Wire sizing, repeater insertion and dominant time constant optimization for a bus line

Moreno Bedoya, David L.

Award date:
2004

TECHNISCHE UNIVERSITEIT EINDHOVEN

Department of Mathematics and Computer Science

MASTER'S THESIS

Wire Sizing, Repeater Insertion and Dominant Time Constant Optimization for a Bus Line.

by
David Leonardo Moreno Bedoya

Supervisors: Prof.dr.ir. W.H.A. Schilders
Dr. T. Heijmen

Eindhoven, August 2004

Dedicated to Alejandra María my companion for many years and now my wife, to my mother Graciela for making me what I am today, to my father Orlando for the great values he gave me, to my uncle Jaime for his continuous support, to Ivonne and Vanessa my two beautiful sisters, to Tina Schneider for her support and many many others that helped and believed in me.

Acknowledgements

Special thanks to the Scientific Computing group of the Technical University of Eindhoven for the opportunity of receiving a great education.

Special thanks to Tino Heijmen, Atul Katoch, and W.H.Schilders which were supervisors the first two at Philips research and the latest at TU/e.

Special thanks to COLFUTURO, the institution that made possible the dream of studying in Holland.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The trend of the continuing scaling of process technology, the increase of the resistance per unit length and the decrease of transistor delay have a growing effect on the dominance of the interconnect delay over transistor delay. Research on optimal interconnect topology has been undertaken in the past decades with the aim to provide an accurate and inexpensive tool for the designer in his quest for performance, area and power optimization. A topology that supplies low power consumption as well as short delay has been the objective of VLSI systems developers and for this many mathematical methods have been proposed.

The conventional approach used for optimal interconnect sizing is the Elmore delay[28] approach. The Elmore delay is a posynomial function of the conductances and capacitances in the circuit. However, the use of Elmore delay leads to convex optimization problems only in a few cases. On the other hand, the dominant time constant [3], which is a quasiconvex function of the design parameters and uses the maximum eigenvalues of the capacitance and conductance matrices, can be used to cast convex optimization problems in many cases. These present a great advantage: they can be optimized very efficiently using recently developed interior point polynomial methods.

Because the Elmore delay has no useful convex properties (except when the RC circuit has a tree topology) that can be used for optimization, we chose the dominant time constant as the delay metric in this research. The dominant time constant approach allows the optimization of topologies others than RC trees, like clock meshes and those including coupling capacitances. Due to smaller feature sizes the coupling capacitance increases and as a result we obtain crosstalk between the wires.

This document deals with research based on the dominant time constant approach and finds a trade-off curve between area, delay and power that provides useful information for the uniform wire sizing topology (UWS). In this topology, (UWS) all the wires have the same width and all the repeater have the same size. This topology has been chosen because the combination of optimal repeater insertion, uniform wire sizing (UWS) and uniform gate sizing generally yields sufficiently accurate results so that any additional benefit from op-

timization using non-uniform wire sizing (NUWS) becomes marginal[22]. This is the main reason why we decided to study UWS instead of NUWS.

This document gives an overview of the results obtained in the past. We investigated the uniform wire sizing approach with the purpose of providing a function the designer could use to find optimal values for the design parameters such as the widths of the wires, repeaters sizes and spaces between wires, such that low power consumption and short delay is provided as stated before.

The structure of this document is as follows: the second chapter reviews convex functions and semidefinite programming, followed by some theory and background on the circuit modelling in chapter three, the fourth chapter presents interior point polynomial methods with the intention of giving an understanding of the process behind the optimization software used in the research, the fifth chapter contains theory on interpolation methods, which will provide new trade-off points between area, power and delay based on the data collected from the simulations. Finally, the sixth chapter will be dedicated mainly to the description of the uniform wire sizing problem, the approach that is used and the results that have been obtained.

1.1 Previous work

The conventional approach to interconnect sizing is based on linear RC models and on the Elmore delay[28] as a measure of signal propagation delay. This approach was introduced by Rubenstein et Al [29].

The Elmore delay is used extensively in circuit optimization because it is a posynomial function of the conductances and capacitances and this allows the use of convex and geometric programming to optimize the signal propagation delay. The called TILOS approach[30] was the first of a family of optimization methods that use Elmore delay and are based on geometric programming.

In 1995, Vandenberghe and Boyd [3] proposed the dominant time constant as an alternative measure of delay and they showed that it has several important advantages over Elmore delay when applied in circuit optimization:

1. A far wider class of circuits can be handled, including circuits with non-grounded capacitances and resistive networks.

2. The dominant time constant approach can be formulated as a convex problem and for this kind of problems very efficient interior point polynomial methods have been developed.

Vandenberghe and Boyd designed and developed a software package called sp [31](Software for semidefinite programming) with the objective of implementing the methods based on

the dominant time constant. The software was extensively used but still it presents great disadvantages, however. One of this is that both, the primal and the dual problem have to be specified. Furthermore, starting points for the primal as well as for the dual problem have to be provided.

T.Heijmen and Van Staalduinen [1] used the dominant time constant approach for real life problems but used a software package called SeDuMi instead of sp[6]. SeDuMi was developed by Sturm and is currently one of the most widely applied tools for semidefinite programming. In SeDuMi only the primal problem has to be specified, the software then finds the solution of the dual problem also. Furthermore, SeDuMi allows to start with a so called "cold start" (not a feasible point). In that case, it first find a feasible region and then selects a feasible point within this region.

Chapter 2

Convex Functions and Semidefinite Programming

2.1 Introduction

This chapter gives an important introduction to the world of convex functions and semidefinite programming. This is important due to the fact that the dominant time constant which is the specific model to study for the delay, is a quasiconvex function of the design parameters and for being so, allows the use of new efficient developed interior point polynomial methods that provide accurate solutions to the problems in consideration.

Furthermore, our main study is based on the semidefinite programming itself which allows the formulation of optimization problems and covers theory ranging from linear classical problems going through square problems to Lorentz cones. For a more detail review of convexity and semidefinite programming, the user is referred to Vandenberghe [5] and [4].

2.2 Convex Functions

Definition

A function $f : R^n \rightarrow R$ is *convex* if its domain is a convex set, and if for all $x, y \in$ **domain**(f):

$$\begin{array}{l} f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \\ \forall 0 \leq \lambda \leq 1 \end{array} \tag{2.1}$$

Geometrically, this can be seen as a *chord* from points $x, f(x)$ to $y, f(y)$ that lies above the graph of f. f is strictly convex if the strict inequality of (2.1) holds and if $x \neq y$. It is called concave if $-f$ is convex.

An important fact is that for an affine function(linear) we always have equality in (2.1), so

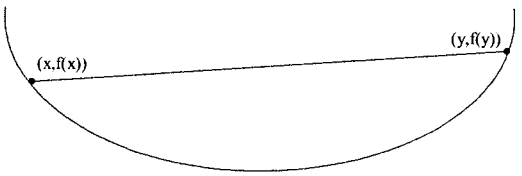


Figure 2.1: Convex Function.

all affine functions are both convex and concave functions and viceversa, because:

$$f \text{ is convex iff} \qquad g(t) = f(x+vt) \quad \text{is convex}$$

$$\forall\ x \in \text{domain}(f) \wedge \forall\ v \in R^n \quad \wedge\{t|x+tv\} \in \text{domain(f)}$$

This property is very useful because it can be used to find the convexity of a set by tracing and restricting the function to a line.

First-Order Conditions

Suppose the function f has a gradient for any point of its domain then f is convex if and only if its domain is convex, and (using taylor series to describe the function in series terms and truncating it at $\mathcal{O}(1)$):

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) \tag{2.2}$$

holds $\{\forall\ x, y \mid x, y \in \text{domain}(f)\}$

This property results very useful to determine the convexity of certain functions through its local information points.[1]

Second-Order Conditions

Assuming now that $\exists \Delta f$, that is its *Hessian* or second derivative exists at each point $\forall$ dom(f), which is open; then f is convex if and only if dom(f) is convex and its *Hessian*

[1] for a proof of the equation (2.2) please refer to [5].

is semidefinite positive, that is.

$$\nabla^2 f(x) \geq 0 \tag{2.3}$$

This two conditions allow us to define in a fast way if a given function is convex or not.

Examples

Here there are some examples of convex and concave functions. We start with some functions on R, with variable x.

- *Exponential.* e^{ax} is convex on $\mathcal{R}$, for any $a \in \mathcal{R}$.
- *Powers.* x^a is convex on $\mathcal{R}_{++}$ when $a \geq 1, \vee, a \leq 0$, and concave for $0 \leq a \leq 1$
- *Powers of absolute value.* $|x|^p$, for $p \geq 1$, is convex on $\mathcal{R}$
- *Logarithm.* $logx$ is concave on $\mathcal{R}_{++}$.
- *Norms.* Every norm on $\mathcal{R}^n$ is convex.
- *Max. Function.* $f(x) = \max(x_1, \cdots, x_n)$ is convex on $\mathcal{R}^n$
- *Quadratic-over-linear function.* The function $F(x, y) = \frac{x^2}{y}$, with

$$\mathbf{dom} f = \mathcal{R}x\mathcal{R}_{++} = \{(x, y) \in \mathcal{R}^2 | y > 0\},$$

  is convex.

