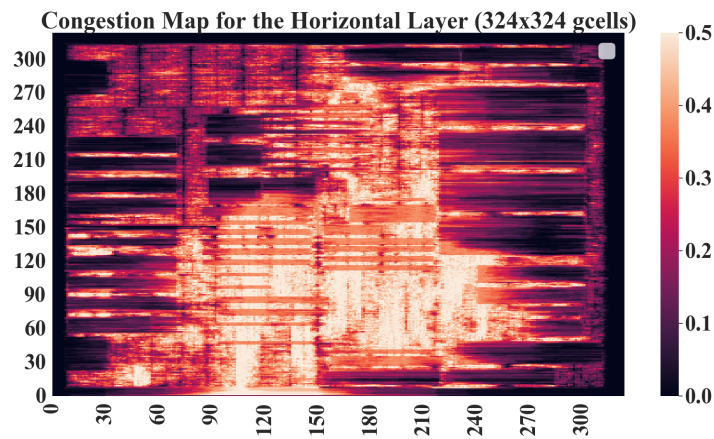


Figure 3.1: Example of a congestion map (horizontal layer).

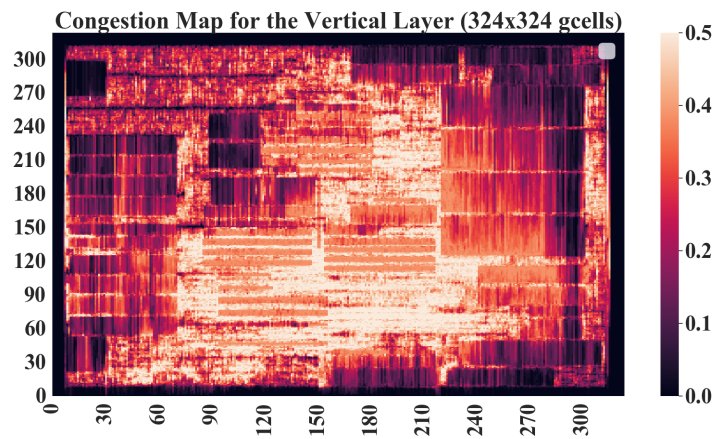


Figure 3.2: Example of a congestion map (vertical layer).

Figures 3.1 and 3.2 show examples of horizontal layer and vertical layer congestion map respectively. It can be observed from the congestion maps that, in regions that are

highly congested (congestion hot spots), there are a lot of nets that are routed closely to each other, making them highly susceptible to coupling effects. We intend to use of this information to estimate net parasitics. For the segment of net  $n$  that passes through a gcell  $i$  with horizontal and vertical layer congestion  $con_h(i)$  and  $con_v(i)$  respectively, the coupling capacitance of that segment can be thought as a function of  $con_h(i)$  and  $con_v(i)$ , i.e.,  $C_c(net) = f(con_h(i), con_v(i))$ .

### 3.1 Estimating Coupling Capacitance from Congestion Maps

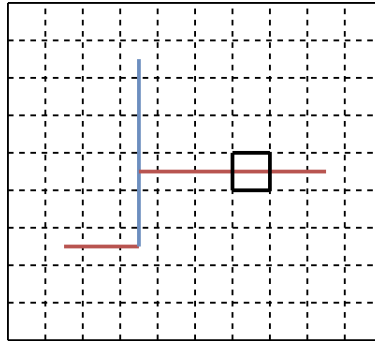


Figure 3.3: Example of a global net in a grid.

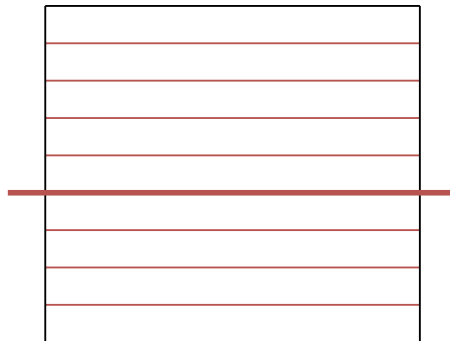


Figure 3.4: Focusing on the highlighted grid cell in figure 3.3.

Figure 3.3 shows a sample global route and figure 3.4 focuses on one of its segments passing through a grid cell  $g$  (highlighted in figure 3.3). The maximum possible same-layer coupling capacitance ( $C_{C_{max}(H)}$ ) that this net segment can experience is when all

the nearby routing tracks (for the same metal layer) of the gcell are occupied, i.e., the gcell is fully congested.  $C_{Cmax(H)}$  can be considered as a technology parameter that can be estimated using design rules for a particular process.

For any other gcell, that is not fully congested, the same-layer coupling capacitance on the net segment is always less than  $C_{Cmax(H)}$ . In a two layer example, for a net segment routed on metal layer  $M_h$ , same-layer coupling capacitance for that segment is roughly  $con_h \cdot C_{Cmax(H)}$ . Consider figures 3.5 and 3.6 which show different horizontal congestions and vertical congestions in a gcell. In figure 3.5 the net is routed in a horizontally drawn metal layer and the horizontal layer congestion ( $con_h$ ) is much less than the vertical layer congestion ( $con_v$ ). In this case, the coupling capacitance for the net segment at metal layer  $M_h$  is slightly lesser than the estimated coupling capacitance ( $con_h \cdot C_{Cmax(H)}$ ) due to inter-layer fringing of coupling capacitance. In figure 3.6, the net is routed in a horizontally drawn metal layer and the horizontal layer congestion ( $con_h$ ) is greater than the vertical layer congestion ( $con_v$ ), and the effect of fringing is minimal. In this framework, the effect of fringing is neglected as it is a secondary effect. Similar argument can be made for the orthogonal-layer coupling capacitance.

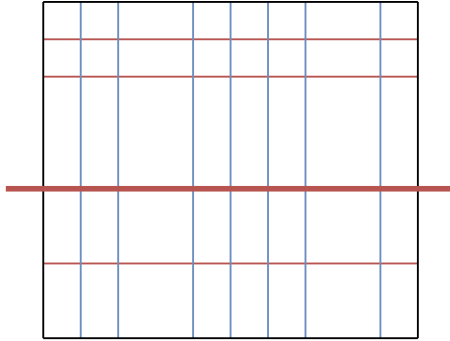


Figure 3.5: Example where horizontal layer congestion is lesser than vertical layer congestion.

For a net segment routed on metal layer  $M_h$  passing through a gcell  $i$  with horizontal layer congestion  $con_{hi}$ , if the orthogonal metal layer is  $M_v$ , congestion at the orthogonal

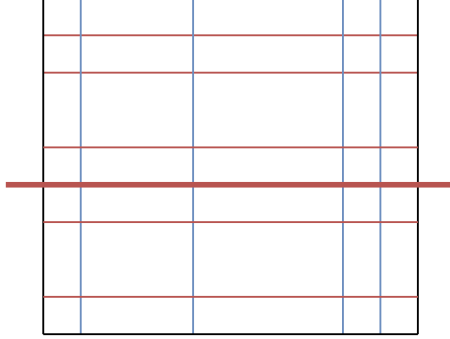


Figure 3.6: Example where horizontal layer congestion is greater than vertical layer congestion.

layer for the grid cell is  $con_{vi}$ , the estimated coupling capacitance for the net segment  $C_C(seg)$  is suggested to be taken as:

$$\overline{C_C(seg)} = (con_{hi} \cdot l_{seg}) \cdot C_{Cmax(H)} + (con_{vi} \cdot l_{seg}) \cdot C_{Cmax(V)} \quad (3.1)$$

where  $C_{Cmax(H)}$  and  $C_{Cmax(V)}$  are the maximum possible same-layer coupling capacitance and orthogonal-layer coupling capacitance respectively and  $l_{seg}$  is the length of the route segment that passes through a given gcell. The estimated coupling capacitance of a net is the sum of coupling capacitance contributions of all the route segments of the net and is given by:

$$\overline{C_C(net)} = \sum_{all\ segments} \overline{C_C(seg)} \quad (3.2)$$

The multiplication factors used with  $C_{Cmax(H)}$  and  $C_{Cmax(V)}$  in equation (3.1) can also be any user defined function of congestion values.

