

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM MICROELETRÔNICA

GUILHERME AUGUSTO FLACH

## **Clock Mesh Optimization**

Thesis presented in partial fulfillment  
of the requirements for the degree of  
Master on Microelectronics

Marcelo de Oliveira Johann  
Advisor

Ricardo Augusto da Luz Reis  
Coadvisor

Porto Alegre, May 2010

## CIP – CATALOGING-IN-PUBLICATION

Flach, Guilherme Augusto

Clock Mesh Optimization / Guilherme Augusto Flach. – Porto Alegre: PGMICRO da UFRGS, 2010.

75 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Microeletrônica, Porto Alegre, BR–RS, 2010. Advisor: Marcelo de Oliveira Johann; Coadvisor: Ricardo Augusto da Luz Reis.

1. Clock. 2. Clock mesh. 3. Skew. 4. Performance. 5. Micro-processor. 6. Variability. 7. Microelectronic. I. Johann, Marcelo de Oliveira. II. Reis, Ricardo Augusto da Luz. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Aldo Bolten Lucion

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do PGMICRO: Prof. Ricardo Augusto da Luz Reis

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*To the beauty of this world, despite its ugliness.*



## **ACKNOWLEDGMENTS**

To my mom and my stepfather for their endless support... To my father who taught me my first computer lessons... And also special thanks to Gustavo Wilke for his unmeasurable contribution to this work.



# CONTENTS

<b>LIST OF ABBREVIATIONS AND ACRONYMS</b> . . . . .	11
<b>LIST OF SYMBOLS</b> . . . . .	13
<b>LIST OF FIGURES</b> . . . . .	15
<b>LIST OF TABLES</b> . . . . .	17
<b>ABSTRACT</b> . . . . .	19
<b>RESUMO</b> . . . . .	21
<b>1 INTRODUCTION</b> . . . . .	23
<b>1.1 Clock Signal</b> . . . . .	23
<b>1.2 Clock Network</b> . . . . .	23
1.2.1 Signal Regeneration . . . . .	24
<b>1.3 Clock Mesh</b> . . . . .	25
<b>1.4 Motivation</b> . . . . .	26
<b>1.5 Contributions</b> . . . . .	26
<b>2 TIMING CONSTRAINTS IN SYNCHRONOUS CIRCUITS</b> . . . . .	29
<b>2.1 Clock Signal</b> . . . . .	29
<b>2.2 Sequential Components: FlipFlops</b> . . . . .	29
<b>2.3 Combinational Paths</b> . . . . .	30
<b>2.4 Clock Signal Uncertainty</b> . . . . .	30
2.4.1 Clock Skew . . . . .	30
2.4.2 Clock Jitter . . . . .	30
<b>2.5 Clock Period Definition</b> . . . . .	31
2.5.1 Ideal Scenario: Neither Skew nor Jitter . . . . .	31
2.5.2 Introducing Clock Skew . . . . .	31
2.5.3 Introducing Clock Jitter . . . . .	32
2.5.4 Putting All Together . . . . .	32
<b>3 SOURCES OF SKEW AND JITTER</b> . . . . .	35
<b>3.1 Summary of Skew and Jitter Sources</b> . . . . .	36

<b>4</b>	<b>A BRIEF REVIEW ON COMMON CLOCK DISTRIBUTION ARCHITECTURES</b>	39
<b>4.1</b>	<b>Single-Path Networks</b>	39
4.1.1	Clock Tree	39
4.1.2	H-Tree	40
4.1.3	Spine-Tree	41
<b>4.2</b>	<b>Multiple-Path Networks</b>	41
4.2.1	Cross-Link	41
4.2.2	Clock Mesh	41
<b>4.3</b>	<b>Tree + Local Mesh (TLM)</b>	42
<b>5</b>	<b>RELATED WORKS</b>	45
<b>5.1</b>	<b>Combinatorial Algorithms for Fast Clock Mesh Optimization</b>	45
5.1.1	Simultaneous Mesh Buffer Placement and Sizing	45
5.1.2	Mesh Reduction	46
5.1.3	Buffer Model	46
<b>5.2</b>	<b>MeshWorks: An Efficient Framework for Planning, Synthesis and Optimization of Clock Mesh Networks</b>	47
5.2.1	Mesh Size Definition	47
5.2.2	Buffer Placement and Sizing	47
5.2.3	Edge Removal	47
5.2.4	Buffer Model	48
<b>6</b>	<b>MESH SIZE SELECTION</b>	49
<b>6.1</b>	<b>Definitions</b>	49
<b>6.2</b>	<b>Motivation</b>	50
<b>6.3</b>	<b>Average Stub Length</b>	50
<b>6.4</b>	<b>Mesh Size for Minimum Wirelength</b>	51
<b>6.5</b>	<b>Mesh Size for Minimum Capacitance</b>	52
<b>6.6</b>	<b>Optimum Mesh Bin Density</b>	52
<b>6.7</b>	<b>Experiments</b>	52
6.7.1	General Simulation Setup	53
6.7.2	Specific Simulation Setup	53
6.7.3	Result Analysis	53
<b>6.8</b>	<b>Conclusions</b>	54
<b>7</b>	<b>SKEW REDUCTION BY MESH BUFFER DISPLACEMENT</b>	57
<b>7.1</b>	<b>Elmore Delay</b>	57
7.1.1	Conductance Matrix Definition	57
<b>7.2</b>	<b>Single-dimension displacement optimization</b>	58
7.2.1	Terms and Conventions	59
7.2.2	Problem Definition	59
7.2.3	Buffer Displacement and Sink Delay Computation	60
7.2.4	Sink Delay Sensitivities	60
7.2.5	Sink Delay Estimation using Sensitivities	60
7.2.6	Skew Reduction as a Linear Program	61
<b>7.3</b>	<b>Two-dimensional displacement optimization</b>	62
<b>7.4</b>	<b>Experiments</b>	62



7.4.1	Setup . . . . .	62
7.4.2	Results . . . . .	63
<b>7.5</b>	<b>Conclusion . . . . .</b>	<b>64</b>
<b>8</b>	<b>MESH EXPLORER: A CLOCK MESH TOOL . . . . .</b>	<b>65</b>
<b>8.1</b>	<b>Programming Language and External Libraries . . . . .</b>	<b>65</b>
<b>8.2</b>	<b>Key Features . . . . .</b>	<b>65</b>
8.2.1	Spice Simulation . . . . .	65
8.2.2	Elmore Simulation . . . . .	66
8.2.3	Sink Properties Visualization . . . . .	66
8.2.4	Changing Buffer and Sink Properties . . . . .	68
8.2.5	Dragging Buffers . . . . .	68
8.2.6	Mesh Optimization Through Buffer Displacement . . . . .	68
<b>9</b>	<b>CONCLUSIONS . . . . .</b>	<b>71</b>
<b>9.1</b>	<b>Future Work . . . . .</b>	<b>71</b>
	<b>REFERENCES . . . . .</b>	<b>73</b>



## **LIST OF ABBREVIATIONS AND ACRONYMS**

GUI	Graphical User Interface
FF	Flip-Flop
FO	Fanout Of
ASIC	Application Specific Integrated Circuit
TLM	Tree + Local Meshes



## LIST OF SYMBOLS

$f$	Femto
$\mu$	Micron/Mean
$m$	Milli
$n$	Nano
$\Omega$	Ohms
$p$	Pico
$\sum$	Summation
$\sigma$	Standard deviation



## LIST OF FIGURES

Figure 1.1:	The clock signal - a periodic signal that synchronizes the circuit components. . . . .	23
Figure 1.2:	Clock signal spanned over all circuit area. . . . .	24
Figure 1.3:	Clock mesh is a grid wire to where clock sinks connect to. . . . .	25
Figure 2.1:	Timing diagram for a positive edge-triggered flipflop (JAN M. RABAEY ANANTHA P. CHANDRAKASAN, 2002) . . . . .	29
Figure 2.2:	A combinational path. . . . .	30
Figure 2.3:	Positive Clock Skew . . . . .	32
Figure 2.4:	Negative Clock Skew . . . . .	32
Figure 2.5:	Clock Jitter . . . . .	32
Figure 3.1:	Clock sinks with different delays due to different RC. . . . .	35
Figure 3.2:	Sources of skew and jitter (JAN M. RABAEY ANANTHA P. CHANDRAKASAN, 2002). . . . .	36
Figure 4.1:	Example of different topologies for a same set of sinks. . . . .	40
Figure 4.2:	H-Tree. . . . .	40
Figure 4.3:	Fishbone. . . . .	41
Figure 4.4:	Spine Tree. . . . .	41
Figure 4.5:	Cross-Link. . . . .	42
Figure 4.6:	Clock Mesh. . . . .	42
Figure 4.7:	Tree + Local Mesh (TLM). . . . .	43
Figure 5.1:	Greedy set cover for mesh buffer placement and sizing (VENKATARAMAN et al., 2006). . . . .	45
Figure 5.2:	Mesh reduction algorithm (VENKATARAMAN et al., 2006). . . . .	46
Figure 5.3:	Driver model used in (VENKATARAMAN et al., 2006). . . . .	46
Figure 5.4:	Algorithm to find a good initial mesh size (RAJARAM; PAN, 2008). . . . .	48
Figure 5.5:	Buffer model used in (RAJARAM; PAN, 2008). . . . .	48
Figure 6.1:	A $4 \times 4$ Clock Mesh . . . . .	49
Figure 6.2:	Mesh wirelength as a function of the mesh size. . . . .	50
Figure 6.3:	Mesh capacitance as a function of the mesh size. . . . .	51
Figure 6.4:	An internal bin region where stubs have the same length $x$ . . . . .	51
Figure 6.5:	The coverage region of a buffer. . . . .	53
Figure 6.6:	Skew and capacitance trade-off for no input skew. . . . .	54
Figure 6.7:	Skew and capacitance trade-off when input skew is applied. . . . .	54

Figure 7.1:	A Simple RC Circuit . . . . .	58
Figure 7.2:	Multi-step optimization algorithm. . . . .	59
Figure 7.3:	The delay of sink $i$ w.r.t the buffer $j$ . . . . .	60
Figure 7.4:	Computation of sensitivity, $\alpha_{ij}$ , of a sink $i$ w.r.t the buffer $j$ . . . . .	61
Figure 7.5:	Displacement direction computation. . . . .	62
Figure 7.6:	ISCAS Optimization Results . . . . .	64
Figure 8.1:	Mesh Explorer Tool. . . . .	66
Figure 8.2:	Main Window of Mesh Explorer Being Designed Using wxForm-Builder. . . . .	67
Figure 8.3:	Electrical Spice Simulation. . . . .	68
Figure 8.4:	Mesh Explorer Delay View Mode. . . . .	69
Figure 8.5:	Clock skew optimization through buffer displacement. . . . .	70



## LIST OF TABLES

Table 3.1:	Sources of skew and jitter. . . . .	37
Table 7.1:	ISCAS Benchmarks . . . . .	63
Table 7.2:	ISCAS Optimization Result . . . . .	63
Table 8.1:	Buffer Properties . . . . .	68
Table 8.2:	Sink Properties . . . . .	69



## ABSTRACT

Clock meshes are a suitable clock network architecture for reliably distributing the clock signal under process and environmental variations. This property becomes very important in the deep sub-micron technology where variations play a main role.

The clock mesh reliability is due to redundant paths connecting clock buffers to clock sinks, so that variations affecting one path can be compensated by other paths. This comes at cost of more power consumption and wiring resources. Therefore it is clear the trade-off between reliably distributing the clock signal (more redundancy) and the power and resource consumption.

The *clock skew* is defined as the difference in the arrival time of clock signal at clock sinks. The higher is the clock skew, the slower is the circuit. Besides slowing down the circuit operation, a high clock skew increases the probability of circuit malfunction due to variations.

In this work we focus on the clock skew problem. We first extract some useful information on how the clock wirelength and capacitance change as the mesh size changes. We present analytical formulas to find the optimum mesh size for both goals and study how the clock skew varies as we move further away from the optimum mesh size.

We also present a method for reducing the clock mesh skew by sliding buffers from the position where they are traditionally placed. This improvement comes at no increasing cost of power consumption since the buffer size and the mesh capacitance are not changed.

**Keywords:** Clock, Clock mesh, Skew, Performance, Microprocessor, Variability, Micro-electronic.



## Otimização de Malhas de Relógio

### RESUMO

Malhas de relógio são arquiteturas de rede de relógio adequadas para distribuir confiavelmente o sinal de relógio na presença de variações de processo e ambientais. Tal propriedade se torna muito importante nas tecnologias submicrônicas onde variações têm um papel importante.

A confiabilidade da malha de relógio é devido aos caminhos redundantes conectando o sinal de relógio até os receptores de forma que variações afetando um caminho possam ser compensadas pelos outros caminhos. A confiabilidade vem ao custo de mais consumo de potência e fiação. Desta forma fica claro o balanceamento necessário entre distribuir confiavelmente o sinal de relógio (mais redundância) e o consumo de potência e aumento de fiação.

O *clock skew* é definido como a diferença entre os tempos de chegada do sinal de clock nos seus receptores. Quanto maior é o clock skew, mais lento o circuito precisa operar. Além de diminuir a velocidade do circuito, um valor alto de clock skew aumenta a probabilidade de o circuito não funcionar devido às variações.

Neste trabalho, nos focamos no problema de clock skew. Inicialmente extraímos informações úteis de como o comprimento da fiação e a capacitância variam a medida que o tamanho da malha varia. São apresentadas fórmulas analíticas que encontram o tamanho ótimo para ambos objetivos e é apresentado um estudo de como o clock skew varia a medida que nos afastamos do tamanho ótimo da malha de relógio.

Um método para a redução de clock skew através do deslocamento dos buffers também é apresentado. Tal melhoria no clock skew não afeta o consumo de potência já que o tamanho dos buffers e a malha não são alterados.

**Palavras-chave:** Relógio, Malha de Relógio, Skew, Performance, Microprocessador, Variabilidade, Microeletrônica.



# 1 INTRODUCTION

Imagine some stage of a milk industry where two independent robots cooperate. Their operation is synchronized by a common signal, which can be either on or off. When the signal is on, the first robot fills the bottle with fresh milk and the second one waits. When the signal is off, the first one stops filling and the second one closes the bottle. Now imagine that for some reason the second robot receives the synchronization signal earlier than the first one. In this case, the bottle is closed before the milk is completely disposed causing the system to fail.

As in our milk industry example, synchronous circuits are composed of several components, where many of them must operate synchronously so that the circuit works properly. Discrepancies in the time the synchronization signal arrives at each synchronous component may cause the circuit malfunction.

## 1.1 Clock Signal

In synchronous circuits, the synchronization is performed via a periodic clock signal, as shown in Figure 1.1, which controls the operation of all synchronous components. Synchronous components are the ones that store the information processed by the circuit. The clock signal defines the moment at which data is read from and written to such components - hereafter, for short, called clock sinks or just sinks.

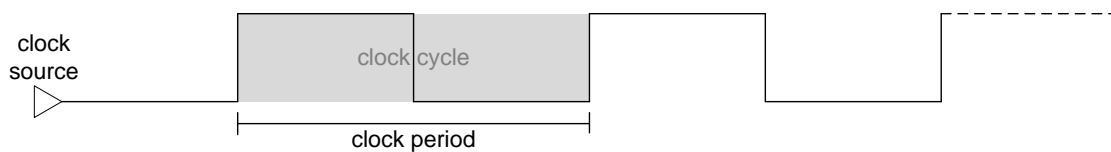


Figure 1.1: The clock signal - a periodic signal that synchronizes the circuit components.

The clock period - the time a clock cycle requires to be completed - defines the velocity of the circuit. The shorter the clock period is, the faster is the circuit operation. The clock period can be reduced (or the clock frequency can be increased) until some bounds are not crossed. These bounds and how they are affected by the clock signal arrival time discrepancy are discussed in the Chapter 2.

## 1.2 Clock Network

The clock signal is carried to the clock sinks throughout a clock network, which commonly spans over all the circuit area. An example of a clock network is shown in Figure

1.2. Such a huge network is more prone to fabrication process and environmental variations and it is one of the most important sources of power consumption of a digital circuit (WARNOCK et al., 2002). Tolerance to variations and power consumption are the two main issues that must be addressed in the clock network design.