- *Log-sum-exp.* The function $f(x) = log(e^{x_1} + \cdots + e^{x_n})$ is convex on $\mathcal{R}^n$

For a detailed review on the functions and some proofs of their convexity please refer to [5].

Jensen's Inequality and Extensions

The inequality

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y),$$

is often called *Jensen's inequality* and can be easily extended to convex combinations of more than two points: if f is convex, $x_1, \cdots, x_k \in$ **dom** f and $\theta_1, \cdots, \theta_k \geq 0$ with $\theta_1 + \cdots + \theta_k = 1$, then:

$$f(\theta_1 x_1 + \cdots + \theta_k x_k) \leq \theta_1 f(x_1) + \cdots \theta_x f(x_k).$$

It can also be extended to infinite sums, integrals, and expected values. If $p(x) \geq 0$ on $S \subseteq$ **dom**f, $\int_S p(x)dx = 1$,then.

$$f(\int_S p(x)xdx) \leq \int_S f(x)p(x)dx$$

provided that the integrals exist.

2.3 Quasiconvex functions

Definition

A function $f : \mathbf{R}^n \to \mathbf{R}$ is called *quasiconvex* (or *unimodal*) if its domain and all its sublevel sets

$$s_\alpha = \{x \in \mathbf{dom}\ f \mid f(x) \leq \alpha\},$$

for $\alpha \in \mathbf{R}$, are convex.

A function that is both *quasiconvex* and *quasiconcave* (-*quasiconvex*) is called *quasilinear*. A function is *quasilinear* if its domain, and every level set $\{x|f(x) = \alpha\}$ is convex.

Properties

Quasiconvexity is a generalization of convexity in some way. i.e A function f is *quasiconvex* if and only if $\mathbf{dom}f$ is convex and for any $x, y \in domf$ and $0 \leq \theta \leq 1$,

$$f(\theta x + (1-\theta)y) \leq max\{f(x), f(y)\},$$

which is called the *Jensen's inequality* for *quasiconvex* functions.

Quasiconvex functions on R

$f : \mathbf{R} \to \mathbf{R}$ is *quasiconvex* if and only if at least one of the following conditions holds:

- f is nondecreasing
- f is nonincreasing
- there is a point $c \in \mathbf{dom}f$ such that for $t \leq c$ (and $t \in \mathbf{dom}f$), f is nonincreasing, and for $t \geq 0$ (and $t \in \mathbf{dom}f$, f is nondecreasing.

The point c can be chosen form its critical points.

First-Order conditions

if $f : \mathbf{R}^n \to \mathbf{R}$ is differentiable. Then f is quasiconvex if and only if $\mathbf{dom}f$ is convex and for all $x, y \in \mathbf{dom}f$:

$$f(y) \leq f(x) \Rightarrow \Delta f(x)^T(y-x) \leq 0$$

Second-Order conditions

if $f : \mathbf{R}^n \to \mathbf{R}$ is twice differentiable. If f is quasiconvex then for all $x \in \mathbf{dom}f$ and all $y \in \mathbf{R}^n$, we have:

$$y^T\Delta f(x) = 0 \Rightarrow y^T\Delta^2 f(x)y \geq 0$$

which reduces to:

$$f'(x) = 0 \Rightarrow f''(x) \geq 0$$

for a quasiconvex function on $\mathbf{R}$.

For proofs please refer to [5].

2.4 Cones, primal and dual problems

Cone Definition

The *cone linear programming* in its standard canonical form:

$$\inf\{c^T x \mid Ax = b, x \in \mathcal{K}\} \tag{2.4}$$

where $x \in \mathbf{R}^n$ is the vector of decision variables, $\mathbf{k} \subseteq \mathbf{R}^n$ is a pre-specified convex cone, and $b \in \mathbf{R}^m, c \in \mathbf{R}^n, A \in \mathbf{R}^{mxn}$ are given data. Recall:

$$\mathcal{R}^n_+ = \{x \in \mathcal{R}^n : x \geq 0\}$$

is defined as a convex cone, and the dual of the cone is $\mathcal{R}^n_+$. In a similar way the set

$$\{(t; x) \in \mathbf{R}^{n+1} : t \geq |x|\}$$

is a convex cone in $\mathbf{R}^{n+1}$, called the second order cone. The dual cone is also the second order cone in $\mathbf{R}^{n+1}$. It is self dual.

Important subclasses of colinear programming are linear programming, semidefinite programming, second order cone programming, and a mixture of these.

Primal and dual problems

The primal-dual self-dual embedding technique was first proposed by Ye, Todd and Mizuno [14] for solving linear programming problems. The advantage of this methods is that allows to take any pre-described interior point as initial feasible solutions for the embedded problem, and the embedded problem is guaranteed to have an optimal solution which can be approximated by using any interior point algorithm.[2]

Consider the following conic optimization problem [8]:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K}, \end{array} \tag{2.5}$$

where $c \in \mathcal{R}^n, b \in \mathcal{R}^m, A \in \mathcal{R}^{mxn}$, and $\mathbf{K} \subseteq \mathbf{R}^n$ is a solid convex cone with its dual cone defined as

$$k^* = \{s | x^T s \geq 0 \ \forall \ x \in \mathbf{K}\}$$

[2]This will be discussed in more detail in chapter 4.

which has an associated dual problem, called

$$\begin{array}{ll} \text{minimize} & b^T y \\ \text{subject to} & A^T y + s = c \\ & s \in \mathcal{K}^*, \end{array} \tag{2.6}$$

equations (2.5) and equation (2.6) are the main goal to optimize in the so called *"the self-dual embedding technique for optimization over self-dual homogeneous cones"* which will be review with more detail in the chapter relevant to interior point polynomial methods. For now, it is enough to mention that the equations above are optimize over symmetric cones which are one step before in hierarchy than semidefinite programming, hereby the importance to review the concepts.

2.5 Semidefinite Programming

Definition

Semidefinite programming (SDP) refers to any optimization problem with any mixture of (symmetric) matrix and scalar-valued variables which has a linear objective function and any combination of linear equality or inequality constraints. Usually this constraints are nonlinear but convex, so a semidefinite programming problem can be cast as a convex optimization problems [4].

We consider the problem of minimizing a linear function of a variable $x \in R^m$ subject to a matrix inequality:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & F(x) \geq 0 \end{array} \tag{2.7}$$

where

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i$$

The objective function is the vector $c^T \in R^m$ tied to $m+1$ symmetric matrices $F_0, \cdots, F_m \in R^{n \times n}$. The inequality sign in (2.7) means that the matrices are semidefinite positive, recall:

$$(Fx, x) \geq 0 \quad \forall x \in R^n \tag{2.8}$$

where

$$(Fx, x) \equiv \text{trace}\left((Fx)^T x\right)$$

also known as the inner product.

A semidefinite program is a convex optimization problem since it is entirely convex, recall from convexity:

$$\text{if } F(y) \geq 0 \wedge F(x) \geq 0$$

$$F(\lambda x + (1-\lambda)y) = \lambda F(x) + (1-\lambda)F(y) \geq 0 \quad \forall \quad 0 \leq \lambda \leq 1$$

Figure (2.2) shows a semidefinite program set with a feasible region $\{x|F(x) \geq 0\}$ depicted by the closed region within the lines:

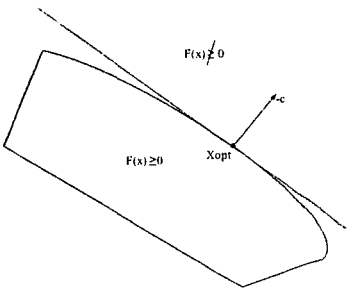


Figure 2.2: Semidefinite program example.

Colloquially speaking the semidefinite programming problem is to move as far as possible in the $-c$ direction within the feasible region. Specifically for the problem show above we see that X_{opt} is the optimal point.

Note that the optimal point is on the boundary of the feasible set, this means that the function evaluated in the point is singular. This also means that there is always an optimal point on the boundary.

In the example discussed above, we observe that the feasible set is not entirely convex, it is composed of piecewise algebraic surfaces, which is the general case. The main idea is as follows. At a point where the boundary is smooth, it is defined locally by some specific minors of the matrix $F(x)$ vanishing. Thus the boundary is locally the zero set of some polynomials in $x_1, \cdots, x_m$ [4].

From the definition of semidefinite programming given above we can infer that it can be cast also as a linear problem:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax + b \geq 0 \end{array} \tag{2.9}$$

where the inequality should be read as component wise inequality, if we express $Ax + b \geq 0$ as $F(x) = \text{diag}(Ax + b) \geq 0$, then we can cast the linear problem (2.9) as a semidefinite problem and viceversa.

We also observe that the semidefinite program can be used in such a way that many constraints can be included by using the concept of linear matrix inequality as long as the resulting block of matrices remains positive semidefinite.

Examples.

