2019-08-09

Development of an Accurate Clock Delay Model with Application in Clock Network Buffer Sizing

Farshidi, Ali

UNIVERSITY OF CALGARY

Development of an Accurate Clock Delay Model with Application in Clock Network Buffer Sizing

by

Ali Farshidi

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING

CALGARY, ALBERTA

AUGUST, 2019

Abstract

Clock network synthesis is an important stage of the Integrated Circuit (IC) design cycle. The performance of the IC highly depends on the clock network synthesis which makes this stage critical where accuracy is very important. In this thesis, a new delay model is proposed for clock networks that is capable of estimating clock signal delay with significantly improved accuracy in a relatively low runtime. This model is developed using least square fitting by employing data oriented training. The developed model is formulated in the form of posynomials which makes it a suitable option for application in geometric programming gate and clock network sizing optimization frameworks. The experimental results demonstrate the effectiveness of the proposed delay model in predicting the delay at the timing critical clock sinks in the clock network, i.e. sinks with minimum and maximum delays, and the estimated values are, on average, 20 ps closer than the Elmore values to the reference circuit simulator tool, ngspice. This is while the runtime of the proposed delay model is negligible compared to the ngspice simulations. This helps designers obtain accurate delay estimations in low runtime for quick optimization iterations. In addition, a clock network buffer sizing approach is developed which includes an objective function with geometric programming format considering two competing objectives, power consumption and clock skew. The clock slew and technology constraints are also integrated into this optimization problem. The clock network buffer sizing experiments show significant improvements compared to the initial clock networks in terms of clock skew, up to 183 ps, while the power consumption improves for all test cases, on average by 54%.

Acknowledgements

I would like to thank all people who helped me in production of this thesis. First, I would like to express my thanks to my supervisor, Dr. Behjat, for her suggestions in making this thesis. I also thank my co-supervisor, Dr. Rakai, for his comments and suggestions. I wish to thank all of my previous teachers who helped me a lot in my education. I would like to thank my family which is the most important factor of my success. I wish to thank my mom and dad who always guide me into the way of success. Finally, thanks to my lab-mates, Yangyang, Nima, Amin, Aysa, Emily, Stephanie, Erfan, Daniel, Upma, Amir, for helping me throughout my MSc studies.

Table of Contents

List of Figures and Illustrations

List of Tables

List of Symbols, Abbreviations and Nomenclature

Acronyms:

AAT	Actual Arrival Time
AO	Area Objective
AWE	Asymptotic Waveform Evaluation
C	Capacitor
CAD	Computer-Aided Design
CTS	Clock Tree Synthesis
EDA	Electronic Design Automation
fF	femtofarad
GP	Geometric Programming
IC	Integrated Circuit
ISPD	International Symposium on Physical Design
mm	millimeter
MO	Multi Objective
MSE	Mean Sequare Error
mW	milliWatt
nm	nanometer
pF	picoFarad
ps	picosecond
RAT	Required Arrival Time
RC	Resistor-Capacitor
RLC	Resistor-Inductor-Capacitor
s	second
SLP	Sequential Linear Program
s.t.	subject to
STA	Static Timing Analysis
VLSI	Very Large Scale Integration
μm	micrometer
μW	microWatt

Sub-scripts:

i	Index of clock sinks, inequality constraints, constant parameters
j	Index of clock sinks, equality constraints, monomial functions
k	Index of segments, buffers
n	Index of nodes
Scalars:	
a	A real number
AT_i	The arrival time of clock signal to the i_{th} clock sink
b	A real number
b_l	Length of a buffer
bufferdelay	The delay of clock signal through buffer
b_w	Width of a buffer
$C_{b_{input}}$	Capacitance of the buffer seen by the upstream buffer
$C_{b_{output}}$	Capacitance of the buffer seen by the downstream buffer
C_i	Capacitance i
$C_{k_{\mathrm{downstream}}}$	Downstream capacitance of resistance segment k
C_{sink}	Clock sink capacitance
C_{uw}	Unit wire capacitance per length (fF/nm)
C_w	Capacitance of the wire
D_f	Domain of function f
D_{max}	maximum of the sink delays
D_{min}	minimum of the sink delays
$\mathrm{Internaldelay}_{\mathrm{Buffer_k}}$	Internal delay of the clock buffer k
j_0	Number of monomials in a posynomial function
L_b	Length of the buffer
l_{min}	Minimum length of buffer
$\mathrm{LumpedRCDelay}_{\mathrm{BA}}$	Lumped RC delay from node A to node B
L_w	Length of the wire
o	The number of inequality constraints
p	The number of equality constraints
R_b	Output resistance of the buffer
R_k	Resistance of segment k
R_{uw}	Unit wire resistance per length (Ohm/nm)
R_w	Resistance of the wire
s_i	Clock sink i
$SinkDelay_{\min}$	The minimum value of the sink delays
Term_{B_1}	First term of buffer delay model
Term_{B_2}	Second term of buffer delay model
Term_{W_1}	First term of wire delay model
Term_{W_2}	Second term of wire delay model
t_{skew}	Target clock skew
t_{slew}	Target clock slew
W_b	Width of the buffer

wiredelay	The delay of clock signal through wire
w_{min}	Minimum width of buffer
y_i	A variable needed for transformation
α	Constant parameter
α_b	A coefficient
α_w	A coefficient
β_b	A coefficient
β_w	A coefficient
γ_1	Constant parameter
γ_2	Constant parameter
γ_3	Constant parameter
γ_4	Constant parameter
Sets:	
B	The set of all the buffers
K	The set of all the resistance segments
S	The set of all the sinks
X	A convex set
Vectors:	
$\mathbf{x}$	The vector of all buffer widths and lengths
Functions:	
$\mathrm{Area}(\mathbf{x})$	Total area of the buffers
$f(.)$	An objective function
$f_1(.)$	A continuous function
$f_2(.)$	A convex function
$f_3(.)$	A non-convex function
$f_{\mathrm{m}}(.)$	A monomial function
$f_{\mathrm{p}}(.)$	A posynomial function
$g_i(.)$	An inequality constraint
$h_j(.)$	An equality constraint
$k(.)$	A twice differentiable function
$m(.)$	A monomial function
$n(.)$	A function which is not monomial
$NodeDelay_i(.)$	Elmore delay at node i
$p(.)$	A posynomial function
$SinkDelay_i(.)$	The time it takes for the clock signal to be delivered to sink i
$SinkDelay_n^s(.)$	The delay from the output of the previous upstream clock buffer to the node n
$slew_n(.)$	The input clock slew at node n
Definitions:	
Clock network:	The clock paths connecting the clock source to the clock sinks
Clock network buffer sizing:	This physical design stage aims to determine the sizes of the clock buffers

Clock network synthesis:	Clock network will be designed during this stage
Clock network wire sizing:	This physical design stage aims to determine the sizes of the wires
Clock signal:	A signal that should be sent from clock source to the clock sinks
Clock sink:	Circuit synchronous components
Clock skew:	The maximum difference of the clock sink delays
Clock slew:	Time difference between 10% and 90% of maximum voltage of the clock signal and vice-versa
Clock tree:	A clock network without cycle in the clock paths
Delay:	The needed time for a clock signal to arrive to a node from another node
Floor planning:	The floor planning stage determines the exact sizes and locations of the partitions and pins
Gate sizing:	This physical design stage aims to determine the sizes of the gates
Partitioning:	This physical design stage aims to determine the partition sizes, numbers and number of the wires needed to connect the partitions
Placement:	The locations of the circuit cells are determined during this stage
Routing:	Paths of wires between the clock cells of the circuit will be determined during this stage
VLSI physical design:	A process that determines the physical attributes of the integrated circuit

Chapter 1

Introduction

The process of designing an Integrated Circuit (IC) is a large scale size and complex problem [49]. This process is referred to Very Large Scale Integration (VLSI) design. The IC design technology (i.e. VLSI) has been growing rapidly in the past few decades [40]. This has made the use of efficient IC design specific software unavoidable. Electronic Design Automation (EDA) refers to the industry that provides IC design software packages. Today, several major EDA companies, such as Cadence [1], Synopsys [7] and Mentor Graphics [4], with headquarters in Silicon Valley are providing software packages and support for IC designers. These software packages are designed to solve the VLSI design problems. VLSI design includes many sub-problems [49]. The design stages for VLSI circuits are shown in Fig 1.1 [49]. The first stages can be considered as high-level steps where the overall requirements of the IC should be determined. The later VLSI design stages are low-level steps of IC design and after fabrication stage and testing, the IC design process is completed [49].

Each design stage includes several sub-problems. The focus of this thesis is the VLSI physical design stage. Physical design is often divided into several sub-problems [49]:

- Partitioning
- Floor planning and placement

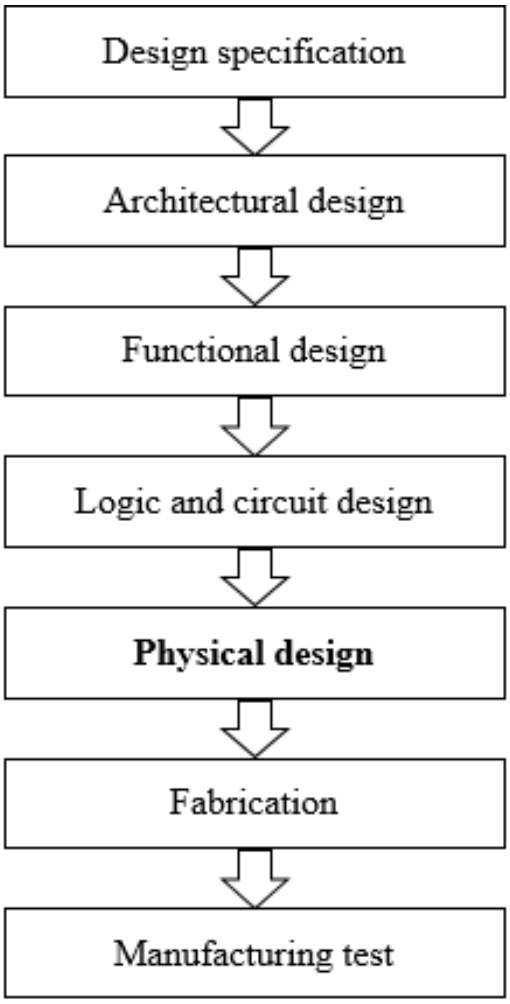

Figure 1.1: Main stages of the VLSI design process [49]

- **Clock network synthesis**
- Routing
- Timing closure

The output of the VLSI physical design is a circuit layout which includes the exact locations of circuit components (i.e. cells) and the paths of the wires (nets) that connect them.

The structure of the rest of this chapter is as follows: VLSI physical design flow stages including partitioning, floor planning, placement, clock network synthesis, routing and timing closure

are introduced briefly in Section 1.1. In Section 1.2, the motivations and contributions of the thesis are discussed. Finally, the layout of the rest of this thesis is presented in Section 1.3.

1.1 VLSI physical design flow

VLSI physical design flow is in a way similar to building a city. Suppose that an imaginary city contains billions of people and unplaced houses and roads that connect the houses. This city needs to be planned perfectly, so that it occupies the least space and uses the least of energy. The solver needs to design the city considering these objectives:

- The size of the city, i.e. area.
- The travel paths distances of the people to their destinations.
- The energy usage of the city, e.g. electricity and gas usage.

The solver is supposed to place all the houses and design roads from the houses to the destinations. This is an example of a complex optimization problem with different objectives and several constraints. Some of the objectives may even be competing objectives in a way that minimizing one may result in increasing another.

Similarly, in VLSI physical design flow, the locations of all the cells, e.g. flip-flops, gates and other circuit components, in the given IC area should be determined. Flip-flop is a component of IC that is able to store binary data as it has two stable states. The wire routing between the connected cells should be designed. These connectivities are given in the circuit description as a netlist. The main VLSI physical design flow objective functions are:

- The circuit area
- Total length of wires
- Total power consumption

- Timing constraints

In the following subsections, a brief explanation of each stage of the physical design flow is presented.

1.1.1 Partitioning

The input to the partitioning optimization problem is the output of the circuit design stage. The process of VLSI physical design starts with the partitioning stage. During the partitioning stage, the input circuit is divided into partitions (i.e. blocks). The partition sizes, numbers and number of wires needed to connect the partitions are important factors in this stage. The output of the partitioning stage includes a list of the partitions and the wire connections between them [49].

1.1.2 Floor planning and placement

During floor planning and placement, the partitions generated in the partitioning stage should be placed in the IC area. The wire tracks for the routing should be made. One of the important objectives that should be considered in this stage is minimizing the final IC area [49].

The floor planning stage determines the exact sizes and locations of the partitions and pins. Floor planning also designs net tracks for placement stage. Placement usually includes global and detailed placement sub-problems where the exact locations of the cells are determined. The placement's quality can significantly affect the quality of the routing stage [49].

1.1.3 Clock network synthesis

The locations of the clock sinks are determined during the placement stage. During the clock network synthesis stage, the clock paths from the clock source to the clock sinks should be routed [82]. In this stage, the goal is to design a clock network to send the clock signal from the clock source to all the synchronous circuit components (i.e. clock sinks) [82].

Ideally, the clock sinks should receive the clock signal at the same time. However, in most cases the clock signal delays at different clock sinks are not same. The maximum difference of the clock signal delay at all the clock sinks is called *clock skew*. Clock skew is one of the objectives of the VLSI physical design that should be minimized.

Clock network buffer insertion and sizing are other parts of the clock network synthesis stage that aim to minimize the clock skew [40]. In the clock network buffer insertion and sizing, the location and the size of the buffers along the clock paths will be determined, respectively [40].

Power consumption is another important objective in this stage [40]. Power consumption and clock skew are competing objectives. The output of clock network synthesis stage aims to find a balanced trade-off between clock skew and power consumption.

1.1.4 Routing

After solving the placement optimization problem, the circuit is needed to be routed. The routing stage is often solved in two stages, global and detailed routing [49]. In these stages, the important objective that needs to be minimized is the length of the connecting wires between the cells while keeping congestion under control [49].

1.1.5 Timing closure

The last stage of the VLSI physical design is to optimize all the timing constraints including hold time constraints [49]. Hold time constraints determine the time that the clock signal is needed to be stable at each synchronous IC components [49]. At the end of this stage, all the timing constraints should be satisfied. This stage includes Static Timing Analysis (STA) process which considers the worst-case scenario for all the gates in all the paths. In this process, the Required and Actual clock signal Arrival Time (i.e. RAT, AAT) and their difference (i.e. slack) will be calculated [49]. Positive slack at all the clock paths means that the main timing constraints are satisfied.

1.2 Research motivations and contributions

Clock network synthesis is an important stage in the VLSI physical design flow. The contributions of this thesis are aimed at improving the outcomes of the clock network synthesis stage. Generally, an IC needs a clock network to distribute a clock signal from the clock source to the synchronous circuit components (i.e. clock sinks). This clock signal is required to be received at all the clock sinks at the same time. However, as most engineering optimization problems, this ideal situation cannot be always achieved in practice. The clock skew, i.e. the maximum difference of the arrival time of the clock signal to the clock sinks, is used as a key factor in this optimization problem. One of the commonly used methods to improve the quality of the clock network design is clock network buffer insertion and sizing which can reduce the clock skew and power consumption.

Based on the clock network structure design methods, clock networks can be divided into two types of clock network categories: clock meshes and clock trees [82]. Clock tree is an special type of clock network where no cycle is allowed for the clock signal in the clock network [82]. In other words, the clock sinks are the end points of the clock signal sent from the clock source [82]. However, clock meshes can include cycles in the clock network [49]. As clock trees are often more efficient but harder to design, in this thesis, the focus is on the clock trees.

The hypothesis of this thesis is that the accuracy of delay estimation can be improved by fitting new posynomial models using the data obtained based on a sample circuit consisting of a source, sink, buffer and wire. This model can then be used to reduce the power consumption during buffer sizing. In this thesis, in order to improve the IC performance, a new mathematical delay model for clock network synthesis is proposed. This model can efficiently predict clock signal delay with a higher accuracy compared to the popular Elmore delay model [25]. This model can be applied in clock network buffer sizing tools. The accuracy of the prediction of the clock signal delay can improve the quality of the sizing algorithms. In addition, a new and efficient clock network buffer sizing framework is proposed which shows improvements in terms of power consumption, clock

skew and runtime compared to the existing clock network buffer sizing methods in the literature.

The main contributions of this thesis are:

- Proposing an accurate clock signal delay model and improving the sink delay prediction significantly.
- Improvement in the minimum and maximum sink delay estimations over the Elmore delay model used in [82].
- GP compatibility of the proposed clock signal delay model.
- Low runtime compared to the *ngspice* circuit simulator.
- Applying the proposed clock signal delay model to a clock network buffer sizing algorithm which results in improvements in the power consumption and the clock skew.
- Developing a new efficient clock network buffer sizing method optimizing power consumption and clock skew while meeting the slew constraints.
- Significant power consumption, clock skew and runtime improvement using the proposed clock network buffer sizing method compared to the existing methods in the literature.

1.3 Thesis structure

The structure of the rest of this thesis is as follows.

Chapter 2: In this chapter, an extended background on clock network synthesis and clock network delay modeling is presented. Different clock network delay models are analyzed and compared in terms of accuracy and modeling complexity.

Chapter 3: In this chapter, convex and Geometric Programming (GP) optimization problems are explained. The clock network buffer sizing literature is reviewed and a clock network buffer sizing framework modeled as a GP optimization problem is discussed.

Chapter 4: The first set of contributions of this thesis is presented in Chapter 4. First, the development of a new clock network delay model is presented. Then, the proposed delay model is applied to a clock network buffer sizing application. The mean square error of the estimated values from the proposed delay model is improved compared to the Elmore delay model. The proposed clock network delay model is also compared to other existing delay models in the literature in terms of accuracy, complexity and modeling effort.

Chapter 5: The second set of contributions of this thesis is presented in Chapter 5, where a new multi-objective formulation based on [41, 82] is developed to simultaneously optimize both clock skew and power objectives during clock network buffer sizing. The proposed formulation can significantly improve the power consumption and clock skew while meeting technology and clock slew constraints. Furthermore, the formulation has relatively low complexity which improves the runtime.

Chapter 6: The last chapter concludes this thesis with presenting the summary, conclusions and future work.

Chapter 2

Clock Network Delay Modeling Background

2.1 Introduction

In an Integrated Circuit (IC), a clock network is synthesized to distribute a clock signal throughout the circuit to synchronize all the computations across the chip [66, 69]. In high-performance clock networks, the insertion delays from the clock source to the registers critically affect the circuit performance and the maximum frequency the circuit can operate at [96]. This requires a highly accurate mathematical model to estimate the insertion delays during optimization. However, many optimization formulations, e.g. [41, 72, 82, 104], still use Elmore delay [25]. Elmore delay has been shown to be inaccurate for the modern technology nodes [13, 52, 55, 70, 78, 103].

Due to lack of more accurate delay models, most Electronic Design Automation (EDA) tools, e.g. [1, 4, 7] are forced to employ timing simulators that add significant runtime overhead. In addition, such timing simulators mostly cannot be efficiently used as part of a convex mathematical optimization framework which could enable more runtime efficient optimization algorithms.

In this chapter, an introduction to delay modeling and its critical role in IC physical design is

given. The rest of this chapter is organized as follows: The basics of clock network synthesis, its challenging design objectives and constraints are discussed in Section 2.2. Then, in Section 2.3, different clock network delay models existing in the literature are discussed. Finally, Section 2.4 summarizes the chapter.

2.2 Clock Network Synthesis Design Objectives and Constraints

Clock signal delay critically affects the performance of an IC, especially its operating frequency and power consumption [96]. According to Moore's law [73], every two years, the number of transistors per unit of area in ICs doubles. The larger transistor density leads to more critical timing path that makes optimizing timing metrics a challenging design factor. On the other hand, increasing clock frequency and decreasing transistor channel sizes have made estimating the clock performance more challenging [96].

In Fig. 2.1, the clock network ispd2009-22 is presented. This network is then used to explain the basics of clock networks. The physical size of this circuit is $5mm \times 5mm$. In Fig. 2.1, si represents the sink i. All the sinks are connected to the clock source $s0$ using vertical and horizontal wires. The clock signal delay of si ($SinkDelay_i$) is defined as the time it takes for the clock signal to be delivered to si ($i > 0$) from the clock source.