## 3.2 Constructing Approximate Parasitics from Congestion Maps and Global Router Solutions

Using congestion maps and the output of the global router, we can transform a net segment  $ab$  shown in figure 3.7 to an RC  $\pi$  element as shown in figure 3.8.  $R_w$  and

$C_w$  are the wire resistance and wire capacitance for that net segment. The multiplier  $M$  is obtained from the congestion map as explained above. Appendices A.1 and A.2 briefly explain how a net’s physical information is interpreted in common global routing solutions. Figure 3.9 shows the basic flow of obtaining the estimated parasitics from global router solution.

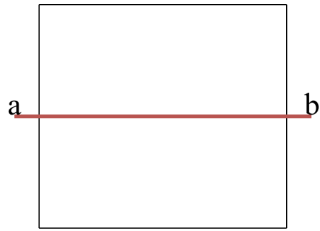


Figure 3.7: Physical representation of a net segment.

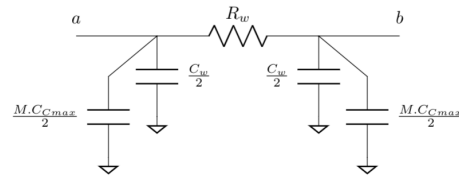


Figure 3.8: Equivalent RC  $\pi$  segment.

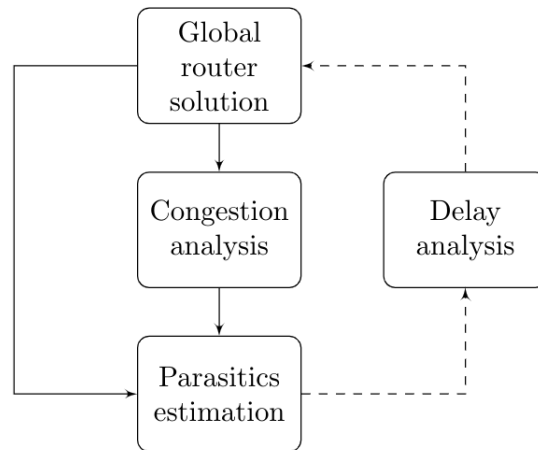


Figure 3.9: Flow chart for extracting estimated parasitics.

The RC  $\pi$  elements of all the net segments are stitched together for a particular net. Via resistances are inserted whenever a route’s metal layer changes and pin capacitances of the receiver gates (obtained from timing libraries) are appended at all the sink nodes. The extracted “approximate” parasitics can be dumped out in either detailed

standard parasitics format (DSPF) or in standard parasitics exchange format (SPEF). The coupling capacitance estimates for a net obtained from the netlist are multiplied with a probabilistic switching factor as explained in section 2.1 to obtain the delay distribution of the net. Given below is a sample of the extracted net in DSPF from a global router solution delivered by BoxRouter [5].

```

...
*Resistances for line segment: (3583,6698,1)-(3623,6698,1)
R336 n3583_6698 n3593_6698 5
R337 n3593_6698 n3303_6698 5
R338 n3603_6698 n3613_6698 5
R339 n3613_6698 n3623_6698 5
*Ground capacitances for line segment: (3583,6698,1)-(3623,6698,1)
C336 n3583_6698 VSS 2
C337 n3593_6698 VSS 4
C338 n3603_6698 VSS 4
C339 n3613_6698 VSS 4
C340 n3623_6698 VSS 2
*Coupling capacitances to ground for line segment:
*(3583,6698,1)-(3623,6698,1)
CC336 n3583_6698 VSS 0.625
CC337 n3593_6698 VSS 1.25
CC338 n3603_6698 VSS 1.14285714
CC339 n3613_6698 VSS 1.14285714
CC340 n3623_6698 VSS 0.57142857
**Line Segment: (2358,7048,1)-(3618,7048,1)
...

```

## Chapter 4

# Estimating Driving-Point Admittance

Modeling the interconnects as a lumped capacitance is no longer accurate, as the wire resistance in advanced technology nodes is comparable to the driver resistance. As a result, the driving gate experiences a lumped load capacitance that is significantly lesser than the total interconnect capacitance. This phenomenon is called interconnect resistive shielding, because the interconnect resistance shields the driver from a portion of the total interconnect capacitance. Obtaining the driving-point admittance is the first step towards estimating the “effective” capacitance that the driver experiences, in the presence of resistance shielding effects.

Parasitics associated with the driver gate are highly nonlinear in nature, whereas the wire parasitics are linear in nature. To tackle this, delay calculation engines calculate the interconnect delay in two steps: (i) First, the waveforms at the driving-point (output of the driver) are estimated using moment matching techniques which also take into account the nonlinear nature of driver resistance, (ii) The waveform obtained at the driving point is propagated to the sinks of the interconnect RC structure (inputs of receiver gates connected to the net) using linear system analysis techniques.



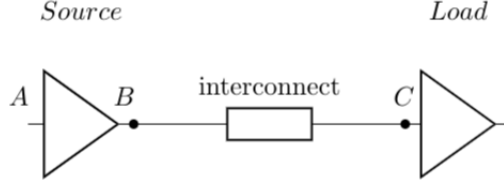


Figure 4.1: Illustration to identify different components of interconnect delay.

Figure 4.1 is used to illustrate the different components of the interconnect delay. Let  $T_g$  denote the source gate delay in the absence of a load and  $T_g(I)$  denote the gate delay in the presence of an interconnect load. Due to loading effects,  $T_g(I) > T_g$ . The interconnect delay  $T_{int}$  is as follows:

$$\begin{aligned}
 T_{int} &= T_{AC} - T_g \\
 &= [T_g(I) + T_{BC}] - T_g \\
 &= [T_g(I) - T_g] + T_{BC}
 \end{aligned} \tag{4.1}$$

where  $T_{AC}$  is the delay from point “A” to point “C”, and  $T_{BC}$  is the delay of the interconnect structure. The driving-point analysis techniques presented below are aimed at modeling the “extra source gate delay” ( $[T_g(I) - T_g]$ ).

O’Brien-Savarino reduction [17] is the process of transforming a general RC tree (figure 4.2) to an RC  $\pi$  model at the driving point. This model is used to compute the effective capacitance iteratively by comparing the average current drawn by the model to the average current drawn by a single capacitance up to the 50% delay mark. The admittance at the driving-point is given by:

$$\begin{aligned}
 Y(S) &= \frac{I(S)}{V(S)} \\
 &= \sum_{i=0}^{\infty} y_i \cdot s^i \quad [\text{If } V(t) \text{ is an impulse excitation}]
 \end{aligned} \tag{4.2}$$

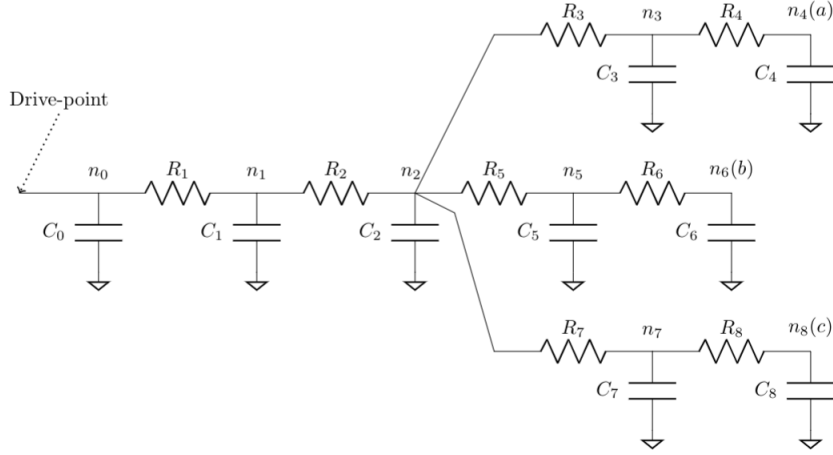
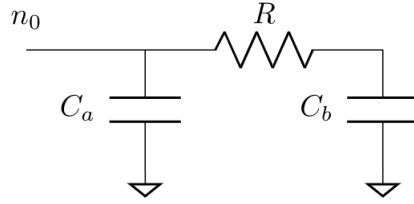


Figure 4.2: A general RC tree.