Quadratically constrained quadratic programming

A convex quadratic constraint $(Ax+b)^T(Ax+b)-c^Tx-d \leq 0$,with $x \in \mathbf{R}^k$, can be written as

$$\begin{bmatrix} I & Ax+b \\ Ax+b & c^Tx+d \end{bmatrix} \geq 0$$

we see that the left side of the equation depends linearly on the vector x and that it can be expressed as

$$F(x) = F_0 + x_1F_1 + \cdots + x_kF_k \geq 0$$

with

$$F_0 = \begin{bmatrix} I & b \\ b^T & d \end{bmatrix}, \quad F_i = \begin{bmatrix} 0 & a_i \\ a_i^T & c_i \end{bmatrix}, i = i, \cdots, k,$$

where $A = [a1, \cdots, a_k]$. Be aware that this problem can be rewritten in such a way that it turns into a semidefinite program. The standard trick here is to include a dummy variable to make the objective linear and then use **Schuur complement** to represent the matrices:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} I & A_0x+b_0 \\ (A_0x+b_0)^T & c_0^Tx+d_0+t \end{bmatrix} \geq 0 \\ & \\ & \begin{bmatrix} I & A_ix+b_i \\ (A_ix+b_i)^T & c_i^Tx+d_i+t \end{bmatrix} \geq 0, \quad i = 1, \cdots, L, \end{array}$$

Maximum eigenvalue and matrix norm minimization

If the matrix $A(x)$ depends affinely on $x \in \mathbf{R}^k : A(x) = A_0 + A_1x_1 + \cdots + A_kx_k$, where $A_i = A_i^T \in \mathbf{R}^{PxP}$, then the problem of minimizing the maximum eigenvalue of the matrix $A(x)$ can be cast as a semidefinite program:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & tI - A(x) \geq 0 \end{array}$$

Problems of this type will arise in our study and that is why it is important to mention them.

2.6 Linear Matrix Inequalities

The most common problem in *semidefinite programming* (SDP) is the minimization of a linear function subject to a linear matrix inequality, given by:

$$\begin{array}{ll} \text{minimize} & C^Tx \\ \text{subject to} & A(x) \geq 0 \end{array} \tag{2.10}$$

where $A(x) = A_0 + x_1A_1 + \cdots + x_mA_m, \quad A_i = A_i^T$.

The objective or cost function vector $C \in R^m$ and $m+1$ symmetric matrices $A_0, A_1, \cdots, A_m \in R^{n\times n}$. Where the inequality sign in $A(x) \geq 0$ means that $A(x)$ is positive semidefinite.

So, problem (2.10) is called a *semidefinite program* and $A(x) \geq 0$ *a linear matrix inequality.* Furthermore, we can handle multiple constraints in SDP by representing them as one big diagonal matrix. We can also incorporate a wide variety of convex constraints on x by representing them as LMIs [3].

In particular we can write equation (2.10) as a LMI plus some linear inequalities:

$$\begin{array}{ll} \text{Minimize} & C^Tx \\ \text{S.T.} & A(x) \geq 0 \\ & f_i^Tx \leq g_i, \quad i = 1, \cdots, p \end{array} \tag{2.11}$$

which can be rewritten as a SDP if:

$$\begin{array}{ll} \text{Minimize} & C^Tx \\ \text{S.T.} & \begin{bmatrix} A(x) & 0 \\ 0 & \text{diag}\left(g_1 - f_1^Tx, \cdots, g_p - f_p^Tx\right) \end{bmatrix} \geq 0 \end{array} \tag{2.12}$$

The matrix in (2.12)is a collection of linear an semidefinite positive constraints. We observe the importance of being able to include linear constraints (being able to impose inferior and superior boundaries on the variables).

Finally, we remark the importance of studying semidefinite programming problems because it has many known applications in engineering and also some very useful problems can be cast as semidefinite programs as well by applying some transformations into it like the linear problem, the quadratically constrained quadratic programming problem and many others. It is worth to mention also that semidefinite programs can be solved very efficiently both in theory and in practice; specially using the recently developed interior point polynomials methods which we will discuss in chapter (4) of this report.

Chapter 3

Circuit Models

3.1 Introduction

In this chapter, we give an introduction to circuit related concepts which are necessary for the modelling within our project In particular we treat RC circuits, Kirchhoff voltage and current laws, definitions for the switching and short circuit power dissipation, the interconnect model, which includes the distribution of the capacitances and conductances, a description of the possible delays to use, and we present a detailed treatment of the time dominant time constant approach.

3.2 General RC Circuits

Consider a conductance $G = \frac{1}{R}$:

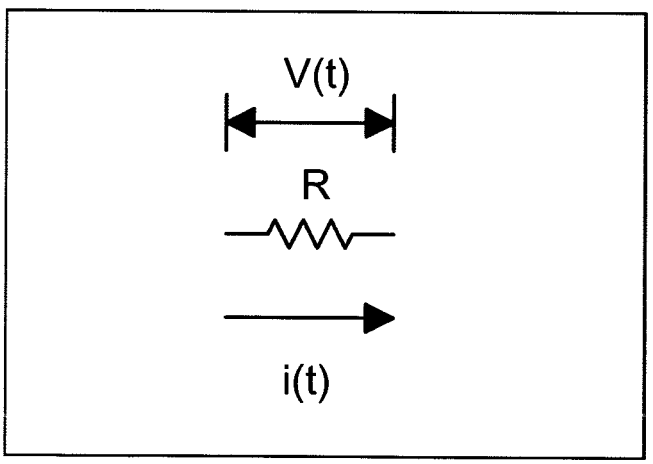


Figure 3.1: Resistor characterized by a current $i(t) = Gv(t)$.

with the voltage and current given by:

$$\begin{aligned} v(t) &= R\ i(t) \\ i(t) &= G\ v(t) \end{aligned}$$

where R is the resistance, $i(t)$ is the current and $v(t)$ is the voltage.

Now, consider a capacitance C with current given by:

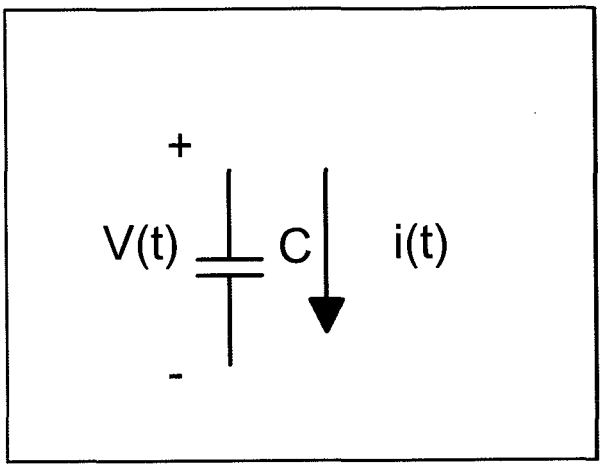


Figure 3.2: Capacitance Characterized by a current $i(t) = C\frac{dv}{dt}$.

$$i(t) = C\frac{dv}{dt}$$

where C is called the capacitance and $\frac{dv}{dt}$ is the variation of the voltage with respect to time.

Using Kirchhoff's current law[1][10] the circuit current can be described by the differential equation:

$$C\frac{dv}{dt} = -G(v(t) - u(t)), \tag{3.1}$$

where $v(t) \in \mathbf{R}^n$ is the vector of node voltages, $u(t) \in \mathbf{R}^n$ is the vector of independent voltage sources, $C \in \mathbf{R}^{n\times n}$ is the capacitance matrix, and $G \in \mathbf{R}^{n\times n}$ is the conductance matrix.

Consider a circuit with N branches and $n+1$ nodes, numbered 0 to n, where node 0 is the ground or reference node (figure(3.3)). We will assume that the capacitance and resistive networks[2] are connected.

[1]The algebraic sum of all the current leaving a gaussian surface φ at time t is equal to zero

[2]The networks obtained by removing all resistors and voltages sources or all capacitors and voltages respectively

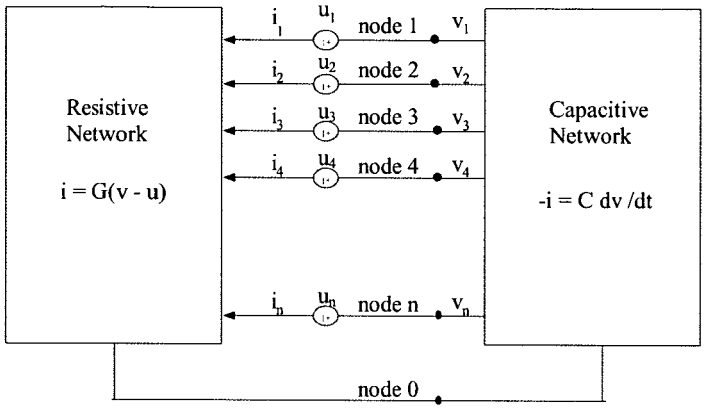


Figure 3.3: General RC circuit with $n+1$ nodes shown as a resistive network, a capacitive network, and voltage sources [3].

Each branch k consist of a capacitor $c_k \geq 0$, and a conductance $g_k \geq 0$ in series with a voltage source U_k (see figure (3.4)

For future use, we note that for this class of circuits C and G [3] have the following well-known form: for $i = 1, \cdots, n$,

$$G_{ii} = \sum_{k \in N(i)} g_k, \quad C_{ii} = \sum_{k \in N(i)} c_k,$$

where the summations extend over all branches to node i, and, for $i, j = 1, \cdots, n$, $i \neq j$,

$$G_{ij} = -\sum_{k \in N(i,j)} g_k, \quad C_{ij} = -\sum_{k \in N(i,j)} c_k,$$

where the summations extend over all branches between nodes i and j.