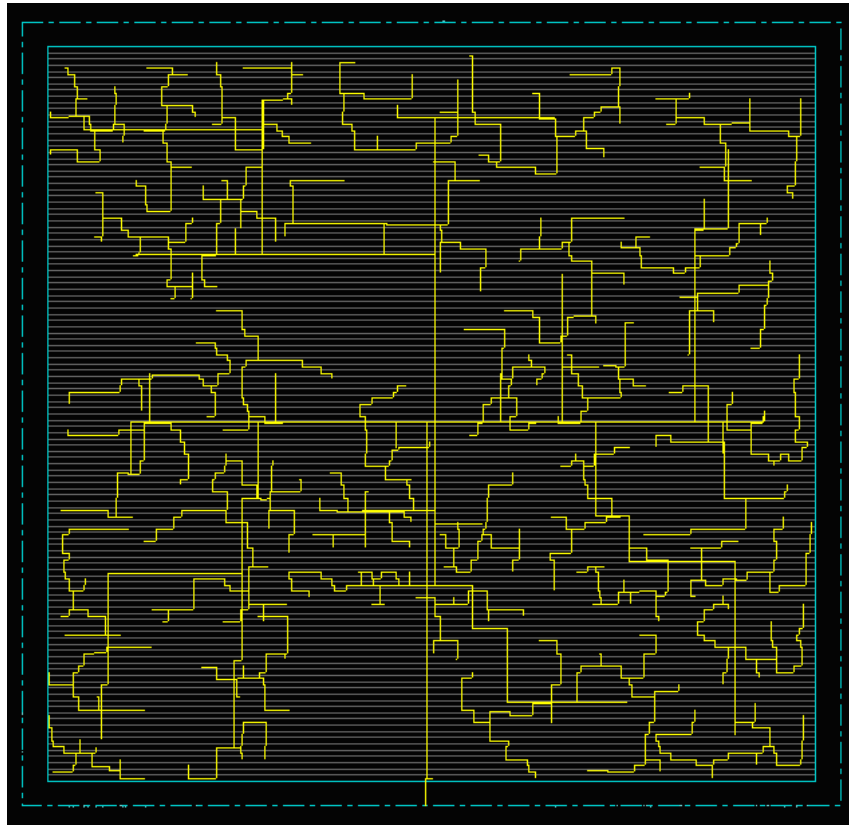


Figure 1.2: Clock signal spanned over all circuit area.

Fabrication process and environmental variations affect the circuit components by changing their electrical parameters in a non-deterministic fashion. In the case of a clock network some paths may become faster and others may become slower disturbing any design-time attempt to balance the arrival times at each clock sink. The clock network must be designed so that even under variations the arrival time discrepancy is confined inside a tolerance limit. Since the variation problems worsen as the technology node shrinks, variation tolerance becomes more and more relevant in the circuit design.

Beside the problems caused by variations, the clock network is commonly the main source of power consumption in a synchronous circuit. At each clock cycle, the clock network and its sinks must be charged and discharged. Since the clock network frequently spans over all circuit area and must drive several components, no wonder it is one of the main sources of power consumption. For instance, the total clock capacitance of the Alpha microprocessor correspond to 40% of the total effective switching capacitance of the circuit (JAN M. RABAEY ANANTHA P. CHANDRAKASAN, 2002) (WARNOCK et al., 2002).

### 1.2.1 Signal Regeneration

Any electrical signal becomes weaker as it transverse a wire. For smaller connections, the signal attenuation may be neglected. However, for a huge network like the



clock network, regeneration elements must be inserted.

Although the regeneration elements - also called *buffers* - allow the signal to travel longer distances, they insert new sources of arrival time discrepancy and power consumption.

### 1.3 Clock Mesh

Some clock network architectures were developed and studied in the literature. They can be divided into two main categories w.r.t the connectivity between clock signal and clock sinks: (1) single path and (2) multiple paths.

In the first category there is only one path connecting each clock sink to the clock source while in the second one multiple paths (or redundant paths) may connect a clock sink to the clock source. Redundant paths allow variations affecting one path to be compensated by other ones. On the other hand, single path architecture consume less circuit resources. Single and multiple path architectures are described in Chapter 4.

Among the redundant path architectures, *clock meshes* emerge as a feasible multiple-path solution to the variability problem for sub-micron technologies. Clock meshes are  $m + 1 \times m + 1$  wire grids formed by  $m + 1$  horizontal and  $m + 1$  vertical wires as shown in Figure 1.3. An  $m + 1 \times m + 1$  mesh is referred in this work as having size  $m$ .

Clock sinks connect to the clock mesh through wires called *stubs*. The clock signal is connected at multiple points of the clock mesh through a high-level clock network architecture. These points are commonly the *mesh nodes* - the intersection of a horizontal and a vertical wire.

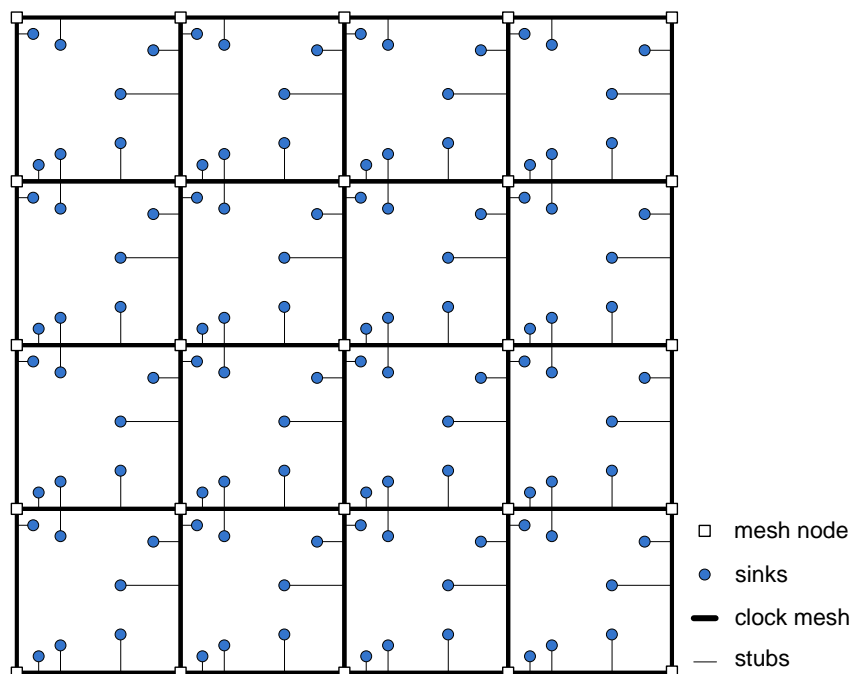


Figure 1.3: Clock mesh is a grid wire to where clock sinks connect to.

Due to the high degree of redundancy, clock meshes have a very high tolerance to variations, which can be increased by increasing the mesh size. The tolerance to variations comes at cost of more wire and thus power consumption.

In Chapter 6 the trade-off between power consumption and variation tolerance is discussed. In Chapter 7 a method for reducing the arrival time discrepancy without increasing the power consumption is presented.

## 1.4 Motivation

Clock meshes impose a great usage of wiring resources and power consumption, which restricted its use to products like microprocessors where the demand for performance further narrows the tolerance to variations. Several microprocessors have used clock meshes to distribute the clock signal:

- Itanium 1st Generation - 2000 (TAM et al., 2000)
- 1.2GHz Alpha Microprocessor - 2001 (XANTHOPOULOS et al., 2001)
- Power4 - 2002 (WARNOCK et al., 2002) (RESTLE et al., 2002)
- Power5 - 2004 (KALLA; SINHAROY; TENDLER, 2004) (CLABES et al., 2004)
- Dual-Core SPARC V9 - 2005 (HART et al., 2006)
- First Cell Processor - 2005 (PHAM et al., 2006) (PHAM et al., 2005) (PHAM et al., 2005)
- Power6 - 2007 (FRIEDRICH et al., 2007) (THOMSON; RESTLE; JAMES, 2006)

In all of those circuits, clock meshes are not used as the sole solution for clock distribution. As already pointed out, the clock mesh itself relies on a high-level clock network architecture, which brings the clock signal to multiple points at mesh. And, in many cases, the sinks of the clock mesh are not directly the synchronous elements of the circuit, but the inputs of a lower-level clock network architecture. Therefore, we must consider that clock meshes do not replace other clock architecture, but they are applied together to make the circuits more robust to process and environmental variations.

However, as the variations still and will increase as the technology advances, even non-microprocessors may consider the use of clock meshes. Since, for non-microprocessor, the design life-cycle is much shorter than for microprocessors, there is a new demand for methods and algorithms, which allow the automatic clock mesh synthesis addressing the time-to-market requirements (RAJARAM; PAN, 2008). The clock mesh architecture has been gaining increasing attention in the literature in the recent years (VENKATARAMAN et al., 2006) (RAJARAM; PAN, 2008) (REDDY; WILKE; MURGAI, 2006) (CHEN et al., 2005) (YEH et al., 2006).

Gustavo Wilke's thesis (WILKE, 2008) analyzed and presented optimization methods for mesh architectures including a sliding window technique for optimizing electric simulation and a new buffer model for decreasing the mesh power consumption. To some extent, this work is an extension of Wilke's work as the analysis and methods presented herein expands the results of that work.

## 1.5 Contributions

The contributions of this work can be summarized as follow:

- A tool for aiding the development of new algorithms for mesh optimization and teaching clock meshes (Chapter 8);
- An analytical formula for finding the mesh size, which minimize the total mesh capacitance (Chapter 6);
- A study on how to define a good mesh size, which provides insights in the right mesh size for trading-off variation tolerance and power consumption (Chapter 6);
- A method for improving the tolerance to variations of clock meshes (Chapter 7) without increasing the total mesh capacitance through linear programming.



## 2 TIMING CONSTRAINTS IN SYNCHRONOUS CIRCUITS

In this chapter, the parts that compose a synchronous circuit and its implications on the clock period,  $T$ , are presented.

### 2.1 Clock Signal

Considering a single-phase positive edge-triggered clocking style, the rising edge of the clock signal defines the end of the current clock cycle and the beginning of the next one. At a rising edge, positive edge-triggered flipflops store data currently held in their inputs. Hereafter, throughout this work, flipflop refers to a positive edge-triggered flipflop and a clock edge to a rising clock edge as well.

### 2.2 Sequential Components: FlipFlops

The time required for a flipflop to change its output after an occurrence of a clock edge is called *propagation time* ( $t_{pFF}$ ). To be correctly stored, data at the input of a flipflop must be stable during a certain amount of time before and after a clock edge arrives. The time required before is defined as the *setup time* ( $t_{setup}$ ) and the time required after is defined as the *hold time* ( $t_{hold}$ ). The time parameters of a flipflop are presented in the time diagram shown in Figure 2.1.

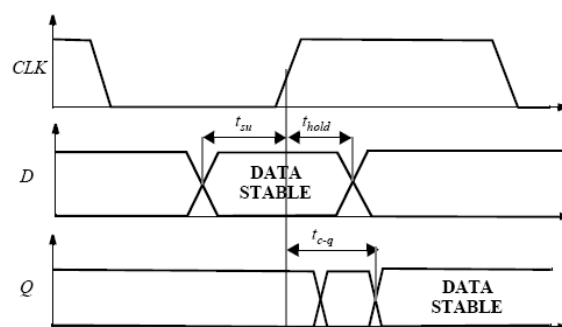


Figure 2.1: Timing diagram for a positive edge-triggered flipflop (JAN M. RABAEY ANANTHA P. CHANDRAKASAN, 2002)

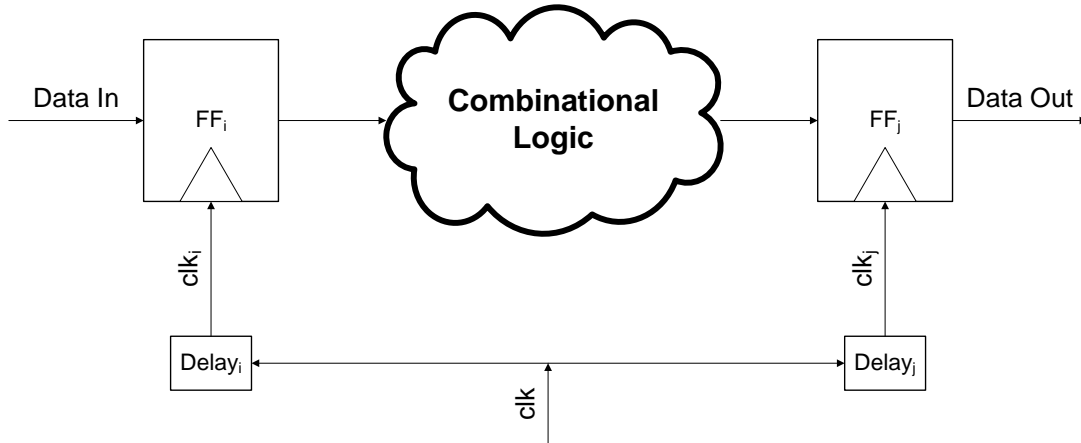


Figure 2.2: A combinational path.

### 2.3 Combinational Paths

A combinational path is composed of one input flipflop  $i$  and one output flipflop  $j$  connected through a combinational logic as shown in Figure 2.2. This is the basic structure found in synchronous circuits. A combinational path may present different delays depending on its input and output values. For timing analysis, we are interested in the minimum ( $t_{comb-min}$ ) and maximum ( $t_{comb-max}$ ) delays of the combinational path. The clock arrival time for flipflop  $i$  and  $j$  are given by  $t_{clk_i}$  and  $t_{clk_j}$  respectively.

### 2.4 Clock Signal Uncertainty

The clock signal may arrive at different times at each clock sink even though it is being generated from a common source. The first reason is that different paths with different lengths and widths present different delays. However, even when the paths are designed with the same electric characteristics, process and environmental variations tend to mismatch the paths delays.

Arrival discrepancies can be classified as static and dynamic. Static variations are caused mainly by the circuit design and by process variations while dynamic variations are commonly effect of environmental variations. The sources of static and dynamic variations are discussed in Chapter 3.

#### 2.4.1 Clock Skew

Static variations cause the clock signal to arrive in different times at each clock sinks. The time mismatch between two clock sinks is called *clock skew*. Since clock skew is a static characteristics of the circuit, by definition, it does not change cycle-to-cycle. For two clock sink  $i$  and  $j$ , clock skew is given by Equation 2.1.

$$\delta(i, j) = t_{clk_i} - t_{clk_j} \quad (2.1)$$

#### 2.4.2 Clock Jitter

Dynamic variations may change the delay of a given node cycle-by-cycle. This discrepancy on the clock signal delay is called *clock jitter*. For a clock sink  $i$ , the clock jitter

for two subsequent clock cycles is given by Equation 2.2.

$$t_{jitter_i} = t_{clk_{1_i}} - t_{clk_{2_i}} \quad (2.2)$$

## 2.5 Clock Period Definition

In this section we walk through the equations that must hold if the circuit is expected to work properly. When designing a circuit, the designer must be aware of two basic timing constraints. Both define a lower bound constraint for the clock period and for the delay of the fastest combinational path.

The first lower bound states that clock period must be sufficiently large so that every combinational path processes and stores correctly its information before the next clock cycle begins. The second one states that the delay of the fastest combinational path inside a combinational path must be sufficiently large to avoid race conditions. A race condition occurs when the current data held in the combinational path is overwritten by the next one.

First we suppose an ideal scenario where neither skew nor jitter acts on the circuit. Next the clock skew and clock jitter implications on the ideal scenario are outlined. And, finally, we put all together and present the timing constraints when skew and jitter apply.

### 2.5.1 Ideal Scenario: Neither Skew nor Jitter

Initially, suppose neither skew nor jitter affects the circuit. In this ideal case, Equation 2.3 and 2.4 must hold such that the circuit works properly.

$$T > t_{pFF} + t_{comb-max} + t_{setup} \quad (2.3)$$

$$t_{comb-min} > t_{hold} - t_{pFF} \quad (2.4)$$

### 2.5.2 Introducing Clock Skew

In the presence of clock skew the ideal timing constraint equations should be adapted to account to the skew. The new clock period lower bound is presented in Equation 2.5 and the new minimum combinational path delay is presented in Equation 2.6.

$$T > t_{pFF} + t_{comb-max} + t_{setup} - \delta \quad (2.5)$$

$$t_{comb-min} > t_{hold} - t_{pFF} + \delta \quad (2.6)$$

As clock skew can be positive or negative there are two possible scenarios: (1) positive clock skew ( $t_{clk_i} > t_{clk_j}$ ) and (2) negative clock skew ( $t_{clk_i} < t_{clk_j}$ ).

#### 2.5.2.1 Positive Clock Skew

A positive clock skew occurs when the clock signal arrives later at flipflop  $j$  than at flipflop  $i$  as shown in Figure 2.3. In this case, clock period can be reduced by  $\delta$  improving the circuit performance as can be noticed in Equation 2.5. On the other hand, the delay of the fastest combinational path must be slowed down by  $\delta$  as can be seen in Equation 2.6.