Clock skew is the maximum difference of the clock sink delays [40]. It can also be defined as the difference of the maximum and minimum delays of all the sinks:

$$Skew = max\{SinkDelay_i(\mathbf{x}) - SinkDelay_j(\mathbf{x})\}\ , \forall i, j \in S \tag{2.1}$$

where S represents the set of all the sinks and $\mathbf{x}$ is the vector of all buffer widths and lengths (b_w, b_l). The clock skew concept is shown in Fig. 2.2. The top waveform in Fig. 2.2, which is shown with green color, is the clock signal waveform at the clock source. The arrived clock signal

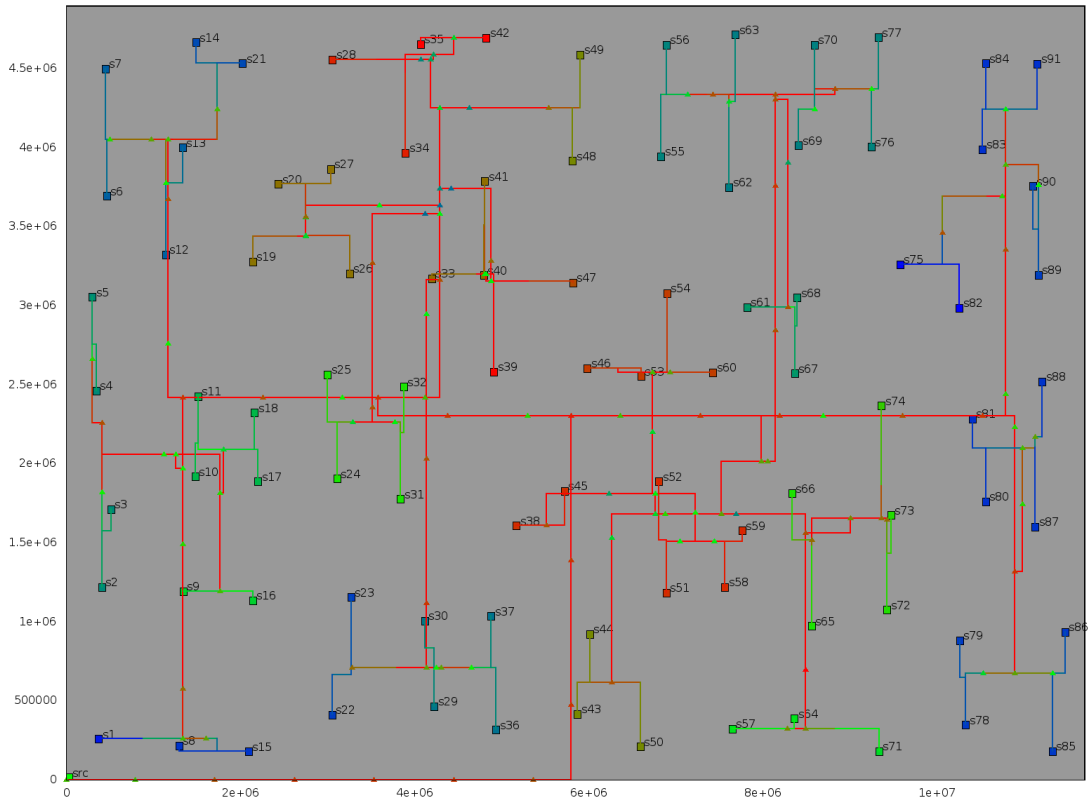

Figure 2.1: An example clock network, ISPD2009, Circuit 22. The color of the sinks represents the relative value of their clock signal delay where red/green/blue means low/medium/high delay. Also, blue/green/red color of the buffers (shown by triangles) represents a low/medium/high input clock slew value.

at each sink includes a specific amount of delay. The red and blue waveforms in the same figure show the arrived clock signals at the sink with minimum and maximum delay, respectively. The minimum delay, maximum delay and their difference (also known as clock skew) are shown in Fig. 2.2.

In Fig. 2.1, the value of the sink delay of the red/blue sinks are low/high and close to the minimum/maximum of sink delays. Therefore, the red and blue sinks are considered to be more critical. The value of the sink delay of the green sinks are not close to the minimum or maximum of the sink delays. Then, in this design stage, the green sinks are less timing critical and their delays may not significantly affect the clock skew.

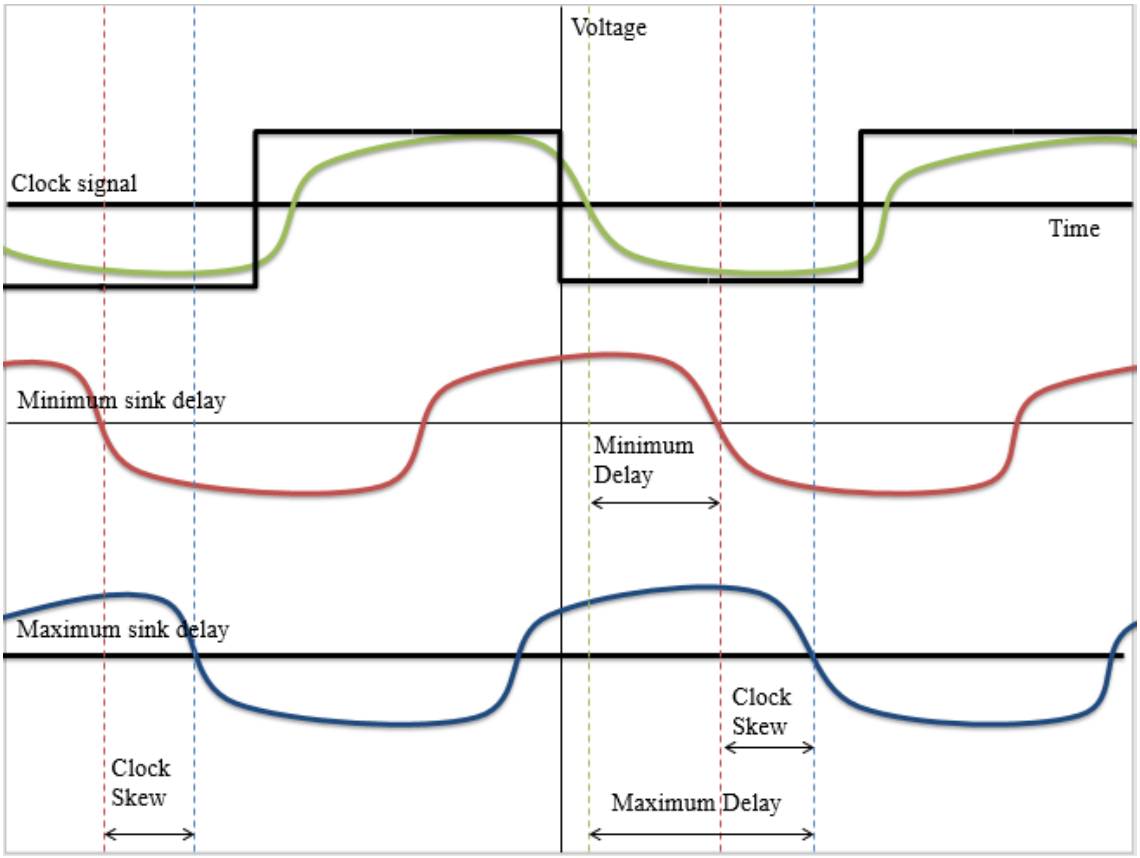

Figure 2.2: An example of clock delay and clock skew.

Although there are a few exceptions such as useful skew [27, 32, 37, 59, 60] where designers prefer to have a special non-zero skew value rather than zero skew, mostly the ideal value of clock skew is zero [22, 30, 39, 46, 56, 99, 100, 101]. This is achieved if all the sinks receive the clock signal at the same time. However, considering that achieving zero skew is challenging, IC designers try to minimize the clock skew. As much as clock skew is minimized, the clock network will be more flexible for working in higher operating frequencies.

Another important feature is the shape of the waveform of the clock signal which affects the IC performance [62]. Ideally, clock signal waveform needs to be as sharp as possible such as the black waveform in Fig. 2.2. However, the green waveform in the same figure is often the real clock signal waveform in clock networks which includes transition time in the rise and fall edges.

In Fig. 2.2, in the green clock signal waveform, the time difference between 10% and 90% of the maximum voltage and vice-versa is defined as clock slew. Clock slew is the widely used metric to evaluate the quality of the waveform of the clock signal. In IC design, usually clock slew needs to meet some upper bound constraints [2, 3, 40, 41, 66, 82, 98] because the quality of the clock signal affects the reliability of the circuit performance.

Buffer insertion is an important and widely used method to reduce clock skew and maintain the quality of the clock signal waveform [40, 41, 82, 97]. In Fig. 2.1, the triangles on the wires represent the buffers. Buffers can be sized during clock network buffer sizing optimization [40, 82]. The red color on the triangles represents a low input clock slew value and the blue ones show high input clock slew values. Therefore, the blue buffers are the critical ones in terms of clock slew which IC designers need to focus on to meet the clock slew constraints while the red and green ones are less critical.

Clock signal delays as well as other timing metrics, e.g. clock skew and clock slew, must be optimized during synthesizing and sizing the clock networks. This requires an accurate mathematical model to estimate the clock signal delay during optimization. However, most Electronic Design Automation (EDA) tools are forced to employ either timing simulators and very complex delay models that are accurate and precise but add significant runtime overhead [110], or the existing runtime efficient delay models that are not accurate for the modern technology nodes [25, 109]. Therefore, developing an accurate yet low complexity delay model can improve the quality of the clock network sizing results and decrease the turnaround time.

2.3 Relevant Works in Clock Delay Model Development

In modern technology nodes, having an accurate delay model is critical [78]. This is even more critical for timing clock trees where every pico-second can directly affect the circuit performance. The reason is that the longest path delay and the clock skew, which include wire delay and gate

delay (e.g. buffer delay), are the main factors limiting the maximum operating clock frequency [49, 96].

In IC design, the estimation of the gate delay and the delay of the wires connecting the gates is required to perform optimization during the clock tree synthesis (CTS). CTS calculates the delay in each node of the clock tree during optimization. In addition, STA methods generally analyze the timing closure by estimating the gate and wire delays and adding them together [110].

Many research works have addressed STA [9, 28, 29, 34, 50, 89, 105]. These works mainly focused on finding better chip or transistor level delay [57, 63, 64, 65, 80, 87, 91, 102] or noise estimation models [21, 64, 95].

Gate and wire delay estimation methods are proposed with various results in terms of accuracy and runtime [110]. Generally, there is a trade-off between accuracy and runtime and IC designers can choose one of the methods to use based on their accuracy and runtime requirements. Some of the most important methods are discussed below.

2.3.1 Lumped delay models

Elmore delay model

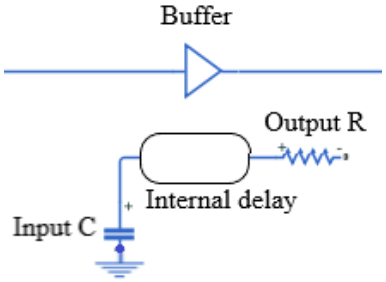

Figure 2.3: Elmore buffer delay model [42, 93]

In 1948, Elmore [25] proposed a mathematical model using a Laplace form function that has been used as a delay model to estimate clock signal delay in the clock network. Today, the Elmore

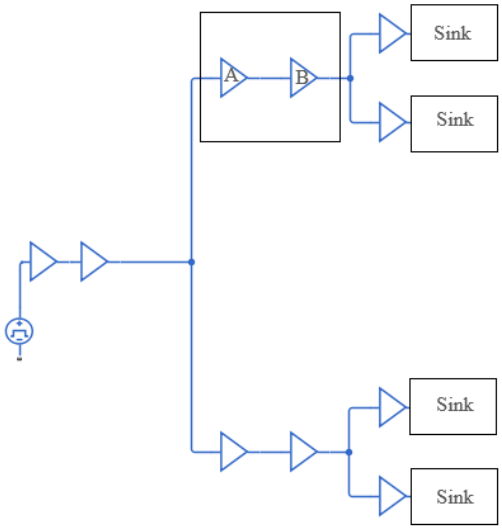

Figure 2.4: A sample clock tree network

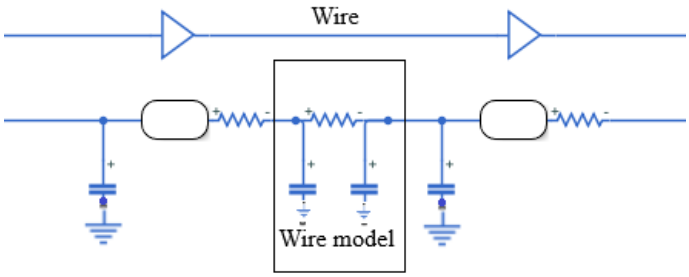

Figure 2.5: Elmore wire delay model [93]

delay model is considered to be inaccurate, especially for estimating gate delays [45], because this method only considers resistors and capacitors of the RC network and ignores the clock slew and inductance's effects [78]. In the Elmore delay model, buffers are modeled as an output resistance and an input capacitance as shown in Fig. 2.3. Then, the value of the buffer delay estimated using the Elmore delay model does not depend on the input clock slew and circuit's inductance which is not an accurate assumption for modern technology nodes.

Mathematically, Elmore delay model ignores the second and higher moments of the impulse

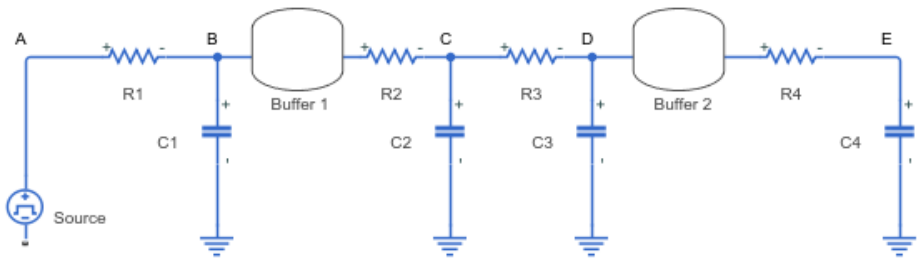

Figure 2.6: RC network using Elmore delay model

response and approximates the first moment as the delay. However, it is shown in [45] that at least the second and third moments should be considered as well to achieve a reasonably accurate delay estimation.

In [25, 45], the calculation of the first moment of the impulse response in an RC network is discussed. In the Elmore delay model, buffers are considered to be isolating the downstream and upstream capacitances [42]. In addition, to calculate the Elmore wire delay, the wire is modeled as a resistor and two capacitors while the value of each capacitor is half of the capacitance of the wire. For example, in Fig. 2.5 the RC equivalent of the two connected buffers, Buffer A and Buffer B in Fig. 2.4, is shown.

The Elmore delay model for an RC network such as the one in Fig. 2.6 can be defined as [103]:

$$\mathrm{NodeDelay_i} = \sum_{k \in K} R_k \times C_{k_{\mathrm{downstream}}} \tag{2.2}$$

where:

$\mathrm{NodeDelay_i}$: the Elmore delay at node i

K: the set of all the resistance segments in the path from source to the node i

R_k: the value of resistance segment k

$C_{k_{\mathrm{downstream}}}$: the downstream capacitance of resistance segment k

For example, in Fig. 2.6, the delay from the clock source to node E can be calculated by adding up the delay values from the clock source shown by node A to node B, node B to node C, node C to node D and node D to node E together. All of the mentioned delay values are calculated based on the Elmore delay as below:

$$
\begin{aligned}
&\mathrm{ElmoreDelay}_{\mathrm{AB}} = \mathrm{R}_1 \times \mathrm{C}_{\mathrm{R1}_{\mathrm{downstream}}} \\
&C_{R1_{\mathrm{downstream}}} = C_1 \\
&\mathrm{ElmoreDelay}_{\mathrm{BC}} = \mathrm{R}_2 \times \mathrm{C}_{\mathrm{R2}_{\mathrm{downstream}}} + \mathrm{Internaldelay}_{\mathrm{Buffer}_1} \\
&C_{R2_{\mathrm{downstream}}} = C_2 + C_3 \\
&\mathrm{ElmoreDelay}_{\mathrm{CD}} = \mathrm{R}_3 \times \mathrm{C}_{\mathrm{R3}_{\mathrm{downstream}}} \\
&C_{R3_{\mathrm{downstream}}} = C_3 \\
&\mathrm{ElmoreDelay}_{\mathrm{DE}} = \mathrm{R}_4 \times \mathrm{C}_{\mathrm{R4}_{\mathrm{downstream}}} + \mathrm{Internaldelay}_{\mathrm{Buffer}_2} \\
&C_{R4_{\mathrm{downstream}}} = C_4
\end{aligned}
\tag{2.3}
$$

$C_{R_{1\mathrm{downstream}}}$ only includes C_1 because Buffer 1 isolates the rest of the RC network. The same applies to any buffers in the RC network.

Delay bounding technique

Approximately 30 years after the introduction of the Elmore model, bounding clock signal delay [75] was developed and shown to be more accurate than Elmore. This model uses a closed form formulation to find the upper and lower bounds of the clock signal delay. Using this method enabled the use of the maximum clock signal delay and the voltage threshold to predict whether a clock network is fast enough or not. This model was further improved in terms of delay bounds in [88].

In [45], it is shown that the Elmore delay model can be considered as an upper bound for half of the delay in an RC network. A lower bound for half of the delay is also determined. Although, these methods especially Elmore delay model, were easy to calculate and used widely in the IC

physical design area [18, 23, 38, 41, 72, 82, 83, 84, 94, 104, 111], these models are not accurate enough for modern technology nodes anymore [13, 18, 52, 55, 70, 72, 78, 103]. The reason of this inaccuracy is the advancements in the fabrication of the nano-technology design and the lower technology nodes [31, 65, 87].

Lumped capacitance (Lumped C) delay model

In [109], four different lumped models are discussed that were previously used widely to calculate the delay. The most simple lumped delay model is Lumped C. In this model, all the extracted capacitances will be simply summed. Although the runtime of Elmore delay model is low, Lumped C model is still faster but less accurate [110].In this wire delay model, the wire resistance and inductance are ignored. Then, the drawback of this method is that compared to Elmore delay model, the accuracy is lower. However, it computes the delay faster. An example of the Lumped C delay model is shown in Fig. 2.7.

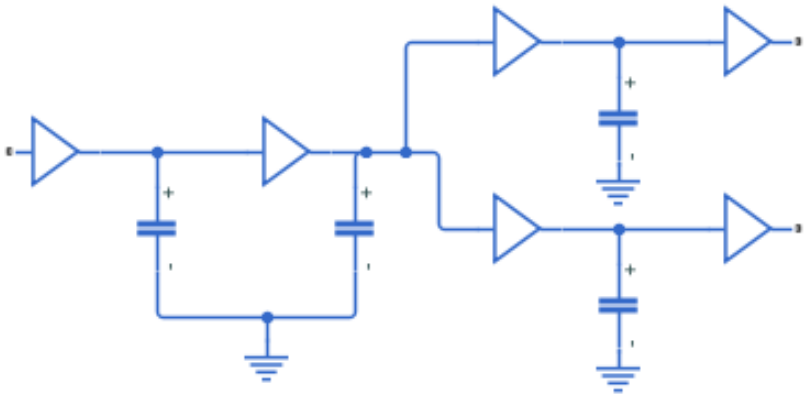

Figure 2.7: An RC network modeled using Lumped C delay model

In addition, an algorithm is proposed in [109] that estimates the delay and output clock slew using an equivalent resistance instead of gates. The value of the effective capacitance between two

buffers can be estimated in this approach. Then, it can determine whether the Lumped C model is accurate enough for that specific RC network or not.

Lumped RC delay model

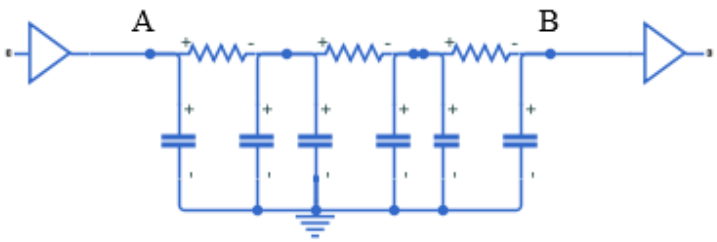

Figure 2.8: An RC network modeled using Lumped RC delay model

In the Lumped RC model, similar to Elmore delay model, the effects of the resistances and capacitances of the wires on the delay are taken into account. This model is more accurate than the Lumped C delay model but still less accurate than the Elmore delay model. The runtime of the Elmore delay model is a bit more but comparable. The Lumped RC delay value for the network in Fig. 2.8 is given in the following:

$$LumpedRCDelay_{BA} = \alpha \times C_{A_{\text{downstream}}} \times (\textstyle\sum_{k \in K} R_k) \tag{2.4}$$

where:

$\text{LumpedRCDelay}_{\text{BA}}$: the Lumped RC delay from node A to node B

α: a constant value (0.69) in the path from source to the node i [78]

$C_{A_{\text{downstream}}}$: The downstream capacitance at node A

Since the Elmore and Lumped RC delay models consider the resistances and capacitances but ignore the inductance, they have low runtime, they are used more frequently in industry.

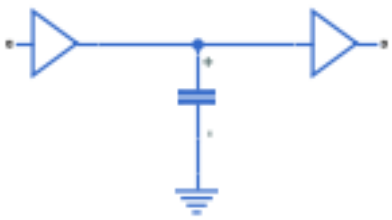

Figure 2.9: The RC equivalent of the network in Fig. 2.8 using Lumped C delay model

Single Pi delay model

In this model, the whole RC network between the buffers is modeled as a resistor and two capacitors. For example, the RC equivalent of the RC network in Fig. 2.8 using the single Pi model is given in Fig. 2.10. This model adds a resistor to the Lumped C model given in Fig. 2.9, while it still ignores the inductance. This model is computationally fast, but not accurate enough for industry applications [109].

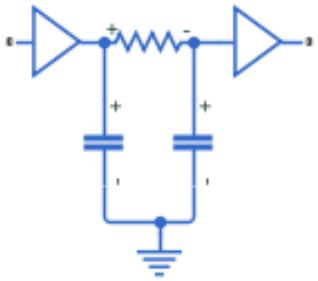

Figure 2.10: The RC equivalent of the network in Fig. 2.8 using single Pi delay model

Lumped RLC delay model

This model is very complex and results in high computation runtime. However, its accuracy is higher than the single Pi, Lumped C, Lumped RC and Elmore delay models since it considers the

resistance, capacitance and inductance effects. This model is demonstrated in Fig 2.11. It was not successful commercially [109], since it is time consuming and inefficient for implementation in the IC physical design tools for delay calculation.