3.2.1 Kirchhoff's law

In lumped circuits, the voltage between any two nodes and the current flowing into any element through a node are well defined and their behavior is given by Kirchhoff's laws.

Kirchhoff's 1st Law states that the current flowing into a junction in a circuit (or node) must equal the current flowing out of the junction. This law is a direct consequence of the conservation of charge. Since no charge can be lost in the junction, any charge that flows in must ultimately flow out. Kirchhoff's 1st Law can be remembered as the rule that uses

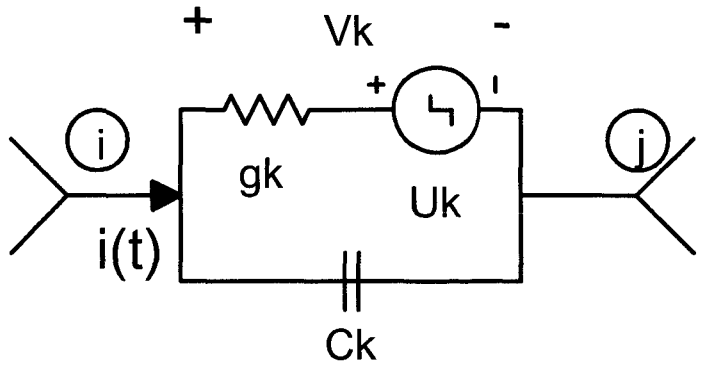


Figure 3.4: Orientation of the *kth* branch current V_k and branch current I_k in an *RC* circuit. Each branch consists of a capacitor $c_k \geq 0$, and a resistor with conductance $g_k \geq 0$ in series with an independent voltage source U_k.

nodes to study the flow of current around a circuit.

Kirchhoff's 2nd Law states that for any closed loop path around a circuit the sum of the voltage gains and voltage drops equals zero. In the circuit shown (figure(3.4)), there is a voltage gain for each electron travelling through the voltage source (symbolized by U) and a voltage drop across the resistor (iR). Applying Kirchhoff's law:

$$I_k = c_k \frac{dV_k}{dt} + g_k(V_K - U_k), \quad k = 1, \cdots, N.$$

Kirchhoff's 2nd Law is based on the principle of conservation of energy. No energy can be lost from or gained by the circuit, so the net voltage change must be 0. Kirchhoff's 2nd Law can be remembered as the rule that uses loops to study the flow of current around a circuit.

With the help of Kirchhoff's laws, specifically its 1st law, we will introduce the *reduce node incidence matrix* of a circuit $A \in \mathbf{R}^{n \times N}$ where n is the number of nodes and N is the number of branches of a digraph of a given circuit (figure(3.5)).
The reduced node incidence matrix is nothing more than the set of algebraic equations obtained from applying Kirchhoff's law

$$Ai = 0 \tag{3.2}$$

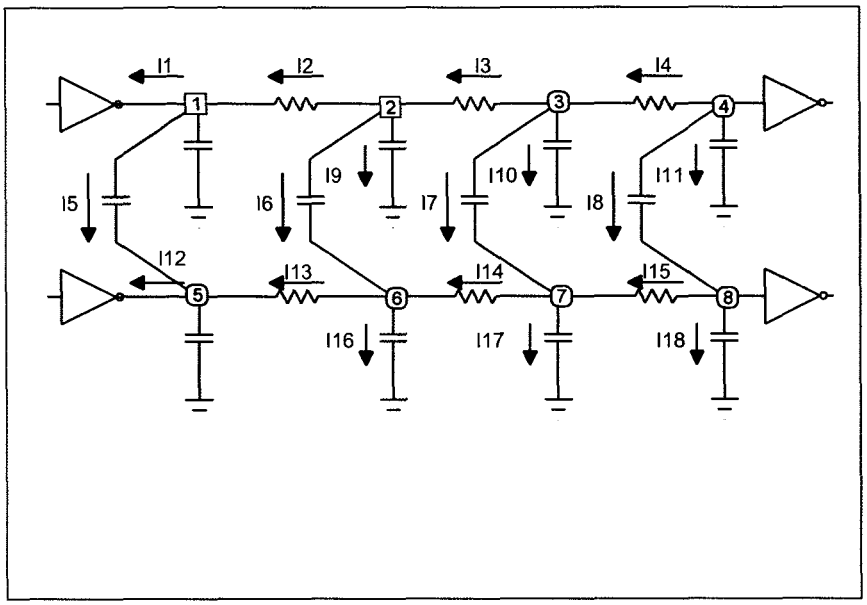


Figure 3.5: Deduction of a reduced node incidence matrix.

For figure (3.5) we see that we have $n = 4$ nodes and 16 current branches. We see that from node 1, for example we have three currents flowing: I_2 that enters to node 1 and I_1, I_5 that go out of node 1. So, because of conservation of energy we can rewrite the equation as:

$$I_1 - I_2 + I_5 = 0$$

In a similar fashion we can write all the ingoing and outgoing currents for node 2. In this case the currents I_2, I_6, I_9 go out and I_3 enters to node 2. We can repeat the same procedure until we get the whole description of the circuit:

$$\begin{array}{ccccccccccc} I_1 & - & I_2 & & & I_5 & & & & & = 0 \\ & & I_2 & -I_3 & & & I_6 & & I_9 & & = 0 \\ & & \cdots & \vdots & & \cdots & & & & \vdots & = 0 \\ & & & & & & & -I_8 & & I_{15} \quad I_{18} & = 0 \end{array}$$

This set of equations can be rewritten in matricial form as mentioned in equation (3.2) with i being the vector of the currents and A the so called reduced node incidence matrix

$A =$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	−1			1													
2		1	−1			1			1									
3			1	−1			1			1								
4				1				1			1							
5					−1							1						
6						−1							1			1		
7							−1							1			1	
8								−1							1			1

The numbers at the left and at the top have been added to assist the reader and are not elements of the matrix.

3.3 Analysis of Capacitances and Conductances

We are interested in design problems in which the capacitance matrix C and the conductance matrix G depend on some design parameters $x \in R^m$. Specifically we assume that the matrices C and G are affine functions of x.

$$\begin{aligned} C(x) &= C_0 + x_1 C_1 + \cdots + x_m C_m \\ G(x) &= G_0 + x_1 G_1 + \cdots + x_m G_m \end{aligned} \tag{3.3}$$

where C_i and G_i are symmetric.

To obtain a description of the form of equation (3.1), we use the reduced node-incidence matrix introduce before $A \in \mathbf{R}^{n \times B}$ (where, $n =$ number of nodes , and $B =$ number of Branches) and define C and G as:

$$\begin{aligned} C &= A \ \mathrm{diag}(c) \ A^T \\ G &= A \ \mathrm{diag}(g) \ A^T \end{aligned} \tag{3.4}$$

where C and G are semidefinite positive and c and g contain the description of the capacitance and conductance matrices respectively for a given circuit. The vector elements c_k and g_k denote the capacitance and the conductance of branch k, respectively, see figure(3.4). It can also be proved that they are also nonsingular if the matrix capacitive and resistive subnetwork are connected.

Recall the definition of semidefinite positive:

$$(CA, A) \geq 0 \quad \forall A \in R^n \tag{3.5}$$

Where A is the reduced-node incidence matrix described in [10].

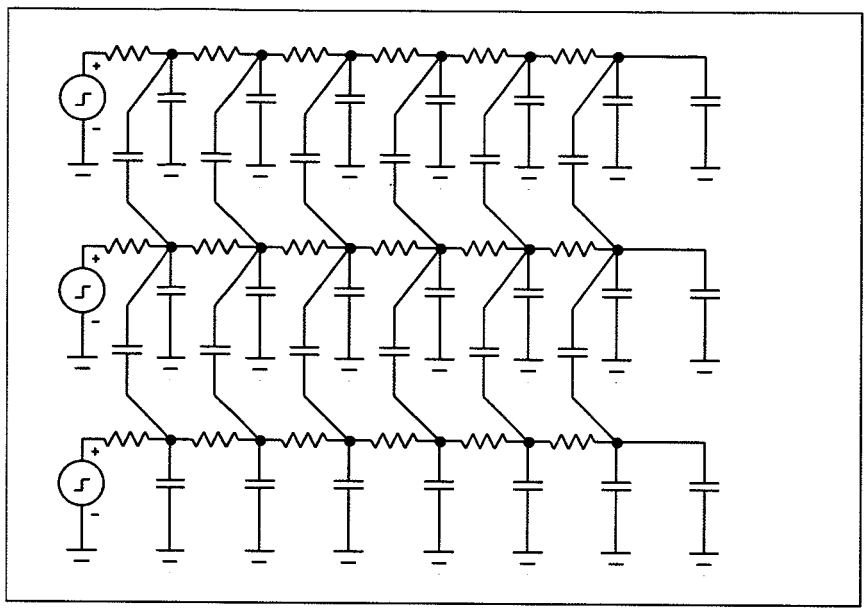


Figure 3.6: Equivalent problem.