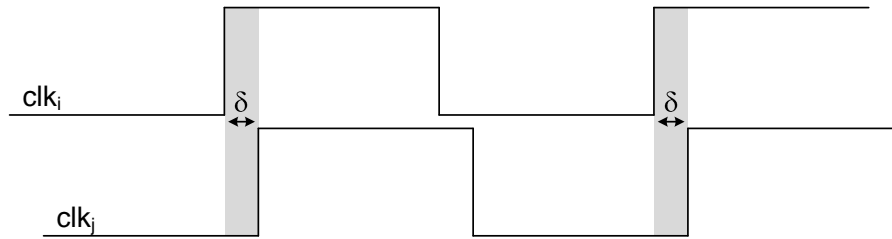


Figure 2.3: Positive Clock Skew

### 2.5.2.2 Negative Clock Skew

A negative clock skew occurs when the clock signal arrives earlier at flipflop  $j$  than at flipflop  $i$  as shown in Figure 2.4. In this case, clock period must be increased by  $\delta$  harming the circuit performance as can be noticed in Equation 2.5. On the other hand, the delay of the fastest combinational can be even faster than in the ideal scenario as can be seen in Equation 2.6.

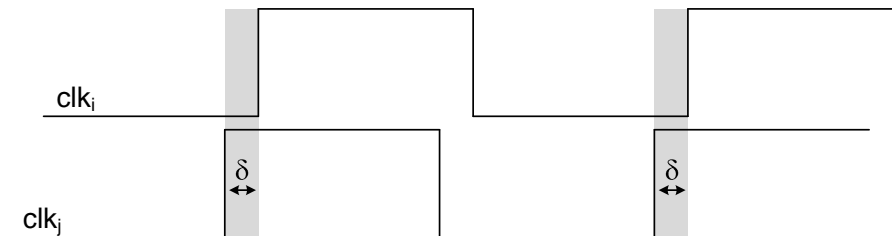


Figure 2.4: Negative Clock Skew

### 2.5.3 Introducing Clock Jitter

In the presence of jitter in the circuit the ideal timing constraint equations should be adapted to account to the jitter. Figure 2.5 shows how the clock jitter affects the clock signal arrival time. The new clock period lower bound is presented in Equation 2.7 and the new minimum combinational path delay is presented in Equation 2.8. Notice that, differently from the clock skew, clock jitter always harms the circuit performance.

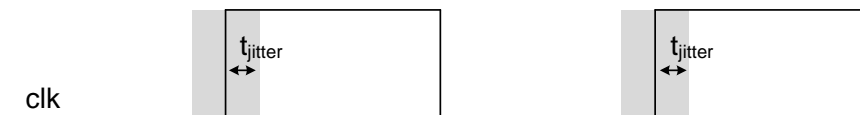


Figure 2.5: Clock Jitter

$$T > t_{pFF} + t_{comb-max} + t_{setup} + 2t_{jitter} \quad (2.7)$$

$$t_{comb-min} > t_{hold} - t_{pFF} + 2t_{jitter} \quad (2.8)$$

### 2.5.4 Putting All Together

As previously presented, when defining the circuit timing constraints the designer must take into account both clock skew and clock jitter. Combining the timing constraints



for clock skew and clock jitter effects, we have the final timing constraints as defined by Equation 2.9 and Equation 2.10.

$$T \geq t_{pFF} + t_{comb-max} + t_{setup} - \delta + 2t_{jitter} \quad (2.9)$$

$$t_{comb-min} > t_{hold} - t_{pFF} - \delta + 2t_{jitter} \quad (2.10)$$

As already pointed out, both skew and jitter may harm the circuit operation velocity. Although, Equation 2.9 states that a positive clock skew improve the circuit velocity, it is almost impracticable to design a circuit with only either positive clock skew. In general, we can assume that in real designs both positive and negative clock skews are present.

Also it is important to keep in mind that a high discrepancy in the clock signal arrival times, not only harms the circuit velocity, but increases the probability of failures. Therefore circuits must be designed so that the clock skew and clock jitter are small.



### 3 SOURCES OF SKEW AND JITTER

Clock skew is intrinsic to all circuit designs. Although, in practice, it cannot be avoided, it can and should be minimized. Many reasons contribute to the different arrival times observed at clock sinks.

The most direct reason is the fact that the clock network is an RC network and such nodes with different RC present different delays. Figure 3.1 shows a simple example of clock skew due to RC. In theory, this source of skew can be eliminated by building a clock tree network in which every node presents the same RC w.r.t the driver node. However, when the network is inserted inside the circuit, the coupling capacitance may be different for each line unbalancing again the RC.

Many skew sources cannot be predicted exactly during design time. These sources are due to process and environmental variability. Process variability is the main source of clock skew, as it affects wire and transistor dimensions changing their electrical characteristics.

- Process Variability

- **Wire Dimension.** The balancing of the RC clock networks is performed using the nominal characteristics of the wires. However, due to process variability, wire dimensions and consequently wire resistance and capacitance differs from the expected values. This acts by unbalancing the RC clock network and potentially increasing the clock skew.
- **Transistor Width.** Variations in transistor width may harm the clock skew by changing the drive strength and the load capacitance of clock buffers. For

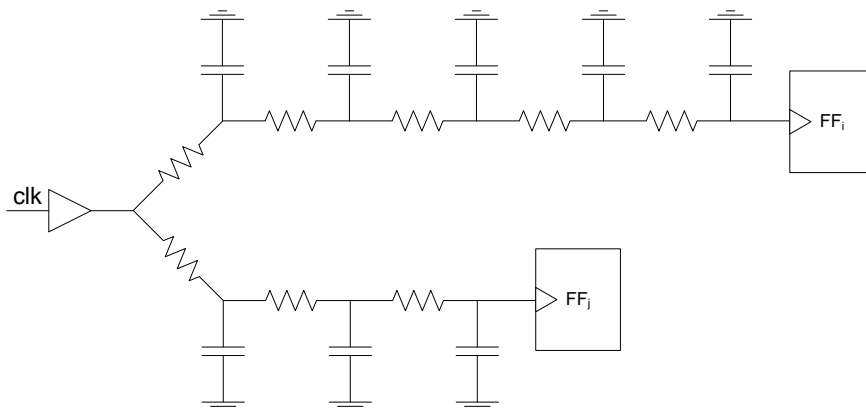


Figure 3.1: Clock sinks with different delays due to different RC.

instance, if the transistor width of a buffer is increased, it loads faster its sinks. On the other hand, if the transistor width of a sink is increased, it takes more time to be loaded.

- **Transistor Length.** Similar to the transistor width variation, the variation of the transistor length may increase skew as the threshold voltage of buffers is changed as well as the capacitance of sinks.

- Environmental Variability

- **Temperature.** The temperature changes the transistor and wire properties acting as a source of skew.

### 3.1 Summary of Skew and Jitter Sources

This section summarizes the sources of clock skew and clock jitter based on the definitions from Rabaey's book. (JAN M. RABAEY ANANTHA P. CHANDRAKASAN, 2002).

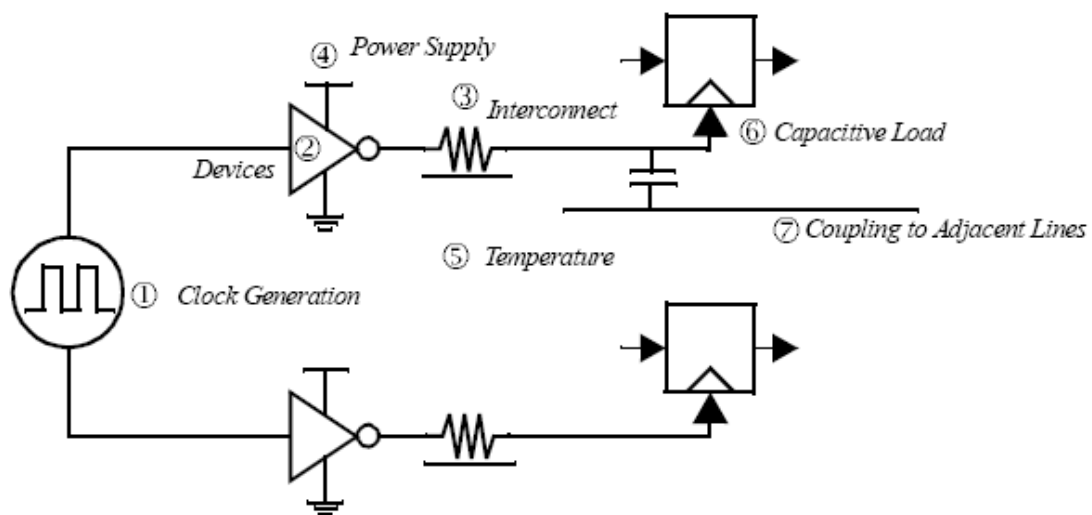


Figure 3.2: Sources of skew and jitter (JAN M. RABAEY ANANTHA P. CHANDRAKASAN, 2002).

Table 3.1: Sources of skew and jitter.

Source	Causes	Description
1 - Clock generation	Jitter	The clock generator is an analog circuit which itself causes jitter. The core of the clock generator is a Voltage-Controlled Oscillator (VCO). This is an analog circuit sensitive to intrinsic device noise and power supply variations.
2 - Devices	Skew	Buffers are critical elements on the clock network and, as any other device, they suffers from process variation (threshold voltage, load capacitance). The buffer variations affect the buffer delay, causing clock skew.
3 - Interconnections	Skew	Variations in vertical and lateral dimension cause the interconnect capacitance and resistance to vary across a chip. Since this variation is static, it affects the clock skew between different paths.
4 - Power Supply	Jitter	The power supply voltage is a strong function of the switching activity. This affects the delay through buffers as it directly affects the drive of the transistors.
5 - Temperature	Skew	Temperature gradients across the chip is a result of variations in power dissipation across the die. Since the device parameters (such as threshold, mobility, etc.) depend strongly on temperature, buffer delay for a clock distribution network along one path can vary drastically compared to another path. Although temperatures varies along the time, this variation is in order of microseconds, which are much larger than the clock period. Therefore, temperature is considered as a static variation, causing skew and not jitter.
6 - Capacitive Load	Jitter	The load capacitance is highly non-linear and depends on the applied voltage. In many latches and registers this translates to the clock load being a function of the current state of the latch/register (e.g., the values stored on the internal nodes of the circuit), as well as the next state. This causes the delay through the clock buffers to vary from cycle-to-cycle, causing jitter.
7 - Coupling to Adjacent Lines	Jitter	Any coupling between the clock wire and adjacent signal results in timing uncertainty. Since the adjacent signal can transition in arbitrary directions and at arbitrary times, the exactly coupling to the clock network is not fixed from cycle-to-cycle. This results in clock jitter.



## 4 A BRIEF REVIEW ON COMMON CLOCK DISTRIBUTION ARCHITECTURES

In this chapter we briefly describe some common clock network architectures as clock trees, clock meshes and hybrid architectures outlining their advantages and disadvantages. They are separated in two main categories - single-path and multiple-path - w.r.t the connection between clock source and clock sinks.

### 4.1 Single-Path Networks

In single-path networks, there is only one path connecting the clock source to each clock sink. In this way, one can work with minimum wirelength networks reducing the wiring resources and thus the power consumption. On the other hand, as there is only one path connecting clock source to clock sinks, variations affecting a path cannot be compensated by other paths.

#### 4.1.1 Clock Tree

A clock tree is a simple way to connect the clock sinks to the clock source by means of a tree structure. A clock tree contains no redundant paths, so that there is only one path connecting the source to any sink.

The clock skew and power consumption can vary greatly depending on the topology and routing of the clock tree. In order to achieve a low skew value all paths connecting the sinks to the clock source should have similar RC constants. In fact, considering the Elmore delay model, zero-clock skew could be generated. Zero-clock skew trees are addressed by (CHAO et al., 1992) (TSAO; KOH, 2000) (KAHNG; ALBERT TSAO, 1996). Minimum wirelength trees are efficient for power reduction but are likely to have a large skew due to unbalanced RC. Figure 4.1 presents two different topologies for the same set of clock sinks.

As clock network commonly drives a huge load capacitance, driving all the structure through a single buffer is impracticable and may lead to large slew. Therefore buffers are inserted breaking down the tree structure. (PULLELA et al., 1993) addresses the problem of generating a zero-skew clock tree at the same time the buffer insertion is performed.

Generic clock trees can be very unstructured, so researchers have developed more predictable tree structures as h-trees (ULLMA, 1984) and x-trees (FRIEDMAN, 2001). Both are easier to be designed and the buffer insertion problem becomes simpler to be solved since the tree itself is a balanced structure.

The tolerance to variability is a key point in the design of clock trees. Since trees have no redundant paths, variations that may affect one branch are not compensated by any

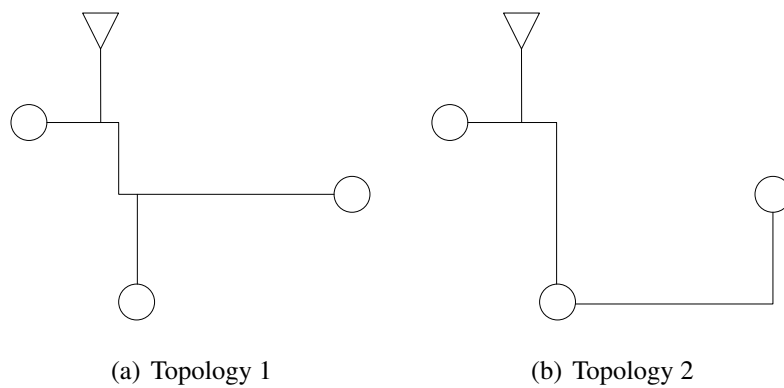


Figure 4.1: Example of different topologies for a same set of sinks.

other. Therefore the actual clock skew, after fabrication, may be much greater than that estimated during design time, causing a circuit malfunction.

#### 4.1.2 H-Tree

An h-tree (ULLMA, 1984) is a symmetric structure in which the wirelength from any leaf to the root is the same. Figure 4.2 depicts an h-tree topology. An h-tree, as a symmetric structure, is more predictable and easy to be designed. The routing is straightforward since the tree structure can be mapped directly to wires. The buffer insertion becomes easier since the wire load capacitance is balanced throughout the branches.

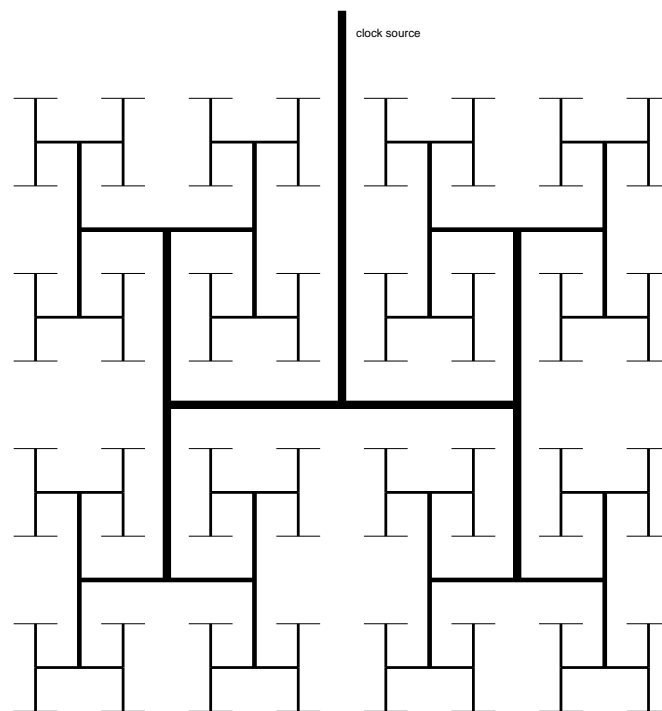


Figure 4.2: H-Tree.

The sinks are connected to the deepest-level branches of the h-tree forming a structure known as fishbone as seen in Figure 4.3.



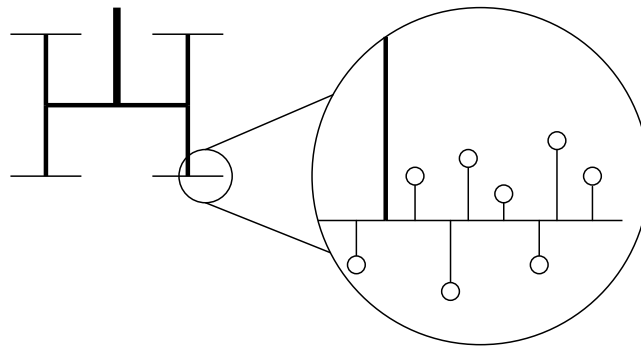


Figure 4.3: Fishbone.

### 4.1.3 Spine-Tree

A spine is a wire to where clock sinks are connected. A circuit may have several spines. These spines are driven by a clock tree. Figure 4.4 presents a spine-tree structure with two spines.