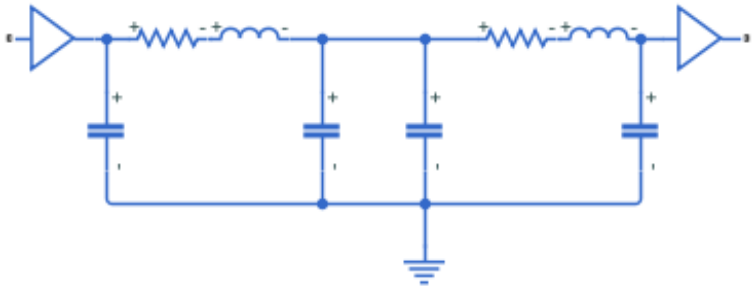

Figure 2.11: An RLC network using Lumped RC delay model

Summary of lumped delay models

Despite of Elmore delay's inaccuracy in delay estimation, it is still the most used lumped delay model in commercial tools [109]. The reason for widely using Elmore delay model is that it is computationally inexpensive. For example, although in [72], it is shown that the Elmore delay model is inaccurate, this model is still used for estimating the actual and required signal arrival times. This is because the model introduces the same amount of error in the estimated values, hence, the errors cancel out when comparing actual and required signal arrival times. However, the shortcoming of this method is the timing violations it leaves after optimization. Therefore, even though this method can be used for simple circuits, today's more complex circuits need more accurate yet computationally inexpensive delay models.

2.3.2 Look-up table methods

Several research works have been conducted to find more accurate delay models than the Elmore model [11, 13, 14, 15, 16, 51, 53, 54, 55, 67, 68, 70, 86, 103, 109]. One of the popular delay estimation methods, especially in the industry, is using a look-up table. Generally, the 2-dimensional look-up table methods take an output capacitance load of the gate and the input clock slew as the inputs to the look-up table and the outputs are the gate delay and output clock slew [110]. For instance, a look-up table method considering different input clock slew values is proposed in [67] for delay calculations.

In [26] and [109], a two-step approach is discussed and compared to the other popular existing delay and clock slew estimation methods. In the two-step approach, the gates and the RC network are separated and the delay of the RC network is found using a lumped delay model such as Elmore delay model. The delay of the gates is found using look-up tables. This method is commercially successful and is used by many STA tools [109]. However, if higher accuracy is needed, this technique might not be sufficient. This is due to dividing the delay modeling problem into two sub-problems which can increase the error [109].

Although the look-up table method is more accurate than Elmore delay model, it needs designing a complex table with many different input clock slew values which is time-consuming [53]. The second disadvantage of this method is that it is expensive in terms of memory usage [58]. As industry has progressed to the nanometer fabrication era, the gate delays and wire delays are more accurate if calculated by an indeterministic delay model [110].

2.3.3 K-factor models

One of the simple delay estimation methods is the K-factor model [110]. This model depends on the input clock slew of the gate and its output capacitance [36, 44]. In this method, the gate delay and output clock slew values are approximately considered as K-factor equations. The accuracy

of this method is significantly better than the Elmore delay because the fitted equations are made using SPICE simulation results [44]. In [44], the K-factor method is used to estimate the gate (i.e. buffer) delays. They use a technique originally proposed in [79] which is then improved in [35] to calculate the output capacitance. The results of this method is more accurate than the Elmore delay method. However, in [36], it is shown that the drawback of K-factor method is that the accuracy of the delay estimation is proportional to the value of the output capacitance, i.e. the smaller the capacitance is, the lower the accuracy will be.

2.3.4 Moment matching methods

As mentioned in section 2.3.1, Elmore delay model approximates the first moment of the impulse response as the delay value. However, moment matching models consider the higher moments as well. The delay estimations using these models are more accurate than the Elmore delay model. The first moment matching model, which is originally proposed in [76], is a technique called Asymptotic Waveform Evaluation (AWE) [110]. PRIMA [74] is a more recent moment matching model.

Moment matching techniques are improved and used in many other works. For example, in [86], a method is developed using the AWE technique to estimate the clock delay. The accuracy of the delay estimation using this method is high and close to the SPICE results. Although the moment matching models are more accurate than the Elmore delay model, these models are more time-consuming due to their complexity [110]. For instance, the method used in [86] needs solving a nonlinear and non-convex problem. Therefore, these models are inefficient to use in IC physical design computer-aided design (CAD) tools for optimization [52]. Then, the demands for more accurate but convex delay estimation models have been growing.

2.3.5 Closed form methods

The first and most popular closed form delay model is the Elmore delay model. The main reason for the popularity of the Elmore delay model compared to the other methods such as moment matching is that the Elmore delay model can be expressed as a closed form function based on the resistors and capacitors of the RC network [52, 88]. Being closed form is one of the main reasons that the Elmore delay model is computationally inexpensive, fast to calculate, and commercially successful.

Several other works [11, 15, 16, 52, 54] have focused on finding a closed form function for delay modeling to address the lack of accuracy of the Elmore delay model with comparable runtime. In [11], an interconnect delay model is proposed by fitting the Elmore delay model function to the SPICE results. The runtime of this model is close to the Elmore delay model and its accuracy is improved. Also, they use this delay model for wire sizing to show the efficiency. However, this model is more applicable for wire sizing applications and not gate sizing, because it is modeled and trained only based on the widths and lengths of the wires and ignores the gates. Then, it is not accurate to be used in other applications such as buffer and gate sizing where the width and length of the buffers and gates are the more important factors.

In [15] and [16], a closed form formulation for delay and clock slew estimation is proposed. This work has matched the moments of the RC network to the log-normal distribution. It also uses a technique called PERI from [53] and [54] to show that the method can be used for either a step input signal or a ramp input signal.

The benefit of the closed form delay models is that they are better applicable to the IC physical design simulator and optimizer tools. In Chapter 4, a new accurate closed form delay model is proposed that is compatible with geometric programming. This means that the proposed delay model is convex and is applicable to be used in clock network buffer sizing applications such as [40, 41, 81, 82] to find the global minimum solution.

2.3.6 SPICE methods

Any component in an IC may affect the overall value of the wire delay or gate delay [110]. SPICE simulation is the most accurate method that considers all the sources that affect the delay values. SPICE models can approximate accurate delay values. However, in [110], it is shown that these methods have the highest runtime and are the most expensive computationally. Many industry tools opt to use accurate timing simulators [6] to evaluate the clock delays but these simulators are relatively slow [89] and cannot be used in a global convex optimization framework.

2.3.7 Summary of the delay models

In this section, a summary of the delay models discussed in this chapter is given. In Table 2.1, the advantages and disadvantages of these delay models are compared. In this table:

- The first column clusters closed and non-closed delay models separately.
- The name of delay models and the most popular references for them are given in second column.
- Third column shows a comparison of the delay models in terms of accuracy.
- Fourth column represents whether the mathematical formulation of each delay model has high complexity or not.
- The modeling effort for each model is evaluated based on the runtime in the fifth column.
- Compatibility to the modern nano-technology is critical for fabrication. The sixth column evaluates each delay model's robustness for fabrication.
- The clock network sizing problems mostly use a delay model to predict clock delay at each node of the clock network tree accurately. The global optimum solution of these optimization problems is not guaranteed to be achieved if the problems are non-convex. Then, convexity

of delay model used in clock network sizing problem is effective. The seventh column represents the compatibility of each delay model with convex optimization.

- Finally in the eighth column, the popularity of these delay models in the industry is compared based on the information and data available in the literature.

Table 2.1: Comparison of the existing delay models in the literature. This table is a summary of the delay models presented in section 2.3. The comparison is based on the information and data available in the literature.

	Delay model	High accuracy	Low complexity	Modeling effort	Robust for fabrication	Applicable to convex opt	Commercially popular
Not closed	Lumped RLC [109]	✓	×	×	✓	×	×
	Look-up table [67]	✓	✓	✓	×	×	✓
	Moment Matching (PRIMA) [74]	✓	×	×	✓	×	×
	SPICE [6]	✓	×	×	✓	×	✓
Closed form	Elmore [25]	×	✓	✓	×	✓	✓
	Fitted Elmore Delay [11]	✓	✓	✓	×	✓	×
	PERI [16]	✓	✓	✓	×	✓	×
	Delay bounding [45, 88]	×	✓	✓	×	✓	×
	Lumped C [109]	×	✓	✓	×	✓	×
	Single Pi [109]	×	✓	✓	×	✓	×
	Lumped RC [18]	×	✓	✓	×	✓	✓

2.4 Summary

In this chapter, a comprehensive background on delay modeling for clock networks is given. The delay modeling problem and its challenges are explained. Different delay models in the literature are reviewed and their shortcomings and advantages are explained. The comparison of all the delay models based on critical design and optimization criteria is also presented in this Chapter.

Chapter 3

Background on Optimization Models for Clock Network Buffer Sizing

3.1 Introduction

Optimization methods are used in many different research fields and their importance is growing everyday. They help solve engineering problems while saving cost and runtime. The main reasons for their importance can be the large size of the problems encountered today as well as lack of good models for the complicated problems.

The optimization methods are widely used in EDA industry. VLSI stands for the very large scale integration which indicates that the sizes of the optimization problems being solved are large. This is because of the billions of components that are integrated into an IC. The high price of the resources used in IC fabrication is another reason that makes optimization methods important. Mostly, the VLSI physical design problems are divided into many sub-problems and many of these sub-problems need to use optimization methods to be efficiently solved.

In the rest of the chapter, first in Section 3.2, the convexity and geometric programming (GP) are introduced. Then, different clock network buffer sizing methods are analyzed in Section 3.3.

In Section 3.4, GP is used to convert the clock network buffer sizing problem to a convex problem. Finally, a summary of this chapter is given in Section 3.5.

3.2 General Optimization Terminology and Definitions

Most optimization problems can be categorized based on whether they are constrained or unconstrained. Mathematically, an unconstrained optimization problem can be defined as:

$$\min \quad f(\mathbf{x}) \tag{3.1}$$

where $f(\mathbf{x})$ is the objective function and $\mathbf{x}$ is a vector of variables. The structure of a constrained optimization problem can be written as:

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0 \ , i \in 1, 2, ..., o \\ & h_j(\mathbf{x}) = 0 \ , j \in 1, 2, ..., p \end{aligned} \tag{3.2}$$

In (3.2), $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ are the inequality and equality constraints, respectively. Parameters o and p are the number of inequality and equality constraints, respectively.

Unconstrained optimization problems can be solved using gradient method, Newton method or Newton-like method if they are continuous and differentiable [43, 71]. Constrained optimization problems need to find a solution in the feasible region. Feasible region of a constrained optimization problem is a region that includes all the feasible solution points which satisfy the constraints of the optimization problem. Some constrained optimization problems can be transformed to unconstrained problems using Lagrangian relaxation [43].

3.2.1 Convex set

A convex set is a set of points where the line that connects the points is entirely within the set. Convex sets are important in defining optimization problems. It is shown in [24] that mathematically, X is a convex set if for any real numbers $a, b \geq 0$;

$$\begin{aligned} &a \times x_1 + b \times x_2 = x_k \ , \forall x_1, x_2 \in X \\ &a + b = 1; \end{aligned} \tag{3.3}$$

then, $x_k \in X$. In Fig. 3.1, the difference between convex and non-convex sets is shown in $\mathbb{R}^2$.

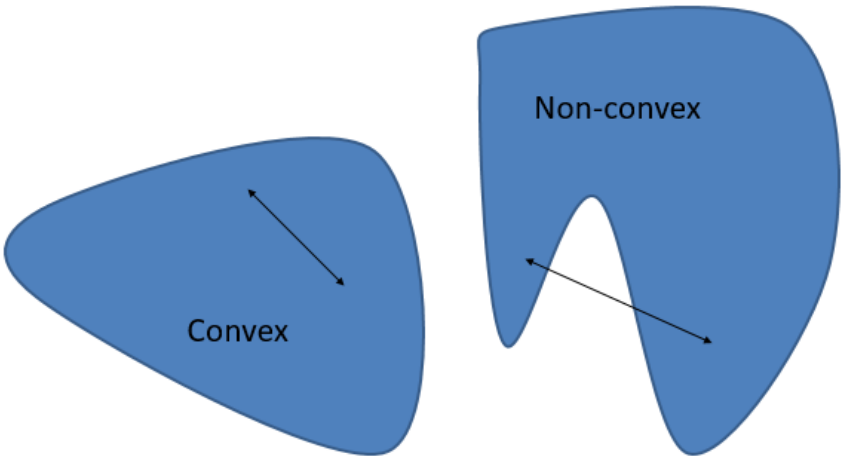

Figure 3.1: An illustration of a convex set versus a non-convex set

3.2.2 Convex function

Similarly, a continuous function $f_1(x)$ is convex on a domain of convex set X if for any $a, b \geq 0$ [24]:

$$\begin{aligned} &f_1(a \times x_i + b \times x_j) \leq a \times f_1(x_i) + b \times f_1(x_j) \ , \forall x_i, x_j \in X \\ &a + b = 1; \end{aligned} \tag{3.4}$$

For a twice differentiable function, convexity can be determined by the second derivative of the function. A function $k(x)$ is convex on a domain of convex set if the second derivative of the function $k(x)$ is:

- available in its domain.
- positive semidefinite.

For example, the convexity of e^x is proven in (3.5):

$$\begin{aligned} &k(x) = e^x \\ &k''(x) = e^x \succeq 0 \end{aligned} \tag{3.5}$$

where $\succeq$ represents the positive semidefinite characteristic.

For simple nonlinear functions, a common way to understand convexity or determine the non-convexity is to visualize the function. If there is a straight joining line between any two points of the graph of a function that intersects with the graph in at least one other point, that function is non-convex. Fig. 3.2 represents a quadratic function given in (3.6) as an example of a convex function while function $f_3(x)$ given in (3.7) is not convex. Fig. 3.3 shows $f_3(x)$ as a non-convex example.

$$\begin{aligned} &f_2(x) = x^2 \\ &f_2''(x) = 2 \succeq 0 \end{aligned} \tag{3.6}$$

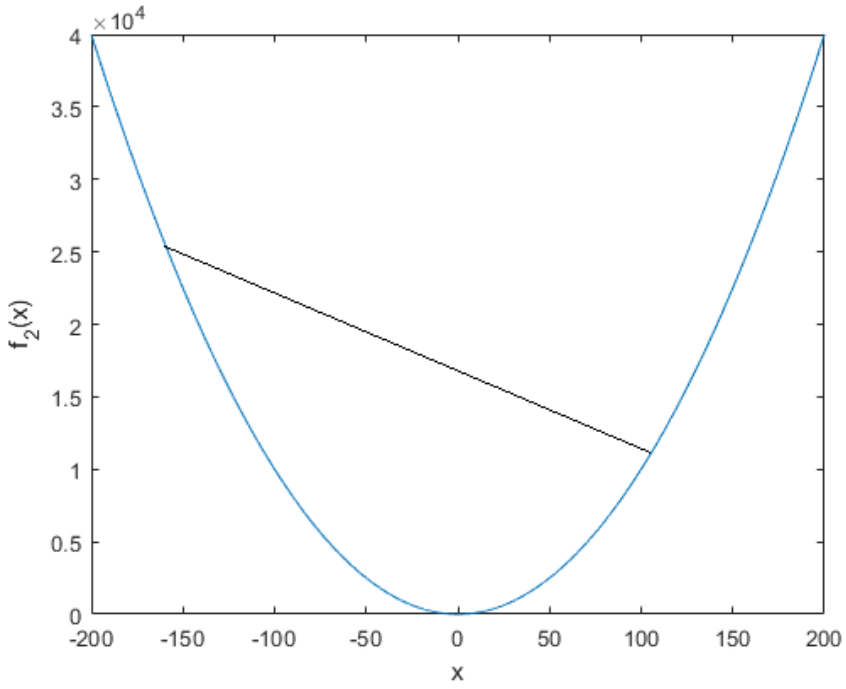

Figure 3.2: An example of a convex function

$$\begin{aligned} f_3(x) &= x^3 \\ f_3''(x) &= 3x \not\succeq 0 \end{aligned} \tag{3.7}$$

3.2.3 Convex optimization problem

Generally, the type of an optimization problem depends on the type of its objective and constraint functions. Optimization problem (3.2) is a convex optimization problem if:

- The domain of f is a convex set.
- Objective function $f(\mathbf{x})$ is convex.
- Inequality functions $g_i(\mathbf{x})$ are convex.
- Equality functions $h_j(\mathbf{x})$ are affine.

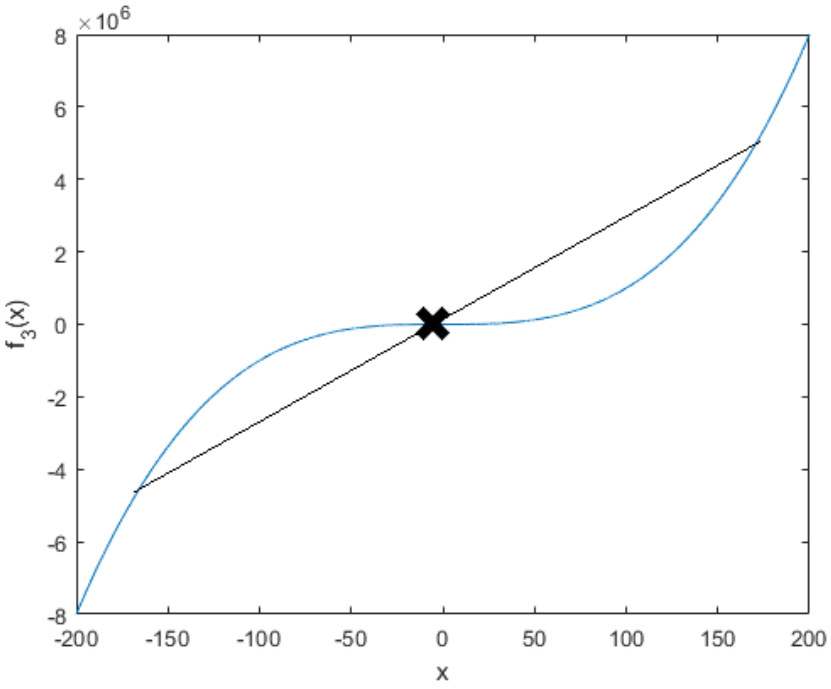

Figure 3.3: An example of a non-convex function

Function $h(\mathbf{x})$ is affine if [24]:

$$h(x_1, x_2, ..., x_n) = k_1 \times x_1 + k_2 \times x_2 + ... + k_n \times x_n + c \tag{3.8}$$

The convexity of an optimization problem is important because the convex optimization problems have only one local minimum which is the global minimum. $x_1 \in D_f$ is a global minimum point if [24]:

$$f(x_1) \leq f(x) \ , \forall x \in D_f \tag{3.9}$$

where D_f is the domain of $f(x)$. Therefore, modeling a problem as a convex optimization can help achieve a global minimum in an efficient runtime while non-convex optimization problems might

not be able to find the global optimum in reasonable runtime.

3.2.4 Geometric programming (GP) optimization problem

A geometric program is a special type of optimization problems. In a GP optimization problem:

- The objective function should be posynomial.
- The inequality constraints should be posynomial.
- The equality constraints should be monomial.
- All the variables should be positive.

A function $f_{\mathrm{m}}(\mathbf{x})$ is monomial if it is in the below global form [24]:

$$f_{\mathrm{m}}(x_1, x_2, ..., x_n) = k \times x_1^{p_1} \times x_2^{p_2} \times ... \times x_n^{p_n} \tag{3.10}$$

where:

$$\begin{aligned} &k > 0 \\ &p_i \in \mathbb{R} \end{aligned} \tag{3.11}$$

A posynomial function $f_{\mathrm{p}}(\mathbf{x})$ is a sum of monomial functions.

$$f_{\mathrm{p}}(x_1, x_2, ..., x_n) = \sum_{j=1}^{j_0} (k_j \times x_1^{p_{1j}} \times x_2^{p_{2j}} \times ... \times x_n^{p_{nj}}) \tag{3.12}$$

where:

$$\begin{aligned} & k_j > 0 \\ & p_{ij} \in \mathbb{R} \\ & j_0 = \text{Number of monomials} \end{aligned} \tag{3.13}$$

For example, function $m(\mathbf{x})$ is a monomial:

$$m(x_1, x_2, x_3, x_4, x_5, x_6) = 12.76 \times x_1^3 \times x_2^{4.5} \times \times x_3^{-2.25} \times x_5^{0.5} \times x_6^0 \tag{3.14}$$

However, function $n(\mathbf{x})$ in (3.15) is not a monomial.

$$n(x_1, x_2, x_3, x_4, x_5, x_6) = -12.76 \times x_1^3 \times x_2^{4.5} \times \times x_3^{-2.25} \times x_5^{0.5} \times x_6^0 \tag{3.15}$$

Finally, an instance of a posynomial function $p(\mathbf{x})$ is given below:

$$p(x_1, x_2, x_3, x_4, x_5, x_6) = 0.63 \times x_2^{-5} \times x_4^{2.5} \times x_6^{-11} + 52 \times x_1^0 \times x_3^{4.3} \times x_5^4 \tag{3.16}$$

The form of a GP optimization problem is shown below:

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 1 \ , i \in 1, 2, ..., o \\ & h_j(\mathbf{x}) = 1 \ , j \in 1, 2, ..., p \\ & \mathbf{x} > 0 \end{aligned} \tag{3.17}$$

where $f(\mathbf{x})$ and $g_i(\mathbf{x})$ are posynomials and $h_j(\mathbf{x})$ are monomials.

Transforming GP optimization problem to a convex optimization problem

Originally, a GP optimization problem is a non-convex optimization problem. However, the main advantage of the GP optimization is that it can be transformed to a convex optimization problem. Hence, it can be solved efficiently with guaranteed global optimality.