Figure 4.3: A reduced RC  $\pi$  model.

For the RC  $\pi$  model as shown in figure 4.3, the driving point admittance is given by:

$$Y = s.C_b + \frac{s.C_a}{1 + s.R.C_a} \quad (4.3)$$

Expanding equation (4.3) using Taylor series approximation and matching the first three non-zero coefficients (moments) from equation (4.2) we get:

$$C_a = \frac{y_2^2}{y_3} \quad (4.4)$$

$$C_b = y_1 - \frac{y_2^2}{y_3} \quad (4.5)$$

$$R = -\frac{y_3^2}{y_2^3} \quad (4.6)$$

For the RC  $\pi$  model to be (realizable), i.e.,  $R > 0, C_a > 0, C_b > 0$ , the following restrictions are imposed on  $y_1, y_2, y_3$ :

$$\begin{aligned} y_1 > 0, y_2 < 0, y_3 > 0 \\ y_1 y_3 > y_2^2 \end{aligned} \quad (4.7)$$

Consider figure 4.2. An algorithm to obtain  $\{y_1, y_2, y_3\}$  for a general RC tree that contains lumped resistors and capacitors is described below. Let  $Y_u(S)$  denote the unknown driving-point admittance looking downstream from the point that is immediately upstream of the circuit element that is to be traversed, let  $Y_d(S)$  denote the driving-point admittance looking downstream from the point that is immediately downstream of the circuit element that is to be traversed. Given below are three rules (assuming there are only lumped parasitic elements in the tree) to correctly propagate the moments upstream. The algorithm starts at the leaves of the RC tree ( $a, b$ , and  $c$  in figure 4.2) and initializes  $\{(y_d)_1, (y_d)_2, (y_d)_3\}$  at the leaves to zero, and our goal is to traverse upstream to the root and obtain  $\{(y_u)_1, (y_u)_2, (y_u)_3\}$  at the root of the RC tree.

**Rule-1:** Rule-1 explains the propagation of moments upstream along a single branch past a series lumped capacitor (figure 4.4).

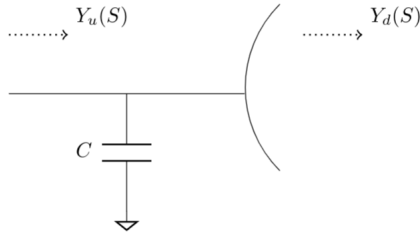


Figure 4.4: Propagation of moments past a lumped capacitor.

$$\begin{aligned}
 (y_u)_1 &= (y_d)_1 + C \\
 (y_u)_2 &= (y_d)_2 \\
 (y_u)_3 &= (y_d)_3
 \end{aligned}
 \tag{4.8}$$

**Rule-2:** Rule-2 explains the propagation of moments upstream along a single branch past a series lumped resistor (figure 4.5).

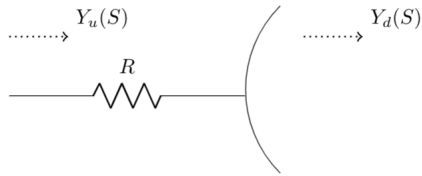


Figure 4.5: Propagation of moments past a lumped resistor.

$$\begin{aligned}
 (y_u)_1 &= (y_d)_1 \\
 (y_u)_2 &= (y_d)_2 - R(y_d)_1^2 \\
 (y_u)_3 &= (y_d)_3 - 2R(y_d)_1(y_d)_2 + R^2(y_d)_1^3
 \end{aligned}
 \tag{4.9}$$

**Rule-3:** Rule-3 explains the combining of 2 or more single-branch moments that are parallel at a branching point in the tree (figure 4.6). Let  $Y_B(S)$  denote the parallel combination of  $b$  moments.

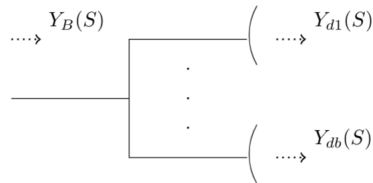


Figure 4.6: Propagation of moments past a branching point.

$$\begin{aligned}
(y_B)_1 &= \sum_{i=1}^b (y_{di})_1 \\
(y_B)_2 &= \sum_{i=1}^b (y_{di})_2 \\
(y_B)_3 &= \sum_{i=1}^b (y_{di})_3
\end{aligned} \tag{4.10}$$

In presence of a probabilistic coupling capacitance, as described in section 2.1, the final moments obtained at root  $\{y_1, y_2, y_3\}$  and the moments obtained at each intermediate step,  $\{(y_d)_1, (y_d)_2, (y_d)_3\}$  and  $\{(y_u)_1, (y_u)_2, (y_u)_3\}$  would also be random variables. All mathematical operations performed in equations (4.8) to (4.10) should be performed in the context of the moments being random variables. Distributions of  $\{y_1, y_2, y_3\}$  can be obtained by performing Monte Carlo simulations of the RC tree with probabilistic coupling capacitance.

## Chapter 5

# Validation of Results with OpenSTA

OpenSTA [4] is the timing analysis/verification component of the OpenROAD project [1], an effort towards democratizing hardware design. The OpenROAD project aims to provide a self-learning open-source ecosystem that offers a tapeout-capable tool chain. OpenSTA is an open-source static timing analysis tool published under the GNU General Public License.

### 5.1 Incorporating Crosstalk Delay

In the presence of coupling capacitance, the net delay can increase (or decrease) depending on the relative switching directions of the aggressor/victim pair. Consider figure 5.1.  $T_{nom}$  represents the nominal net delay, which is calculated when the aggressor net is quiet (switching factor is 1). In this case, the net delay is computed using 1X coupling capacitance.  $T_{min}$  and  $T_{max}$  are the minimum and maximum possible net delays for the worst-case crosstalk coupling respectively. For example, the worst-case timing may be computed using 0X coupling capacitance (aggressor and victim are switching together in the same direction, switching factor is 0) for worst-case minimum delay and 2X coupling capacitance (aggressor and victim are switching together in the opposite

direction, switching factor is 2) for worst-case maximum delay.  $T'_{min}$  and  $T'_{max}$  represents the actual minimum and maximum net delays which are obtained after updating the worst-case net delays in multiple iterations as described in [15, 19] .

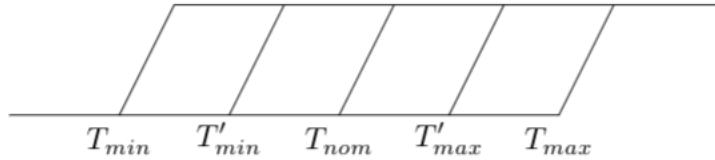


Figure 5.1: Illustration of crosstalk delay on a net.