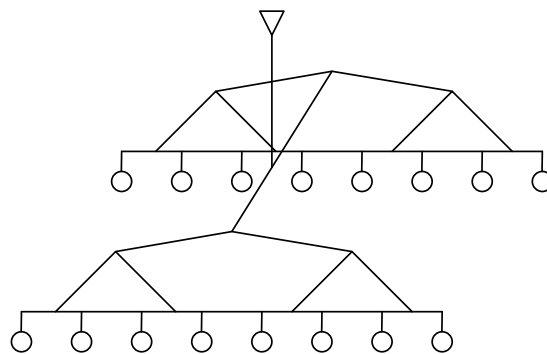


Figure 4.4: Spine Tree.

## 4.2 Multiple-Path Networks

In multiple-path networks, a clock sink may be connected to the clock source by multiple or redundant paths. This allows the variations affecting one path to be compensated by other ones, although more wiring is necessary and thus more power is dissipated when compared to single-path networks.

### 4.2.1 Cross-Link

Cross links are wires used to make shortcuts between branches of a tree. In this way the variation affecting some branch can be compensated by another branch. Figure 4.5 shows an example of a cross link topology where bowed wires are the cross links.

### 4.2.2 Clock Mesh

A clock mesh is a grid structure to which clock sinks are connected. In this work, a mesh composed by  $m + 1$  rows and  $m + 1$  columns is referred as having  $m$  size. Figure 4.6 shows a  $6 \times 6$  clock mesh.

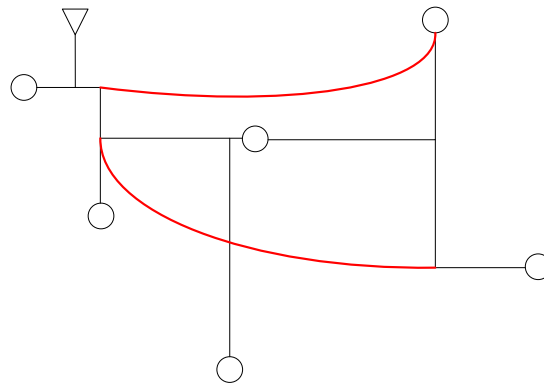


Figure 4.5: Cross-Link.

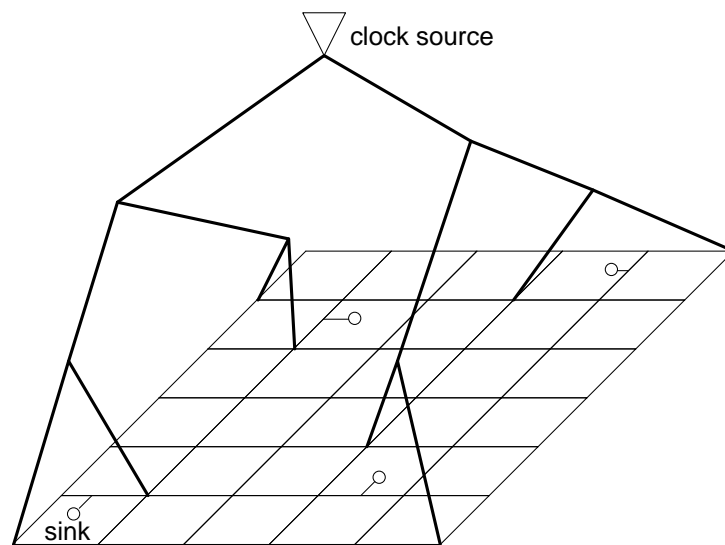


Figure 4.6: Clock Mesh.

The clock mesh is driven by mesh buffers which are commonly placed in the intersection of a horizontal and vertical line. Those mesh buffers, by their turn, are driven by another clock network. This top-level network may be another clock mesh or, more commonly, a clock tree.

The main characteristic of a clock mesh is that there are multiple paths from buffers to each clock sinks. This redundancy means more robustness to variations at the cost of more power consumption.

### 4.3 Tree + Local Mesh (TLM)

When there are more than one clock domain or sinks are clustered in more than one region, multiple clock meshes can be used to drive each clock domain or each cluster of sinks individually. Using multiple clock meshes allows the designer to switch off the clock for some part of the circuit. Figure 4.7 demonstrates the TML topology.

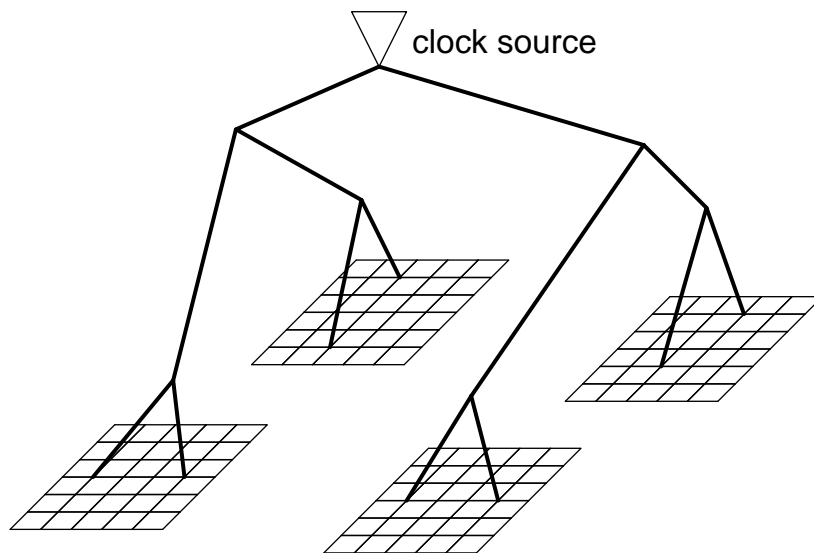


Figure 4.7: Tree + Local Mesh (TLM).



## 5 RELATED WORKS

There are many works in clock network design. Let us consider, in this chapter, the two most relevant to this work (VENKATARAMAN et al., 2006) (RAJARAM; PAN, 2008).

### 5.1 Combinatorial Algorithms for Fast Clock Mesh Optimization

The work "Combinatorial Algorithms for Fast Clock Mesh Optimization" (VENKATARAMAN et al., 2006) presents a set-cover based algorithm for simultaneous mesh buffer placement and sizing. An edge removal algorithm based on survivable network theory is also presented.

#### 5.1.1 Simultaneous Mesh Buffer Placement and Sizing

The main goal of the set-cover algorithm is to find whether a buffer is required at a grid node or not and if so the size of such buffer minimizing the total sum of the buffer sizes. The size of a buffer is such that it drives less than the maximum load capacitance it can load.

Let  $I$  be the set of grid nodes and  $B$  be the set of buffer sizes. Let us define the covering region of a node  $i$  while driven by buffer  $j$ ,  $CR_i^j$ , as the maximal set of nearest nodes of  $i$  such that the total capacitance, including mesh wire capacitances, are less than the total capacitance that buffer  $j$  can drive. The greedy set coverage algorithm is shown in Figure 5.1.

Greedy set cover for mesh buffer placement/sizing.
Input : $SCR = \cup CR_i^j$ for each $i \in I$ and $j \in B$
Output : $M =$ set of covering regions that are picked
<ol style="list-style-type: none"> <li>1. <math>M \leftarrow \phi</math></li> <li>2. While <math>M</math> does not cover <math>I</math> do <ol style="list-style-type: none"> <li>2.1 For each unpicked covering region <math>CR_i^j</math> define <math>C_{eff} = \frac{b_j}{ CR_i^j - M }</math></li> <li>2.2 Pick set <math>C</math> with least <math>C_{eff}</math>.</li> <li>2.3 <math>M \leftarrow M \cup C</math></li> </ol> </li> <li>3. For each node <math>i \in I</math>, if there exists <math>j \in B</math> and <math>l \in B</math> both <math>CR_i^j</math> and <math>CR_i^l</math> are picked and <math>b_j &gt; b_l</math> drop <math>CR_i^l</math> from the solution.</li> </ol>

Figure 5.1: Greedy set cover for mesh buffer placement and sizing (VENKATARAMAN et al., 2006).

### 5.1.2 Mesh Reduction

The mesh reduction acts by removing edges from the mesh so that the power dissipation is reduced and a certain level of redundancy is still maintained. The edges are removed until the two following constraints are not met:

- There exist at least  $k$  node locations such that the distance between these nodes and any sink,  $s$ , is less or equal to  $L_{max}$ . This constraints are implicitly taken into count by the connectivity requirement.
- There exist at least  $l$  disjoint paths connecting a grid node to a clock sink.

Figure 5.2 shows the algorithm used for mesh reduction.

Greedy Steiner Network for Mesh Reduction
Input : $G = (V, E)$ , and connectivity requirements
Output : $E' \subset E$ satisfies connectivity requirements
1. For each $e \in E$ , set $c(e) = 1$ . 2. For each sink $s_i \in S$ 2.1 Find $k$ closest buffers locations 2.2 Identify $l$ minimum cost disjoint paths (denoted by $P_i$ ) between $s_i$ and identified buffer locations 2.3 For each $e \in P_i$ , 2.3.1 $E' \rightarrow E' \cup e$ , $c(e) = 0$ . 3. Output $E'$ .

Figure 5.2: Mesh reduction algorithm (VENKATARAMAN et al., 2006).

### 5.1.3 Buffer Model

Clock mesh simulation is a very timing consumption problem. Often a simplified buffer model is used to speedup the simulation time. Besides of providing a model that can accurately compute slew delay times, the mesh buffer model has to accurately model the interaction between the different mesh buffers. In the paper, the authors propose to use the model presented in Figure 5.3. The first stage is composed by a variable current source in parallel to a variable capacitance. The variable capacitance and the voltage dependent current source are both described by two dimensional look-up tables. Both capacitance and current are dependent on the output voltage and on the voltage at the output of the second stage.

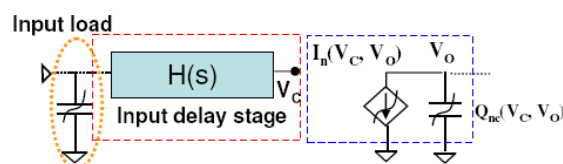


Figure 5.3: Driver model used in (VENKATARAMAN et al., 2006).

## 5.2 MeshWorks: An Efficient Framework for Planning, Synthesis and Optimization of Clock Mesh Networks

MeshWorks (RAJARAM; PAN, 2008) claims to be the first automatic tool for planning and synthesizing mesh networks. The tool can be divided into three major steps: mesh size definition; buffer placement/sizing and edge removal. Its results demonstrate a improvement of 26% in buffer area, 19% in wirelength and 18% in power can be achieved in comparison to (VENKATARAMAN et al., 2006).

### 5.2.1 Mesh Size Definition

Mesh Size must trade-off the total wirelength and the skew reduction. As we increase the total wirelength, we increase the mesh redundancy therefore reducing the skew. On the other hand, when we increase the wirelength we increase the power and routing resources.

The total wirelength is composed by the mesh wirelength, plus the stubs wirelength. So we can write the mesh wirelength of an  $m \times n$  clock mesh as a function of the mesh size as in Equation 5.1 where  $X_{bound}$  and  $Y_{bound}$  are the mesh dimensions.

$$L_{tot} = L_{mesh} + L_{stub} = m \times X_{bound} + n \times Y_{bound} + \sum_{i=1}^n L_{stub}^i \quad (5.1)$$

A bound for the clock skew,  $Sk_{bound}$ , can be written as a function of the mesh size as shown in Equation 5.2 where  $L_{stub}^{max}$  is the maximum stub length,  $CL^{max}$  is the maximum sink capacitance and  $D_{max}$  is the maximum distance between a sink and the nearest mesh buffer. The Equation 5.2 is composed by the three components:

$$Sk_{bound} = \underbrace{[max \{D_p (CL_p^{max})\} - min \{D_q (CL_{q-1}^{max})\}]}_I + \underbrace{Delay (D_{max})}_{II} + \underbrace{IntDel (L_{stub}^{max}, CL_L^{max})}_{III} \quad (5.2)$$

1. (I) accounts for the skew due to the difference of buffer loading capacity.
2. (II) accounts for the skew caused by the difference of path lengths connecting sink to the nearest buffer.
3. (III) accounts for the skew caused by the maximum stub length driven the maximum sink capacitance.

Figure 5.4 shows the algorithm to find a good initial mesh size.

### 5.2.2 Buffer Placement and Sizing

As in the work "Combinatorial Algorithms for Fast Clock Mesh Optimization", MeshWorks uses a set-cover heuristic to place and chose buffers sizes from an input library.

### 5.2.3 Edge Removal

For edge removal, MeshWorks uses a network sensitivity approach. Network sensitivity theory aims to efficiently evaluate sensitivities of a given output parameter (voltage or current) to changes in the circuit parameters. If the sensitivity of clock arrival time at the clock sinks with respect to the width of mesh edges are computed the mesh edges with smaller influence can be removed. The required steps to perform the sensitivity analysis are:

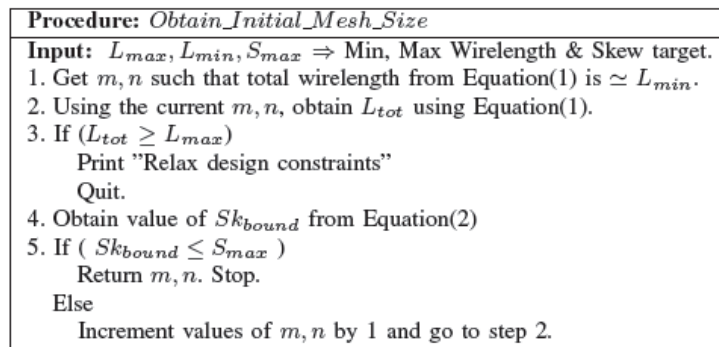


Figure 5.4: Algorithm to find a good initial mesh size (RAJARAM; PAN, 2008).

1. Identify and lump clock sink clusters. This reduces the total number of elements in the mesh model speeding up the sensitivity analysis.
2. Replace all mesh buffers by a linear model.
3. Obtain Elmore delay sensitivities for every lumped sink cluster with respect to every mesh edge.
4. Sort mesh edges in a sensitivity increasing order. Mesh edges are removed such that removed edges are at least  $N$  mesh nodes away from each other. This requirement prevents interactions between removed mesh segments, i.e., only mesh segments far away from other removed mesh segments can be removed to make sure that the mesh segment sensitivity was not affected.
5. By removing mesh edges the total load capacitance driven by each mesh buffer is reduced. A post processing sizing step is performed to reduce the size of mesh buffers whose covering region overlaps with another mesh buffer.

#### 5.2.4 Buffer Model

For overcoming the high simulation time, MeshWorks uses the buffer model presented in Figure 5.5. This model is simpler than the one of the work (VENKATARAMAN et al., 2006) and the error because of their buffer models is around 4% for delays and 1% for skew. However we were not able to achieve the same results when input skew is applied to the mesh buffers.

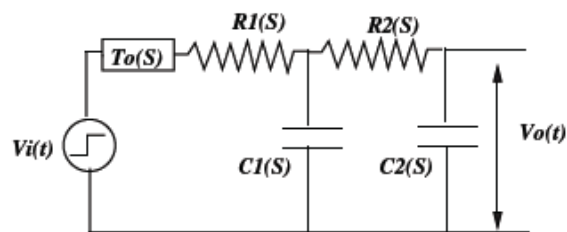


Figure 5.5: Buffer model used in (RAJARAM; PAN, 2008).



## 6 MESH SIZE SELECTION

As we shall view in this chapter, we can extract very useful theoretical results about the clock mesh size taking into account only the number of clock sinks that must be driven. We start by studying the total clock mesh wirelength and the total clock mesh capacitance as a function of the mesh size and present analytical formulas to find the optimum mesh size for both goals. Later we use those formulas for analyzing the clock skew change as we move away from the optimum size.

Finally we conclude the chapter by providing guidelines for clock mesh designers on the clock mesh size selection.

### 6.1 Definitions

In our analysis we assume randomly placed clock sinks over an  $L \times L$  die area. All clock sinks have the same capacitance. A clock mesh is built over this region with  $m + 1$  rows and  $m + 1$  columns evenly spaced and it is referred as having an  $m$  size. The stubs connect sinks to the nearest grid edges. Figure 6.1 shows an example of a  $4 \times 4$  clock mesh.

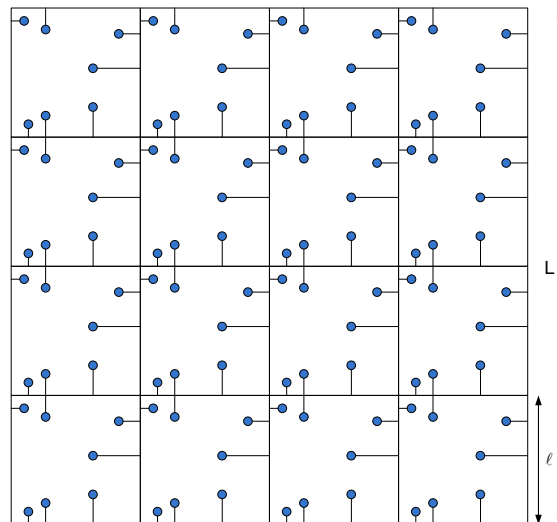


Figure 6.1: A  $4 \times 4$  Clock Mesh

Although the buffer placement affects the clock skew, it can be ignored for the following discussion since we are interested only in the total mesh wirelength and capacitance. In the experiments, we shall describe the placement and buffer sizing strategy. For now,

with no loss of generality, consider that mesh is driven by a single buffer sized according to the fanout-of- $n$  rule: the size of the buffer is such that its input capacitance is equal to the total mesh capacitance divided by  $n$ .