Considering the constraint $\mathbf{x} > 0$ given in (3.17), a variable can be defined as $y_i = \ln(x_i)$ to transform the non-convex GP optimization problem to a convex optimization problem. Based on (3.18), instead of each of the variables $\mathbf{x}$ in the GP optimization problem, $e^{\mathbf{y_i}}$ can be replaced.

$$\begin{aligned} y_i &= \ln(x_i) \\ e^{y_i} &= e^{\ln(x_i)} \\ x_i &= e^{y_i} \end{aligned} \tag{3.18}$$

For instance, the monomial function in (3.12) can be rewritten as (3.19) to be used in the convex form of the GP optimization problem.

$$f(x_1, x_2, ..., x_n) = e^{k'} \times e^{y_1 \times p_1} \times e^{y_2 \times p_2} \times ... \times e^{y_n \times p_n} \tag{3.19}$$

Similarly, the new form of posynomials can be rewritten as the summation of the transformed monomials given in (3.19). Then, the new form of the GP optimization problem is as:

$$\begin{aligned} \min \quad & f(\mathbf{y}) = \textstyle\sum_{l=1}^{l_0} [e^{k'_{l0}} \times e^{y_{1l0} \times p_{1l0}} \times e^{y_{2l0} \times p_{2l0}} \times ... \times e^{y_{nl0} \times p_{nl0}}] \\ \text{s.t.} \quad & g_i(\mathbf{y}) = \textstyle\sum_{l=1}^{l_t} [e^{k'_{li}} \times e^{y_{1li} \times p_{1li}} \times e^{y_{2li} \times p_{2li}} \times ... \times e^{y_{nli} \times p_{nli}}] \leq 1; \ \forall i \in 1, 2, ..., o \\ & h_j(\mathbf{y}) = e^{k'_j} \times e^{y_{1j} \times p_{1j}} \times e^{y_{2j} \times p_{2j}} \times ... \times e^{y_{nj} \times p_{nj}} = 1 \ ; \ \forall j \in 1, 2, ..., p \end{aligned} \tag{3.20}$$

Then, a natural logarithm operator is applied to take the logarithm of all the functions:

$$\begin{aligned} \min \quad & \ln f(\mathbf{y}) = \ln \textstyle\sum_{l=1}^{l_0} [e^{k'_{l0}} \times e^{y_{1l0} \times p_{1l0}} \times e^{y_{2l0} \times p_{2l0}} \times ... \times e^{y_{nl0} \times p_{nl0}}] \\ \text{s.t.} \quad & \ln g_i(\mathbf{y}) = \ln \textstyle\sum_{l=1}^{l_t} [e^{k'_{li}} \times e^{y_{1li} \times p_{1li}} \times e^{y_{2li} \times p_{2li}} \times ... \times e^{y_{nli} \times p_{nli}}] \leq 0; \ \forall i \in 1, 2, ..., o \\ & \ln h_j(\mathbf{y}) = \ln [e^{k'_j} \times e^{y_{1j} \times p_{1j}} \times e^{y_{2j} \times p_{2j}} \times ... \times e^{y_{nj} \times p_{nj}}] = 0 \ ; \ \forall j \in 1, 2, ..., p \end{aligned} \tag{3.21}$$

The new monomials are given in below:

$$hn_j(\mathbf{y}) = k'_j + \ln e^{y_{1j} \times p_{1j}} + \ln e^{y_{2j} \times p_{2j}} + ... + \ln e^{y_{nj} \times p_{nj}} = 0; \ \forall j \in 1, 2, ..., p \tag{3.22}$$

Then, the new inequality constraints (3.23) are affine.

$$hn_j(\mathbf{y}) = k'_j + y_{1j} \times p_{1j} + y_{2j} \times p_{2j} + ... + y_{nj} \times p_{nj} = 0 \tag{3.23}$$

In a similar way, the new posynomials after transformation are convex functions, because they always have positive definite second derivative in the domain of the optimization problem.

Because of the logarithm variable transforms, the variables and coefficients of the transformed monomials and posynomials given in (3.21) are positive. In addition, after taking the natural logarithm:

- The posynomials of the objective and inequality functions are transformed to log-sum-exp functions that are shown to be convex in [24].
- The equality constraints become affine.
- The domain of transformed optimization problem is convex.

Then, the transformed GP optimization problem is a convex optimization problem that can be efficiently solved using convex optimization techniques.

3.3 Clock network buffer sizing background

3.3.1 Clock network buffer sizing challenges

An important task in the clock tree synthesis design stage is reducing clock skew to close the IC timing [22, 85]. The quality of the clock signal at each clock node is another important constraint. To achieve this, the clock slew needs to meet certain design-specific limits.

Recently with the increasing demands for portable devices, optimizing power consumption has become an important challenge in VLSI circuit design. It is shown in different works that the clock network is one of the major power consumers in an IC [48, 82]. The main reason for this high power consumption is that the clock signal drives a high capacitance load in an IC [97]. Therefore, minimizing resources such as power and clock network area is another important objective while designing clock networks. Clock network area is defined as the summation of the area of each buffer. In clock networks, power consumption is positively correlated to the area. Therefore, clock network area has been used as a proxy for power consumption in several works [41, 82, 107].

Clock network buffer sizing is an effective way to achieve lower power consumption [12, 107]. Fundamentally, a non-inverting buffer is formed by two series-connected inverter gates that (a) improve the delay and (b) decrease the clock slew [49]. During clock network buffer sizing optimization problem the optimal sizes of all the buffers are determined [82].

Clock network buffer sizing is an optimization problem with major objectives and constraints listed below:

- Clock network area
- Power consumption
- Clock skew
- Clock slew

- Technology constraints

3.3.2 Literature review

The rapid improvement in the performance of digital ICs is mainly achieved by the fast growth in the number of transistors in the ICs and the shrinking of transistor sizes [10]. However, this progress in the fabrication technology has made it harder to meet the timing constraints such as clock skew and clock slew.

In clock networks, using buffers and sizing them is one of the main solutions to improve the clock slew by restoring the clock signal level [90, 97], meet clock skew constraints [40, 42], reduce power consumption [40, 97]. Clock network buffer sizing is often formulated as an optimization problem [40, 41, 82]. The proposed clock network buffer sizing method developed as part of this thesis can be transformed to a convex optimization problem using GP. In this work, the main advantages of using a convex optimization problem compared to other types are:

- It can be used efficiently and reliably to find the global optimum solution.
- This optimization method can be solved in a reasonable runtime.

In this section, the most efficient models for clock network buffer sizing existing in the literature are discussed. These models mostly include multi-objective approaches [33, 41] and single objective approaches such as area objective [82, 107] and clock skew objective [17, 77] frameworks.

3.3.3 Single Objective Methods

The single objective convex optimization problems have only one global optimum [20]. In the proposed algorithm in [98], the clock slew rate is used as an upper bound constraint and the number of buffers is minimized. Their method ignores the short circuit power. However, in [97], a buffer

insertion and sizing algorithm considering clock slew is proposed which minimizes total power consumption including the short circuit power.

In [107], in order to reduce the complexity of the objective function of power consumption, a Sequential Linear Program (SLP) is formed and the value of the objective function is reduced in each iteration. SLP is also used in [17] in a clock skew objective method to iteratively linearize the nonlinear problem in order to minimize the clock skew subject to a power consumption constraint. The disadvantage of the SLP algorithm is that it doesn't guarantee to reach the global optimal solution.

3.3.4 Multi-Objective Methods

Power consumption and clock skew are competing objectives. Therefore, it is necessary to include power and clock skew in the optimization objective at the same time to find a balanced trade-off between them. Often, multi-objective problem formulations minimize a weighted sum of the objectives and in most cases, designers have to pre-determine the weights of the objectives based on their experience [82]. However, a self tuning multi-objective formulation is analyzed in [82] that can obtain large improvements in terms of power and clock skew. In [82], all the widths and lengths of the buffers are considered as variables of the optimization problem and a new clock skew variable is added to the objective function to transform the clock skew formulation in (2.1) to a GP format since (2.1) cannot be directly used in the problem objective [82].

In [41], a buffer and wire sizing optimization problem is formulated using GP. The improvement of the power consumption and clock skew using this method is significant relative to the state-of-the-art methods. However, the complexity of this method's formulation has increased leading to significantly larger runtime that made the algorithm not suitable for large designs.

3.3.5 Other Approaches

In [112], a new model for clock skew is proposed considering variations which is tested on 180nm technology designs. The minimization of clock skew variation across corners is evaluated in [47]. In [22, 30, 61, 101], different zero skew algorithms under Elmore delay model [11] are proposed which can achieve zero skew. However, the zero skew algorithms are not robust to variations introduced during the fabrication process. In [77], a method is proposed to minimize clock skew and clock slew based on a greedy algorithm. The first shortcoming of this method is that the designers cannot constrain the power consumption. Another issue of this method is that as a greedy algorithm, it may not obtain the global optimal solution.

3.4 Modeling clock network buffer sizing as a GP optimization problem

The large scale of the clock networks is the reason for the high runtime of clock network buffer sizing optimization problem. GP has been used for solving large scale circuit design problems and several works show that GP is fast and reliable [24, 41, 82]. Clock network buffer sizing has been modeled as a GP and solved efficiently in [81]. In this section, the process of transforming clock network buffer sizing to a GP optimization problem is explained.

The mathematical formulation of a clock network buffer sizing optimization problem minimiz-

ing clock network area is given below [81]:

$$
\begin{aligned}
\min \quad & \text{Area}(\mathbf{x}) = \textstyle\sum_{k \in B} (b_{w_k} \times b_{l_k}) \\
\text{s.t.} \quad & max\{SinkDelay_i(\mathbf{x}) - SinkDelay_j(\mathbf{x})\} \leq t_{Skew} \ , \forall i, j \in S \\
& slew_n(\mathbf{x}) \leq t_{slew} \ , \forall n \in B \bigcup S \\
& l_{min} \leq b_{l_k} \ , k \in B \\
& w_{min} \leq b_{w_k} \ , k \in B
\end{aligned}
\tag{3.24}
$$

where:

- $\text{Area}(\mathbf{x})$: total area of the buffers
- B: set of all buffers
- S: set of all sinks
- $slew_n(\mathbf{x})$: the input clock slew at node n

and the variables are defined as:

- b_{w_k}: width of buffer k
- b_{l_k}: length of buffer k

l_{min} and w_{min} are the constant values based on the fabrication technology. t_{slew} and t_{skew} are the target clock slew and clock skew, respectively, that are given in design specifications by the designer.

The objective function of the formulation in (3.24) is the total area of all the buffers in the clock network that includes multiplications of buffer widths and lengths. Then, the summation of all these multiplications is a posynomial that is already in the GP format.

The clock skew constraints determine the limitation on the clock skew where this limitation is a target clock skew set by designer. The clock skew constraint is formulated as the following [81]:

$$\max\{SinkDelay_i(\mathbf{x}) - SinkDelay_j(\mathbf{x})\} \leq t_{Skew} \ , \forall i, j \in S \tag{3.25}$$

The maximum operator in (3.25) cannot be included and handled in GP. Therefore, every difference between the values of the clock signal delays of the clock network sinks should be considered as a constraint in the first step to transform (3.25) into a GP format. Then, clock skew constraint can be rewritten as several constraints as:

$$SinkDelay_i(\mathbf{x}) - SinkDelay_j(\mathbf{x}) \leq t_{\text{skew}} \ , \forall i, j \in S \tag{3.26}$$

In the second step, to remove the negative terms, an approximation is used in new clock skew constraints by replacing $SinkDelay_j(x)$ with the minimum value of the delays, i.e. $SinkDelay_{\min}$, of the clock network sinks.

$$SinkDelay_i(\mathbf{x}) - SinkDelay_{\min} \leq t_{\text{skew}} \ , \forall i \in S \tag{3.27}$$

where $SinkDelay_{\min}$ is a constant value for the minimum sink delay. Then, the negative component in (3.27) which is now a constant is moved to the right hand side of the constriants to transform these constraints to the standard GP format:

$$SinkDelay_i(\mathbf{x}) \times (SinkDelay_{\min} + t_{\text{skew}})^{-1} \leq 1 \ , \forall i \in S \tag{3.28}$$

The new constraints in (3.28) are in the GP format. Therefore, these clock skew constraints can be added to the final GP formulation.

An approximation is used in [82] based on a method in [19] to model the clock slew of each

node. This approximation which is given in (3.29) is used in the formulation.

$$slew_n(\mathbf{x}) \approx \ln(9) \times SinkDelay_n^s(\mathbf{x}) \tag{3.29}$$

where, $SinkDelay_n^s(x)$ is the delay from the output of the previous upstream buffer to node n. In a typical clock network, in order to have synchronization and performance, clock slew should be maintained under a target clock slew (t_{slew}).

$$slew_n(\mathbf{x}) \leq t_{\text{slew}} \ , \forall n \in B \bigcup S \tag{3.30}$$

This constraint can be transformed to the GP format:

$$slew_n(\mathbf{x}) \times (t_{\text{slew}})^{-1} \leq 1 \ , \forall n \in B \bigcup S \tag{3.31}$$

In addition, technology constraints are monomials and can be transformed to the GP format as in (3.32).

$$\begin{aligned} l_{min} \times (b_{l_k})^{-1} \leq 1 \ , k \in B \\ w_{min} \times (b_{w_k})^{-1} \leq 1 \ , k \in B \end{aligned} \tag{3.32}$$

Then, the final formulation of the area objective clock network buffer sizing [81] as a GP

optimization problem is given below:

$$\begin{aligned} \min \quad & \textstyle\sum_{k\in B} b_{w_k} \times b_{l_k} \\ \text{s.t.} \quad & SinkDelay_i(\mathbf{x}) \times (SinkDelay_{\min} + t_{\text{skew}})^{-1} \leq 1 \ , \forall i \in S \\ & slew_n(\mathbf{x}) \times (t_{\text{slew}})^{-1} \leq 1 \ , \forall n \in B \bigcup S \\ & l_{min} \times (b_{l_k})^{-1} \leq 1 \ , k \in B \\ & w_{min} \times (b_{w_k})^{-1} \leq 1 \ , k \in B \end{aligned} \tag{3.33}$$

3.5 Summary

In this chapter, first, an introduction to the optimization problems with specific focus on the convex and GP optimization problem formulations are presented. The challenges in clock network buffer sizing and the existing methods are also discussed. Finally, the formulation of the clock network buffer sizing as a GP optimization problem is explained.

Chapter 4

Development of a New Efficient GP-Compatible Clock Delay Model

4.1 Introduction

In this chapter, a new and efficient delay model is developed. The proposed delay model is able to accurately estimate the clock signal delay in different nodes of the clock network in a reasonable runtime. The least square fitting is investigated for the delay modeling and the proposed delay model is evaluated using a set of benchmarks form the ISPD 2009 [2] and 2010 high-performance clock tree synthesis contests [3]. The advantage of the proposed delay model is that it fits in the GP format which has been successfully used for buffer and wire sizing in optimization models [41, 81, 82].

Once the model is developed, the estimated values by the proposed delay model are compared against an accurate circuit simulation tool, ngspice [6], as a reference. The results demonstrate that the accuracy of estimated delay values have significantly improved. The proposed model achieves up to 59 ps improvement in accuracy of the minimum and maximum sink delay estimation.

The experiments on the clock network buffer sizing also demonstrate that the proposed delay

model shows 15 ps delay estimation improvement, on average, compared to the Elmore delay model used in [82]. Application of the proposed delay model in clock network buffer sizing also improves the power consumption and clock skew of the clock network.

The main contributions of this chapter are as follows:

- Developing an accurate delay model for clock signal which results in improvement in sink delay prediction.
- Improving the minimum and maximum sink delay estimations significantly.
- Achieving a GP compatible format for the proposed delay model.
- Achieving low runtime compared to the *ngspice* circuit simulator.
- Improving clock network buffer sizing results by applying the proposed delay model.

The rest of this chapter is organized as follows: In Section 4.2, the modeling of the proposed delay model is presented. In Section 4.3, the experimental results are analyzed. Finally, in Section 4.4, the summary of the chapter is given.

4.2 Proposed GP-Compatible Delay Model

4.2.1 RC Network Model

The focus of this section is on clock network delay modeling. Consider a simple clock network such as the one presented in Fig. 4.1 where a clock source is connected to a buffer by a single wire and the buffer is connected to a sink directly. The RC equivalent of this simple circuit considering the Elmore delay model is given in Fig. 4.2.

Using the Elmore delay model, the values of the resistance and capacitances of the wire model and output resistance and input capacitance of the buffer model in Fig. 4.2 can be calculated based

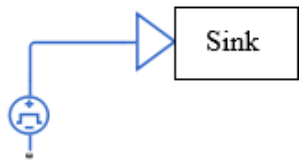

Figure 4.1: A simple clock network

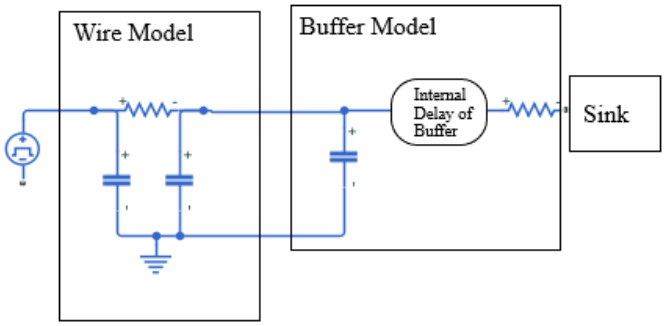

Figure 4.2: RC equivalent of the clock network in Fig. 4.1

on the length and width of the wire and the buffer. Also, according to the Elmore delay model, the internal delay of the buffer is considered as a constant value. These formulations that are important to calculate the wire delay model and buffer delay model are given below:

$$
\begin{aligned}
R_w &= L_w \times R_{uw} \\
C_w &= L_w \times C_{uw} \\
R_b &= \gamma_1 \times L_b / W_b \\
C_{b_{output}} &= \gamma_2 \times L_b \times W_b + \gamma_3 \\
C_{b_{input}} &= \gamma_4 \times L_b \times W_b
\end{aligned}
\tag{4.1}
$$

where:

- R_w: Resistance of the wire

- C_w: Capacitance of the wire
- R_{uw}: Unit wire resistance per length (Ohm/nm)
- C_{uw}: Unit wire capacitance per length (fF/nm)
- L_w: Length of the wire
- R_b: Output resistance of the buffer
- $C_{b_{output}}$: Capacitance of the buffer seen by the downstream buffer
- $C_{b_{input}}$: Capacitance of the buffer seen by the upstream buffer
- L_b: Length of the buffer
- W_b: Width of the buffer
- γ_3: A constant parameter to model the internal delay of buffer as a constant part of the delay model
- γ_i ($\forall i \in 1, 2, 4$): A constant parameter based on the fabrication technology

The unit wire resistance and capacitance per length are constants but are different for various type of wires since any type of wire has its specific characteristics. Also, different types of buffers have special characteristics and internal delay. Then, γ_1, γ_2, γ_3 and γ_4 are different constant values for each type of buffer. They can be calculated based on the input load capacitance, output parasitic capacitance and output resistance of the buffer which are given in the technology files for the benchmarks provided in the contests [2, 3]. According to these technology files:

- There are two different types of wires (i.e. wire type 0 and 1).
- There are two different types of buffers (i.e. buffer type 0 and 1).

4.2.2 Mathematical formulation

The mathematical formulations of the wire and buffer delay model are developed based on (2.2). In the proposed delay model, the buffer is modeled to isolate the upstream capacitance from the downstream capacitance in the clock network. Also, according to the contests [2, 3], buffers include an output parasitic capacitance which means in Fig. 4.2, after modeling the buffer using the Elmore delay model, between the output resistor of the buffer and the sink, another capacitance ($C_{b_{output}}$) connected to the ground should be added as shown in Fig. 4.3.

Figure 4.3: RC equivalent of Fig. 4.2

Then, wire and buffer delay models are analyzed separately. The internal buffer delay is modeled as γ_3 as part of the output parasitic capacitance and it is considered as a constant value. In the wire model in Fig. 4.2, the values of each of the two capacitances are considered half of the capacitance of the wire (C_w) based on the single pi delay model presented in Section 2.3.1. Then, the wire delay model according to the Elmore delay model is formulated as below:

$$\begin{aligned} &\text{wiredelay}= \\ &L_w \times L_b \times W_b \times (R_{uw} \times \gamma_4) \\ &+L_w^2 \times (R_{uw} \times C_{uw}/2) \end{aligned} \tag{4.2}$$

Fig. 4.4, shows the effect of the wire length on the wire delay. According to this figure, wire delay is proportional to the wire length because the resistance of the wire increases with its wire length.