3.4 Interconnect Modelling

Figure(3.7) shows a simplified representation of the interconnect capacitor structure. Its equivalent representation in circuits is given in figure(3.6). This representation has been used in our method. The three parallel wires have length L. The capacitance to ground of the wire at the left of figure (3.7) is given by:

$$C_1(w_1, s_1) = C_B(w_1) + C_{EB}^{\infty} + C_{EB}(s_1) \tag{3.6}$$

where C_{EB}^{∞} and $C_{EB}(s_1)$ are the values of C_{EB} at infinite spacing and at spacing s_1, respectively. For the wire segments in the middle holds,

$$C_2(w_2, s_1, s_2) = C_B(w_2) + C_{EB}(s_1) + C_{EB}(s_2) \tag{3.7}$$

The capacitance to ground for the segment at the right is given by,

$$C_3(w_3, s_2) = C_B(w_3) + C_{EB}(s_2) + C_{EB}^{\infty} \tag{3.8}$$

The coupling capacitance C_{EC} decreases with increasing spacing. The edge-to-bottom capacitance C_{EB} also has a spacing-dependent component,

$$C_{EB}(s) = C_{EB}^{\infty} + [C_{EB}(s) - C_{EB}^{\infty}] = C_{EB}^{\infty} + \Delta C_{EB}(s) \tag{3.9}$$

the sum of $C_{EC}(s)$ and $\Delta C_{EB}(s)$ is a (nearly) linear function in the reciprocal spacing $1/s$. Let's define the function

$$\Delta C_E(s) = C_{EC}(s) + \Delta C_{EB}(s) = C_{EC}(s) + C_{EB}(s) - C_{EB}^{\infty} \tag{3.10}$$

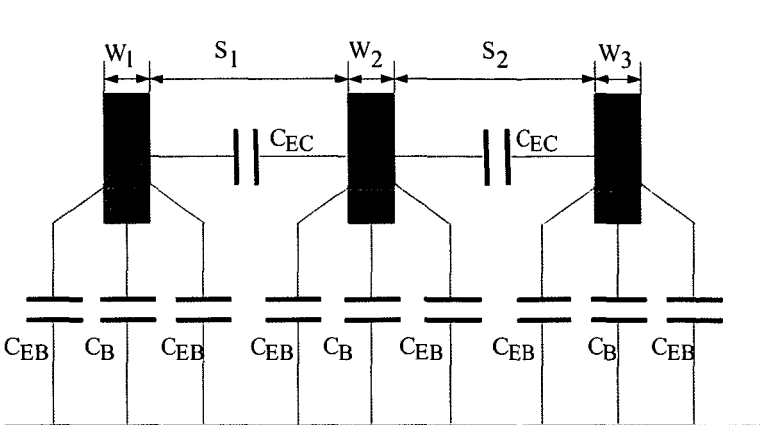


Figure 3.7: Interconnect capacitance structure analyzed.

Both the function and the coupling capacitance C_{EC} can be modelled by linear functions in $1/s$. Therefore, for the different contributions to the capacitance the following expressions are applied

$$\begin{array}{lllll} C_B & = & c_b wl & = & \beta wl, \\ C_{EB}^{\infty} & = & c_{eb}^{\infty} l & = & \varphi l \\ \Delta C_E & = & c_{\Delta e} \frac{l}{s} & = & \delta \frac{l}{s} \\ C_{EC} & = & c_{ec} \frac{l}{s} & = & \gamma \frac{l}{s} \end{array} \tag{3.11}$$

Consider a given bus composed of one or several wires and repeaters (figure(3.8)). Our desire is to simulate the behavior of the bus by using πRC elements to describe each wire. Figure(3.9) shows a typical πRC element, which is used to model the interconnect line. The πRC element is composed of two grounded capacitances and a resistance between them. The capacitances are proportional to the width of the element w, the resistance is inversely proportional to w. All πRC elements in the circuit have a fixed length l.

It has been proven for Elmore delay that an optimal approximation method for an interconnect line within an error of 3% and less is given by using 3 πRC elements or more, for a proof please refer to [27].

The driver element has been modelled by an input capacitance $c_{in}d$, one signal generator, a resistance $R_{driver} = \frac{1}{G_{driver}} = \frac{1}{gd}$ and an output capacitance $c_{out}d$, where d is the size of the repeater (figure(3.10)).

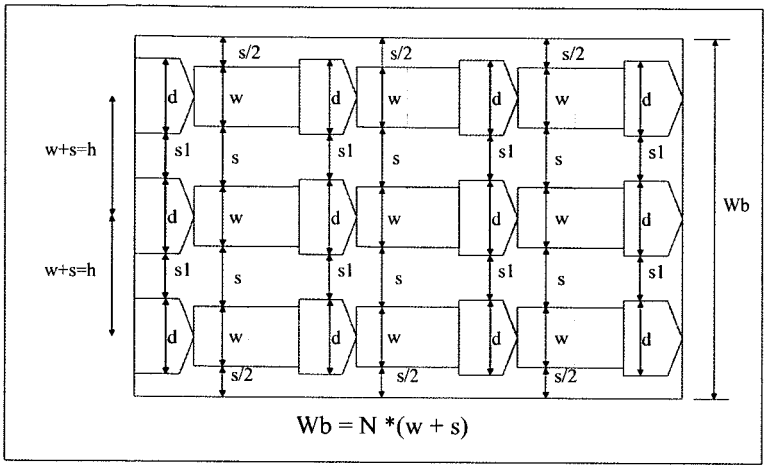


Figure 3.8: Interconnect topology.

We assume that the conductances and the ground capacitances are affine with the width of the segment w_i and the coupling capacitances are affine to the inverse of the space between the wires s_i. In particular, for the case of a πRC element of length l, without neighbors the values or r_i, c_i and g_i are given by

$$r_i = R_s \frac{l}{w_i} \tag{3.12}$$

$$c_i = (c_b w_i + c_f) l \tag{3.13}$$

$$g_i = \frac{1}{r_i} = \alpha w_i \tag{3.14}$$

$$\alpha = \frac{1}{R_s\, l} \tag{3.15}$$

where R_s, c_b and c_f denote the sheet resistance, the unit area bottom capacitance, and the unit length fringing capacitance, respectively.

3.5 Example

For a topology of 2 wires and 4 nodes with no repeaters figure(3.11), we find the corre-

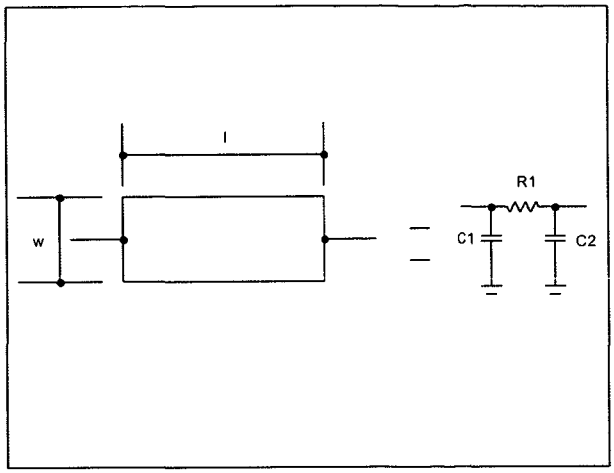


Figure 3.9: πRC element represented by two capacitances $C_1 = C_2 = \frac{1}{2}(c_b w + c_f)\, l$ and one resistance $R = \frac{1}{\alpha w}$.

sponding node incidence matrix, as discussed in section (3.2.1).

$$A = \begin{Bmatrix} 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{Bmatrix}$$

The corresponding matrices for the capacitances and the conductances are given by equation(3.3). If all segments have the same width w, the repeaters the same size d and the spacing is uniform, we have three independent variables: $x_1 = w, x_2 = t = 1/s$, and $x_3 = d$. The capacitance and conductance matrices are then given by

$$C = C_0 + C_1 w + C_2 t + C_3 d \tag{3.16}$$

$$G = G_0 + G_1 w + G_2 t + G_3 d \tag{3.17}$$

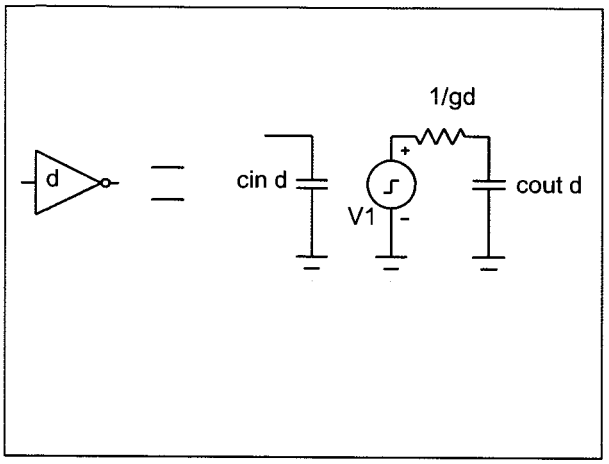


Figure 3.10: Driver modelled by an input capacitance $c_{in}d$, an output capacitance $c_{out}d$, one signal generator and a conductance $Gdriver\ R = \frac{1}{G_{driver}} = \frac{1}{gd}$.