Timing reports presented in Appendix B illustrate the calculation of the worst-case net delays. The slacks shown in timing reports 1 and 2 (B.1 and B.2) represent delay calculations performed to obtain the nominal net delays (1X coupling capacitance) for setup and hold analysis respectively. The slacks shown in timing reports 3 and 4 (B.3 and B.4) represent the delay calculations performed to obtain the worst-case maximum (2X coupling capacitance) and minimum (0X coupling capacitance) net delays for setup and hold analysis respectively. For setup timing analysis, the slack calculated in timing report B.3 is less than the slack calculated in B.1 because the net delays increase due to positive crosstalk delay (switching factor is 2). For hold timing analysis, the slack calculated in timing report B.4 is less than the slack calculated in B.2 because the net delays decrease due to negative crosstalk delay (switching factor is 0).

## 5.2 Validation of Results

In this section, we provide the following comparisons:

- We compare the total coupling capacitance of the net obtained from SPEF of the design to the total coupling capacitance of the net obtained from methods explained in chapter 3.

- We compare the worst-case net delays obtained from OpenSTA to the worst-case net delays obtained from methods described in chapter 2.

The analysis is performed on the “AES cipher” circuit, designed in ASAP 7nm process design kit (PDK). Table 5.1 shows the list of randomly selected nets that were chosen for in-depth analysis. It also shows the total net coupling capacitance and the worst-case maximum net delay obtained from OpenSTA path reports for each net.

Net Name	Name map in SPEF	Total $C_c$ (pF)	$T_{max}$ (pS)
FE_OCPN11453_FE_OFN19269_net_5624	*464	1.10E-04	8.05E-02
FE_OCPN11485_FE_OFN20277_n	*433	1.78E-05	3.77E-02
FE_OCPN11401_net_12566	*515	6.85E-05	2.95E-02
FE_OCPN11476_net_1361	*442	4.90E-05	2.75E-02
FE_OCPN11484_FE_OFN18956_net_16651	*434	5.64E-05	2.64E-02
FE_OCPN11499_FE_OFN20326_net_14843	*419	4.68E-05	2.55E-02
FE_OCPN11518_net_18647	*400	2.47E-05	2.48E-02
FE_OCPN11444_FE_RN_230	*473	3.09E-05	1.67E-02
FE_OCPN11488_net_17127	*430	5.46E-06	1.50E-02
FE_OCPN11400_net_9726	*516	9.11E-06	1.46E-02

Table 5.1: Sample set of nets chosen for analysis.

Figure 5.2 compares the total coupling capacitance on the net obtained through two processes: (i) In Process-1, we obtain the total coupling capacitance on the net by subtracting the total wire capacitance of the net from the total net capacitance, (ii) In Process-2, we obtain the coupling capacitance using methods explained in section 3.1. We see a strong correlation for the two cases explained above. The slight deviations can be attributed to the uncertainty in estimation of  $C_{Cmax(H)}$  and  $C_{Cmax(V)}$ , since different gcells can have different values for  $C_{Cmax(H)}$  and  $C_{Cmax(V)}$  depending on the edge capacities assigned to that gcell.

Figure 5.3 compares the worst-case (maximum) net delays obtained through two processes: (i) In Process-1, we obtain the worst-case net delays using OpenSTA, (ii) In Process-2, we first extract the estimated parasitics using methods described in section



3.2. We then perform Elmore delay analysis on the extracted parasitics to obtain the worst-case net delays. Table 5.2 shows a comparison of delay values obtained through OpenSTA and the summary of the final delay distribution obtained using estimated parasitics. For the delay distribution using estimated parasitics the switching factor distribution is taken to be {0:10%, 1:80%, 2:10%}. For OpenSTA, the worst-case minimum delay (0X coupling capacitance), nominal delay (1X coupling capacitance), and worst-case maximum delay (2x coupling capacitance) are provided. For the final delay distribution obtained using the estimated parasitics, the minimum and maximum values of the delay distribution, mean of the delay distribution, and the standard deviation( $\sigma$ ) of the delay distribution are provided. The post-layout net timing window and the timing window obtained through estimated parasitics strongly overlap.

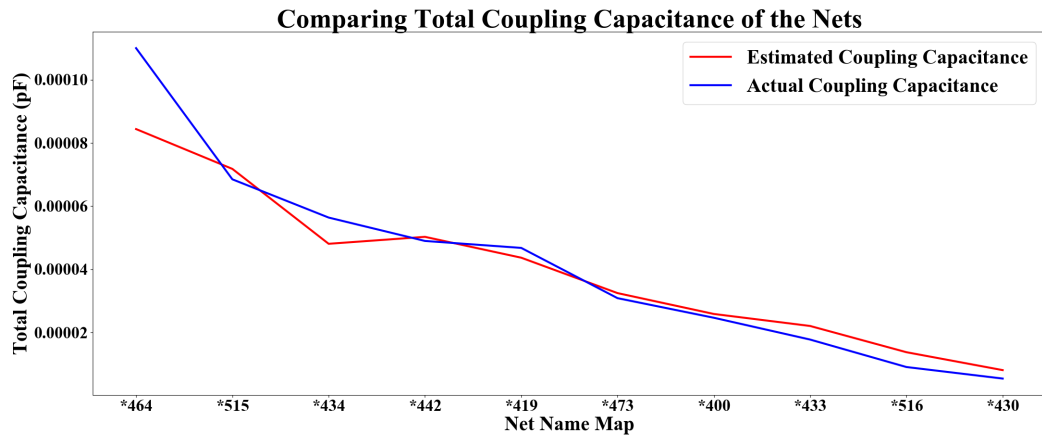


Figure 5.2: Comparison of a net's total coupling capacitance.

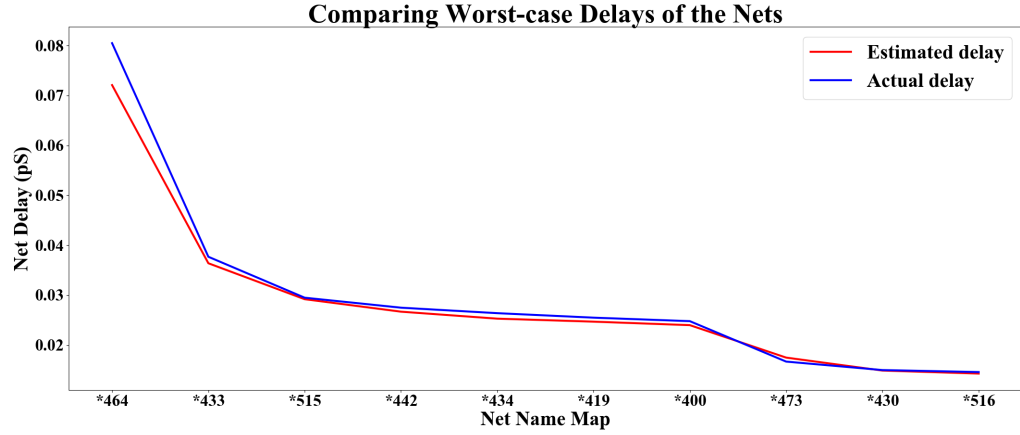


Figure 5.3: Comparison of a worst-case maximum net delays.