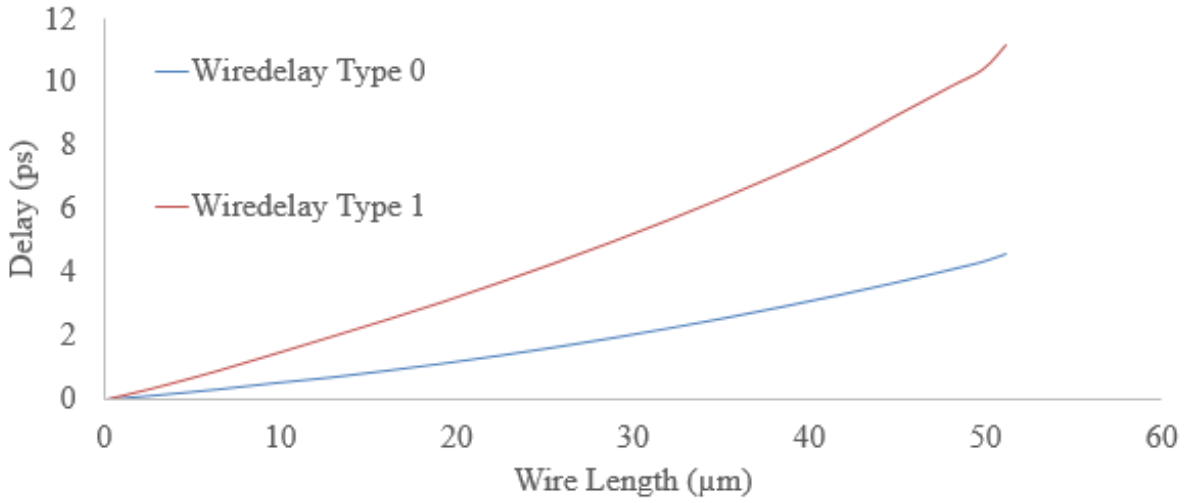

Figure 4.4: Wire delay plotted by wire length

The buffer delay model considering the Elmore delay model is formulated as:

$$\begin{aligned} &\text{bufferdelay}= \\ &L_b^2 \times (\gamma_1 \times \gamma_2) \\ &+L_b/W_b \times (\gamma_1 \times (\gamma_3 + C_{sink})) \end{aligned} \tag{4.3}$$

where C_{sink} represents the capacity of the sink and in Fig. 4.1 is the output capacitance load seen by the buffer.

In Fig. 4.5, the buffer delay based on the output capacitance load for buffer type 0 and 1 is shown. In this experiment, the buffer input slew is kept almost the same for different output capacitance values. From this figure, it can be seen that the buffer delay is increased by the output capacitance load which is represented by C_{sink}.

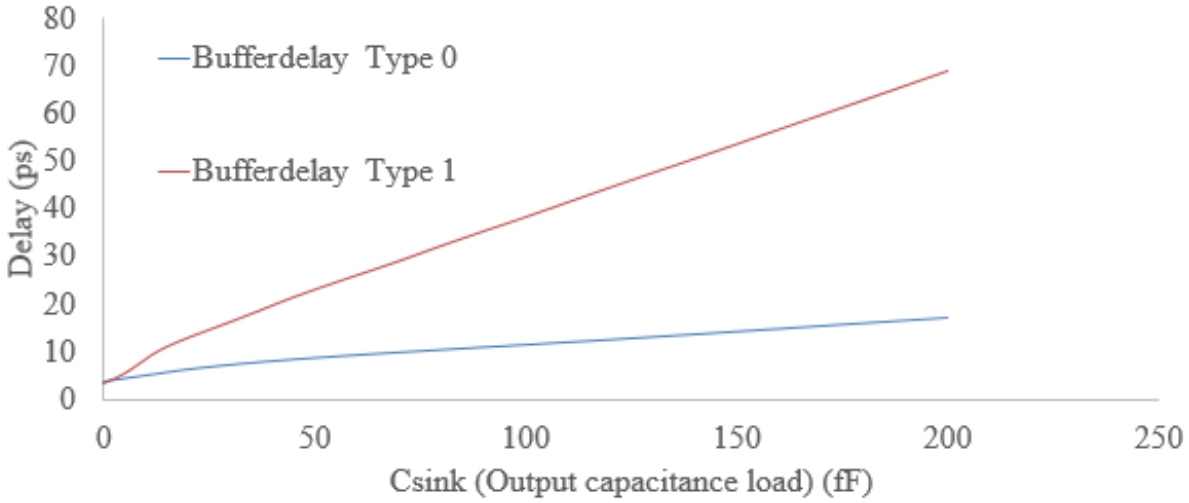

Figure 4.5: Buffer delay based on the output capacitance

Each equation of (4.2) and (4.3) include two terms. For wire delay model, the two terms are:

$$\text{Term}_{W_1} = \text{L}_w \times L_b \times W_b \times (R_{uw} \times \gamma_4)$$

$$\text{Term}_{W_2} = \text{L}_w^2 \times (R_{uw} \times C_{uw}/2) \tag{4.4}$$

and for the buffer delay model, the terms are:

$$\text{Term}_{B_1} = \text{L}_b^2 \times (\gamma_1 \times \gamma_2)$$

$$\text{Term}_{B_2} = \text{L}_b/W_b(\gamma_1 \times (\gamma_3 + C_{sink})) \tag{4.5}$$

According to the $Term_{W_2}$ in (4.4), the wire delay is supposed to have quadratic growth as the

wire length increases. Unfortunately, the buffer input slew violations made it impossible to get the data with larger than $0.5\mu m$ wire length. However, the red curve in Fig. 4.4, which shows the wire delay for wire type 1, is starting to increase quadratically after $0.4\mu m$ wire length. This shows that the trend of the Equation (4.2) is close to the actual delay curve (i.e. red curve) provided based on the ngspice result.

In addition, the $Term_{B_2}$ in (4.5) shows a linear relationship between buffer delay and C_{sink} considering other terms are constants. In Fig. 4.5, the curves are close to linear in higher capacitance load, but not for lower capacitance load close to 0 fF, which is not a common case in the ICs. Then, the curves in Fig. 4.5 show that the linear relationship between buffer delay and C_{sink} is consistent with the simulation data and can be used for least square fitting.

In this chapter, the clock network in Fig. 4.1 is evaluated with different input values of L_w, L_b, W_b, C_{sink}. As discussed before, all the other elements in (4.2) are constants. The ngspice [6] is used to find accurate values for the wire, buffer and sink delays to populate the training data. In [52], it is shown that the inaccuracy of the Elmore delay model increases as nodes get closer to the buffers. In addition, in [92], it is shown that the average value of the distance between the buffers in an IC is getting smaller. Then, the focus of the training data is more on this region to achieve more accurate delay model.

In this thesis, it is proposed to use least square fitting to find a suitable delay model. This model includes coefficients (i.e. α_w, β_w, α_b, β_b) which are determined using the training data. Then, the best delay model will be calculated using the following:

$$\begin{aligned}
&\text{ActualSinkDelay}=\text{ActualWireDelay}+\text{ActualBufferDelay}\\
&\text{ActualWireDelay}= \alpha_w\times\text{Term}_{W_1} + \beta_w\times\text{Term}_{W_2} \qquad (4.6)\\
&\text{ActualBufferDelay}= \alpha_b\times\text{Term}_{B_1} + \beta_b\times\text{Term}_{B_2}
\end{aligned}$$

Algorithm 1: Delay modeling algorithm: training, development and validation algorithm

1 **Input:** $numTrainingData$, $numBenchmarks$
2 **Output:** Error (mse) of the estimated delays compare to the actual delays for each benchmark
3 $num \leftarrow 0$
4 **while** *num* $\leq numTrainingData$ **do**
5 Synthesize a new 1-sink simple clock network
6 Run ngspice on the clock network
7 Extract the wire and buffer delay values and add them to the $trainingData$
8 Construct the delay modeling terms based on (4.4) and (4.5) and add them to the $trainingData$
9 $num \leftarrow num + 1$
10 **end**
11 Apply least square fitting using the $trainingData$
12 Develop the proposed delay model using the coefficients from the least square fitting solution based on (4.6)
13 $i \leftarrow 0$
14 **while** $i \leq numBenchmarks$ **do**
15 $estDelays_i \leftarrow$ Calculate the delays at the sinks of benchmark i using the proposed delay model
16 $actDelays_i \leftarrow$ Calculate the delays at the sinks of benchmark i using ngspice
17 $mse_i \leftarrow$ MSE($estDelays_i$, $actDelays_i$)
18 Output mse_i
19 $i \leftarrow i + 1$
20 **end**
21

In proposed Algorithm 1, the proposed delay modeling algorithm including data training, model development and validation is presented. The inputs to this algorithm are the number of data points used for training ($numTrainingData$) and the number of benchmarks from the ISPD 2009 [2] and 2010 [3] contests ($numBenchmarks$). The first part of the algorithm includes the data training steps. In this part of the algorithm, the goal is to extract a $trainingData$ set. For each data point, a simple clock network such as the one presented in Fig. 4.1 is synthesized by varying the buffer location, sink location, or sink capacitance (line 5). This part can be performed for any other technology nodes to populate suitable $trainingData$ set. Then, ngspice is run on the synthesized clock network (line 6), wire and buffer delay values are extracted (line 7) and the

delay modeling terms are constructed based on (4.4) and (4.5) (line 8). The $trainingData$ set is populated considering all the different delay ranges (especially smaller than 250 ps).

In the rest of this algorithm, the proposed delay model is developed using least square fitting based on equation proposed in (4.6) (line 11 and line 12). Then, the proposed delay model is validated using the contest benchmarks where the delays at all the sinks of each benchmark are calculated using the proposed delay model (line 15) and ngspice (line 16). Finally, the mean square error (mse) of the sink delays estimated using the proposed delay model are compared to the ngspice sink delay values and it is returned as the output of the algorithm. It is important to mention that since the algorithm is deterministic, the cross validation is not necessary.

4.3 Experimental Results

4.3.1 Experimental Setup

Hardware setup

All the experiments of this chapter are performed using a 64-bit server which includes:

- Two Intel 2.00-GHz Xeon E5-2620 processors
- 64 GB of memory

Software setup

The C++ programming language is used for all the codes and to populate the training data. MATLAB R2018a [8] is used for least square fitting and data visualization. Mosek 9.0 [5] is used to solve the clock network buffer sizing optimization problem. The reference clock skew and power consumption values were obtained by ngspice [6]. The power consumption of the clock networks is calculated by adding up all the power consumption components including:

- Switching power
- Short circuit power
- Leakage power

All the experiments in this chapter consider the variations in buffer sizes.

Benchmarks

The clock networks available through ISPD 2009 [2] and ISPD 2010 [3] contests were used to evaluate the proposed GP-compatible delay model. These clock networks are based on the 45 nm technology. It is important to mention that these clock networks are the latest available to academia. In Table 4.1, statistics of the ISPD 2009 and 2010 networks are presented. In this table:

Table 4.1: Statistics of the ISPD 2009 and 2010 input clock networks.

Cir.	Height (mm)	Width (mm)	S.cap (pF)	$\|S\|$	$\|B\|$	Power (mW)	Skew (ps)
09-11	11	11	4.2	0.1k	3.5k	20	48
09-12	8	13	4.1	0.1k	3.5k	20	91
09-21	13	12	3.6	0.1k	3.6k	21	51
09-22	12	5	3.4	0.1k	2.1k	12	27
09-31	17	17	9.6	0.3k	7.8k	43	228
09-32	17	17	6.7	0.2k	5.9k	33	143
09-nb1	3	2	5.9	0.3k	1.3k	5	25
Avg.	-	-	-	-	-	22	88
10-1	8.0	8.0	19	1.1k	11.4k	50	5
10-2	7.0	7.0	39	2.2k	22.1k	99	5
10-3	0.5	0.5	18	1.2k	3.0k	11	9
10-4	2.7	2.7	12	1.8k	3.9k	14	4
10-5	2.5	2.5	5	1.0k	1.9k	7	29
10-6	0.9	0.9	13	9.8k	2.5k	9	26
10-7	1.4	1.4	18	1.9k	3.9k	14	5
10-8	1.8	1.6	13	1.1k	2.9k	10	11
Avg.	-	-	-	-	-	27	12

- The clock networks' names are presented in the first column.

- Height and width of the clock networks are given in the second and third columns, respectively.
- The sink capacitance of the clock networks is presented in the fourth column.
- The number of sinks and number of buffers for the circuits are given in the fifth and sixth columns, respectively.
- Power consumption and clock skew are given in the seventh and eighth columns, respectively.

4.3.2 Experimental Analysis

To quantify the efficiency of the proposed delay model, several experiments were performed. In the first experiment, the minimum delay (D_{min}) and the maximum delay (D_{max}) are calculated in each clock network using ngspice, the Elmore and the proposed delay model. These values are given in Table 4.2. In this table, the clock networks' names are given in Column 1, and the calculated D_{min} using ngspice is presented in Column 2. In Columns 3 and 4, changes from the ngspice values ΔD_{min} when using the Elmore and the proposed delay model are given, respectively. In Column 5, the improvement in the delay estimation using the proposed delay model over the Elmore delay model is presented.

From Table 4.2, it can be seen that the proposed method can produce significantly more accurate results in terms of maximum and minimum delay of the sink delay over the Elmore delay model for the majority of test cases. The maximum improvement is 59 ps which is achieved for maximum sink delay of ISPD09-32. Generally, the highest improvements can be seen for the 2009 ISPD benchmarks which have higher delay, on average.

In Fig. 4.6, the advantage of the proposed delay model over the Elmore delay model in the delay range larger than 250 ps is shown. Also, the behaviour of the proposed delay model in the delay range between 100 ps and 250 ps is shown in the Fig. 4.7.

Table 4.2: Comparison of the difference (Δ) of the proposed delay model and the Elmore delay model with reference data (ngspice) for the minimum of sink delays (D_{min}) and maximum of sink delays (D_{max}) in the input clock networks.

Cir.	D_{min}				D_{max}			
	Spice (ps)	Elmore Δ	Proposed Δ	Imp (ps)	Spice (ps)	Elmore Δ	Proposed Δ	Imp (ps)
09-11	519	38	**0**	38	563	45	**5**	40
09-12	433	34	**7**	27	517	41	**6**	35
09-21	581	44	**1**	43	627	53	**10**	43
09-22	397	23	**9**	14	421	33	**3**	30
09-31	643	46	**1**	45	856	72	**15**	57
09-32	743	58	**11**	47	878	70	**11**	59
09-nb1	120	13	**4**	9	146	**5**	9	-4
Avg.	491	37	**5**	32	573	46	**8**	37
10-1	501	23	**6**	17	507	35	**15**	20
10-2	630	29	**7**	22	634	43	**19**	24
10-3	136	4	**2**	2	144	8	**1**	7
10-4	174	**2**	5	-3	176	9	**0**	9
10-5	146	12	**6**	6	172	15	**8**	7
10-6	105	**2**	3	-1	131	4	4	0
10-7	167	5	**4**	1	170	11	**1**	10
10-8	130	5	**2**	3	142	6	**1**	5
Avg.	249	10	**4**	6	260	16	**6**	10

The data for these figures are provided based on all the sink delays of the clock networks in Table 4.1. The ordering of the sink index is based on the sink delays calculated by ngspice simulator. These figures show the improvement of the delay prediction using the proposed delay model compared to the Elmore delay model in high and low delay ranges, respectively.

In many works in the VLSI design area, Mean Square Error method is used to evaluate the results of the STA and delay modeling algorithms [28, 108, 109]. The improvements obtained by the calculations for D_{min} and D_{max} only show the improvements obtained in the two most critical paths. In order to show a complete picture for the improvements in all clock delay paths, the Mean Square Error (MSE) of all the sink delays of the input clock networks compared to the values obtained by ngspice are calculated and presented in Table 4.3.

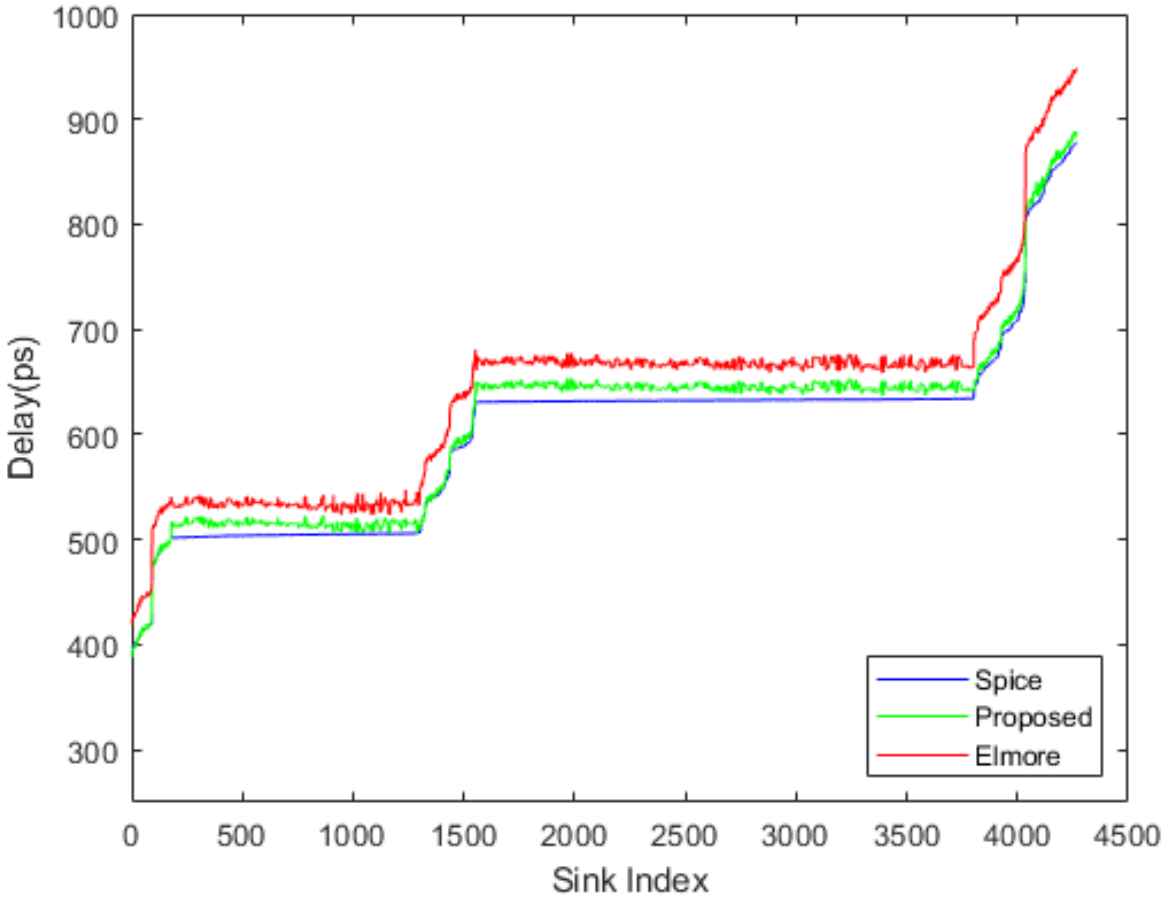

Figure 4.6: Sink delay estimation for delay range larger than 250 ps. The delay values are based on all of the sink delays.

In this table:

- The first column represents the names of the clock networks.
- The second and third columns show the MSE of the Elmore and the proposed delay models compared to the ngspice reference values, respectively.
- The MSE per sink of the Elmore and the proposed delay models compared to the ngspice reference values are given in the fourth and fifth columns, respectively.

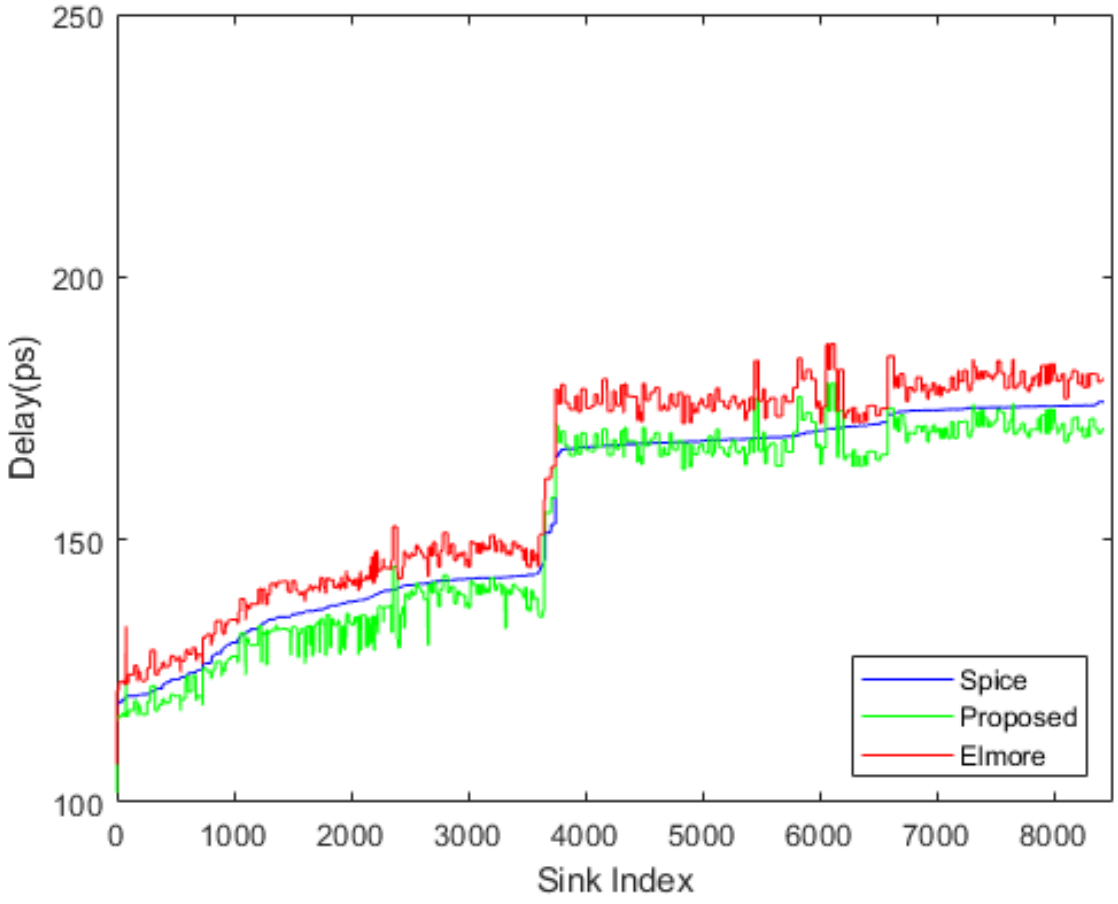

Figure 4.7: Sink delay estimation for delay range between 100 ps and 250 ps. The delay values are based on all of the sink delays.