The matrices on the right hand side of equation(3.16) are given by,

$$C_0 = \begin{Bmatrix} \varphi l & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2\varphi l & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\varphi l & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \varphi l & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \varphi l & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\varphi l & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2\varphi l & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi l \end{Bmatrix} \tag{3.18}$$

where, φ is part of the fringing capacitance (equation(3.11))

$$C_1 = \begin{Bmatrix} \beta l & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2\beta l & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\beta l & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta l & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta l & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\beta l & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2\beta l & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta l \end{Bmatrix} \tag{3.19}$$

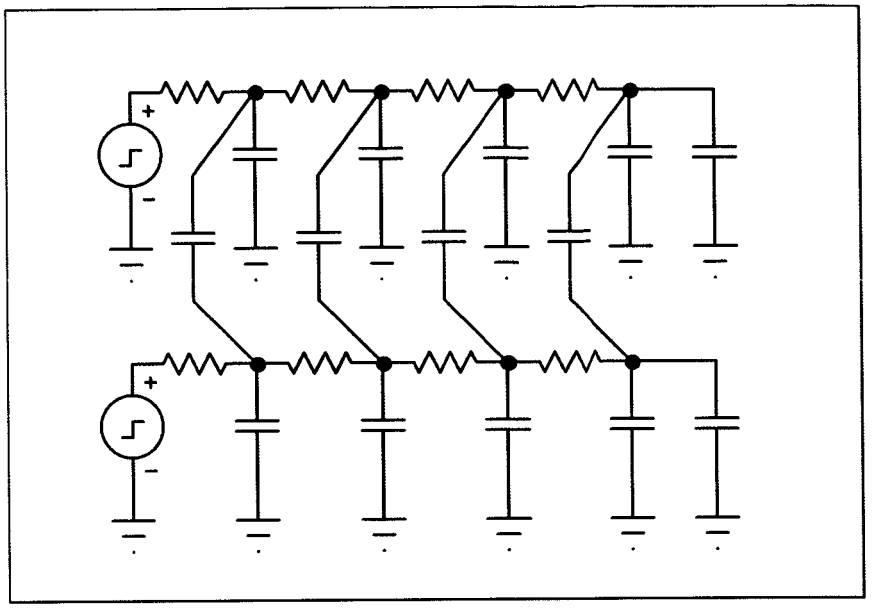


Figure 3.11: Two wires four nodes.

β is the capacitance bottom to ground,

$$C_2 = \begin{Bmatrix} (\gamma+\delta)l & 0 & 0 & 0 & -\gamma l & 0 & 0 & 0 \\ 0 & 2(\gamma+\delta)l & 0 & 0 & 0 & -2\gamma l & 0 & 0 \\ 0 & 0 & 2(\gamma+\delta)l & 0 & 0 & 0 & -2\gamma l & 0 \\ 0 & 0 & 0 & (\gamma+\delta)l & 0 & 0 & 0 & -\gamma l \\ -\gamma l & 0 & 0 & 0 & (\gamma+\delta)l & 0 & 0 & 0 \\ 0 & -2\gamma l & 0 & 0 & 0 & 2(\gamma+\delta)l & 0 & 0 \\ 0 & 0 & -2\gamma l & 0 & 0 & 0 & 2(\gamma+\delta)l & 0 \\ 0 & 0 & 0 & -\gamma l & 0 & 0 & 0 & (\gamma+\delta)l \end{Bmatrix}$$

where γ is the coupling capacitance and δ is the space dependent term of the bottom capacitance.

$$C_3 = \begin{Bmatrix} c_{out} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{in} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{out} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{in} \end{Bmatrix} \tag{3.20}$$

c_{in} is the input capacitance of the driver and c_{out} is the output capacitance, per unit length.

On the other hand, the matrices on the right-hand side of equation(3.17) are given by:

$$G_1 = \begin{Bmatrix} \alpha & -\alpha & 0 & 0 & 0 & 0 & 0 & 0 \\ -\alpha & 2\alpha & -\alpha & 0 & 0 & 0 & 0 & 0 \\ 0 & -\alpha & 2\alpha & -\alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & -\alpha & \alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha & -\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & -\alpha & 2\alpha & -\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & -\alpha & 2\alpha & -\alpha \\ 0 & 0 & 0 & 0 & 0 & 0 & -\alpha & \alpha \end{Bmatrix} \tag{3.21}$$

where α is the width dependent capacitance term, see equation(3.15)

$$G_3 = \begin{Bmatrix} g & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{Bmatrix} \tag{3.22}$$

G_0 is the zero matrix because G does not contain constant terms. G_2 is the zero matrix as the elements of the conductance matrix G are independent of the wire spacing. g is the load capacitance of the driver(equation(3.14)).

3.6 Delay

We are interested in how fast a change in the input u propagates to the different nodes of the circuit, and in how this propagation delay varies as a function of the resistances and capacitances.

Here we will introduce two typical delay measures and the dominant time constant delay.

Solving equation (3.1) for $u = 0$

$$\begin{aligned} &\tfrac{d}{dt}[e^{-C^{-1}\ G\ t}\ v(t)] = 0 \\ &e^{-C^{-1}\ G\ t}\ v(t) = C_0 \\ &v(t) = e^{-C^{-1}\ G\ t} v(0) \end{aligned}$$

where $v(t) \geq 0$ if $v(0) \geq 0$, we obtain an expression to calculate the response to a step input signal.

3.6.1 Threshold Delay

The measure of the delay at node k is the first time after which $v_k(t)$ increases above a given threshold level $\alpha > 0$, i.e.,

$$T_k^{\text{thres}} = \inf \{T | \ |v_k(t)| \leq \alpha \quad \text{for } t \geq T\}$$

and we call the maximum threshold delay to any node the *critical threshold delay* of the circuit [3]:

$$T^{\text{thres}} = \max \{T_1^{\text{thres}}, \cdots, T_n^{\text{thres}}\} = \inf \{T | \ ||v_k(t)||_\infty \leq \alpha \ \text{ for } \ t \geq T\}$$

where $||\cdot||_\infty$ denotes the infinity norm, defined by $||z||_\infty = \max_i |z_i|$. The critical threshold delay is the first time after which all node voltages are less than α.

3.6.2 Elmore Delay

The Elmore delay of a linear network is defined to be the first moment of the network impulse response [12]. The Elmore delay to node k is defined as [3]:

$$T^{\text{elm}} = \int_0^\infty v_k(t) \ dt. \tag{3.23}$$

which in the case of a ladder (figure(3.12)) it has been shown [11] that it is equal to the sum, over all the resistors, of that resistance times all its downstream capacitance.

$$T^{\text{elm}} = \int_0^\infty v_k(t) \ dt = R_1(C_1 + \cdots + C_n) + R_2(C_2 + \cdots + C_n) + \cdots + R_n C_n \tag{3.24}$$

Therefore, we can express the Elmore delay in terms of G, C and $v(0)$ as

$$T_k^{\text{elm}} = e_k^T G^{-1} C v(0)$$

where e_k is the kth unit vector. Thus the vector of Elmore delays is given by the simple expression $RCv(0)$, where $R = G^{-1}$ is the resistance matrix. We define the critical Elmore delay as the largest Elmore Delay at any node

$$T^{\text{elm}} = \max_k \ T_k^{\text{elm}}$$

For a grounded RC circuit with $v(0) \geq 0$ we can express the critical Elmore delay as

$$T^{\text{elm}} = ||G^{-1} C v(0)||_\infty$$

Note that $G^{-1}C = RC$ is element wise nonnegative.

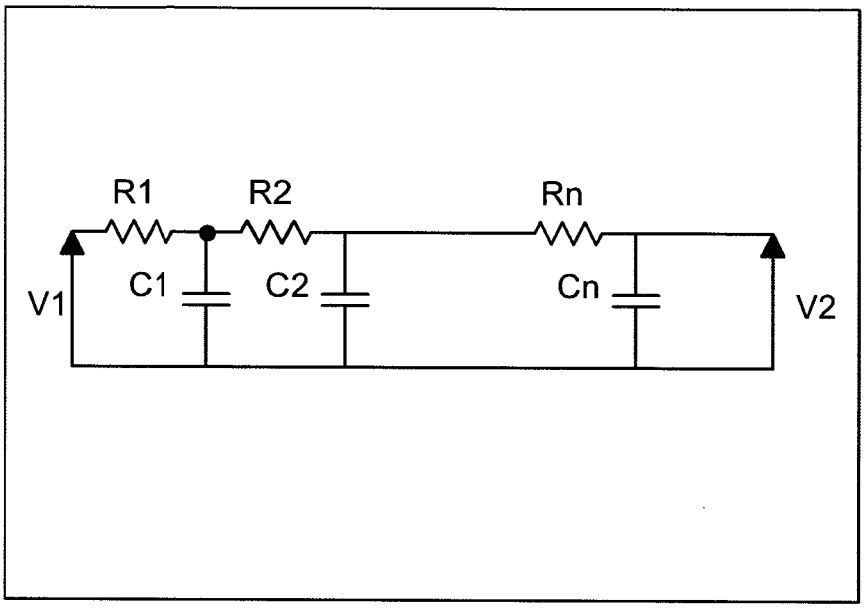


Figure 3.12: RC-Ladder.