Name Map	OpenSTA (pS)			Expedited Parasitic Extraction (pS)			
	$T_{D(min)}$	$T_{D(nom)}$	$T_{D(max)}$	$T_{D(min)}$	$T_{D(max)}$	Mean Delay	$\sigma$
*464	5.97E-02	7.01E-02	8.05E-02	5.66E-02	7.21E-02	6.43E-02	2.34E-03
*433	3.66E-02	3.72E-02	3.77E-02	3.49E-02	3.64E-02	3.57E-02	2.16E-04
*515	2.41E-02	2.68E-02	2.95E-02	2.32E-02	2.92E-02	2.62E-02	8.53E-04
*442	2.23E-02	2.49E-02	2.75E-02	2.11E-02	2.67E-02	2.39E-02	8.08E-04
*434	2.28E-02	2.46E-02	2.64E-02	2.20E-02	2.53E-02	2.36E-02	4.71E-04
*419	2.25E-02	2.40E-02	2.55E-02	2.16E-02	2.47E-02	2.32E-02	4.28E-04
*400	2.32E-02	2.40E-02	2.48E-02	2.22E-02	2.40E-02	2.31E-02	2.54E-04
*473	1.61E-02	1.64E-02	1.67E-02	1.57E-02	1.75E-02	1.66E-02	2.52E-04
*430	1.49E-02	1.50E-02	1.50E-02	1.46E-02	1.49E-02	1.47E-02	4.67E-05
*516	1.41E-02	1.44E-02	1.46E-02	1.36E-02	1.43E-02	1.39E-02	1.07E-04

Table 5.2: A comparison of delay values obtained through OpenSTA and expedited parasitic extraction.

## Chapter 6

# Conclusion

Signal integrity at advanced technology nodes has turned out to be a major concern for IC designers and timing engineers. As the technology node is pushed further into the deep sub-micron region, the effect of crosstalk is amplified. Elaborate crosstalk analysis methodologies requires sign-off accurate parasitics of the fully routed design. Since concrete parasitic extraction is a time consuming task, executing any engineering change orders (ECOs) towards resolving timing uncertainties caused due to crosstalk effects can develop into a cumbersome exercise. A quick method to weed out highly coupled nets is essential for a timely tapeout.

In this thesis, we have proven that congestion maps could be regarded as a quick and reliable metric to gauge the severity of coupling effects. The delay metrics obtained through expedited extraction of estimated parasitics shows strong correlation to the post-layout delay metrics obtained from an open-source timing analyzer. This thesis also presents the framework for a robust crosstalk analysis methodology under the assumption that coupling capacitances could be modeled as random variables.

# References

- [1] T. Ajayi et al. “OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain”. In: *Government Microcircuit Applications and Critical Technology Conference*. 2019. URL: <https://vlsicad.ucsd.edu/Publications/Conferences/370/c370.pdf>.
- [2] M. T. Bohr. “Interconnect Scaling-The Real Limiter to High Performance ULSI”. In: *Proceedings of International Electron Devices Meeting*. 1995, pp. 241–244. DOI: 10.1109/IEDM.1995.499187.
- [3] H. Chen, C. Hsu, and Y. Chang. “High-Performance Global Routing with Fast Overflow Reduction”. In: *Asia and South Pacific Design Automation Conference*. 2009, pp. 582–587. DOI: 10.1109/ASPDAC.2009.4796543.
- [4] J. Cherry. *OpenSTA (OpenROAD Project)*. <https://github.com/abk-openroad/OpenSTA>. 2018.
- [5] M. Cho and D. Z. Pan. “BoxRouter: A New Global Router Based on Box Expansion and Progressive ILP”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 26.12 (2007), pp. 2130–2143. ISSN: 0278-0070. DOI: 10.1109/TCAD.2007.907003.
- [6] C. Chu and Y. Wong. “FLUTE: Fast Lookup Table Based Rectilinear Steiner Minimal Tree Algorithm for VLSI Design”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27.1 (2008), pp. 70–83. ISSN: 0278-0070. DOI: 10.1109/TCAD.2007.907068.

- [7] F. Dartu and L. T. Pileggi. “Calculating Worst-Case Gate Delays Due to Dominant Capacitance Coupling”. In: *Proceedings of the 34th Design Automation Conference*. June 1997, pp. 46–51. DOI: 10.1109/DAC.1997.597115.
- [8] W. C. Elmore. “The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers”. In: *Journal of Applied Physics* 19.1 (1948), pp. 55–63. DOI: 10.1063/1.1697872.
- [9] B. Franzini et al. “Crosstalk Aware Static Timing Analysis: A Two Step Approach”. In: *Proceedings IEEE First International Symposium on Quality Electronic Design (Cat. No. PR00525)*. 2000, pp. 499–503. DOI: 10.1109/ISQED.2000.838935.
- [10] L. Gal. “On-Chip Crosstalk-The New Signal Integrity Challenge”. In: *Proceedings of the IEEE Custom Integrated Circuits Conference*. 1995, pp. 251–254. DOI: 10.1109/CICC.1995.518179.
- [11] P. D. Gross et al. “Determination of Worst-Case Aggressor Alignment for Delay Calculation”. In: *IEEE/ACM International Conference on Computer-Aided Design. Digest of Technical Papers (IEEE Cat. No.98CB36287)*. 1998, pp. 212–219. DOI: 10.1145/288548.288616.
- [12] A. B. Kahng, S. Muddu, and E. Sarto. “On Switch Factor Based Analysis of Coupled RC Interconnects”. In: *Proceedings of the 37th Annual Design Automation Conference*. DAC '00. Los Angeles, California, USA: ACM, 2000, pp. 79–84. ISBN: 1-58113-187-9. DOI: 10.1145/337292.337318. URL: <http://doi.acm.org/10.1145/337292.337318>.
- [13] W. H. Kao, M. Basel, and R. Singh. “Parasitic Extraction: Current State of the Art and Future Trends”. In: *Proceedings of the IEEE* 89.5 (2001), pp. 729–739. ISSN: 0018-9219. DOI: 10.1109/5.929651.
- [14] D. A. Kirkpatrick and A. L. Sangiovanni-Vincentelli. “Techniques for Crosstalk Avoidance in the Physical Design of High-Performance Digital Systems”. In: *Proceedings of the IEEE/ACM International Conference on Computer-aided Design*.

- ICCAD '94. San Jose, California, USA: IEEE Computer Society Press, 1994, pp. 616–619. ISBN: 0-89791-690-5. URL: <http://dl.acm.org/citation.cfm?id=191326.191587>.
- [15] Y. Kukimoto and K. Keutzer. “Refining Switching Window by Time Slots for Crosstalk Noise Calculations”. In: *IEEE/ACM International Conference on Computer Aided Design*. Nov. 2002, pp. 583–586. DOI: 10.1109/ICCAD.2002.1167591.
- [16] G. E. Moore. “Cramming More Components onto Integrated Circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff.” In: *IEEE Solid-State Circuits Society Newsletter* 11.3 (2006), pp. 33–35. ISSN: 1098-4232. DOI: 10.1109/N-SSC.2006.4785860.
- [17] P. R. O’Brien and T. L. Savarino. “Modeling The Driving-Point Characteristic of Resistive Interconnect for Accurate Delay Estimation”. In: *IEEE International Conference on Computer-Aided Design. Digest of Technical Papers*. 1989, pp. 512–515. DOI: 10.1109/ICCAD.1989.77002.
- [18] L. T. Pileggi. “Coping with RC(L) Interconnect Design Headaches”. In: *Proceedings of the IEEE/ACM International Conference on Computer-aided Design. ICCAD '95*. San Jose, California, USA: IEEE Computer Society, 1995, pp. 246–253. ISBN: 0-8186-7213-7. URL: <http://dl.acm.org/citation.cfm?id=224841.225048>.
- [19] S. S. Sapatnekar. “On the Chicken-And-Egg Problem of Determining the Effect of Crosstalk on Delay in Integrated Circuits”. In: *IEEE 8th Topical Meeting on Electrical Performance of Electronic Packaging (Cat. No.99TH8412)*. 1999, pp. 245–248. DOI: 10.1109/EPEP.1999.819235.
- [20] S. S. Sapatnekar. *Timing*. Berlin, Germany: Springer-Verlag, 2004. ISBN: 1402076711.
- [21] D. Sylvester. “Analytical Modeling and Characterization of Deep-Submicrometer Interconnect”. In: *Proceedings of the IEEE* 89.5 (2001), pp. 634–664. ISSN: 0018-9219. DOI: 10.1109/5.929648.