- The sixth, seventh and eighth columns represent the runtime of the ngspice simulator, the Elmore and the proposed delay model, respectively.

From this table, it can be seen that in 2009 ISPD benchmarks, the mean square error of the Elmore delay model is 50 times higher than the mean square error of the proposed delay model. The mean square error per sink is significantly improved in all test cases with the exception of 09-nb1. This circuit has a relatively small chip area any many unique characteristics. It can be seen in Table 4.1 that this circuit has the most number of sinks in the ISPD 2009 circuits and the

Table 4.3: Comparison of mean square error (MSE) of the proposed delay model and the Elmore delay model considering all the sink delays of input networks comparing to reference data (ngspice).

Cir.	MSE		MSE per Sink		runtime (s)		
	Elmore	Proposed	Elmore	Proposed	Spice	Elmore	Proposed
09-11	1701	**11**	14.05	**0.09**	83	< 1	< 1
09-12	1362	**5**	11.64	**0.05**	90	< 1	< 1
09-21	2309	**37**	19.73	**0.31**	86	< 1	< 1
09-22	902	**6**	9.91	**0.06**	41	< 1	< 1
09-31	3272	**77**	11.98	**0.28**	312	< 1	< 1
09-32	4497	**57**	23.67	**0.30**	200	< 1	< 1
09-nb1	23	78	0.07	0.24	25	< 1	< 1
Avg.	2009	**39**	13.01	**0.19**	120	< 1	< 1
10-1	863	**127**	0.78	**0.11**	583	< 1	< 1
10-2	1259	**174**	0.56	**0.08**	1919	< 1	< 1
10-3	34	**9**	0.03	**0.01**	72	< 1	< 1
10-4	32	**15**	0.02	**0.01**	120	< 1	< 1
10-5	80	**28**	0.08	**0.03**	45	< 1	< 1
10-6	18	**10**	0.02	**0.01**	62	< 1	< 1
10-7	63	**5**	0.03	**0.00**	130	< 1	< 1
10-8	26	**8**	0.02	**0.01**	71	< 1	< 1
Avg.	297	**47**	0.19	**0.03**	375	< 1	< 1

second most number of sinks in all the ISPD 2009 and 2010 circuits. However, this circuit has the least number of buffers among all the circuits. Also, the last three columns of this table, show that the runtime of the proposed and the Elmore delay models for all test cases are significantly lower compared to the ngspice timing simulator.

In Fig. 4.8, the improvement in delay estimation of all the sink delays of all the clock networks using the proposed delay model is plotted. The data is extracted by measuring all the sink delays using ngspice, the Elmore and the proposed delay model.

In order to evaluate the proposed delay model in different scenarios, the ISPD 2009 and ISPD 2010 benchmarks are considered separately in Fig. 4.9 and Fig. 4.10, respectively. The comparison of the proposed delay model and the Elmore delay model shown in these figures, shows the

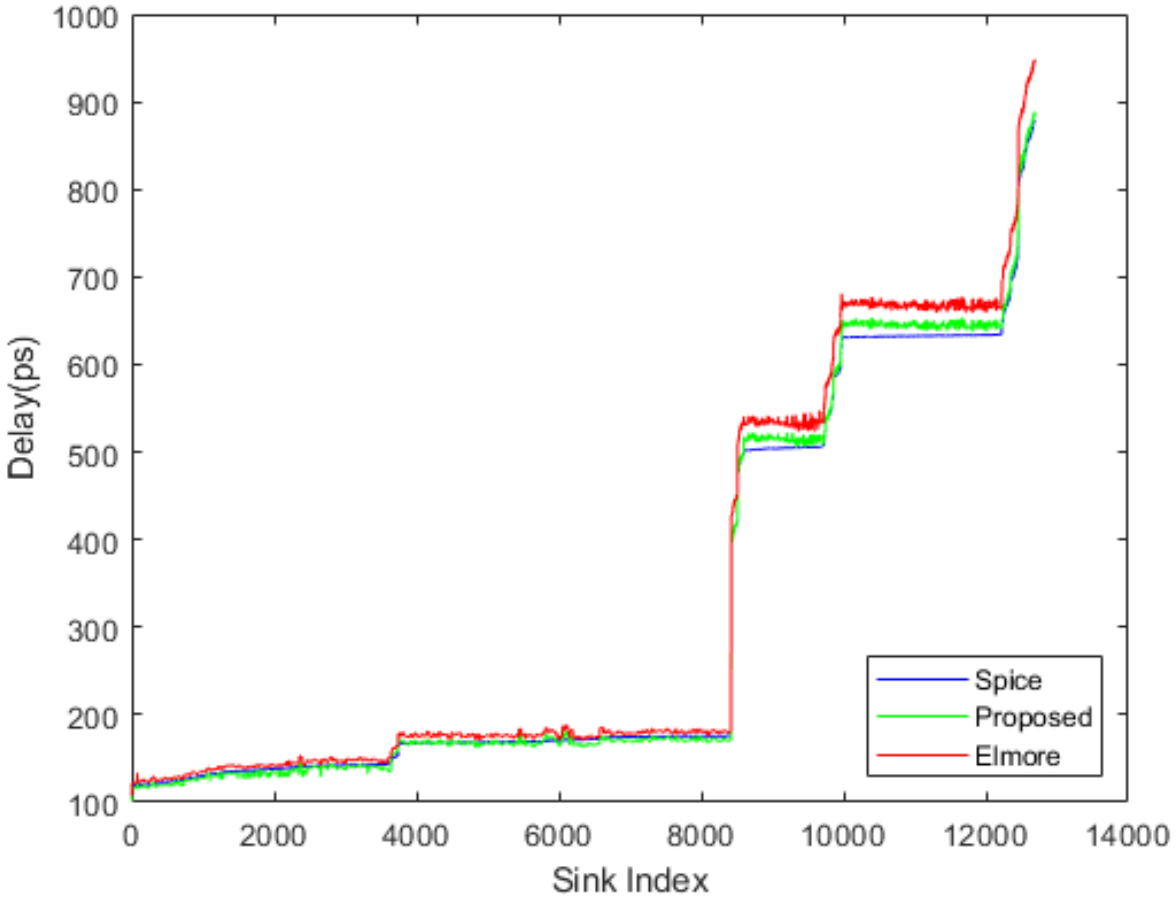

Figure 4.8: Sink delay estimation using ngspice, the proposed delay model and the Elmore delay model

improvement in the sink delay prediction in both benchmark sets.

In addition, in Fig. 4.8, Fig. 4.9 and Fig. 4.10, a big delay jump can be seen between 200 ps and 400 ps. In all 2009 and 2010 clock networks, the number of clock sinks with sink delay smaller than 200 ps is about 8000 and 4000 sinks have sink delay larger than 400 ps. However, there are not many sinks with the delay between 200 ps and 400 ps. This is due to the incoming placement solution. During the placement stage, if a clock sink is not connected to a fix object such as IO (Input-Output) components, it will most likely be placed close to the clock source. However, if it is connected to a fix object such as IO components, it tends to move away from the clock source.

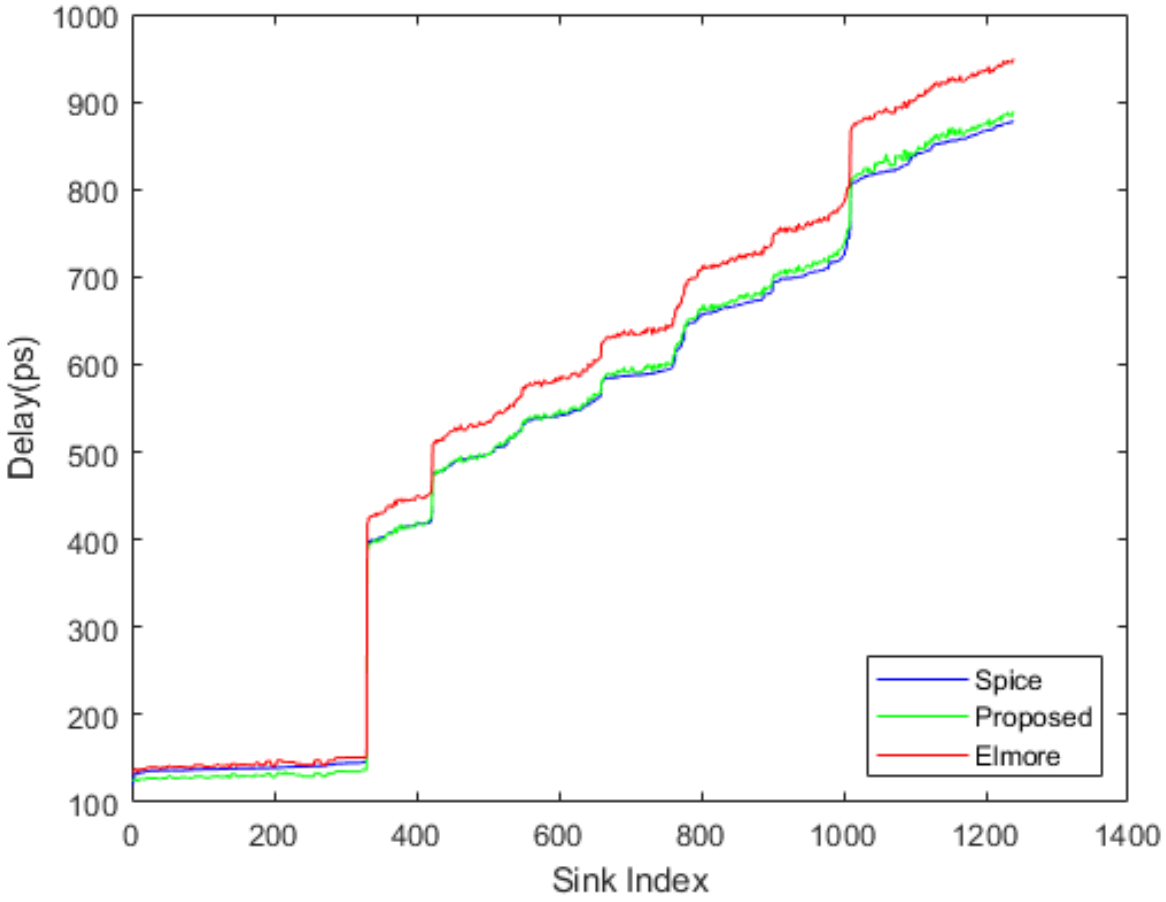

Figure 4.9: Sink delay estimation for all 2009 clock networks using ngspice, the proposed delay model and the Elmore delay model

This explains the different ranges of the sink delays.

In the next experiment, clock network buffer sizing is performed using the Area Objective (AO) [81] method to evaluate the performance of the proposed delay model in clock network buffer sizing applications. Table 4.4 presents the results of these experiments. In this table:

- The clock networks' names are given in the first column.
- The power consumption improvement after clock network buffer sizing using the Elmore and the proposed delay models are represented in the second and third columns, respectively.

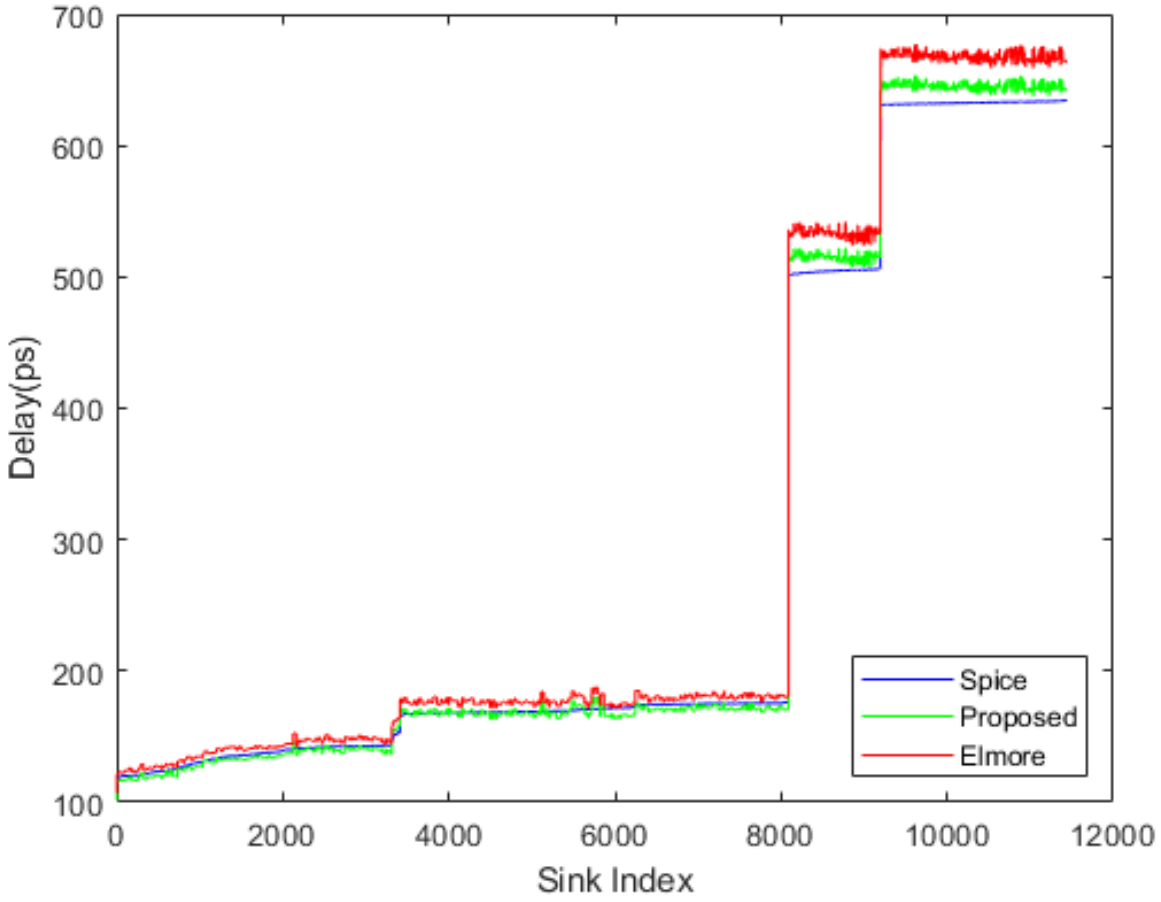

Figure 4.10: Sink delay estimation for all 2010 clock networks using ngspice, the proposed delay model and the Elmore delay model

- The fourth column shows the power consumption percentage improvement of the sized clock networks using the proposed delay model over the Elmore delay model.
- The fifth and the sixth columns show the clock skew values of the clock networks sized using the Elmore and the proposed delay models, respectively.
- The performance of the proposed delay model in term of clock skew improvement is compared in the seventh column.
- The runtime of the optimizer solving the sizing optimization problem using the Elmore and

proposed delay models are represented in the eighth and ninth columns, respectively.

- Finally, the last column shows the runtime improvement of the proposed delay model compared to the Elmore delay model.

Table 4.4: Comparison of the power consumption and clock skew after sizing with Area Objective (AO) clock network buffer sizing method in [81] using the proposed delay model and the Elmore delay model. All network data are calculated using ngspice.

Cir.	Power			Clock Skew			Runtime		
	Elmore (μW)	Proposed (μW)	Imp %	Elmore (ps)	Proposed (ps)	Imp (ps)	Elmore (s)	Proposed (s)	Imp (s)
09-11	10170	8192	**19**	23	14	**9**	15	15	0
09-12	9288	7462	**20**	63	39	**24**	8	13	-5
09-21	9959	8185	**18**	29	16	**13**	17	19	-2
09-22	6458	5076	**21**	18	12	**6**	7	6	1
09-31	16271	13969	**14**	127	57	**70**	157	163	-6
09-32	15288	12534	**18**	102	55	**47**	107	68	39
09-nb1	1965	1737	**12**	26	26	0	4	8	-4
Avg.	9914	8165	**17**	55	31	**24**	45	42	**3**
10-1	8260	6797	**18**	20	23	-3	275	265	**10**
10-2	14271	11576	**19**	29	33	-4	1014	881	**133**
10-3	2754	1993	**28**	15	15	0	31	18	**13**
10-4	3277	2438	**26**	13	15	-2	32	55	-23
10-5	1738	1299	**25**	53	65	-12	21	26	-5
10-6	1999	1501	**25**	45	52	-7	26	25	**1**
10-7	3165	2278	**28**	18	17	**1**	70	60	**10**
10-8	2510	1785	**29**	10	11	-1	30	30	0
Avg.	4747	3708	**25**	25	29	-4	187	170	**17**
Avg. of all	7158	5788	**21**	39	30	**9**	121	110	**11**

In Table 4.4, the power consumption shows reduction for all test cases while clock skew is improved by up to 133 ps. The runtime comparison shows that the proposed delay model formulation is as fast as the Elmore delay model. Then, clock network buffer sizing experiments show the efficiency and scalability (i.e. low runtime) of the proposed delay model.

The ISPD 2009 circuits shown in Table 4.1 have large clock skew values while their power

consumption is not as large as ISPD 2010 circuits. On the other hand, the ISPD 2010 circuits have very low clock skew values. However, the large power consumption values of these circuits is the reason that the average of power consumption reduction by Area Objective (AO) clock network buffer sizing method for ISPD 2010 circuits is larger compared to the ISPD 2009 circuits.

It is important to mention that the Area Objective (AO) clock network buffer sizing method in [81] considers clock skew as a constraint while power consumption which is modeled as total clock buffer area is considered as the only objective function. The power consumption reduction in all the test cases (up to 29% for circuit 10-08) shows that the proposed delay model is effective and applicable to clock network buffer sizing tools.

4.4 Summary

In this chapter, the proposed delay model is developed by tailoring the model to the training data from timing a small sample clock network using the circuit simulator tool, ngspice. A mathematical formulation of the proposed delay model is developed. Then, the efficiency of the proposed delay model is compared to the Elmore delay model for different conditions such as:

- Sinks close to the clock source while the delay value is small.
- Sinks with long distances from the clock source when the value of the delay is large.

In Table 4.5, the proposed delay model performance is compared to the other existing delay models in the literature.

The advantages of the proposed delay model include:

- The accuracy of estimated insertion delays at the sinks in the clock network has significantly improved.
- Considering more accurate minimum and maximum sink insertion delays, the global clock skew is also estimated more reliably using the proposed delay model.

Table 4.5: Comparison of the proposed delay model with the existing delay models in the literature. The comparison is based on the information and data available in the literature.

	Delay model	High accuracy	Low complexity	Modeling effort	Robust for fabrication	Applicable to convex opt	Commercially popular
Not closed	Lumped RLC [109]	✓	×	×	✓	×	×
	Look-up table [67]	✓	✓	✓	×	×	✓
	Moment Matching (PRIMA) [74]	✓	×	×	✓	×	×
	SPICE [6]	✓	×	×	✓	×	✓
Closed form	Elmore [25]	×	✓	✓	×	✓	✓
	Fitted Elmore Delay [11]	✓	✓	✓	×	✓	×
	PERI [16]	✓	✓	✓	×	✓	×
	Delay bounding [45, 88]	×	✓	✓	×	✓	×
	Lumped C [109]	×	✓	✓	×	✓	×
	Single Pi [109]	×	✓	✓	×	✓	×
	Lumped RC [18]	×	✓	✓	×	✓	✓
	Proposed delay model	✓	✓	✓	×	✓	×

- When comparing against the reference timing data from ngspice, on average, the mean square error of sink insertion delays is improved by up to 98% over the Elmore delay model used in [81, 82].
- The model complexity remains unchanged compared to the Elmore delay model.
- Clock network buffer sizing results improve using the proposed delay model.

Finally, it is shown that the trained model can then be used to accurately estimate delays at the nodes of the clock trees synthesized in the same technology node.

Chapter 5

Efficient Clock Network Buffer Sizing With Slew Consideration

5.1 Introduction

The clock network often consumes about 40% of the power in an integrated circuit [82]. The reason for this high power consumption is that the switching activity and the capacitance load of the clock network are often high [17]. Then, designing a low power clock network is a critical challenge to meet the power consumption limit for the IC. On the other hand, clock skew minimization significantly improves the IC reliability and performance. Clock network buffer sizing is an important optimization stage in IC design flow. This stage determines the sizes of clock buffers so that the power consumption, clock slew and clock skew of the clock network are optimized.

In this chapter, a new and efficient clock network buffer sizing method is proposed as an optimization problem. Mathematically, the proposed clock network buffer sizing method is a multi-objective GP optimization formulation. The proposed formulation is designed to improve the power consumption and clock skew while considering clock slew and other technology constraints. The clock skew and power consumption are the optimization objectives of the proposed

optimization formulation.

The main contribution of this chapter is the development of an efficient multi-objective GP optimization formulation based on the multiplication of the problem objectives to solve the clock network buffer sizing problem. Runtime results show that the proposed formulation is scalable compared to the other existing clock network buffer sizing formulations.

The organization of the rest of this chapter is as follows: In Section 5.2, the mathematical formulation of the proposed optimization problem is presented. Then, in Section 5.3, the experimental results showing the efficiency of the proposed formulation are discussed and compared to other existing methods. Finally, a conclusion and summary of this chapter are given in Section 5.4.

5.2 The proposed clock network buffer sizing multi-objective formulation

Buffer insertion is a method to improve the timing and clock slew. However, it increases the area and clock network power [49]. The proposed clock network buffer sizing multi-objective formulation is an optimization problem which seeks to minimize clock skew and power consumption considering clock slew and other technology constraints. However, clock skew and power consumption are competing objectives. Therefore, the proposed formulation aims to find a balanced trade-off between the two competing design objectives. The objective functions and the inequality constraints are explained in the following subsections.