3.6.3 Dominant time constant

In [3] it is proposed to use the *dominant time constant* of the RC circuit as a measure of the delay:

Let $\lambda_1, \cdots, \lambda_n$ denote the eigenvalues of the circuit, the roots of the characteristic polynomial $\det(sC + G)$, which are real and negative since they are also the eigenvalues of the symmetric negative definite matrix

$$C^{\frac{1}{2}}(-C^{-1}G)C^{\frac{1}{2}} = -C^{-\frac{1}{2}}GC^{-\frac{1}{2}}$$

We assume that they are ordered decreasingly as

$$0 > \lambda_1 \geq \cdots \geq \lambda_n.$$

The largest eigenvalue, λ_1 is called the *dominant eigenvalue*[3] or *dominant pole* of the RC circuit.
Each node voltage can be represented by

$$v_k(t) = \sum_{i=1}^{n} \alpha_{ik} e^{\lambda_i t}, \tag{3.25}$$

[3]The maximum eigenvalue of a symmetric matrix is convex: the function $f(X) = \lambda_{\max}$, with **dom** $f = S^m$.

$$f(X) = \sup \{y^T X y | \ ||y||_2 = 1\},$$

i.e., as the pointwise supremum of a family of linear functions of X indexed by $y \in \mathbf{R}^m$.

which is a sum of decaying exponentials with rates given by the eigenvalues.

We define the dominant time constant at the kth node as:

$$T_k^{\text{dom}} = -\frac{1}{\lambda_p}$$

where p denotes the index of the first nonzero term in (3.25).

in most cases, v_k contains a term associated with the largest eigenvalue λ_1, in which case we have:

$$T_k^{\text{dom}} = -\frac{1}{\lambda_1}.$$

The dominant time constant T_k^{dom} measures the asymptotic rate of decay of $v_k(t)$.

The dominant time constant can also be expressed in another form that will be more useful to us:

$$T^{\text{dom}} = \min\{T | TG - C \geq 0\} \tag{3.26}$$

Also, the dominant pole which is convex can be expressed as:

$$\lambda_1 = \inf \{\lambda |\ \lambda C(x) + G(x) \geq 0\},$$

T^{dom} can be seen as a quasiconvex function, because:

$$T^{\text{dom}}(x) \leq T_{\text{max}} \Longleftrightarrow T_{\text{max}} G(x) - C(x) \geq 0$$

and,

$$\{x | T^{\text{dom}}(x) \leq T^{\text{max}}\}$$

and its sublevel sets are convex sets for all T_{max}. Quasiconvexity can also be expressed as:

$$T^{\text{dom}}(\theta_x + (1-\theta)\bar{x}) \leq \max\{T^{\text{dom}}(x), T^{\text{dom}}(\bar{x})\}$$

which can be seen as a linear matrix inequality.

In the current research we will be using the time dominant constant approach as a measure of delay, because of the advantages it presents, like the ability of studying non-tree topologies and the fact that can always lead to convex problems which are solved very efficiently by using recently developed interior point polynomial methods [3].

3.7 Power

The voltage source delivers an energy of $\mathbf{v}^T C \mathbf{v}$ which is dissipated half in the resistors during a transition from initial voltage 0 to final voltage $\mathbf{v}$ and half is stored on the capacitors, which is called charging. By the reverse process (discharge) the capacitors loose the energy, which is finally dissipated through the resistors, this is called dissipation power.

Furthermore, during the switching of a static CMOS gate, a direct path from the power supply to the ground is temporarily established, resulting in short circuit power dissipation. For the short circuit power an expression for a single-transition has been obtained from [23]:

$$\begin{aligned} P_{sc} &= t_{rise} V_{dd} I_{peak} \\ &= \frac{\beta}{12}(V_{dd} - 2V_T)^3 t_{rise} \end{aligned} \tag{3.27}$$

where we assume that the inverter is symmetric: $\beta_n = \beta_p = \beta$ and $V_{Tn} = -V_{Tp} = V_T$. t_{rise} is given by [19]

$$t_{rise} = r'_d(c'_0 + c'_p) + \frac{R'_d}{d} c' l + r' l d c'_0 + \frac{1}{2} r' c' l^2$$

where

$$\begin{aligned} r'_d &= \frac{1}{g_d} \\ c'_0 &= c_{in} d \\ c'_p &= c_{out} d \\ c' &= c_b w + c_f + \gamma t \quad \text{nominal case} \\ &= c_b w + c_f + 2\gamma t \quad \text{worst case} \\ &= c_b w + c_f \quad \text{best case} \\ r' &= R_s / w \end{aligned} \tag{3.28}$$

Where c' is the sum of bottom, fringe and coupling capacitances of the wires and

$$\beta' = \frac{\beta}{l_d} d \tag{3.29}$$

where l_d is the transistor length,

$$\beta = 9.80 \times 10^{-6} + 2 \times 10^{-3} d$$

and β is approximated by a linear function that is obtained from a least-squares fit to Pstar simulation results ([24],[26]). Note that actually Equation(3.27) represents the *energy* dissipated per cycle. However, we will use the term power *dissipation* throughout the text.

We take into account that the short circuit power is not a linear function in the design parameters (d, w, t) and also that the short-circuit power is much smaller than the dissipation power, $P_{sc} \approx P_{switch}/10$ in most practical situations.

We optimize the switching power and then we add the short circuit power to the optimized result, we use the following parameter values in equation(3.30) corresponding to the CMOS12 technology [26].

$$\begin{aligned} V_{dd} &= 1.2 \ V \\ V_T &= 0.36 \ V \ (\text{ CMOS12, average for } d = 0.15 - 1\mu m) \\ d &= \text{ Width of NMOS transistor} \end{aligned} \tag{3.30}$$

The value for V_T is the average for d which lies between 0.15 and $1\mu m$, where d represents the inverter size, defined as the width of the NMOS transistor.

Chapter 4

Interior Point Polynomial Methods

4.1 Introduction

For the current research we used Jos. F. Sturm's software **SeDuMi** which stands for Self-Dual Minimization Technique [7], **SeDuMi** is based on interior point polynomial methods which also includes the so-called self-dual embedding technique developed by Ye et al. [14]. The purpose of this chapter is just to give an overview of the method, for complete details the reader is asked to go to [7].

4.2 Primal and Dual Problem

We study *cone linear programming* problems in the standard canonical form:

$$\inf \{c^T x \mid ; Ax = b, x \in \mathcal{K}\} \tag{4.1}$$

where $x \in \Re^n$ is the vector of decision variables, $\mathcal{K} \subseteq \Re^n$ is a convex cone, and $b \in \Re^n$, $c \in \Re^n$, $A \in R^{mxn}$ are given data. Cone linear programs are non-linear, since $\mathcal{K}$ need not be polyhedral.

Cone linear programming can be cast also as linear programming, semidefinite programming, second order cone programming, and a mixture of these. These subclasses arise by letting $\mathcal{K}$ in (4.1) be the nonnegative orthant ($\mathcal{K} = \Re^n_+$), the cone of positive semidefinite matrices, a Cartesian product of Lorentzs cones, or a symmetric cone, respectively.

The cone of positive semi-definite matrices should be considered as a set of vectors in $\Re^n$.

$$\mathcal{K}^s = \{vec(X) \mid X \in \Re^{vxv} \text{ is symmetric positive semi-definite } \}.$$

where $\mathcal{K}^s \subset \Re^n$ with $n = v^2$. $\mathcal{K}^s$ is actually a cone in an $\frac{(n+v)}{2}$ dimensional subspace of $\Re^n$.

The Lorentz cone in $\Re^n$ is defined as

$$\mathcal{K}^q = \left\{ x \in \Re^n \mid x_1 \geq \sqrt{\sum_{i=2}^{n} x_i^2} \right\} \tag{4.2}$$

where the superscript 'q' stands for quadratic.

Non-trivial second order programs involve multiple Lorentz cone (second order constraints). Similarly, linear programs involve non-negativity constraints. In these cases, the cone $\mathcal{K}$ is a Cartesian product of so-called primitive symmetric cones.

A *mixed semidefinite and second order cone optimization problem* can be formulated as a standard cone linear program (4.1) with the following structure:

$$\begin{array}{ll} \text{minimize} & (c^l)^T x^l + (c^q)^T x^q + (c^s)^T x^s \\ \text{such that} & A^l x^l + A^q x^q + A^s x^s = b \\ & x^l \in \Re_+^{\mathcal{K}(l)},\ x^q \in \mathcal{K}^q,\ x^s \in \mathcal{K}^s \end{array} \tag{4.3}$$

where:

- $\mathcal{K}(l) - >$ Denotes the number of nonnegative variables
- $\mathcal{K}^q = \mathcal{K}_1^q \times \cdots \times \mathcal{K}_{\mathcal{K}(q)}^q - >$ Is a Cartesian product of $\mathcal{K}(q)$ Lorentz cones
- $\mathcal{K}^s = \mathcal{K}_1^s \times \cdots \times \mathcal{K}_{\mathcal{K}(s)}^s - >$ Is a Cartesian product of $\mathcal{K}(s)$ cones of positive semidefinite matrices.
- And, $\mathcal{K} = \mathcal{K}(l) + \mathcal{K}(q) + \mathcal{K}(s)$ denote the number of primitive symmetric constraints.