## 6.2 Motivation

Let's take a look at the Figure 6.2, which shows the mesh total wirelength and its components as a function of the mesh size (RAJARAM; PAN, 2008). The data were obtained from a  $1000\mu\text{m} \times 1000\mu\text{m}$  mesh driving 1200 sinks randomly placed on the circuit area. Although it is a snapshot for a given design, different designs present the same behavior as the mesh size changes.

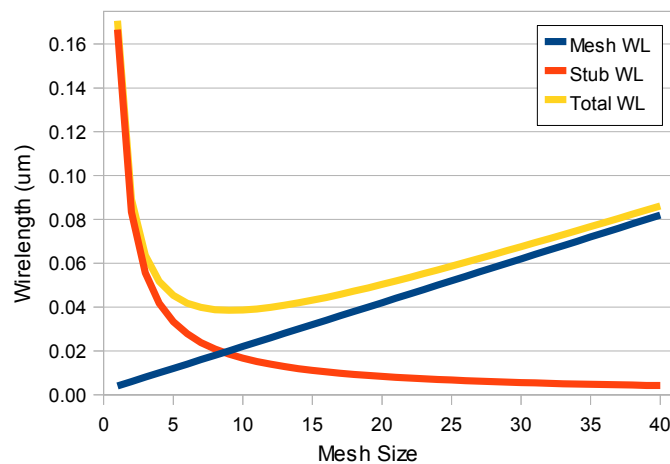


Figure 6.2: Mesh wirelength as a function of the mesh size.

As it can be seen in Figure 6.2 the total mesh wirelength has a global minimum, which means that we waste routing resources if the appropriate mesh size is not chosen. This global minimum emerges due to the reduction on the average stub length as the mesh size increases in contrast with the mesh wirelength that increases linearly as the mesh size increases.

Similarly, the total mesh capacitance has a global minimum as it can be noticed in Figure 6.3. For this chart, the buffers are sized to match the fanout-of-2, e.g. the total input capacitance of the buffers is equal to the total mesh capacitance divided by two. However, as we shall see, the fanout rule does not affect the optimum mesh size for total mesh capacitance reduction.

## 6.3 Average Stub Length

The stubs connect sinks to the nearest mesh edge so that the sinks are always connected to one of the four edges which enclose it. The boundary of an internal, concentric square defines a region in which the stubs have the same length. The greater the boundary is, the greater is the probability of a sink to lie on it. Figure 6.3 shows a mesh bin and a region where stubs have length equal to  $x$ .

By summing up the perimeter of the internal, concentric square regions we obtain the bin area as seen in Equation 6.1.

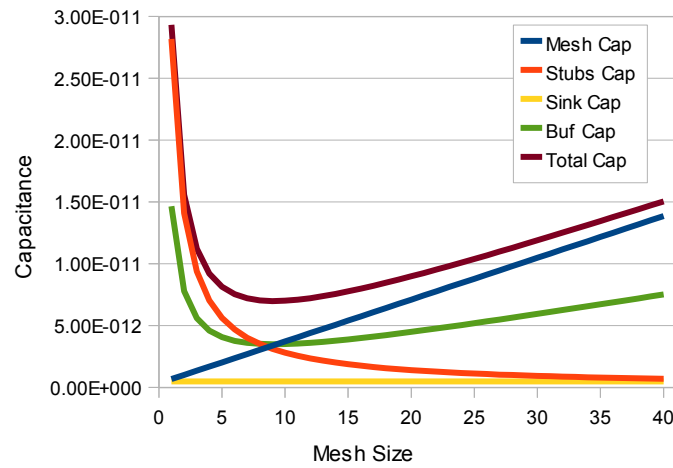


Figure 6.3: Mesh capacitance as a function of the mesh size.

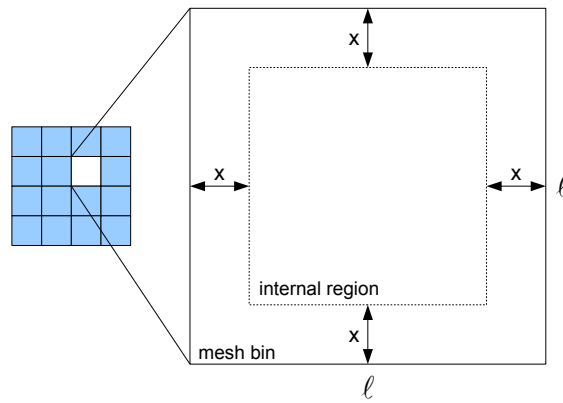


Figure 6.4: An internal bin region where stubs have the same length  $x$ .

$$\int_{l/2}^0 8 \left( \frac{l}{2} - x \right) dx = l^2 \quad (6.1)$$

So we can use the ratio between the perimeter of the region and the bin area as the probability of a sink to lie on the boundary of such region. Finally the average stub length is given by equation 6.2.

$$\int_{l/2}^0 \frac{8 \left( \frac{l}{2} - x \right)}{l^2} dx = \frac{l}{6} \quad (6.2)$$

## 6.4 Mesh Size for Minimum Wirelength

The total mesh wirelength is composed of both the mesh wirelength and the stub wirelength. As seen in the Figure 6.2 the mesh wirelength increases linearly and the stub wirelength decreases in a  $1/m$  rate. It is also clear by Figure 6.2 that the total mesh wirelength has a global minimum.

Equation (6.3) gives us the estimation of the total mesh wirelength as a function of mesh size.

$$W_{total}(m) = \underbrace{2L(m+1)}_{\text{mesh}} + \underbrace{(kL/6m)}_{\text{stub}} \quad (6.3)$$

Now taking  $\frac{\partial W_{total}(m)}{\partial m} = 0$  yields the mesh size that minimizes the total wirelength,  $m_{wire}^*$ , which is given by Equation 6.4;

$$m_{wire}^* = \sqrt{\frac{k}{12}} \quad (6.4)$$

## 6.5 Mesh Size for Minimum Capacitance

We define  $C_{total}$  as the sum of the mesh capacitance (mesh wires and stubs) and buffer capacitance. Taking  $\frac{\partial C_{total}(m)}{\partial m} = 0$ , the mesh capacitance,  $C_{mesh}$ , is given by Equation 6.5.

$$C_{mesh}(m) = \underbrace{C_{wire}[2L(m+1)]}_{\text{mesh wire cap}} + \underbrace{C_{stub}(kL/6m)}_{\text{stub cap}} + \underbrace{kC_{sink}}_{\text{sink cap}} \quad (6.5)$$

Since we size buffers based on the fanout-of- $n$  rule, the total buffer capacitance is proportional to the mesh capacitance. Finally Equation 6.6 gives the total mesh capacitance.

$$C_{total}(m) = \underbrace{C_{mesh}(m)}_{\text{mesh cap}} + \underbrace{C_{mesh}(m)/n}_{\text{buffer cap}} \quad (6.6)$$

The Equation 6.7 shows that the mesh size which minimizes the total mesh capacitance,  $m_{cap}^*$ , is proportional to the square root of the number of sinks and it is not affected by the fanout-of- $n$  rule. This is to say that we can change the fanout used for sizing buffers without affecting the optimum mesh size and so we can trade off power consumption and skew reduction (ABE; HASHIMOTO; ONOYE, 2008).

$$m_{cap}^* = \sqrt{\frac{k}{12}} \sqrt{\frac{C_{stub}}{C_{wire}}} \quad (6.7)$$

Notice also that if have  $C_{stub} = C_{wire}$  then  $m_{cap}^* = m_{wl}^*$ . However generally  $C_{stub} < C_{wire}$  so that  $m_{cap}^* < m_{wl}^*$ .

## 6.6 Optimum Mesh Bin Density

An  $m \times m$  mesh has  $m^2$  bins so that the mesh bin density,  $D_{bin}$ , is given by Equation 6.8. Setting  $m = m_{wire}^*$  in Equation 6.8 states that in a optimum sized mesh, the number of sinks inside a mesh bin is 12.

$$D_{bin} = \frac{k}{m^2} \quad (6.8)$$

## 6.7 Experiments

We perform a set of simulations to analyze the behavior of the clock skew as the mesh size goes away from the optimum mesh size.

### 6.7.1 General Simulation Setup

A buffer is placed at each intersection of a vertical and a horizontal grid wire. The buffers are sized based on the fanout-of-4 rule (SUTHERLAND; SPROULL; HARRIS, 1999) addressing the load capacitance due to direct connected edges as well as the stubs and sink's capacitance connected to those edges. Figure 6.5 presents an example of the region addressed by a buffer.

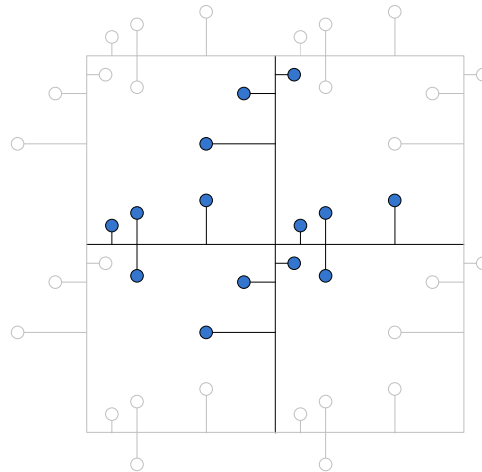


Figure 6.5: The coverage region of a buffer.

The 65nm PTM (ZHAO; CAO, 2007) technology parameters are used in the electrical simulations. Wires smaller than  $100\mu m$  are modeled with  $1\pi$  and the remaining ones are modeled with  $3\pi$  model. In these experiments we set  $C_{stub} = C_{wire}$ .

As discussed previously, the clock signal may arrive at different times at each mesh buffer. We define the clock mesh *input skew* as the maximum skew at mesh buffers. For emulating the effect of input skew, the clock signal arrival time at each mesh buffer is modeled as a uniform random variable. Spatial correlation is modeled through Principal Component Analysis (PCA) (CHANG; SAPATNEKAR, 2003) where the circuit area is divided into three levels. When variability is applied, 1000 Monte Carlo iterations are performed.

### 6.7.2 Specific Simulation Setup

In this chapter, a die of  $1000\mu m \times 1000\mu m$  was considered where 1000 sinks are randomly placed. For this mesh, by Equation 6.4, we have  $m_{wire}^* = 9$ . The input skew applied is in the range  $[0, 200]ps$  with steps of  $25ps$ .

### 6.7.3 Result Analysis

First let us analyze the case where no input skew is applied to the clock mesh, that is, all mesh buffers receive the clock signal at same time. The log-plot of results is shown in Figure 6.6. As we can see, as the mesh size increases further from the optimum mesh size, we still obtain significant clock skew reduction. Although the total mesh capacitance increases and more power is dissipated, it is still worthwhile to increase the mesh size until the desired clock skew is achieved. The ripple presented by the skew curve happens because only one flipflop distribution was used. If several different flipflop distributions were used, the skew curve should be smoother.

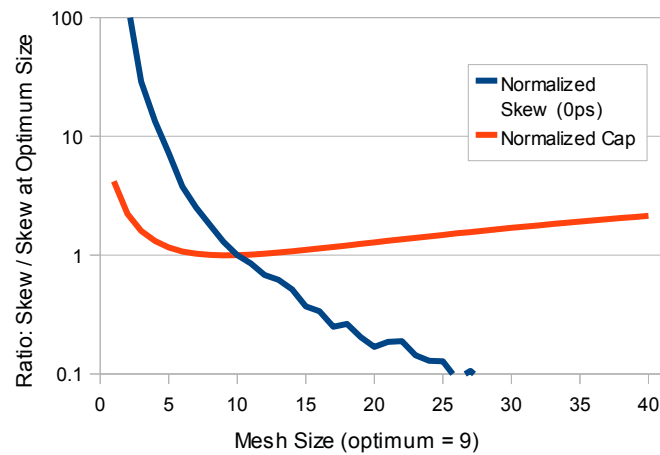


Figure 6.6: Skew and capacitance trade-off for no input skew.

This scenario changes when input skew is applied to the mesh buffers. Figure 6.7 shows the log-plot of the average skew for each scenario where input skew is greater than zero as the mesh size increases. By analyzing Figure 6.7 we can observe that no significant skew reduction is achieved by increasing the mesh size further from the optimum size. This contrasts with the rapidly increasing of the total mesh capacitance. The higher is the total mesh capacitance, the higher is the mesh power consumption. The power consumption also increases by the increase of short circuit current (WILKE, 2008).

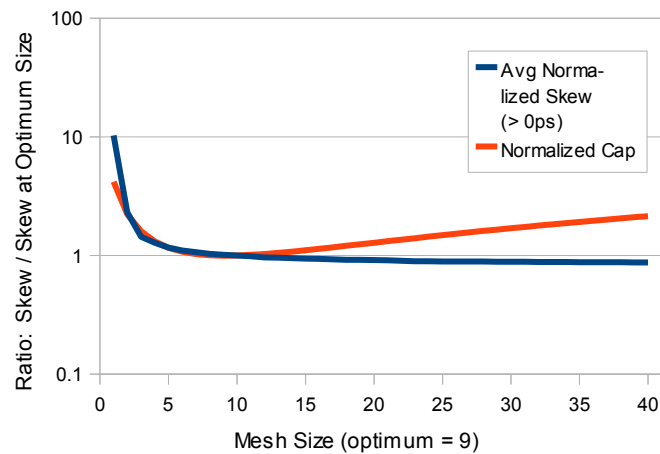


Figure 6.7: Skew and capacitance trade-off when input skew is applied.

## 6.8 Conclusions

As we observed the optimum mesh size for wirelength reduction is only dependent on the number of sinks. The optimum mesh size for capacitance is also dependent on the number of sinks, but it is also affected by the ratio between the mesh and the stub wire capacitances. An interesting property is that in the optimum sized mesh w.r.t the mesh wirelength the average number of sinks inside a mesh bin should be 12.

We analyzed the skew reduction as a function of the mesh size concerning the optimum mesh size. We noticed that a mesh presents different behavior when no input

skew is applied and when it does. When no input skew is applied we still obtain significant skew reduction paying with an increased mesh capacitance. This scenario is quite different when input skew is applied. In this case we can observe that we obtain little skew improvement when moving further away from the optimum mesh size, which do not compensate the increase in total mesh capacitance and as a consequence the power consumption.





## 7 SKEW REDUCTION BY MESH BUFFER DISPLACEMENT

Mesh buffers are usually placed over the mesh grid nodes and sized according to the load present in their vicinity. However, due to the non-symmetric characteristics of the clock sink distribution, some clock sinks are charged faster than the others.

Intuitively, if the mesh buffers are moved closer to the clock sinks with later clock arrival times and farther from the clock sinks with earlier arrival times the clock skew should be reduced.

In this chapter we propose and analyze a method for displacing buffers from their original position in order to reduce mesh clock skew. We develop a sensitivity based method to displace buffers using Elmore Delay (GUPTA et al., 1995). We start by creating an approach to displace buffers over the x-axis only and later extend it to the y dimension.

### 7.1 Elmore Delay

Spice electrical simulation is a very timing consuming step when simulating non-linear components (e.g. transistors). Generally the timing information is required several times inside the design flow, and therefore, for highly-iterative methods, Spice simulation may be prohibitive.

The Elmore model (GUPTA et al., 1995) can be used as a fast, upper bound approximation of the actual delay of an RC network. It can be computed throughout a nodal analysis replacing capacitances by current sources of same magnitude and removing all voltage sources (MUSTAFA CELIK LARRY PILEGGI, 2002). The node voltage is the Elmore delay of the respective node.