5.2.1 First design objective: power consumption

Circuit over-heating thermal breakdown and short battery life issues generally occur due to the high power consumption. Therefore, proposing low-power design techniques is critical in VLSI physical design. In clock network buffer sizing, the summation of areas of clock buffers can be

used as a proxy of clock network power consumption. Therefore, the mathematical model of the first optimization objective (power consumption) is given below based on the total buffer area:

$$\text{Area}(\mathbf{x}) = \sum_{k \in B} (b_{w_k} \times b_{l_k}) \tag{5.1}$$

where:

$\mathbf{x}$: the vector of all buffer widths and lengths (b_w, b_l).

The formulation aims to determine the values of the widths and the lengths of all the buffers leading to the minimum clock network buffer area.

5.2.2 Second design objective: clock skew

Clock skew is usually considered as a constraint as discussed in Chapter 3. However, power consumption minimization often increases the clock skew. Depending on the tightness of the clock skew constraints, the resultant sized clock networks may not be optimized in terms of clock skew. Clock skew optimization can be critical for clock network performance improvement. Also, in order to move to higher operating frequencies, reducing the clock skew is necessary.

In this chapter, clock skew is considered as an objective to be optimized. However, the clock skew model in (2.1) cannot be used directly in the GP format as an objective [82]. Therefore, an upper bound variable is defined in the objective function and a constraint is added to the optimization formulation to maintain the clock skew targets. The upper bound variable of the clock skew is defined as u_{skew} and the clock skew constraints are given below:

$$\text{S}inkDelay_i(\mathbf{x}) - SinkDelay_j(\mathbf{x}) \leq u_{\text{skew}} \ , \forall i, j \in S \tag{5.2}$$

5.2.3 Technology constraints

Technology constraints are applied to the clock network buffer sizing formulation to limit the minimum size of the buffers and account for the limitations in the fabrication technology process:

$$l_{min} \leq b_{l_k} \ , k \in B$$
$$w_{min} \leq b_{w_k} \ , k \in B$$

where:

w_{min}: the minimum allowed width of a buffer

l_{min}: the minimum allowed length of a buffer

5.2.4 The proposed multi-objective formulation

The main motivation for the proposed multi-objective formulation is to find a balanced trade-off between the two competing objective, power consumption and clock skew. In [82], a multi-objective buffer sizing model (D-MO) that minimizes the weighted sum of the objectives is proposed as:

$$\min \quad \alpha \times \mathrm{Area}(\mathbf{x}) + \beta \times u_{\mathrm{skew}} \tag{5.3}$$

where α and β are the multi-objective variable weights. One shortcoming of (5.3) is that this posynomial adds the multi-objective variable weights (i.e. α and β) to the optimization problem which slightly increases the problem complexity. In addition, since the units of the objectives are different, the weights need to compensate for the objective scales although this is done automatically by the multi-objective weight variables in [82]. Another requirement is that the equality constraint given in (5.4) should be added to the optimization problem to account for the inherent

trade-off between the objectives:

$$\alpha \times \beta = 1 \tag{5.4}$$

To address these shortcomings, in this chapter, it is proposed to adopt an optimization objective that includes the multiplication of area (as a proxy for power) and clock skew. This way, there is no need for extra multi-objective variables or objective scaling. The proposed multi-objective clock network buffer sizing optimization formulation is written as:

$$\begin{aligned} \min \quad & \mathrm{Area}(\mathbf{x}) \times u_{\mathrm{skew}} \\ \text{s.t.} \quad & SinkDelay_i(\mathbf{x}) - SinkDelay_j(\mathbf{x}) \leq u_{\mathrm{skew}} \ , \forall i, j \in S \\ & slew_n(\mathbf{x}) \leq t_{slew} \ , \forall n \in B \bigcup S \\ & l_{min} \leq b_{l_k} \ , k \in B \\ & w_{min} \leq b_{w_k} \ , k \in B \end{aligned} \tag{5.5}$$

where:

- $slew_n(\mathbf{x})$: the clock slew at node n
- t_{slew}: the maximum allowed value of clock slew

In the proposed formulation, the multi-objective weights and the weight constraint in the D-MO clock network buffer sizing in [82] are removed. Therefore, the complexity of the optimization problem is slightly lowered. By changing the summation of the weighted design objectives to the multiplication of the design objectives, both objectives can be simultaneously minimized without requiring any weight tuning or objective scaling while the problem can still be modeled as a GP and its convexity does not change. The experimental results in Section 5.3 demonstrate that the proposed formulation offer a better runtime compared to the existing works.

Another advantage of the proposed clock network buffer sizing formulation over the D-MO

method in [82] is that its objective function is a summation of multiplication of the size of each buffer and the clock skew upper bound variable. This means that the objective function is a posynomial with several monomial terms that all include the clock skew upper bound variable. However, in the D-MO method in [82], the objective function is a posynomial with several monomial terms where only one includes the clock skew upper bound variable. This advantage leads to a lower clock skew when clock network is sized using the proposed clock network buffer sizing. The numerical results in Section 5.3 show this.

5.2.5 GP optimization formulation

The proposed multi-objective optimization formulation in (5.5) can be transformed to the GP optimization formulation in (5.6) based on a method originally proposed in [81, 82] and explained in Chapter 3, to be solved efficiently using convex optimization techniques with a guarantee to achieve a global optimum solution.

$$
\begin{aligned}
\min \quad & \mathrm{Area}(\mathbf{x}) \times v_{\mathrm{skew}} = (\textstyle\sum_{k \in B} b_{w_k} \times b_{l_k}) \times v_{\mathrm{skew}} \\
\text{s.t.} \quad & SinkDelay_i(\mathbf{x}) \times (SinkDelay_{\min} \times v_{\mathrm{skew}})^{-1} \leq 1 \ , \forall i \in S \\
& slew_n(\mathbf{x}) \times (t_{\mathrm{slew}})^{-1} \leq 1 \ , \forall k \in B \bigcup S \\
& l_{min} \times (b_{l_k})^{-1} \leq 1 \ , k \in B \\
& w_{min} \times (b_{w_k})^{-1} \leq 1 \ , k \in B
\end{aligned}
\tag{5.6}
$$

where:

- $SinkDelay_{min}$ is the minimum clock signal arrival time at the clock sinks.
- u_{skew} is replaced with $v_{\mathrm{skew}} = (u_{\mathrm{skew}} + SinkDelay_{\min})/SinkDelay_{\min}$ to transform the skew constraint in (5.2) to a posynomial (GP format inequality constraint).

Converting the GP optimization formulation to a convex optimization problem

The proposed multi-objective formulation in (5.6) is a geometric program that needs to be converted to a convex optimization problem to be solved efficiently. According to [24], the first step for converting GP to a convex optimization includes variable transformations as in (5.7):

$$\begin{aligned}
&b_{w_k} = e^{y_{w_k}} \\
&b_{l_k} = e^{y_{l_k}} \\
&v_{\text{skew}} = e^{y_{\text{skew}}} \\
&\mathbf{x} = e^{\mathbf{z}} \\
&SinkDelay_{\min} = e^{d_{\min}} \\
&t_{\text{slew}} = e^{y_{\text{slew}}} \\
&l_{min} = e^{l_{min_y}} \\
&w_{min} = e^{w_{min_y}}
\end{aligned} \tag{5.7}$$

Then, the formulation in (5.6) can be transformed using the new variables and after taking a natural logarithm of the objective and constraint functions, it becomes:

$$\begin{aligned}
\min \quad & \ln((\textstyle\sum_{k\in B} e^{y_{w_k}} \times e^{y_{l_k}}) \times e^{y_{\text{skew}}}) \\
\text{s.t.} \quad & \ln(SinkDelay_i(e^{\mathbf{z}}) \times (e^{d_{\min}} \times e^{y_{\text{skew}}})^{-1}) \leq \ln(1)\ , \forall i \in S \\
& \ln(slew_n(e^{\mathbf{z}}) \times (e^{y_{\text{slew}}})^{-1}) \leq \ln(1)\ , \forall k \in B \bigcup S \\
& \ln(e^{l_{min_y}} \times (e^{y_{l_k}})^{-1}) \leq \ln(1)\ , k \in B \\
& \ln(e^{w_{min_y}} \times (e^{y_{w_k}})^{-1}) \leq \ln(1)\ , k \in B
\end{aligned} \tag{5.8}$$

In (5.8), the transformed objective function is a convex function since it is a log-sum-exp function and in [24], it is shown that log-sum-exp functions are convex. The convex form of the porposed multi-objective buffer sizing formulation is presented in (5.9) where all the inequality

constraints are transformed to either log-sum-exp functions or affine functions which are convex. Then, the convexity condition for the proposed buffer sizing formulation in (5.9) is satisfied.

$$
\begin{aligned}
\min \quad & \ln \textstyle\sum_{k \in B} e^{y_{w_k} + y_{l_k} + y_{\text{skew}}} \\
\text{s.t.} \quad & \ln(SinkDelay_i(e^{\mathbf{z}}) \times (e^{d_{\min} + y_{\text{skew}}})^{-1}) \leq 0 \ , \forall i \in S \\
& \ln(slew_n(e^{\mathbf{z}}) \times (e^{y_{\text{slew}}})^{-1}) \leq 0 \ , \forall k \in B \bigcup S \\
& l_{min_y} - y_{l_k} \leq 0 \ , k \in B \\
& w_{min_y} - y_{w_k} \leq 0 \ , k \in B
\end{aligned}
\tag{5.9}
$$

The runtime results in Section 5.3 show the effectiveness and scalability of the proposed formulation in achieving an optimum solution.

5.3 Experimental analysis

5.3.1 Experimental setup

Designing low power clock networks that meet the clock skew constraints is an important challenge in VLSI design. To emphasize the importance of this challenge, the International Symposium on Physical Design (ISPD) organized two clock contest in 2009 [2] and 2010 [3]. These contests provide the most recent academic benchmarks to the researchers. The proposed GP optimization formulation is applied on a set of benchmarks from the 2009 and 2010 ISPD clock network synthesis contests and the details of initial circuits before sizing are given in Table 5.1. In this table, the clock networks' name, height, width and sink capacitance are presented in Columns 1 to 4. The number of sinks and number of buffers in the circuits are given in Columns 5 and 6, respectively. Power consumption and average clock skew are given in Columns 7 and 8, respectively. As stated in the 2009 ISPD clock network synthesis contest [2], the value of the clock slew upper bound (t_{slew}) is set to 100 ps.

Mosek 9.0 [5] is used to solve the GP optimization problem and the values of the power and clock skew are calculated by ngspice 24 [6]. C++ language is used to write the implementations of the formulation in this chapter. All the results are based on discrete sizing by rounding the continuous sizes from solving GP to discrete sizes. The experiments for D-MO buffer sizing in [82] are conducted on the same machine as the proposed clock network buffer sizing formulation. The machine features of machine are described in 4.3.1. The results of D-MO buffer sizing [82] experiments are reported to be compared against the proposed formulation. It is explained in [106] that variations make the clock skew improvement more difficult. Therefore, all the experimental results of this chapter are measured considering the variations in buffer sizes using Monte Carlo simulations.

Table 5.1: Circuit names and statistics of ISPD 2009 and 2010 initial clock networks.

Cir.	Height (mm)	Width (mm)	S.cap (pF)	$\|S\|$	$\|B\|$	Power (mW)	Skew (ps)
09-11	11	11	4.2	0.1k	3.5k	20	48
09-12	8	13	4.1	0.1k	3.5k	20	91
09-21	13	12	3.6	0.1k	3.6k	21	51
09-22	12	5	3.4	0.1k	2.1k	12	27
09-31	17	17	9.6	0.3k	7.8k	43	228
09-32	17	17	6.7	0.2k	5.9k	33	143
09-nb1	3	2	5.9	0.3k	1.3k	5	25
Avg.	-	-	-	-	-	22	88
10-1	8.0	8.0	19	1.1k	11.4k	50	5
10-2	7.0	7.0	39	2.2k	22.1k	99	5
10-3	0.5	0.5	18	1.2k	3.0k	11	9
10-4	2.7	2.7	12	1.8k	3.9k	14	4
10-5	2.5	2.5	5	1.0k	1.9k	7	29
10-6	0.9	0.9	13	9.8k	2.5k	9	26
10-7	1.4	1.4	18	1.9k	3.9k	14	5
10-8	1.8	1.6	13	1.1k	2.9k	10	11
Avg.	-	-	-	-	-	27	12

5.3.2 Experimental results

ISPD 2009

The experimental results on ISPD 2009 benchmarks are given in Table 5.2. In this table, comparisons of the improvement in power, clock skew and runtime of the resultant networks sized by the proposed formulation and D-MO buffer sizing formulation in [82] are given.

The maximum clock skew reduction is 183 ps which is seen for circuit 11. The average clock skew reduction is 67 ps (more than 70%). The improvement in power consumption is more than 50%, on average. These significant improvements in clock skew and power consumption show the effectiveness of the proposed formulation.

Generally, the runtime for the proposed formulation depends on the number of the buffers. The growth in the number of buffers increases the number of variables and constraints of the formulation which makes the average runtime longer. The average runtime for the proposed formulation is 23 s.

Table 5.2: ISPD 2009 resultant clock networks. Power and clock skew are calculated using ngspice. The reported power and clock skew values are based on the improvement over initial networks.

Cir.	Power consumption		Clock Skew		Runtime	
	D-MO [82]	Proposed	D-MO [82]	Proposed	D-MO [82]	Proposed
	%	%	(ps)	(ps)	(s)	(s)
09-11	46	**47**	16	**42**	49	**10**
09-12	53	53	39	**69**	18	**9**
09-21	48	**51**	2	**42**	37	**6**
09-22	45	**47**	-9	**18**	15	**3**
09-31	61	**62**	83	**183**	251	**93**
09-32	51	**54**	42	**105**	179	**34**
09-nb1	58	**62**	5	**10**	12	**3**
Avg.	52	**54**	25	**67**	80	**23**

ISPD2010

The proposed formulation is evaluated on the clock network synthesized by the best team in ISPD 2010 contest. These clock networks are synthesized with a single type of buffer and one type of wire. The sizes of buffers are all 30 (30 parallel buffers). The results are presented in Table 5.3. The improvements in power, clock skew and runtime for the clock networks sized by D-MO buffer sizing [82] and the proposed formulation are compared. The maximum power reduction using the proposed formulation is 86% observed for circuit 2. The average runtime is improved significantly compared to buffer sizing formulation in [82]. On the other hand, the proposed formulation is effective in improving the power consumption by more than 50%, on average.

Table 5.3: ISPD 2010 resultant clock networks. Power and clock skew are calculated using ngspice. They are measured based on the improvement over initial networks.

Cir.	Power consumption		Clock Skew		Runtime	
	D-MO [82]	Proposed	D-MO [82]	Proposed	D-MO [82]	Proposed
	%	%	(ps)	(ps)	(s)	(s)
10-1	56	52	-11	-10	146	**75**
10-2	53	**86**	-12	-18	711	**255**
10-3	51	48	-1	-1	69	**16**
10-4	56	53	-5	-5	111	**36**
10-5	52	50	-4	-3	47	**14**
10-6	54	51	-1	-1	51	**18**
10-7	56	51	-4	-3	120	**43**
10-8	51	49	-2	-1	48	**16**
Avg.	54	**55**	-5	-5	163	**59**

Clock slew constraints

In high performance clock network designs, the clock slew rate affects the timing delays and circuit performance significantly [40]. Fig. 5.1 represents the nonlinear effect of the input clock slew on the buffer delay.

In Table 5.4, the maximum clock slew and average clock slew in all the nodes of ISPD 2009 and 2010 clock networks sized by the proposed formulation are given. In this table:

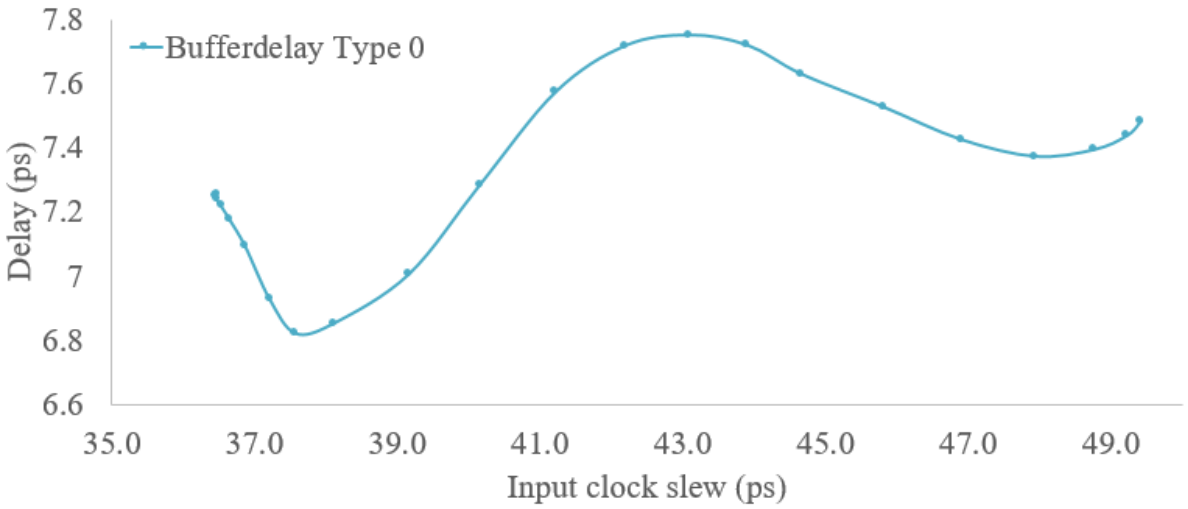

Figure 5.1: Delay plotted by the input clock slew

- The first column represents the name of the clock networks.
- The second column represents the maximum clock slew in the clock networks.
- The third column shows the average clock slew in all the nodes of the clock networks.

Based on the requirements of the ISPD clock network synthesis contests, the target clock slew is set to 100 ps. The average clock slew for ISPD 2009 clock networks sized by the proposed clock network buffer sizing formulation is 50 ps. The only case that has a maximum clock slew larger than 100 ps (i.e. clock slew violation) is circuit 09-31. The reason is that the clock slew constraints are optimized using the delay model by the optimization solver. On the other hand, the reported results in Table 5.4 are evaluated using ngspice circuit simulator. This violation is due to the inherent inaccuracy of the delay model compared to ngspice. However, in most cases, the delay model and the proposed buffer sizing formulation are effective and the clock slew constraints requirements are met when measured by ngspice. For the ISPD 2010 clock networks, which are larger circuits, the clock slew constraints are met and the average clock slew in clock networks is 35 ps. All the sized 2010 clock networks meet the clock slew requirements. This improves not

Table 5.4: Clock slew in ISPD 2009 and 2010 sized clock networks.

Cir.	Clock Slew$_{Max}$(ps)	Clock Slew$_{Avg}$(ps)
09-11	65	44
09-12	70	45
09-21	72	49
09-22	66	41
09-31	161	74
09-32	86	49
09-nb1	61	46
Avg.	83	50
10-1	71	35
10-2	66	33
10-3	57	32
10-4	62	36
10-5	68	46
10-6	46	30
10-7	62	35
10-8	47	32
Avg.	60	35

only the quality of the clock signal but also the performance of the circuit.

5.4 Summary

In this chapter, clock network buffer sizing problem is studied and analyzed. A new multi-objective framework is developed to optimize the upper bound of the clock skew and the area of the clock buffers simultaneously. GP optimization is used to solve this constrained optimization problem as a convex problem. The clock slew target is met in almost all the benchmarks with one exception. Technology fabrication limitations are applied in the formulation and after clock network buffer sizing, all the continuous sizes of the buffers are rounded to the nearest discrete value. On average, the improvement in power consumption is more than 50% which is significant for low power designs.

The runtime results of the proposed formulation are compared with an existing work in the

literature and show significant improvement by removing the extra constraints and developing the new objective formulation.

Chapter 6

Conclusion and Future Work

6.1 Summary and Contributions

With the growing transistor density in digital ICs, the use of complex timing simulators is becoming more limited to the final digital sign-off stages. On the other hand, using faster mathematical delay estimation models are becoming more popular during optimization. However, the existing delay models quickly lose accuracy as the technology nodes advance towards smaller feature sizes. This emphasizes the importance of enhanced mathematical delay models that are able to predict clock delay more accurately and adapt to the new technology nodes.

To this end, in Chapter 4, least squares fitting is applied to improve the accuracy and adaptability of the mathematical delay estimation model. The developed delay model efficiently improves the clock signal delay prediction. It also improves the result of the clock network buffer sizing optimization problems. The results show that the high accuracy and scalability of the proposed delay model leads to improving the clock power consumption and improving the timing metrics such as clock skew and clock slew.

The other contribution of this thesis is developing a new efficient clock network buffer sizing method which is a multi-objective formulation. The proposed clock network buffer sizing method

considers slew constraints and improves the power consumption, clock skew and runtime results compared to the existing methods in the literature. These contributions are published as a research paper in [40].

The list of the main contributions of this thesis is given below:

- Proposing an accurate delay model for clock signal to be used in clock network synthesis.
- Significant improvement in the sink delay prediction.
- Improvement in the critical sink delay estimations (the minimum and maximum sink delay estimations).
- GP compatibility of the proposed clock network delay model.
- Low runtime compared to the *ngspice* circuit simulator.
- Applying the proposed delay model to a clock network buffer sizing algorithm which results in improvements in the power consumption and the clock skew.
- Developing an efficient clock network buffer sizing method optimizing power consumption and clock skew while meeting the slew constraints.
- Significant power consumption, clock skew and runtime improvement using the proposed clock network buffer sizing method compared to the existing methods in the literature.