- [22] T. Xue, E. S. Kuh, and D. Wang. “Post Global Routing Crosstalk Risk Estimation and Reduction”. In: *Proceedings of International Conference on Computer Aided Design*. Nov. 1996, pp. 302–309. DOI: 10.1109/ICCAD.1996.569714.
- [23] H. You and M. Soma. “Crosstalk Analysis of Interconnection Lines and Packages in High-Speed Integrated Circuits”. In: *IEEE Transactions on Circuits and Systems* 37.8 (1990), pp. 1019–1026. ISSN: 0098-4094. DOI: 10.1109/31.56075.

## Appendix A

# BoxRouter and DEF File Formats

This appendix is dedicated to explain how physical information of nets is interpreted from global router solutions in BoxRouter format and DEF format. Specific syntaxes for two file formats are explained here that are relevant to the topics discussed in this thesis.



## A.1 BoxRouter File Format

### A.1.1 Input File Format

The input file that is given to the BoxRouter global router describes the following:

- Locations and sizes of gcells.
- Horizontal and vertical capacity for each metal layer in a gcell.
- Source/sink(s) coordinates for each net.

Given below is an example of an input, with comments included in italics (comments not included in original file), that is supplied to the BoxRouter. Figure A.1 illustrates the construction of the routing grids as per specifications provided in the input.

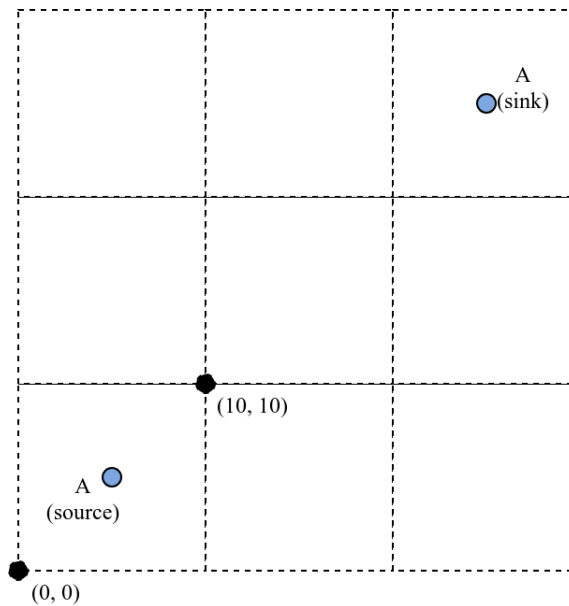


Figure A.1: Routing grid structure from BoxRouter input.

```
#3x3 grid that supports two metal layers
grid 3 3 2
#Vertical edge capacity of each metal layer layer
```

```
vertical capacity 0 1
#Horizontal edge capacity of each gcell
horizontal capacity 1 0
minimum width 1 1
minimum spacing 0 0
<lower-left X> <lower-left Y> <tile width> <tile height>
0 0 10 10
#Number of nets
1
#<net name> <net number> <number of pins> <width>
A 0 2 1
#routing points begin
# <x> <y> <metal layer>
5 5 0
25 25 0
#routing points end
```

### A.1.2 Output File Format

The output file represents the global router's solution. It contains the routing information for all the nets. Each net is broken down into line segments and routed together. Given below is an example of an output given by BoxRouter for the input in section A.1.1. Figure A.2 illustrates the construction of the net routes as per specifications provided in the output.

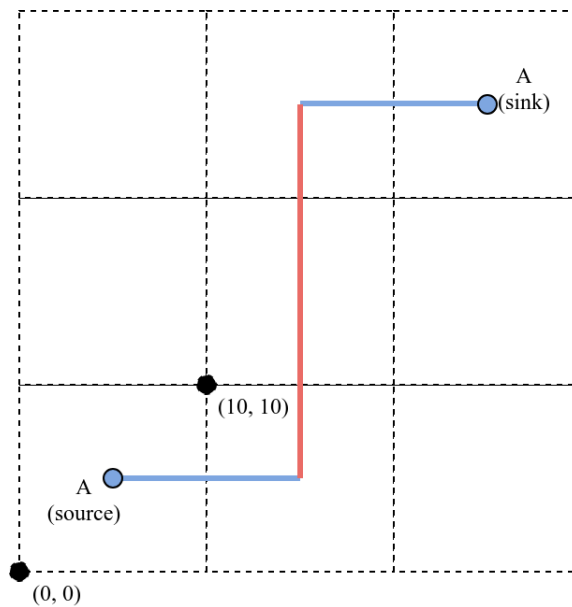


Figure A.2: Net's route obtained from BoxRouter output.

```
A 0          #<net name> <net number>
(5,5,0)-(15,5,0)
(15,5,0)-(15,5,1)  #indicates layer change at the routing point
(15,5,1)-(15,15,1)
(15,15,1)-(15,25,1)
(15,25,1)-(15,25,0)
(15,25,0)-(25,25,0)
!
```

## A.2 DEF File Format

The Design Exchange Format (DEF) file contains design-specific information of a circuit at all points during the layout process. The grid information for congestion analysis and the net's physical information are both available in DEF. Figure A.3 illustrates the physical structure of the net for the DEF specification given below. Comments are included in italics (comments not included in original file).

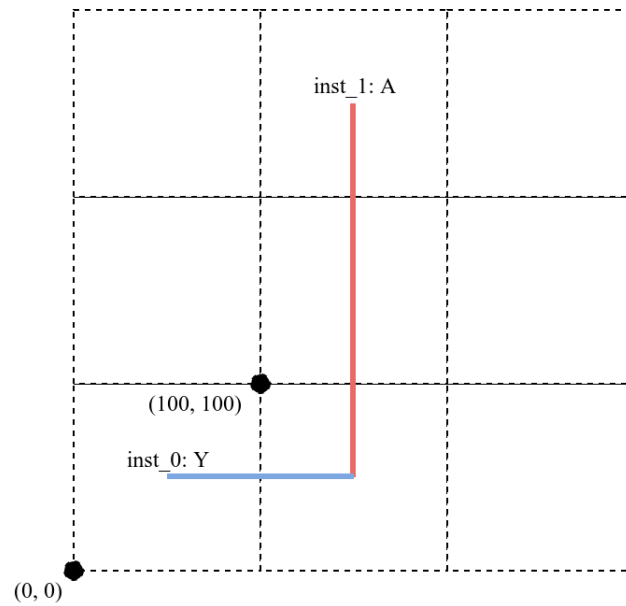


Figure A.3: Net's route obtained from DEF file.

```

...
#X <start> DO <num_columns+1> STEP <space>
GCELLGRID X 0 DO 4 STEP 100 ;
#Y <start> DO <num_rows+1> STEP <space>
GCELLGRID Y 0 DO 4 STEP 100 ;
...