Therefore we can find the Elmore delay for each RC network node,  $\vec{\tau}$ , by solving the linear system of equations 7.1 where  $G$  is the conductance matrix of the RC network and  $\vec{c}$  is the capacitance associated with each network node.

$$G\vec{\tau} = \vec{c} \quad (7.1)$$

Since the Elmore delay is related to the RC value of nodes, we can write the Equation 7.1 in a more straightforward fashion. The inverse of  $G$  is the resistance matrix,  $R$ , so we can write Equation 7.1 as in Equation 7.2.

$$G\vec{\tau} = \vec{c} \rightarrow \vec{\tau} = G^{-1}\vec{c} \rightarrow \vec{\tau} = R\vec{c} \quad (7.2)$$

#### 7.1.1 Conductance Matrix Definition

For an RC network with  $n$  nodes, the conductance matrix  $G$  is an  $n \times n$  symmetric matrix, which is built as follow. Let  $r_{ij}$  be the resistance connecting node  $i$  to node  $j$  and

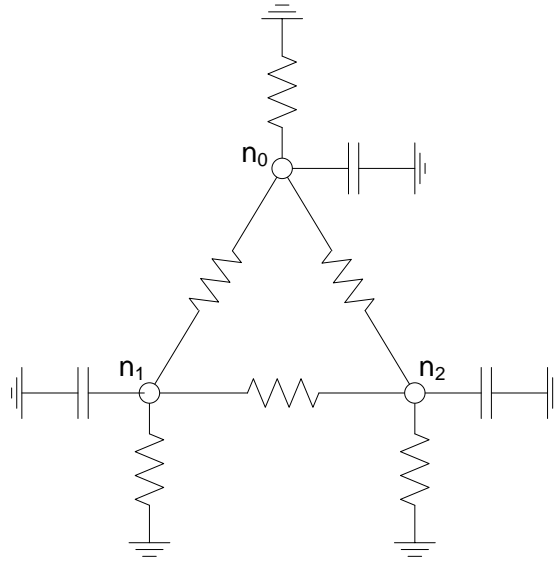


Figure 7.1: A Simple RC Circuit

$r_i$  the resistance connecting node  $i$  to ground (drive resistance) so the conductance matrix,  $G = [g_{ij}]$ , is defined by Equation 7.3.

$$g_{ij} = \begin{cases} -\frac{1}{r_{ij}} & i \neq j \\ \frac{1}{r_i} + \sum_{k=1, k \neq i}^n \frac{1}{r_{ik}} & i = j \end{cases} \quad (7.3)$$

Now we present a simple example on how to build the Elmore system for the RC network shown in Figure 7.1. The corresponding linear system is shown in Equation 7.4.

$$\begin{bmatrix} \frac{1}{R_0} + \frac{1}{R_{01}} + \frac{1}{R_{02}} & -\frac{1}{R_{01}} & -\frac{1}{R_{02}} \\ -\frac{1}{R_{01}} & \frac{1}{R_1} + \frac{1}{R_{01}} + \frac{1}{R_{12}} & -\frac{1}{R_{12}} \\ -\frac{1}{R_{02}} & -\frac{1}{R_{12}} & \frac{1}{R_2} + \frac{1}{R_{02}} + \frac{1}{R_{12}} \end{bmatrix} \begin{bmatrix} \tau_0 \\ \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} \quad (7.4)$$

Since matrix  $G$  is sparse and positive semi-definite (BHATIA, 2006) we can use the Incomplete Cholesky Conjugate Gradient method (SHEWCHUK, 1994) to solve the linear system in Equation 7.1.

## 7.2 Single-dimension displacement optimization

The goal of the displacement algorithm is to move buffers closer to sinks with greater delays and farther from sinks with lower delays so that clock skew is reduced. However, since in a mesh, elements are strongly connected, i.e. there are several paths connecting every pair of nodes, changing a single element property (e.g. buffer position) affects the delays of all mesh nodes. This high dependency makes the buffer displacement problem very complex to be solved exactly and then some heuristic is more appropriate to cope with it.

In this section a sensitivity based approach is presented which, although not exact, can be tuned to trade-off between accuracy and execution time. The main idea is to compute delay changes (sensitivities) for each sink w.r.t a buffer displacement while other buffers are kept in their original position. These sensitivities are put together in a linear program, which finds a displacement for each buffer minimizing the clock skew.

### 7.2.1 Terms and Conventions

We start by defining all terms used in this chapter and present some labeling conventions to facilitate our discussion.

Let:

- $S = \{s_1, s_2, \dots, s_{n_s}\}$  be the set of all  $n_s$  clock sinks, where  $s_i$  is the  $i^{th}$  sink;
- $B = \{b_1, b_2, \dots, b_{n_b}\}$  be the set of all  $n_b$  clock buffers, where  $b_j$  is the  $j^{th}$  clock buffer;
- $\delta_{max}$  be the maximum allowed buffer displacement;
- $\vec{p} = [p_1, p_2, \dots, p_{n_b}]^T$  be the buffer original positions;
- $\vec{\delta} = [\delta_1, \delta_2, \dots, \delta_{n_b}]^T$  be the buffer displacements;
- $\vec{\tau} = [\tau_1, \tau_2, \dots, \tau_{n_s}]^T$  be the sink delay vector, where  $\tau_i$  is the delay of sink  $s_i$  when all buffers are placed at their original positions;
- $\tau_{ij}^+$  be the delay of sink  $s_i$  when buffer  $b_j$  is displaced by  $+\delta_{max}$  over x-dimension and other buffers are kept in their original positions;
- $\tau_{ij}^-$  be the delay of sink  $s_i$  when buffer  $b_j$  is displaced by  $-\delta_{max}$  over x-dimension and other buffers are kept in their original positions.

Throughout this chapter, we use  $i$  for indexing sinks and  $j$  for indexing buffers.

### 7.2.2 Problem Definition

For a given  $m \times m$  clock mesh and  $b$  buffers placed initially at  $\vec{p} = [p_1, p_2, \dots, p_b]^T$  driving a set  $S$  of  $k$  sinks, find a displacement  $\vec{\delta} = [\delta_1, \delta_2, \dots, \delta_b]^T$  in the x-axis for every mesh buffer such that skew is minimized. Figure 1 presents the algorithm in pseudo code.

**Input:**  $m$  - mesh grid size

$S$  - set of sinks

**Output:**  $\delta x$  - set of displacements

```

1 repeat
2    $Init\_sk = compute\_skew(m, S, 0);$ 
3    $A = compute\_sensitivities(m, S);$ 
4    $\delta x = linear\_solver(A);$ 
5    $Final\_sk = compute\_skew(m, S, \delta x);$ 
6 until no significant skew improvement occurs ;

```

Figure 7.2: Multi-step optimization algorithm.

### 7.2.3 Buffer Displacement and Sink Delay Computation

The algorithm starts by computing the sink delays,  $\vec{\tau}$ , when all buffers are placed in their original positions  $\vec{p}$ . After that, each buffer,  $b_j$ , is displaced in x-dimension w.r.t. its original position,  $p_j$ , by  $+\delta_{max}$  and next by  $-\delta_{max}$  keeping the remaining buffers at their original positions.

Every time a buffer is displaced, the sink delay vary. The delay of a sink  $i$  w.r.t. a positive displacement of buffer  $j$  is stored in  $\tau_{ij}^+$ . For a negative displacement, the delay is stored in  $\tau_{ij}^-$ . Figure 7.3 shows buffer  $j$  being displaced and the delay variations in sink  $i$ .

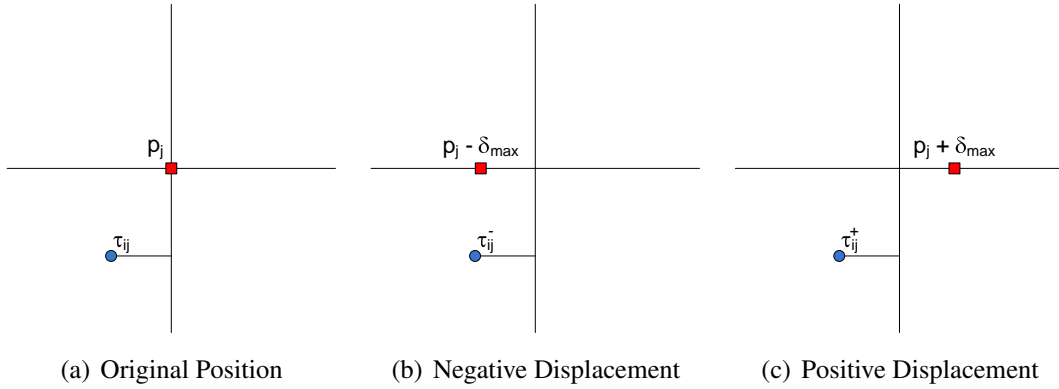


Figure 7.3: The delay of sink  $i$  w.r.t the buffer  $j$ .

### 7.2.4 Sink Delay Sensitivities

Now delay sensitivity can be estimated using  $\tau_{ij}^+$  and  $\tau_{ij}^-$  through linear interpolation. The sensitivity of sink  $s_i$  due to buffer  $b_i$  begin displaced is given by Equation 7.5. Figure 7.4 presents the idea behind  $\alpha$  parameters.

$$\alpha_{ij} = \frac{\tau_{ij}^- - \tau_{ij}^+}{2\delta_{max}} \quad (7.5)$$

We can write sensitivities in matrix form,  $A$ , as in Equation 7.6. The matrix  $A$  has its number of lines determined by  $n_s$  and the number of columns determined by  $n_b$ .

$$A = [\alpha_{ij}] \quad (7.6)$$

### 7.2.5 Sink Delay Estimation using Sensitivities

When buffers are displaced, we can estimate delay of the sink  $i$ ,  $d_i$ , as a fixed part plus a linear combination of delay sensitivities w.r.t the displacements of each mesh buffer as in Equation 7.7. The fixed part is the delay of sink,  $\tau_i$ , when buffers are placed at their original position.

$$d_i = \tau_i + \sum_{j=1}^{n_b} \alpha_{ij} \times \delta_j \quad (7.7)$$

Delay estimative can be also written in matrix form as in Equation 7.8 where  $\vec{d} = [d_1, d_2, \dots, d_{n_s}]^T$ .

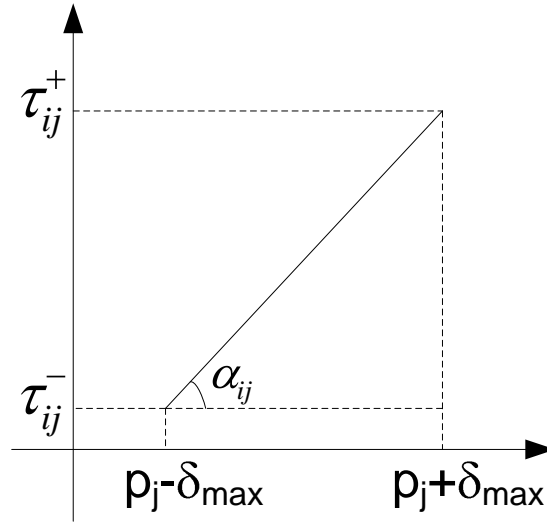


Figure 7.4: Computation of sensitivity,  $\alpha_{ij}$ , of a sink  $i$  w.r.t the buffer  $j$ .

$$\vec{d} = A\vec{\delta} + \vec{\tau} \quad (7.8)$$

### 7.2.6 Skew Reduction as a Linear Program

By expressing the clock sink delays as a function of the displacement an optimization problem can be formulated to reduce the clock skew. The Equation 7.9 shows the optimization problem formulation.

Two different types of constraints are added: constraints to limit the buffer displacement only to valid positions and constraints to limit the maximum displacement in a single optimization round. The variables  $xmin$  and  $xmax$  represent the interval of valid positions for each mesh buffer. The vector  $\vec{p} = [p_1, p_2, \dots, p_b]^T$  represents the absolute position for each mesh buffer.  $\delta xmax$  represents the maximum allowed buffer displacement.

$$\begin{aligned} &\text{minimize} && \max(A\delta x + \tau) - \min(A\delta x + \tau) \\ &\text{subject to} && \delta x + p \geq xmin \\ &&& \delta x + p \leq xmax \\ &&& |\delta x| \leq \delta xmax \end{aligned} \quad (7.9)$$

The optimization problem shown in the Equation 7.9 can be easily solved using linear programming. It should be noticed that Equation 7.7 is a rough approximation to capture the dependency of the clock sinks delays as a function of the mesh buffer displacements.

This equation is valid only for small displacement values, therefore the maximum displacement values had to be defined. Since a set of displacement values may contain values too large to be treated in a single optimization round, several optimization rounds have to be performed. During each iteration sensitivities are computed for all clock sinks w.r.t all mesh buffers considering their current positions. New displacements are computed using linear programming. At last the skew improvement is computed. The algorithm stops when the skew improvement is not substantial.

### 7.3 Two-dimensional displacement optimization

In the previous section we presented an algorithm to optimize the mesh buffer displacements in order to reduce the clock skew. The presented solution allows the mesh buffers to be displaced only in a predefined direction, either in the  $x$ -axis or in the  $y$ -axis. It is necessary to determine in which direction the mesh buffers should be displaced. Once the displacement directions are determined the algorithm presented in Listing 1 can be applied. A simple manner of determining the displacement direction of a given mesh buffer relies on the application of the one dimensional optimization in both directions and then evaluating which one has produced the largest displacement. Algorithm 2 shows the algorithm used to define the displacement direction for each mesh buffer.

**Input:**  $m$  - mesh grid size  
 $S$  - set of sinks  
**Output:**  $\delta xy$  - set of displacements directions

```

1 set all displacement directions towards the x-axis;
2  $\delta x = \text{compute\_displacements}(m, S)$ ;
3 set all displacement directions towards the y-axis;
4  $\delta y = \text{compute\_displacements}(m, S)$ ;
5 foreach mesh buffer  $i$  do
6   | if  $\delta x > \delta y$  then
7   |   | set  $\delta xy(i) = x$ ;
8   | end
9   | else
10  |   | set  $\delta xy(i) = y$ ;
11  | end
12 end

```

Figure 7.5: Displacement direction computation.

After the displacement direction for all mesh buffers is defined the *compute\_displacements* algorithms is executed again using the chosen displacement directions and stops only after the algorithm has converged. Although we have observed that the proposed algorithm usually yields a better solution than executing the displacements in a single orientation this cannot be guaranteed.

## 7.4 Experiments

In this section we perform experiments to show how the displacement technique improves clock skew and its limits.

### 7.4.1 Setup

For testing our method, we have used the ISCAS89 benchmarks (ISCAS89 BENCHMARKS, 1989). The placement of the benchmarks are obtained from (ISCAS89 PLACEMENT, 1989), which used the UCLA Dragon placer (TAGHAVI; YANG; CHOI, 2005). The mesh size for each benchmarks was set so that it minimizes total mesh capacitance. The details of the benchmarks used in this section are presented in Table 7.1.

To analyze the technique under more realistic scenarios, different input arrival times in each mesh buffer are applied. The input arrival times for mesh buffers are modeled

Table 7.1: ISCAS Benchmarks

	#Sinks	Mesh Size
s5378	179	4
s9234.1	211	4
s13207.1	638	7
s15850.1	534	7
s3858.1	1426	11

by a random variable with a maximum variation set in the range  $[0, 50ps]$ . This random variable can simulate roughly clock skew, clock jitter and buffer variations, which are the most harmful variations for clock mesh (ABE; HASHIMOTO; ONOYE, 2008). For each experiment, 1000 Monte Carlo simulations are performed.

Note that, although the displacement technique uses Elmore delay to compute sensitivities, the results are extracted using SPICE simulation.

#### 7.4.2 Results

Table 7.2 presents average results for each ISCAS89 benchmarks and the overall average results for each maximum input delay. As it can be noticed, the average improvement is about 23% when no input delay is applied. On the other hand the improvements go down as the maximum input arrival time increases. This trend is clear in Figure 7.6.

Table 7.2: ISCAS Optimization Result

InputSkew (ps)	s5378	s9234.1	s13207.1	s15850.1	s3858.1	Avg
0	28.90%	25.00%	15.79%	31.19%	18.67%	23.91%
5	19.07%	20.19%	9.49%	22.63%	10.91%	16.46%
10	13.01%	15.25%	5.94%	15.04%	5.92%	11.03%
15	9.46%	11.85%	4.45%	10.44%	3.59%	7.96%
20	7.32%	9.66%	3.76%	7.69%	2.35%	6.16%
25	5.92%	8.11%	3.28%	5.90%	1.64%	4.97%
30	4.92%	6.99%	2.98%	4.67%	1.20%	4.15%
35	4.18%	6.12%	2.72%	3.80%	0.91%	3.55%
40	3.60%	5.42%	2.52%	3.21%	0.68%	3.09%
45	3.18%	4.87%	2.35%	2.75%	0.53%	2.74%
50	2.82%	4.41%	2.20%	2.42%	0.41%	2.45%

As the maximum input skew increases, it tends to overcome the intrinsic mesh skew caused by the unevenly sink distribution over the circuit area. Since the buffer displacement method acts basically by reducing the intrinsic mesh skew, its improvements becomes less significant.