6.2 Future Works

A future work can include development separate models for different delay ranges, e.g. the linear domain and the non-linear domains. Another future direction may be applying the proposed delay model to the clock wire sizing problem. Researchers may pursue this direction to improve the clock wire sizing quality of results by integrating the proposed delay model.

Bibliography

[1] Cadence. https://www.cadence.com.

[2] ISPD 2009 high performance clock network synthesis contest. http://www.sigda.org/ispd/contests/09/ispd09cts.html.

[3] ISPD 2010 high performance clock network synthesis contest. http://www.sigda.org/ispd/contests/10/ispd10cns.html.

[4] Mentor graphics software. https://www.mentor.com.

[5] Mosek 9.0. http://www.mosek.com.

[6] ngspice 24. http://ngspice.sourceforge.net/.

[7] Synopsys. https://www.synopsys.com.

[8] Matlab: Version r2018a, 2018. https://www.mathworks.com.

[9] S. Abbaspour, M. Pedram, A. Ajami, and C. Kashyap. Fast interconnect and gate timing analysis for performance optimization. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 14(12):1383–1388, Dec 2006.

[10] A. Abdollahi and F. Fallah. Runtime mechanisms for leakage current reduction in cmos vlsi circuits. In *Low Power Electronics and Design, 2002. ISLPED '02.*, pages 213–218, Aug 2002.

[11] A. I. Abou-Seido, B. Nowak, and C. Chu. Fitted elmore delay: a simple and accurate interconnect delay model. In *Proceedings. IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pages 422–427, 2002.

[12] S. Ahn, M. Kang, M. C. Papaefthymiou, and T. Kim. Synthesis of resonant clock networks supporting dynamic voltage / frequency scaling. In *The 20th Asia and South Pacific Design Automation Conference*, pages 484–489, Jan 2015.

[13] C. J. Alpert, A. Devgan, and C. Kashyap. A two moment rc delay metric for performance optimization. In *Proceedings of the 2000 International Symposium on Physical Design*, ISPD '00, pages 69–74, New York, NY, USA, 2000. ACM.

[14] C. J. Alpert, A. Devgan, and C. V. Kashyap. Rc delay metrics for performance optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20(5):571–582, May 2001.

[15] C. J. Alpert, F. Liu, C. Kashyap, and A. Devgan. Delay and slew metrics using the lognormal distribution. In *Proceedings 2003. Design Automation Conference*, pages 382–385, June 2003.

[16] C. J. Alpert, F. Liu, C. V. Kashyap, and A. Devgan. Closed-form delay and slew metrics made easy. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23(12):1661–1669, Dec 2004.

[17] K. Athikulwongse, X. Zhao, and S. Lim. Buffered clock tree sizing for skew minimization under power and thermal budgets. In *Proceedings of the Asia and South Pacific Design Automation Conference*, pages 474–479, 2010.

[18] M. Avci and S. Yamacli. An improved elmore delay model for vlsi interconnects. *Mathematical and Computer Modelling*, vol. 51:908–914, 04 2010.

[19] H. Bakoglu. *Circuits, Interconnections, and Packaging for VLSI*. Addison–Wesley, 1990.

[20] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb. A simulated annealing-based multiobjective optimization algorithm: Amosa. *IEEE Transactions on Evolutionary Computation*, vol. 12(3):269–283, June 2008.

[21] D. Blaauw, S. Sirichotiyakul, and C. Oh. Driver modeling and alignment for worst-case delay noise. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11(2):157–166, April 2003.

[22] K. D. Boese and A. B. Kahng. Zero-skew clock routing trees with minimum wirelength. In *[1992] Proceedings. Fifth Annual IEEE International ASIC Conference and Exhibit*, pages 17–21, Sep 1992.

[23] M. Borah, R. M. Owens, and M. J. Irwin. A fast algorithm for minimizing the elmore delay to identified critical sinks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16(7):753–759, July 1997.

[24] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[25] W C. Elmore. The transient response of damped linear networks with particular regard to wideband amplifiers. 19:55 – 63, 02 1948.

[26] M. Celik, L. Pileggi, and A. Odabasioglu. *IC Interconnect Analysis*. Kluwer Academic Publishers, 2002.

[27] T. Chan, A. B. Kahng, and J. Li. Nolo: A no-loop, predictive useful skew methodology for improved timing in ic implementation. In *Fifteenth International Symposium on Quality Electronic Design*, pages 504–509, March 2014.

[28] H. Chang and S. S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single pert-like traversal. In *ICCAD-2003. International Conference on Computer Aided Design*, pages 621–625, Nov 2003.

[29] H. Chang, V. Zolotov, S. Narayan, and C. Visweswariah. Parameterized block-based statistical timing analysis with non-gaussian parameters, nonlinear delay functions. In *Proceedings. 42nd Design Automation Conference, 2005.*, pages 71–76, June 2005.

[30] T. H. Chao, Y. C. Hsu, and J. M. Ho. Zero skew clock net routing. In *[1992] Proceedings 29th ACM/IEEE Design Automation Conference*, pages 518–523, Jun 1992.

[31] C. Chen, X. Yang, and M. Sarrafzadeh. Potential slack: An effective metric of combinational circuit performance. In *Proceedings of the 2000 IEEE/ACM International Conference on Computer-aided Design*, ICCAD '00, pages 198–201, Piscataway, NJ, USA, 2000. IEEE Press.

[32] H. Chou, H. Yu, and S. Chang. Useful-skew clock optimization for multi-power mode designs. In *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 647–650, Nov 2011.

[33] J. Cong, C. Koh, and K. Leung. Simultaneous buffer and wire sizing for performance and power optimization. In *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 271–276, 1996.

[34] J. F. Croix and D. F. Wong. Blade and razor: cell and interconnect delay analysis using current-based models. In *Proceedings 2003. Design Automation Conference*, pages 386–389, June 2003.

[35] F. Dartu, N. Menezes, and L. T. Pileggi. Performance computation for precharacterized cmos gates with rc loads. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15(5):544–553, May 1996.

[36] F. Dartu, N. Menezes, J. Qian, and L. T. Pillage. A gate-delay model for high-speed cmos circuits. In *31st Design Automation Conference*, pages 576–580, June 1994.

[37] S. Do, S. Kim, and S. Kang. Skew control methodology for useful-skew implementation. In *2016 International SoC Design Conference (ISOCC)*, pages 221–222, Oct 2016.

[38] V. Dvorák. Relation between the delay at direct current and elmore's delay. *Electronics Letters*, vol. 8(15):377–378, July 1972.

[39] M. Edahiro. A clustering-based optimization algorithm in zero-skew routings. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 612–616, 1993.

[40] Ali Farshidi, Logan Rakai, and Laleh Behjat. An efficient optimal clock network buffer sizing with slew consideration. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–4, April 2017.

[41] Amin Farshidi, Laleh Behjat, Logan Rakai, and David Westwick. A multiobjective cooptimization of buffer and wire sizes in high-performance clock trees. *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64(4):412–416, April 2017.

[42] Amin Farshidi, Logan Rakai, Laleh Behjat, and David Westwick. Variation-aware clock network buffer sizing using robust multi-objective optimization. *Optimization and Engineering*, vol. 17(2):473–500, Jun 2016.

[43] R. Fletcher. *Practical Methods of Optimization; (2Nd Ed.)*. Wiley-Interscience, New York, NY, USA, 1987.

[44] Y. Gao and D. F. Wong. A fast and accurate delay estimation method for buffered interconnects. In *Proceedings of the ASP-DAC 2001. Asia and South Pacific Design Automation Conference*, pages 533–538, Feb 2001.

[45] R. Gupta. The elmore delay as a bound for rc trees with generalized input signals. In *32nd Design Automation Conference*, pages 364–369, June 1995.

[46] M. Guthaus, D. Sylvester, and R. Brown. Process-induced skew reduction in nominal zero-skew clock trees. In *Proceedings of the ACM/IEEE Asia and South Pacific Design Automation Conference*, pages 84–89, 2006.

[47] K. Han, A. B. Kahng, J. Lee, J. Li, and S. Nath. A global-local optimization framework for simultaneous multi-mode multi-corner clock skew variation reduction. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, June 2015.

[48] R. Islam and M. R. Guthaus. Cmcs: Current-mode clock synthesis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, PP(99):1–9, 2016.

[49] A. Kahng, J. Lienig, I. Markov, and J. Hu. *VLSI Physical Design: From Graph Partitioning to Timing Closure*. Springer, 1st edition, 2011.

[50] A. B. Kahng, S. Kang, H. Lee, S. Nath, and J. Wadhwani. Learning-based approximation of interconnect delay and slew in signoff timing tools. In *2013 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP)*, pages 1–8, June 2013.

[51] A. B. Kahng and S. Muddu. An analytical delay model for rlc interconnects. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16(12):1507–1514, Dec 1997.

[52] C. V. Kashyap, C. J. Alpert, and A. Devgan. An ”effective” capacitance based delay metric for rc interconnect. In *IEEE/ACM International Conference on Computer Aided Design. ICCAD - 2000. IEEE/ACM Digest of Technical Papers*, pages 229–234, Nov 2000.

[53] C. V. Kashyap, C. J. Alpert, F. Liu, and A. Devgan. Peri: A technique for extending delay and slew metrics to ramp inputs. In *Proceedings of the 8th ACM/IEEE International Work-*

shop on Timing Issues in the Specification and Synthesis of Digital Systems, TAU '02, pages 57–62, New York, NY, USA, 2002. ACM.

[54] C. V. Kashyap, C. J. Alpert, F. Liu, and A. Devgan. Closed-form expressions for extending step delay and slew metrics to ramp inputs for rc trees. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(4):509–516, April 2004.

[55] R. Kay and L. Pileggi. Primo: Probability interpretation of moments for delay calculation. In *Proceedings of the 35th Annual Design Automation Conference*, DAC '98, pages 463–468, New York, NY, USA, 1998. ACM.

[56] D. C. Keezer and V. K. Jain. Design and evaluation of wafer scale clock distribution. In *[1992] Proceedings International Conference on Wafer Scale Integration*, pages 168–175, Jan 1992.

[57] I. Keller, K. Tseng, and N. Verghese. A robust cell-level crosstalk delay change analysis. In *IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004.*, pages 147–154, Nov 2004.

[58] M. C. Ketkar. *Delay modeling and optimization in VLSI circuit synthesis*. PhD thesis, 2002. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2016-05-10.

[59] J. Kim and T. Kim. Adjustable delay buffer allocation under useful clock skew scheduling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36(4):641–654, April 2017.

[60] S. Kim, S. Do, and S. Kang. Fast predictive useful skew methodology for timing-driven placement optimization. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, June 2017.

[61] T. Y. Kim and T. Kim. Clock tree embedding for 3d ics. In *2010 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 486–491, Jan 2010.

[62] S. Lerner, E. Leggett, and B. Taskin. Slew-down: analysis of slew relaxation for low-impact clock buffers. In *2017 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP)*, pages 1–4, June 2017.

[63] P. Li, Z. Feng, and E. Acar. Characterizing multistage nonlinear drivers and variability for accurate timing and noise analysis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15(11):1205–1214, Nov 2007.

[64] X. Li, J. Mao, and M. Swaminathan. Transient analysis of cmos-gate-driven$rlgc$interconnects based on fdtd. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30(4):574–583, April 2011.

[65] Z. Li and S. Chen. Transistor level timing analysis considering multiple inputs simultaneous switching. In *2007 10th IEEE International Conference on Computer-Aided Design and Computer Graphics*, pages 315–320, Oct 2007.

[66] M. Lin, H. Sun, and S. Kimura. Power-efficient and slew-aware three dimensional gated clock tree synthesis. In *2016 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, pages 1–6, Sept 2016.

[67] T. Lin, E. Acar, and L. Pileggi. h-gamma: an rc delay metric based on a gamma distribution approximation of the homogeneous response. In *1998 IEEE/ACM International Conference on Computer-Aided Design. Digest of Technical Papers*, pages 19–25, Nov 1998.

[68] F. Liu, C. Kashyap, and C. J. Alpert. A delay metric for rc circuits based on the weibull distribution. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23(3):443–447, March 2004.

[69] J. Lu, W. Chow, and C. Sham. Fast power- and slew-aware gated clock tree synthesis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(11):2094–2103, Nov 2012.

[70] A. K. Mal and A. S. Dhar. Modified elmore delay model for vlsi interconnect. In *2010 53rd IEEE International Midwest Symposium on Circuits and Systems*, pages 793–796, Aug 2010.

[71] Juan C. Meza. Newton's method. *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 3(1):75–78.

[72] K. Moiseev, A. Kolodny, and Sh. Wimer. *Multi-Net Optimization of VLSI Interconnect*. Springer Publishing Company, Incorporated, 2014.

[73] G. E. Moore. Cramming more components onto integrated circuits, reprinted from electronics, vol. 38, number 8, april 19, 1965, pp.114 ff. *IEEE Solid-State Circuits Society Newsletter*, 11(3):33–35, Sept 2006.

[74] A. Odabasioglu, M. Celik, and L. T. Pileggi. Prima: passive reduced-order interconnect macromodeling algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17(8):645–654, Aug 1998.

[75] P. Penfield and J. Rubinstein. Signal delay in rc tree networks. In *18th Design Automation Conference*, pages 613–617, June 1981.

[76] L. T. Pillage and R. A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9(4):352–366, April 1990.

[77] P. Punia, Rouble, N. Shukla, and M. Singh. Minimizing skew and delay with buffer resizing and relocation during clock tree synthesis. *International Journal of Computer Applications*, vol. 95(23):30–35, 2014.

[78] R. Puri, D. Kung, and D. Drumm. Fast and accurate wire delay estimation for physical synthesis of large asics. In *Proceedings of the 12th ACM Great Lakes Symposium on VLSI*, GLSVLSI '02, pages 30–36, New York, NY, USA, 2002. ACM.

[79] J. Qian, S. Pullela, and L. Pillage. Modeling the "effective capacitance" for the rc interconnect of cmos gates. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13(12):1526–1535, Dec 1994.

[80] S. Raja, F. Varadi, M. Becer, and J. Geada. Transistor level gate modeling for accurate and fast timing, noise, and power analysis. In *2008 45th ACM/IEEE Design Automation Conference*, pages 456–461, June 2008.

[81] Logan Rakai, Amin Farshidi, Laleh Behjat, and David Westwick. Buffer sizing for clock networks using robust geometric programming considering variations in buffer sizes. In *Proceedings of ISPD*, pages 154–161, 2013.

[82] Logan Rakai, Amin Farshidi, David Westwick, and Laleh Behjat. Variation-aware geometric programming models for the clock network buffer sizing problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33(4):532–545, 2014.

[83] U. Ramadass, I. , Krishnappriya, J. Ponnian, and P Dhavachelvan. A novel interconnect structure for elmore delay model with resistance-capacitance-conductance scheme. *American Journal of Applied Sciences*, vol. 10:881–892, 07 2013.

[84] U. Ramadass, I. Krishnappriya, and P. Dhavachelvan. New modified elmore delay model for resistance-capacitance-conductance (rcg) interconnect network scheme. *Journal of Theoretical and Applied Information Technology*, vol. 54:361–371, 08 2013.

[85] P. Ramanathan, A. J. Dupont, and K. G. Shin. Clock distribution in general vlsi circuits. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol, 41(5):395–404, May 1994.

[86] C. L. Ratzlaff and L. T. Pillage. Rice: rapid interconnect circuit evaluation using awe. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13(6):763–776, June 1994.

[87] A. Rjoub and A. B. Alajlouni. Graph modeling for static timing analysis at transistor level in nano-scale cmos circuits. In *2012 16th IEEE Mediterranean Electrotechnical Conference*, pages 80–83, March 2012.

[88] J. Rubinstein, P. Penfield, and M. A. Horowitz. Signal delay in rc tree networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2(3):202–211, July 1983.

[89] A. E. Ruehli and J. Hayes. Solver for current source type driver(s) and interconnect with linear or nonlinear loads. In *2009 IEEE 18th Conference on Electrical Performance of Electronic Packaging and Systems*, pages 89–92, Oct 2009.

[90] S. Saini, A. M. Kumar, S. Veeramachaneni, and M. B. Srinivas. An alternative approach to buffer insertion for delay and power reduction in vlsi interconnects. In *2010 23rd International Conference on VLSI Design*, pages 411–416, Jan 2010.

[91] T. Sakurai and A. R. Newton. Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas. *IEEE Journal of Solid-State Circuits*, vol. 25(2):584–594, April 1990.

[92] P. Saxena, N. Menezes, P. Cocchini, and D. A. Kirkpatrick. Repeater scaling and its impact on cad. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23(4):451–463, April 2004.

[93] N. Sherwani. *Algorithm for VLSI Physical Design Automation, Third Edition*. Kluwer Academic Publishers, Massachusetts, USA, 1999.

[94] A. Soltan, A. G. Radwan, and A. Yakovlev. Elmore delay in the fractional order domain. In *2017 European Conference on Circuit Theory and Design (ECCTD)*, pages 1–4, Sept 2017.

[95] H. Su, D. Widiger, C. Kashyap, F. Liu, and B. Krauter. A noise-driven effective capacitance method with fast embedded noise rule calculation for functional noise analysis. In *Proceedings 42nd Design Automation Conference, 2005*, pages 186–189, June 2005.

[96] M. S. Sulaiman. A balanced clock network design algorithm for clock delay, skew, and power optimization with slew rate constraint. In *ICONIP '02. Proceedings of the 9th International Conference on Neural Information Processing. Computational Intelligence for the E-Age*, pages 62–66, Dec 2002.

[97] S. A. Tawfik and V. Kursun. Buffer insertion and sizing in clock distribution networks with gradual transition time relaxation for reduced power consumption. In *2007 14th IEEE International Conference on Electronics, Circuits and Systems*, pages 845–848, Dec 2007.

[98] G. E. Tellez and M. Sarrafzadeh. Minimal buffer insertion in clock trees with skew and slew rate constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16(4):333–342, Apr 1997.

[99] J. Tsai, T. Chen, and CC-P Chen. Zero skew clock-tree optimization with buffer insertion/sizing and wire sizing. *IEEE Transactions on CAD*, vol. 23(4):565–572, 2004.

[100] R. Tsay. Exact zero skew. In *Proceedings of the IEEE International Conference on Computer Aided Design*, pages 336–339, 1991.

[101] R. S. Tsay. An exact zero-skew clock routing algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(2):242–249, Feb 1993.

[102] B. Tutuianu, R. Baldick, and M. S. Johnstone. Nonlinear driver models for timing and noise analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23(11):1510–1521, Nov 2004.

[103] B. Tutuianu, F. Dartu, and L. Pileggi. An explicit rc-circuit delay approximation based on the first three moments of the impulse response. In *33rd Design Automation Conference Proceedings, 1996*, pages 611–616, June 1996.

[104] L. van Ginneken. Buffer placement in distributed RC-tree networks for minimal elmore delay. In *Proceedings of ISCAS*, volume vol. 2, pages 865–868, 1990.

[105] V. Veetil, D. Sylvester, and D. Blaauw. Fast and accurate waveform analysis with current source models. In *9th International Symposium on Quality Electronic Design (isqed 2008)*, pages 53–56, March 2008.

[106] G. Venkataraman, N. Jayakumar, J. Hu, P. Li, S. Khatri, A. Rajaram, P. McGuinness, and C. Alpert. Practical techniques to reduce skew and its variations in buffered clock networks. In *Proceedings of the 2005 IEEE/ACM International Conference on Computer-aided Design*, ICCAD '05, pages 592–596, Washington, DC, USA, 2005. IEEE Computer Society.

[107] K. Wang and M. Marek-Sadowska. Buffer sizing for clock power minimization subject to general skew constraints. In *Proceedings of the Design Automation Conference*, pages 159–164, 2004.

[108] Z. Wang. *Accurate Wire Endpoint Delay and Slope Estimation*. 2012.

[109] Z. Wang and C. Sechen. A new algorithm for accurate wire endpoint delay estimation. *Journal of Algorithms and Optimization*, vol. 2(2):1–14, 2014.

[110] L. Xiu. *VLSI Circuit Design Methodology Demystified: A Conceptual Taxonomy*. Wiley-IEEE Press, 2007.

[111] K. Yamakoshi and M. Ino. Generalised elmore delay expression for distributed rc tree networks. *Electronics Letters*, vol. 29(7):617–618, April 1993.

[112] P. Zarkesh-Ha, T. Mule, and J. D. Meindl. Characterization and modeling of clock skew with process variations. In *Proceedings of the IEEE 1999 Custom Integrated Circuits Conference*, pages 441–444, 1999.

Appendix A

Copyright Permissions

In this appendix, the copyright permissions related to Chapter 5 are given:

From: Laleh Behjat

Sent: Fri 8/9/2019 5:47 AM

To: Ali Farshidi

Subject: Re: Copyright permission from co-authors

Ali,

Sorry for the delay in responding. I give permission for this.

Best,

Laleh

From: logan rakai

Sent: Wed 8/7/2019 10:18 PM

To: Ali Farshidi

Subject: Re: Copyright permission from co-authors

Permission granted

——— Original message ———

From: Ali Farshidi

Date: Wed 8/7/2019 9:22 PM

To: logan rakai , Laleh Behjat

Subject: Copyright permission from co-authors

Hi Dr. Behjat, Dr. Rakai,

I am going to ask Faculty of Graduate Studies to put my thesis on the Vault. Since part of thesis, Chapter 5, was part of a manuscript that you also co-authored, I need permissions from you as well. Please reply to this email indicating your permission.

Thanks in advance,

Ali