```

```
#number of nets
NETS 19617 ;
#net name
- FE_OCPN11527_net_17432
#connectivity information
( inst_0 A ) ( inst_1 Y )
#"*" indicates corresponding coordinate is copied
+ ROUTED M3 ( 150 50 ) ( * 250 ) VIA23
NEW M2 ( 50 50 ) ( 150 * ) VIA12
NEW M3 ( 150 50 ) VIA23
...
```

## Appendix B

# Timing Reports

This appendix is dedicated to illustrate the working of the basic signal integrity framework implemented on the open-source timing tool (OpenSTA) through timing reports, as explained in chapter 5.

## B.1 Timing Report 1

Startpoint: uram0\_aram\_0\_u0\_ra\_reg\_8\_\_u0  
 (rising edge-triggered flip-flop clocked by clk)  
 Endpoint: udcom0\_dcom0\_r\_reg\_data\_\_1\_\_u0  
 (rising edge-triggered flip-flop clocked by clk)  
 Path Group: clk  
 Path Type: max

Delay	Time	Description
0.00	0.00	clock clk (rise edge)
0.00	0.00	clock network delay (ideal)
0.00	0.00	^ uram0_aram_0_u0_ra_reg_8__u0/CLK (DFFHQx4_ASAP7_75t_SRAM)
60.62	60.62	v uram0_aram_0_u0_ra_reg_8__u0/Q (DFFHQx4_ASAP7_75t_SRAM)
69.17	129.79	^ U481336/Y (NAND2x1_ASAP7_75t_R)
74.84	204.63	v U365661/Y (NOR3xp33_ASAP7_75t_R)
59.49	264.12	v FE_OFC6570_n483424/Y (BUFx3_ASAP7_75t_SRAM)
87.74	351.86	^ U481355/Y (INVx1_ASAP7_75t_SRAM)
62.14	413.99	^ FE_OFC6736_n308163/Y (BUFx2_ASAP7_75t_R)
92.33	506.33	^ FE_OFC6739_n308163/Y (BUFx2_ASAP7_75t_R)
123.17	629.50	v U365372/Y (NOR2xp33_ASAP7_75t_SRAM)
49.05	678.54	^ U628155/Y (NAND2xp33_ASAP7_75t_SRAM)

56.81	735.35	v	U628156/Y	(NAND2xp5_ASAP7_75t_L)
78.69	814.04	v	FE_OFC11729_n469029/Y	(BUFx3_ASAP7_75t_R)
50.13	864.17	^	U628162/Y	(NOR3xp33_ASAP7_75t_SRAM)
47.34	911.51	v	U628163/Y	(NAND3xp33_ASAP7_75t_R)
65.92	977.43	^	FE_OFC11956_n469059/Y	(INVx1_ASAP7_75t_R)
58.34	1035.78	v	FE_OFC11957_n469059/Y	(INVx1_ASAP7_75t_L)
80.83	1116.61	^	U628189/Y	(NOR3xp33_ASAP7_75t_SRAM)
43.03	1159.64	v	U628190/Y	(NAND2xp33_ASAP7_75t_SRAM)
56.81	1216.46	^	U628242/Y	(NOR3xp33_ASAP7_75t_R)
52.29	1268.75	^	FE_OFC13549_n469133/Y	(HB1xp67_ASAP7_75t_R)
47.17	1315.92	^	FE_OFC3771_n469133/Y	(BUFx3_ASAP7_75t_R)
68.24	1384.15	v	U628256/Y	(NAND3xp33_ASAP7_75t_SRAM)
33.67	1417.83	^	U628672/Y	(NOR3xp33_ASAP7_75t_SRAM)
25.84	1443.66	v	U628673/Y	(NAND2x1_ASAP7_75t_SL)
85.81	1529.48	^	U630341/Y	(NOR3xp33_ASAP7_75t_SRAM)
89.14	1618.62	v	U630970/Y	(OAI222xp33_ASAP7_75t_R)
70.78	1689.40	v	FE_OFC12432_n472021/Y	(BUFx3_ASAP7_75t_R)
64.50	1753.89	^	U630971/Y	(OAI22xp33_ASAP7_75t_SRAM)
26.73	1780.62	v	U630973/Y	(NAND2xp33_ASAP7_75t_SRAM)
18.22	1798.84	^	U441087/Y	(OAI21xp33_ASAP7_75t_SRAM)
27.21	1826.05	v	U630974/Y	(NOR2xp33_ASAP7_75t_SRAM)
19.17	1845.22	^	U630975/Y	(NAND2xp33_ASAP7_75t_SRAM)
22.87	1868.09	v	U630978/Y	(NAND3xp33_ASAP7_75t_SRAM)
0.07	1868.16	v	udcom0_dcom0_r_reg_data__1__u0/D	



(DFFHQx4\_ASAP7\_75t\_SRAM)

	1868.16		data arrival time
6000.00	6000.00		clock clk (rise edge)
0.00	6000.00		clock network delay (ideal)
0.00	6000.00		clock reconvergence pessimism
	6000.00	^	udcom0_dcom0_r_reg_data__1__u0/CLK (DFFHQx4_ASAP7_75t_SRAM)
-8.26	5991.74		library setup time
	5991.74		data required time
-----			
	5991.74		data required time
	-1868.16		data arrival time
-----			
	4123.58		slack (MET)

## B.2 Timing Report 2

Startpoint: uram0\_aram\_0\_u0\_ra\_reg\_8\_\_u0  
 (rising edge-triggered flip-flop clocked by clk)

Endpoint: udcom0\_dcom0\_r\_reg\_data\_\_1\_\_u0  
 (rising edge-triggered flip-flop clocked by clk)

Path Group: clk

Path Type: min

Delay	Time	Description
0.00	0.00	clock clk (rise edge)
0.00	0.00	clock network delay (ideal)
0.00	0.00	^ uram0_aram_0_u0_ra_reg_8__u0/CLK (DFFHQx4_ASAP7_75t_SRAM)
67.09	67.09	^ uram0_aram_0_u0_ra_reg_8__u0/Q (DFFHQx4_ASAP7_75t_SRAM)
37.39	104.48	v U481638/Y (NOR3xp33_ASAP7_75t_SRAM)
64.24	168.72	v FE_OF5352_n481128/Y (BUFx3_ASAP7_75t_R)
95.33	264.05	^ U627685/Y (NAND3xp33_ASAP7_75t_SRAM)
17.73	281.78	v U627696/Y (NAND3xp33_ASAP7_75t_SRAM)
24.10	305.89	^ U627722/Y (NOR3xp33_ASAP7_75t_SRAM)
40.60	346.49	v U627840/Y (NAND3xp33_ASAP7_75t_SRAM)
32.15	378.63	^ U627841/Y (NOR2x1p5_ASAP7_75t_L)
120.53	499.16	v U630970/Y (OAI222xp33_ASAP7_75t_R)
63.28	562.44	v FE_OF12432_n472021/Y (BUFx3_ASAP7_75t_R)

```

61.46      623.90 ^ U630971/Y (OAI22xp33_ASAP7_75t_SRAM)
19.13      643.03 v U630973/Y (NAND2xp33_ASAP7_75t_SRAM)
15.10      658.14 ^ U441087/Y (OAI21xp33_ASAP7_75t_SRAM)
14.07      672.20 v U630974/Y (NOR2xp33_ASAP7_75t_SRAM)
15.69      687.90 ^ U630975/Y (NAND2xp33_ASAP7_75t_SRAM)
16.81      704.71 v U630978/Y (NAND3xp33_ASAP7_75t_SRAM)
0.07      704.78 v udcom0_dcom0_r_reg_data__1__u0/D
              (DFFHQx4_ASAP7_75t_SRAM)

```

```

704.78      data arrival time

```

```

0.00      0.00      clock clk (rise edge)
0.00      0.00      clock network delay (ideal)
0.00      0.00      clock reconvergence pessimism
0.00      0.00 ^ udcom0_dcom0_r_reg_data__1__u0/CLK
              (DFFHQx4_ASAP7_75t_SRAM)