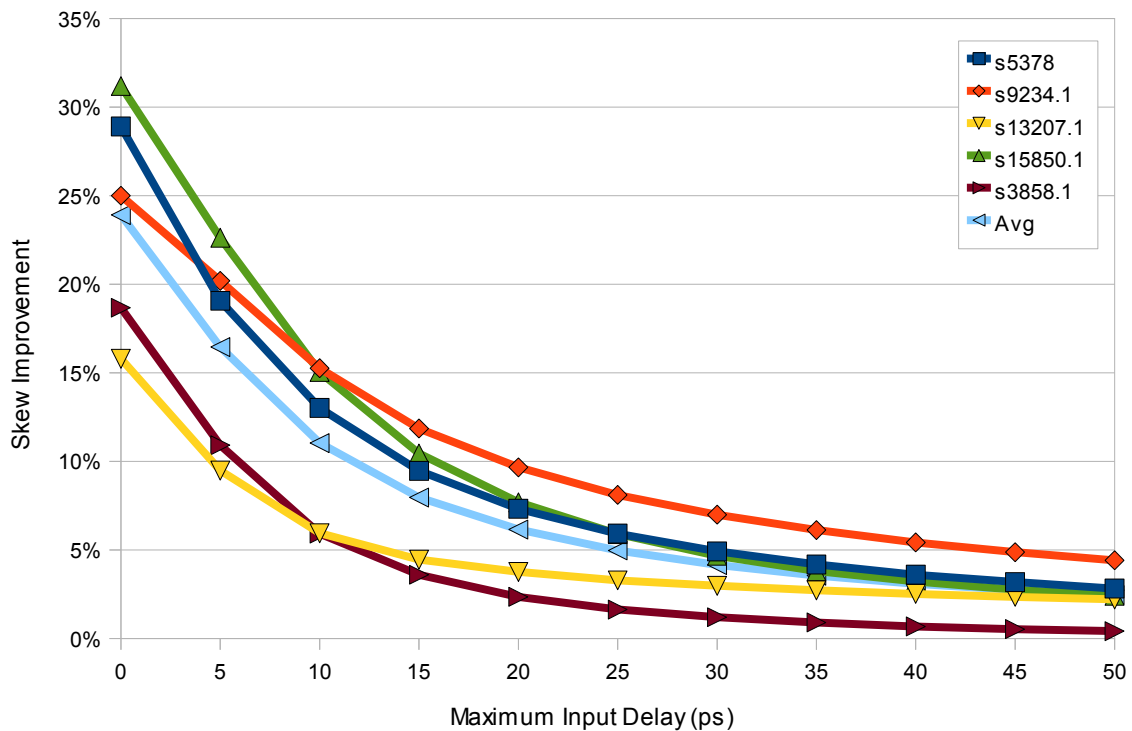


Figure 7.6: ISCAS Optimization Results

## 7.5 Conclusion

The proposed algorithm is effective to reduce the clock skew in mesh based architectures even when variability is applied. For no input skew, the improvement can reach up to 23%. When input skew is applied, the improvements reach in average 11% for 10ps input skew and 5% for 25ps. It should be noticed that the optimization strategy was designed focusing on the nominal 0 skew case, and, for that reason, its improvements are reduced when the skew at the mesh buffers is increased. In practice, for high performance designs, clock skew constraints are very tight. Hence the skew observed at the inputs of the clock mesh is expected to be small.

There is no overhead associated with this technique since the total mesh capacitance is not changed as the buffer sizes are kept constant. In fact, by reducing clock skew, the total mesh power consumption is reduced. Less skew means that less short circuit current flows through the mesh so that the total power consumption is lesser when compared to the non-optimized mesh.



## 8 MESH EXPLORER: A CLOCK MESH TOOL

In this chapter we present the Mesh Explorer tool created to guide the development and analysis of new algorithms and methods on clock meshes. A screen shot of Mesh Explorer is presented in Figure 8.

### 8.1 Programming Language and External Libraries

The Mesh Explorer was developed in C++ using the wxWidgets (SMART; HOCK; CSOMOR, 2005) library for Graphical User Interface (GUI). wxWidgets is a platform independent GUI library, and so allows Mesh Explorer to be compiled for most popular platforms like Windows, Mac OS, Linux. In this work we used a Windows-compiled version of the Mesh Explorer.

The wxFormBuilder (WXFORMBUILDER, 2010) tool was used to design the layout of the user interface. By using wxFormBuilder, the programmer create the interface visually and then generate C++ code automatically. Figure 8.1 shows the main window of the Mesh Explorer being designed with wxFormBuilder tool.

### 8.2 Key Features

Mesh Explorer exposes to the user several useful features, which allow researchers to quickly get insights about how mesh properties are affected by modifications in the mesh structure. Besides guiding the development of new algorithms for mesh optimization, Mesh Explorer can also be used as a didactic tool for introducing students in this clock distribution architecture. In next topics, we present the key features of the Mesh Explorer.

#### 8.2.1 Spice Simulation

Mesh Explorer allows the user to perform Spice electrical simulation by means of hpsice software installed in a remote server. A daemon process is installed in the server side, which waits for simulation requests from the Mesh Explorer tool. After setup, all simulation process is done transparently to the user and it is executed entirely by clicking in a single button. Figure 8.3 shows the process behind an electrical simulation.

User are able to select Monte Carlo simulation for simulating process variability. For variability, there are two options:

- **Input skew variability** simulates the process variability, which causes the clock signal to arrive unevenly in clock mesh buffers.
- **Process variability** simulates variability in transistor widths, transistor lengths,

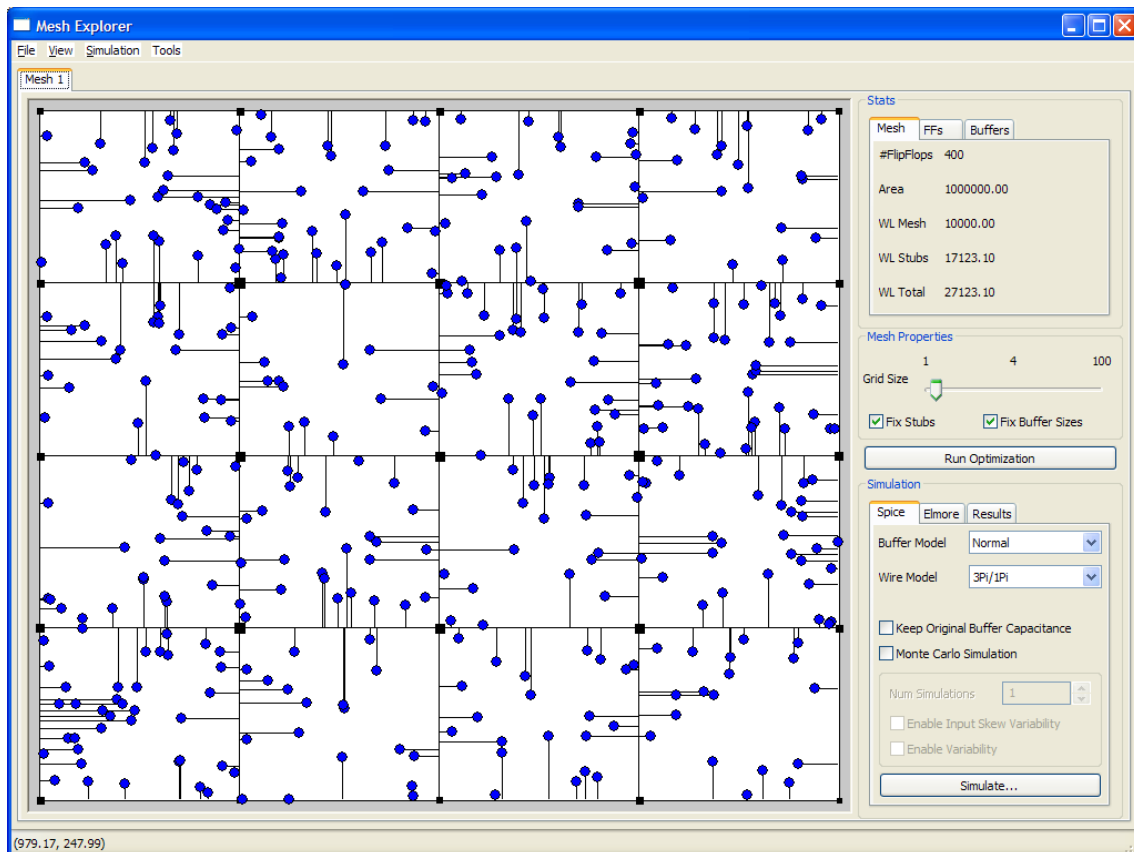


Figure 8.1: Mesh Explorer Tool.

voltage source, wire width and sink capacitances.

### 8.2.2 Elmore Simulation

For a fast delay estimate, Mesh Explorer allows the user to perform an Elmore Simulation of the mesh. Elmore delay provides accurate results, but are not able to simulate input skew.

### 8.2.3 Sink Properties Visualization

User can analyze sink properties graphically using different view modes. Each view mode use a darker color to identify sinks with lesser property values and lighter colors to identify greater ones. The sink with minimum property value is highlighted with a green circle and the sink with maximum property value is highlighted by a yellow circle. The available view modes are:

- **Buffer Dist:** Sinks are painted based on their distance to the nearest buffer. Sinks with lesser distance are painted darker.
- **Stub Length:** Sinks are painted based on their stub length. Sinks with lesser stub lengths are painted darker.
- **Delay:** Sinks are painted based on their delay values. Sinks with lesser delay are painted darker.

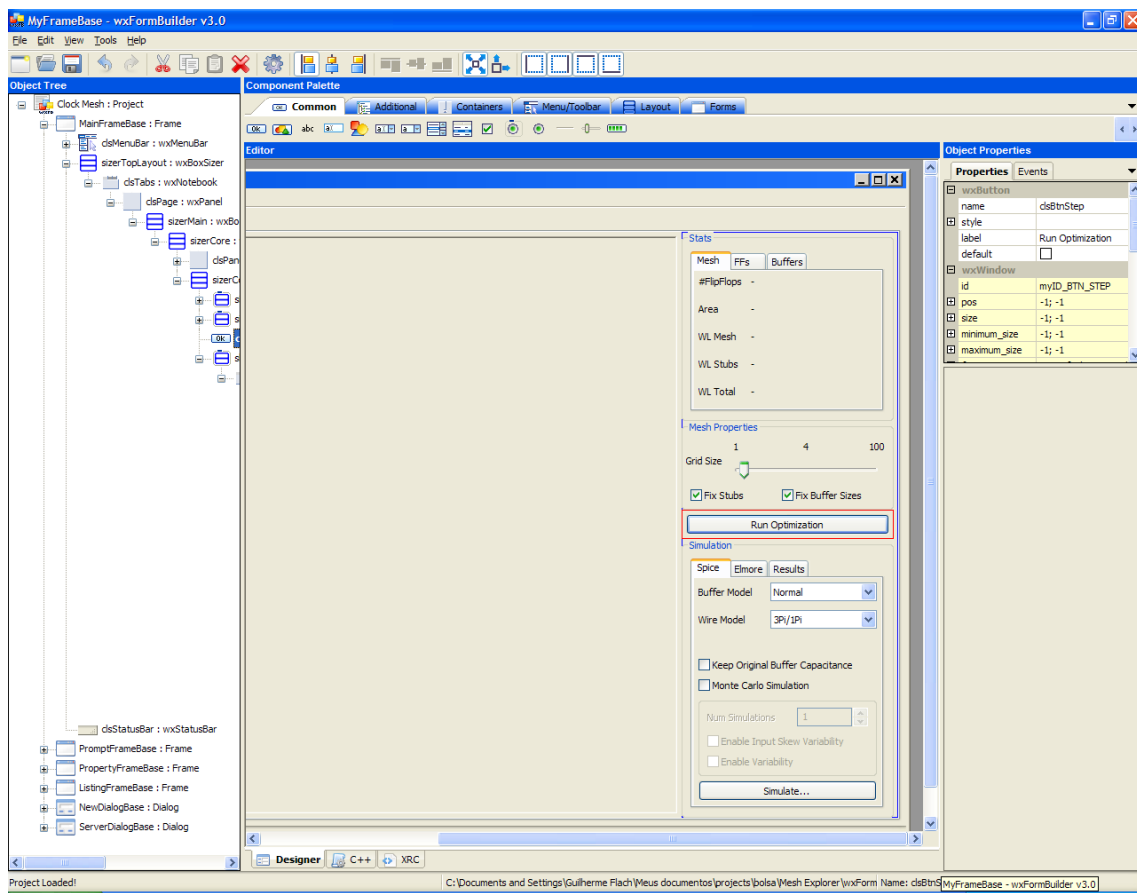


Figure 8.2: Main Window of Mesh Explorer Being Designed Using wxFormBuilder.

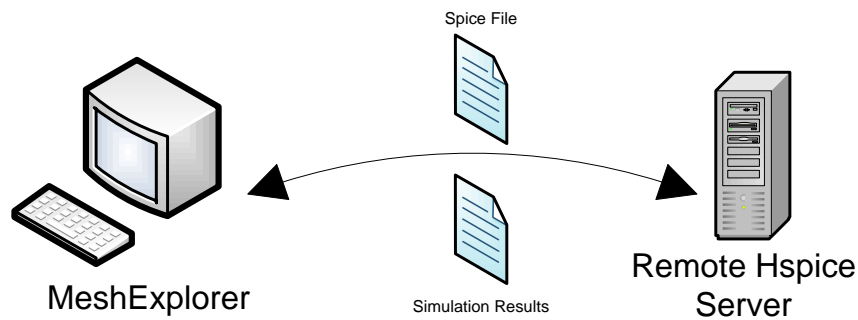


Figure 8.3: Electrical Spice Simulation.

- **Fall Slew:** Sinks are painted based on their fall slew values. Sinks with lesser fall slew are painted darker.
- **Rise Slew:** Sinks are painted based on their rise slew values. Sinks with lesser rise slew are painted darker.

Figure 8.4 presents sinks in the delay view mode.

#### 8.2.4 Changing Buffer and Sink Properties

By double-clicking on a buffer or on a sink, a pop-up window displays their respective properties. Some properties can be changed manually by the designer. Table 8.1 and Table 8.2 present the available properties for buffers and sinks respectively.

Table 8.1: Buffer Properties

Name	Type	Description
Size	Read/Write	Change the buffer size.
Load Capacitance	Read Only	Display the total load capacitance, which is used to size the buffer.
Input Skew	Read/Write	Used to apply an initial delay on buffer.

#### 8.2.5 Dragging Buffers

In traditional mesh design, buffers are placed on the intersection of horizontal and vertical wires of the mesh. However, Mesh Explorer allows user to move buffers away from its original position. By moving buffers sink delays are changed and this can be useful for clock skew reduction as shown in Chapter 7.

#### 8.2.6 Mesh Optimization Through Buffer Displacement

The most important feature of the Mesh Explorer is to perform clock mesh optimization through buffer displacement. As viewed in Section 8.2.5, user can move buffers manually. Although this is an important feature to get insights about how buffer displacement affects the clock mesh properties, it is a complex task to try to reduce skew using

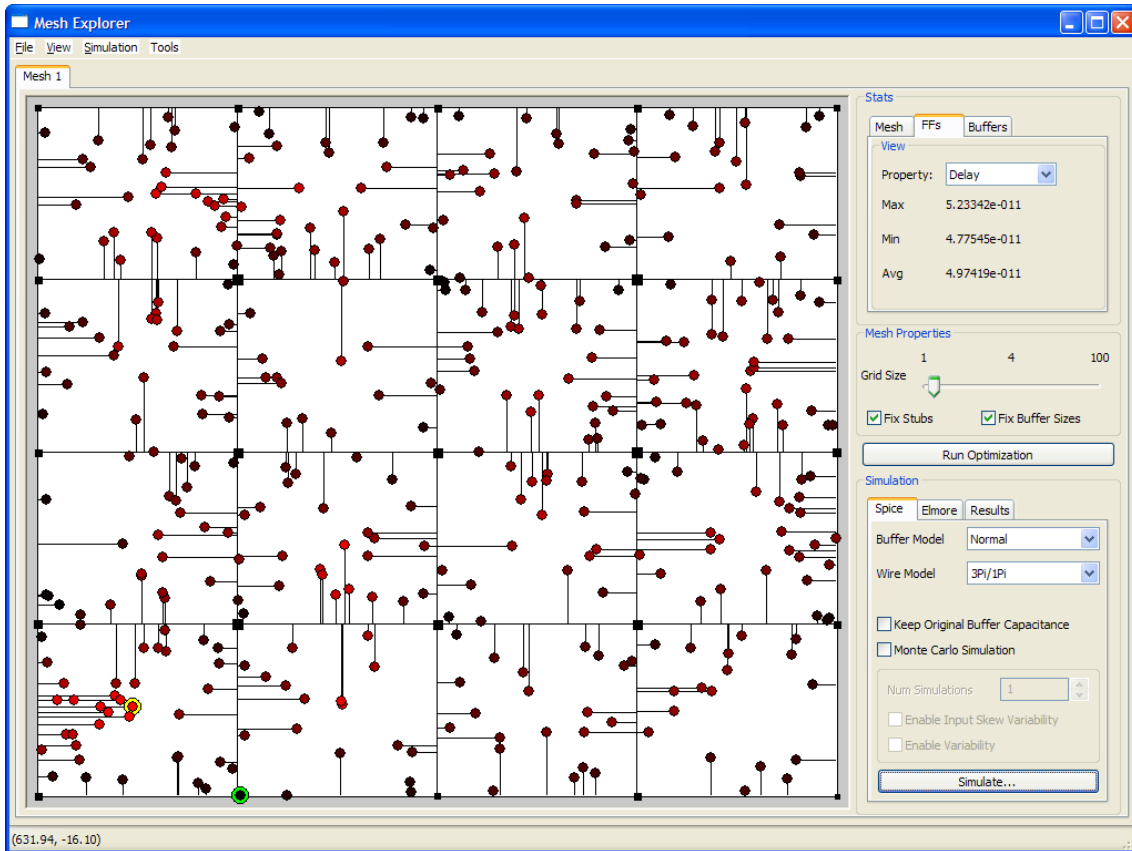


Figure 8.4: Mesh Explorer Delay View Mode.

Table 8.2: Sink Properties

Name	Type	Description
Distance	Read Only	Display the distance from the sink to the nearest buffer.
Stub Length	Read Only	Display the stub length.
Delay	Read Only	Display the delay of sink for the last simulation.
Rise Slew	Read Only	Display the rise slew of sink for the last simulation. Only for Spice simulation.
Fall Slew	Read Only	Display the fall slew of sink for the last simulation. Only for Spice simulation.
Capacitance	Read/Write	Sink capacitance.

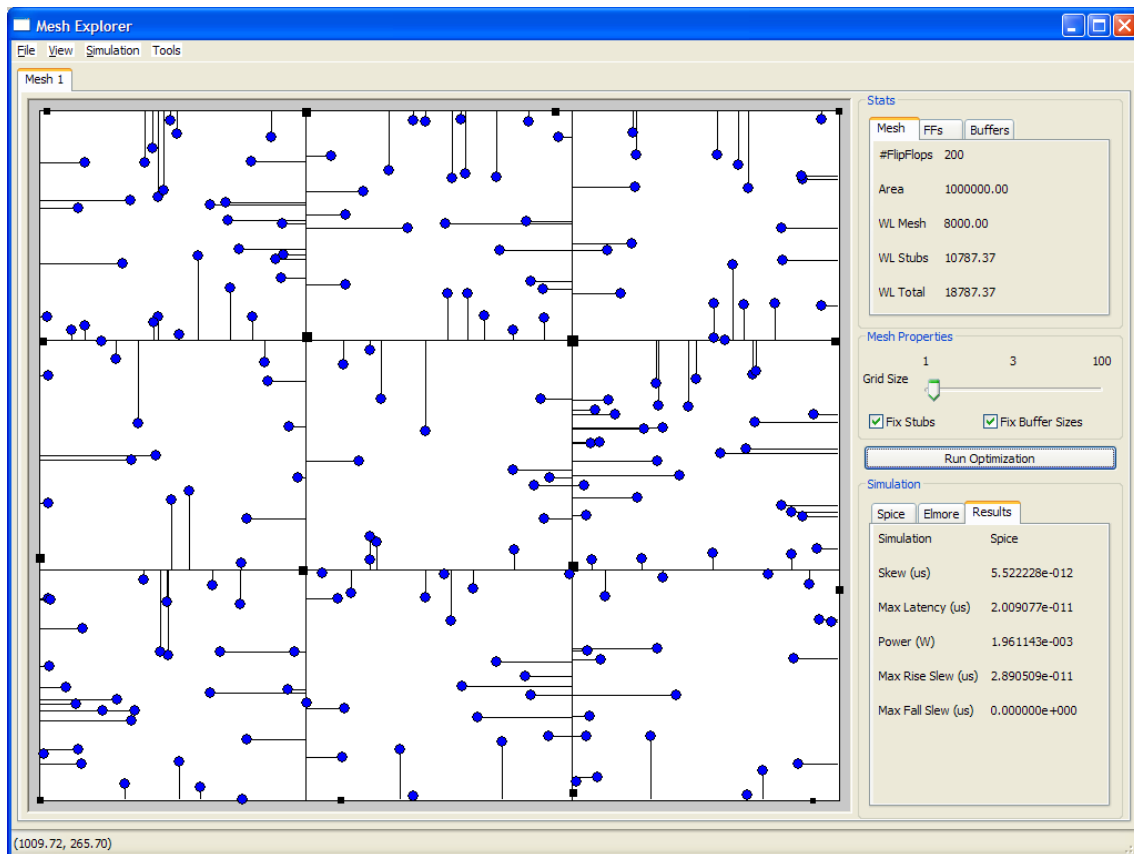


Figure 8.5: Clock skew optimization through buffer displacement.

only intuitive movements. This is because, in a clock mesh, there are many paths connecting every pair of nodes and so changes in a single mesh element may affect all mesh node properties.

The buffer displacement method allows the designer to improve clock skew automatically. This method iteratively builds and solves a linear program based on sink sensitivities w.r.t buffer displacement. The solution of the linear program is the buffer displacement. An example of mesh optimization can be viewed in Figure 8.5. The buffer displacement technique is detailed in Chapter 7.

## 9 CONCLUSIONS

In this work we have briefly reviewed some of the common clock network distribution architectures. We have pointed out that the power consumption and the clock signal arrival times discrepancy at each clock sink are the two main issues which designers must deal with when designing the clock network.

Among the clock network architectures, clock meshes arise as an effective way to reliably distribute clock signal under process and environmental variations. As there are many paths connecting clock signal to clock sinks, variations affecting one path may be compensated by others. This reliability comes at the cost of more power consumption since more wire resources are required, which in turn increases the total load capacitance driven by the clock network. The short circuit current caused by redundant paths are another source of power consumption. Therefore it is clear the trade-off between reliably distributing the clock signal and power consuming savings.

For dealing with the power consumption, we developed an analytical formula to find the optimum mesh size for total mesh capacitance minimization. This formula shows that the optimum mesh size basically depends on the number of sinks that must be driven, although using different types of wires for routing the mesh and stubs affects the optimum mesh size. We have also shown that meshes denser than the optimum mesh achieve little skew reduction at cost of increased power consumption.

Traditionally buffers are placed in the intersection of horizontal and vertical wires of a mesh. In this work, we have investigated how the clock skew can be reduced by moving buffers away from their original positions. This led to the development of the displacement buffer technique, which uses sink delay sensitivities calculation and linear programming to move the buffers. Results show that the skew can be reduced by up to 23% for meshes where no input skew is applied. When input skew is applied, the improvements in clock skew reach in average 11% for 10ps input skew and 5% for 25ps. Although the improvements decrease when input skew increases, this method does not increase the total mesh power consumption since the total mesh capacitance is not changed.

### 9.1 Future Work

A next step in this work is to investigate new techniques for reducing the clock skew under higher input skews along with mesh power consumption.

As shown in this work, denser meshes may not be worthwhile for reducing clock skew under higher input skew of the mesh buffers. In fact, high skew in the mesh buffers may indicate that the wrong clock network was chosen to drive such buffers. An interesting point to investigate is what is the best mix of clock architectures to compose the complete

clock network. For example, the mesh buffers themselves may be driven by a finer clock mesh in a hierarchical fashion.

An important way to reduce mesh power consumption is by removing some mesh edges which may not contribute significantly for skew reduction. In this case, the best mesh size may be slightly greater than the optimum mesh size to allow the removing technique to explore a large solution space.



## REFERENCES

ABE, S.; HASHIMOTO, M.; ONOYE, T. Clock Skew Evaluation Considering Manufacturing Variability in Mesh-Style Clock Distribution. **Quality Electronic Design, International Symposium on**, Los Alamitos, CA, USA, v.0, p.520–525, 2008.

BHATIA, R. **Positive Definite Matrices (Princeton Series in Applied Mathematics)**. [S.l.]: Princeton University Press, 2006.

CHANG, H.; SAPATNEKAR, S. S. Statistical Timing Analysis Considering Spatial Correlations using a Single Pert-Like Traversal. In: ICCAD '03: PROCEEDINGS OF THE 2003 IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, Washington, DC, USA. **Anais...** IEEE Computer Society, 2003. p.621.

CHAO, T.-H. et al. Zero Skew Clock Routing With Minimum Wirelength. **IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing**, USA, v.39, n.11, p.799–814, 1992.

CHEN, H. et al. A sliding window scheme for accurate clock mesh analysis. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, ICCAD. **Anais...** [S.l.: s.n.], 2005. p.939–946.

CLABES, J. et al. Design and Implementation of the POWER5 Microprocessor. In: DESIGN AUTOMATION CONFERENCE, DAC, 41. **Anais...** [S.l.: s.n.], 2004. p.670–672.

FRIEDMAN, E. G. Clock Distribution Networks in Synchronous Digital Integrated Circuits. **Proceeding of IEEE**, [S.l.], v.89, n.5, p.665–692, May 2001.

FRIEDRICH, J. et al. Design of the Power6 Microprocessor. In: IEEE INTERNATIONAL SOLID STATE CIRCUITS CONFERENCE, ISSCC. **Proceedings...** [S.l.: s.n.], 2007. p.96–97.

GUPTA, R. et al. The Elmore delay as bound for RC trees with generalized input signals. In: DAC '95: PROCEEDINGS OF THE 32ND ANNUAL ACM/IEEE DESIGN AUTOMATION CONFERENCE, New York, NY, USA. **Anais...** ACM, 1995. p.364–369.

HART, J. M. et al. Implementation of a fourth-generation 1.8-GHz dual-core SPARC V9 microprocessor. **IEEE Journal of Solid-State Circuits**, [S.l.], v.41, n.1, p.210–217, Jan. 2006.

ISCAS89 Benchmarks. Available from Internet: <<http://www.ece.wisc.edu/~vlsi/tools/iscas-placement/index.html>>. Cited 2010 Jul 5.

ISCAS89 Placement. Available from Internet: <<http://www.ece.wisc.edu/~vlsi/tools/iscas-placement/index.html>>. Cited 2010 Jul 5.

JAN M. RABAEY ANANTHA P. CHANDRAKASAN, B. N. **Digital Integrated Circuits**. [S.l.]: Prentice-Hall, 2002.

KAHNG, A. B.; ALBERT TSAO, C. wen. Planar-DME: a single-layer zero-skew clock tree router. **IEEE Trans. Computer-Aided Design**, [S.l.], v.15, p.8–19, 1996.

KALLA, R.; SINHAROY, B.; TENDLER, J. M. IBM Power5 chip: a dual-core multi-threaded processor. **IEEE Micro**, [S.l.], v.24, n.2, p.40–47, Jan. 2004.

MUSTAFA CELIK LARRY PILEGGI, A. O. **IC Interconnect Analysis**. [S.l.]: Springer, 2002.

PHAM, D. C. et al. Overview of the architecture, circuit design, and physical implementation of a first-generation cell processor. **IEEE Journal of Solid-State Circuits**, [S.l.], v.41, n.1, p.179–196, 2006.

PHAM, D. et al. The design and implementation of a first-generation CELL processor. In: SOLID-STATE CIRCUITS CONFERENCE, ISSCC. **Anais...** [S.l.: s.n.], 2005. p.184–592.

PULLELA, S. et al. Skew and Delay Optimization for Reliable Buffered Clock Trees. In: IEEE INTL. CONF. ON COMPUTER-AIDED DESIGN. **Proceedings...** [S.l.: s.n.], 1993. p.556–562.

RAJARAM, A.; PAN, D. Z. MeshWorks: an efficient framework for planning, synthesis and optimization of clock mesh networks. In: CONFERENCE ON ASIA AND SOUTH PACIFIC DESIGN AUTOMATION, ASP-DAC, Seoul, Korea. **Anais...** Los Alamitos: IEEE Computer Society Press, 2008. p.250–257.

REDDY, S. M.; WILKE, G. R.; MURGAI, R. Analyzing timing uncertainty in mesh-based clock architectures. In: CONFERENCE ON DESIGN, AUTOMATION AND TEST IN EUROPE, DATE, 9., Munich, Germany. **Anais...** New York: ACM Sigda, 2006. p.1097–1102.

RESTLE, P. et al. The clock distribution of the Power4 microprocessor. In: IEEE INTERNATIONAL SOLID STATE CIRCUITS CONFERENCE, ISSCC. **Proceedings...** [S.l.: s.n.], 2002. p.144–145.

SHEWCHUK, J. R. **An Introduction to the Conjugate Gradient Method Without the Agonizing Pain**. Available from Internet: <<http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf>>. Cited 2010 Jul 5.

SMART, J.; HOCK, K.; CSOMOR, S. **Cross-Platform GUI Programming with WxWidgets**. [S.l.]: Pearson Professional Education Prentice Hall PTR, 2005.

SUTHERLAND, I.; SPROULL, B.; HARRIS, D. **Logical effort**: designing fast cmos circuits. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.

TAGHAVI, T.; YANG, X.; CHOI, B.-K. Dragon2005: large-scale mixed-size placement tool. In: ISPD '05: PROCEEDINGS OF THE 2005 INTERNATIONAL SYMPOSIUM ON PHYSICAL DESIGN, New York, NY, USA. **Anais...** ACM, 2005. p.245–247.

TAM, S. et al. Clock generation and distribution for the first IA-64 microprocessor. **IEEE Journal of Solid-State Circuits**, [S.l.], v.35, n.11, p.1545–1552, Nov. 2000.

THOMSON, M. G. R.; RESTLE, P. J.; JAMES, N. K. A 5GHz Duty-Cycle Correcting Clock Distribution Network for the POWER6 Microprocessor. In: IEEE INTERNATIONAL SOLID STATE CIRCUITS CONFERENCE, ISSCC. **Proceedings...** [S.l.: s.n.], 2006. p.1522–1529.

TSAO, C.-W. A.; KOH, C.-K. UST/DME: a clock tree router for general skew constraints. **Computer-Aided Design, International Conference on**, Los Alamitos, CA, USA, v.0, p.400, 2000.

ULLMA, J. D. **Computational Aspects of VLSI**. New York, NY, USA: W. H. Freeman & Co., 1984.

VENKATARAMAN, G. et al. Combinatorial algorithms for fast clock mesh optimization. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, ICCAD, San Jose, California. **Anais...** New York: ACM Press, 2006. p.563–567.

WARNOCK, J. D. et al. The circuit and physical design of the POWER4 microprocessor. **IBM Journal of Research and Development**, [S.l.], v.46, n.1, p.27–52, Jan. 2002.

WILKE, G. **Analysis and Optimization of Mesh-based Clock Distribution Architectures**. 2008. PhD on Microelectronics — Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, RS - Brazil.

WXFORMBUILDER. Available from Internet: <<http://wxformbuilder.org/>>. Cited 2010 Jul 5.

XANTHOPOULOS, T. et al. The Design and Analysis of the Clock Distribution Network for a 1.2GHz Alpha Microprocessor. In: IEEE INTERNATIONAL SOLID STATE CIRCUITS CONFERENCE, ISSCC. **Proceedings...** [S.l.: s.n.], 2001. p.402–403.

YEH, C. et al. Clock Distribution Architectures: a comparative study. In: INTERNATIONAL SYMPOSIUM ON QUALITY ELECTRONIC DESIGN, ISQED, 7., San Jose, CA. **Anais...** Los Alamitos: IEEE Computer Society, 2006. p.85–91.

ZHAO, W.; CAO, Y. Predictive technology model for nano-CMOS design exploration. **J. Emerg. Technol. Comput. Syst.**, New York, NY, USA, v.3, n.1, p.1, 2007.

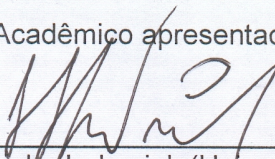


**“Clock Mesh Optimization”**

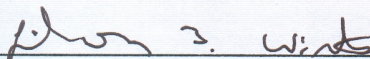
por

**Guilherme Augusto Flach**


Dissertação de Mestrado Acadêmico apresentada aos Senhores:



Prof. Dr. Leandro Indrusiak (Universidade de York)



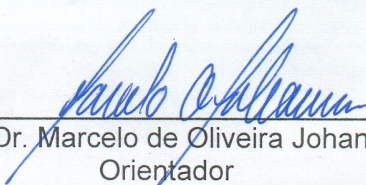
Prof. Dr. Gilson Inácio Wirth (PGMicro/UFRGS)



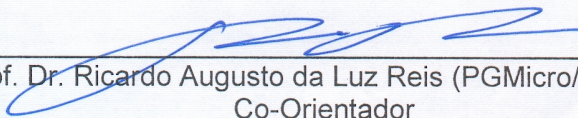
Prof. Dr. Renato Perez Ribas (PGMicro/UFRGS)

Vista e permitida a impressão.

Porto Alegre, 06/07/2019



Prof. Dr. Marcelo de Oliveira Johann  
Orientador



Prof. Dr. Ricardo Augusto da Luz Reis (PGMicro/UFRGS)  
Co-Orientador