```

```

3.34      3.34      library hold time
3.34      3.34      data required time

```

```

-----
3.34      data required time
-704.78   data arrival time
-----

```

```

701.44      slack (MET)

```

### B.3 Timing Report 3

Startpoint: uram0\_aram\_0\_u0\_ra\_reg\_8\_\_u0  
 (rising edge-triggered flip-flop clocked by clk)  
 Endpoint: udcom0\_dcom0\_r\_reg\_data\_\_1\_\_u0  
 (rising edge-triggered flip-flop clocked by clk)  
 Path Group: clk  
 Path Type: max

Delay	Time	Description
0.00	0.00	clock clk (rise edge)
0.00	0.00	clock network delay (ideal)
0.00	0.00	^ uram0_aram_0_u0_ra_reg_8__u0/CLK (DFFHQx4_ASAP7_75t_SRAM)
67.09	67.09	^ uram0_aram_0_u0_ra_reg_8__u0/Q (DFFHQx4_ASAP7_75t_SRAM)
58.38	125.47	v U481336/Y (NAND2x1_ASAP7_75t_R)
130.89	256.37	^ U365661/Y (NOR3xp33_ASAP7_75t_R)
91.05	347.41	^ FE_OFC6570_n483424/Y (BUFx3_ASAP7_75t_SRAM)
89.47	436.89	v U481355/Y (INVx1_ASAP7_75t_SRAM)
118.78	555.67	^ U365391/Y (NOR2xp33_ASAP7_75t_SRAM)
95.56	651.23	v U440252/Y (INVx1_ASAP7_75t_R)
114.18	765.40	v FE_OFC7876_n306938/Y (BUFx3_ASAP7_75t_R)
95.27	860.67	^ U481762/Y (NOR2xp33_ASAP7_75t_L)
53.21	913.88	v U628387/Y (NAND2xp33_ASAP7_75t_SRAM)

90.34	1004.22	^	U628389/Y	(NAND2xp5_ASAP7_75t_R)
103.76	1107.98	v	U628395/Y	(NOR3xp33_ASAP7_75t_SRAM)
42.14	1150.12	^	U628396/Y	(NAND3xp33_ASAP7_75t_SRAM)
21.78	1171.91	v	U439240/Y	(NOR3xp33_ASAP7_75t_SRAM)
38.49	1210.40	^	U628397/Y	(NAND2xp33_ASAP7_75t_SRAM)
48.94	1259.34	v	U628450/Y	(NOR3x1_ASAP7_75t_R)
75.66	1335.00	^	U628463/Y	(NAND3xp33_ASAP7_75t_R)
46.96	1381.96	v	FE_OFC7748_n469576/Y	(INVx1_ASAP7_75t_R)
78.16	1460.12	^	FE_OFC7749_n469576/Y	(INVx1_ASAP7_75t_L)
99.52	1559.64	v	U628672/Y	(NOR3xp33_ASAP7_75t_SRAM)
41.24	1600.88	^	U628673/Y	(NAND2x1_ASAP7_75t_SL)
98.53	1699.40	v	U630341/Y	(NOR3xp33_ASAP7_75t_SRAM)
105.68	1805.09	^	U630970/Y	(OAI222xp33_ASAP7_75t_R)
63.76	1868.85	^	FE_OFC12432_n472021/Y	(BUFx3_ASAP7_75t_R)
89.72	1958.58	v	U630971/Y	(OAI22xp33_ASAP7_75t_SRAM)
29.46	1988.03	^	U630973/Y	(NAND2xp33_ASAP7_75t_SRAM)
23.12	2011.15	v	U441087/Y	(OAI21xp33_ASAP7_75t_SRAM)
23.31	2034.45	^	U630974/Y	(NOR2xp33_ASAP7_75t_SRAM)
18.76	2053.21	v	U630975/Y	(NAND2xp33_ASAP7_75t_SRAM)
16.64	2069.85	^	U630978/Y	(NAND3xp33_ASAP7_75t_SRAM)
0.07	2069.92	^	udcom0_dcom0_r_reg_data__1__u0/D	(DFFHQx4_ASAP7_75t_SRAM)
	2069.92		data arrival time	
6000.00	6000.00		clock clk (rise edge)	
0.00	6000.00		clock network delay (ideal)	

0.00	6000.00	clock reconvergence pessimism
	6000.00	^ udcom0_dcom0_r_reg_data__1__u0/CLK (DFFHQx4_ASAP7_75t_SRAM)
-14.14	5985.86	library setup time
	5985.86	data required time
-----		
	5985.86	data required time
	-2069.92	data arrival time
-----		
	3915.94	slack (MET)

## B.4 Timing Report 4

Startpoint: uram0\_aram\_0\_u0\_ra\_reg\_8\_\_u0  
 (rising edge-triggered flip-flop clocked by clk)

Endpoint: udcom0\_dcom0\_r\_reg\_data\_\_1\_\_u0  
 (rising edge-triggered flip-flop clocked by clk)

Path Group: clk

Path Type: min

Delay	Time	Description
0.00	0.00	clock clk (rise edge)
0.00	0.00	clock network delay (ideal)
0.00	0.00	^ uram0_aram_0_u0_ra_reg_8__u0/CLK (DFFHQx4_ASAP7_75t_SRAM)
67.09	67.09	^ uram0_aram_0_u0_ra_reg_8__u0/Q (DFFHQx4_ASAP7_75t_SRAM)
37.39	104.48	v U481638/Y (NOR3xp33_ASAP7_75t_SRAM)
63.52	168.00	v FE_OF5352_n481128/Y (BUFx3_ASAP7_75t_R)
92.43	260.42	^ U627685/Y (NAND3xp33_ASAP7_75t_SRAM)
17.73	278.16	v U627696/Y (NAND3xp33_ASAP7_75t_SRAM)
24.10	302.26	^ U627722/Y (NOR3xp33_ASAP7_75t_SRAM)
40.60	342.86	v U627840/Y (NAND3xp33_ASAP7_75t_SRAM)
31.16	374.02	^ U627841/Y (NOR2x1p5_ASAP7_75t_L)
94.86	468.88	v U630970/Y (OAI222xp33_ASAP7_75t_R)
50.50	519.38	v FE_OF12432_n472021/Y (BUFx3_ASAP7_75t_R)

```

54.12      573.50 ^ U630971/Y (OAI22xp33_ASAP7_75t_SRAM)
19.13      592.63 v U630973/Y (NAND2xp33_ASAP7_75t_SRAM)
15.10      607.73 ^ U441087/Y (OAI21xp33_ASAP7_75t_SRAM)
14.07      621.80 v U630974/Y (NOR2xp33_ASAP7_75t_SRAM)
15.69      637.49 ^ U630975/Y (NAND2xp33_ASAP7_75t_SRAM)
16.81      654.31 v U630978/Y (NAND3xp33_ASAP7_75t_SRAM)
0.07      654.38 v udcom0_dcom0_r_reg_data__1__u0/D
              (DFFHQx4_ASAP7_75t_SRAM)

```

```

654.38      data arrival time

```

```

0.00      0.00      clock clk (rise edge)
0.00      0.00      clock network delay (ideal)
0.00      0.00      clock reconvergence pessimism
0.00      0.00 ^ udcom0_dcom0_r_reg_data__1__u0/CLK
              (DFFHQx4_ASAP7_75t_SRAM)

```

```

3.34      3.34      library hold time
3.34      3.34      data required time

```

```

-----
3.34      data required time
-654.38   data arrival time
-----

```

```

651.03     slack (MET)

```