Associated with (4.1) is a dual problem, viz.

$$\sup \{ b^T y | A^T y + z = c, z \in \mathcal{K}^* \} \tag{4.4}$$

where $y \in \Re^m$ and $z \in \Re^n$ are the decision variables, and

$$\mathcal{K}^* = \{ z \in \Re^n \mid z^T x \geq 0 \text{ for all } x \in \mathcal{K} \} \tag{4.5}$$

is the dual cone to $\mathcal{K}$.

For mixed semidefinite and second order cone optimization, the dual problem has the following structure [7].

$$\begin{array}{ll} \text{maximize} & b^T y \\ \text{such that} & (A^l)^T y + z^l = c^l \\ & (A^q)^T y + z^q = c^q \\ & (A^s)^T y + z^s = c^s \\ & z^L \in \Re_+^{\mathcal{K}(l)}, Z^q \in \mathcal{K}^q, z^s \in \mathcal{K}^{f*} \end{array} \tag{4.6}$$

where $(\mathcal{K}^s)^* = (\mathcal{K}_1^s)^* \times \cdots \times (\mathcal{K}_{\mathcal{K}(s)}^s)^*$. Furthermore, $(\mathcal{K}^s)^*\mathcal{K}^s$ if we restrict z in definition (4.5) to a proper lower dimensional Euclidean space to take care of symmetry.

$$\mathcal{K}_i^s = \{z = vec(Z) \mid Z = Z^T,\ z^T x \geq 0 \text{ for all } x \in \mathcal{K}_i^s\}$$

This requires that c and the rows A are also in this lower dimensional Euclidean space. If this requirement is given then $\mathcal{K}$ is self-dual.

4.3 Primal-Dual Interior Point Method

If we consider the primal-dual pair (4.3),(4.6) of mixed semidefinite and second order cone programming, then the method we enounce here [7] applies to possibly infeasible problems using the technique of self-dual embedding, as discussed later on.

A basic iteration of the interior point method consist of finding a searchg direction that is added to the current iterative solution (x, y, z) with a certain step length $t > 0$ such that:

$$(x^+, y^+, z^+) = (x, y, z) + t(\delta x, \Delta y, \Delta z). \tag{4.7}$$

where (x^+, y^+, z^+) is the next feasible solution.

The search direction $(\Delta x, \Delta y, \Delta z)$ is implicitly defined by a system of equations, as follows:

$$\begin{array}{rcrcl} \Delta x & + & \Pi\ \Delta z & = & r \\ & & A\ \Delta x & = & 0 \\ A^T\ \Delta y & + & \Delta z & = & 0 \end{array} \tag{4.8}$$

which depends on πRC, an invertible $n \times n$ block diagol matrix and a vector $r \in \Re^n$. In all usual approaches, the invertible matrix πRC is such that

$$\Pi^T z = x \tag{4.9}$$

The diagonal blocks $\Pi^s[k]$, $k = 1, 2, \cdots, k(s)$, should also map the $\mathcal{V}(K_i^s) \times (\mathcal{V}(\mathcal{K}_i^s + 1))/2$ Euclidean space of symmetric matrices onto itself, in order to maintain symmetry of the primal variables $x^s[k]$.

The algorithm below outlinbes the basic scheme of a primal-dual interior point method. The algorithm determines its step length based on a neighbourhood $\mathcal{N} \subset \Re_+^{\mathcal{V}}$.

4.3.1 Interior Point Method

Algorithm

Step 0 Initial solution $(x, y, z) \in \mathcal{K} \times \Re^m \times \mathcal{K}$ with $Ax = b$ and $A^T y + z = c$ such that $\lambda(P(x)^{1/2} z) \in \mathcal{N}$.

Step 1 If $x^T z \leq \epsilon$ then **STOP**.

Step 2 Choose Π and r according to the algorithmic settings. Compute the search direction $(\Delta x, \Delta y, \Delta z)$ from (4.8). Then determine a 'large enough' step length $t > 0$ such that $\lambda(P(x + t\Delta x)^{1/2}(z + t\Delta z)) \in \mathcal{N}$

step 3 Update

$$(x, y, z) \longleftarrow (x + t\Delta x, y + t\Delta y, z + t\Delta z)$$

and return to step 1.

With

$$\lambda(P((x)^{1/2}z) = \lambda(P(z)^{1/2}x)$$

by similarity, see Sturm [13]. The above algorithm is thus symmetric in duality.

The choice of $\mathcal{N}$ in the Algorithm above is important due that it determines the step length strategy.

Some well studied neighborhoods are the

$$\mathcal{N}_2(\beta) = \left\{ w \in \Re^{\mathcal{V}} \mid ||w - \mu e||_2 \leq \beta\mu,\ \mu = \textstyle\sum_{i=1}^{\mathcal{V}} w_i/\mathcal{V} \right\}$$

$$\mathcal{N}_\infty^-(\theta) = \left\{ w \in \Re^{\mathcal{V}} \mid w_j \geq \theta\mu \text{ for all } j = 1, \cdots, \mathcal{V},\ \mu = \textstyle\sum_{i=1}^{\mathcal{V}} w_i/\mathcal{V}, \mu \geq 0 \right\}$$

$\mathcal{N}_2-$ and the $\mathcal{N}_\infty^-$ respectively shown above.

SeDuMi, which is the software used in the present research build by J.F. Sturm uses a combination of both:

$$\mathcal{N}^{wri}(\theta, \beta) = \left\{ w \in \Re^{\mathcal{V}} \mid dist(w, \mathcal{N}_\infty^-(\theta)) \leq \beta\theta\mu\ \mu = \textstyle\sum_{i=1}^{\mathcal{V}} w_i/\mathcal{V} \right\}$$

and uses the individual benefits[7] from the both neighborhoods seen before.

The distance function has to be in accordance with the centering component of the search direction.

$$dist(w, \mathcal{N}_\infty^-(\theta) = min \left\{ \sum_{i=1}^{v} (\sqrt{w_i} - \sqrt{(\hat{w}_i)}) \mid \hat{w} \in \mathcal{N}_\infty^-(\theta) \right\}$$

Another important algorithmic choice is the definition of Π and r in (4.8), since this determines the actual search direction[1].

[1]This will be discussed in the next section

Finally, for the *large enough* expression in step 2 of the algorithm above, Sturm [7] used a bisection search to approximate the step length t^* to the boundary of $\mathcal{N}^{wr}$. Assuming that $z^T r = -x^T z$, the bisection procedure terminates with a t satisfying

$$\tfrac{t^*}{2} \leq t \leq t^*, \qquad 1 - t \leq 2(1 - t^*).$$

4.3.2 The Scaling Operator

The NT-scaling was introduced for semidefinite programming by Nesterov and Todd [15], and derived differently later by Sturm and Zhang [16]. Its generalization to the symmetric cone setting is straightforward, namely

$$\Pi = P(z)^{-1/2} P(P(z)^{1/2} x)^{1/2} P(z)^{-1/2} \tag{4.10}$$

or equivalently

$$\Pi = P(d),$$

where $d \in \mathcal{K}$ is implicitly defined by the equation $P(d)z = x$. An explicit formula for d is

$$d = P(x^{1/2})(P(x^{1/2})z)^{-1/2} \tag{4.11}$$

where we can easily verify that $\Pi^T z = x$ for the choice of Π chosen by J.F. Sturm.

An appeling property of NT scaling is its scale invariance. It holds that $\Pi\mathcal{K} = \Pi^{\mathcal{K}}\mathcal{K} = \mathcal{K}$ for any power $k \in \Re$. This means that we may locally scale the data (A, b, c) to $(A\Pi^{1/2}, b, \Pi^{1/2}c$ so that the current primal solution x and dual solution z are mapped onto the same vector $v = \Pi^{1/2}z = \Pi^{-1/2}x$. The cone $\mathcal{K}$ remains invariant under this transformation. This makes possible to extend the primal-dual from linear programming to semidefinite or symmetric cone programming.

4.3.3 Building the Normal Equations

the systemn of equations (4.8) can be reduced to

$$A\Pi A^T \Delta y = -Ar \tag{4.12}$$

by multiplying the last equation of (4.8) by $A\Pi$ yielding

$$0 = A\Pi(A^T\Delta y + \Delta z) = A\Pi A^T \Delta y + A(r - \Delta x) = A\Pi A^T \Delta y + Ar \tag{4.13}$$

Then, from the first equation of (4.8) we get the third equality, finally we use the second equation $A\Delta x = 0$ to obtain the desired reduced system (4.12). So from a system of equations we finally arrive to one equation dependent on the direction Δy. The directions for Δz and Δx can be solved by solving Δy and then using it into equations (4.8) to get the desire direction coordinate.

Now, there are several ways of solving the system (4.12) when Π is symmetric positive definite. Given a factor ϕ such that $\Pi = \phi\phi^T$, one can solve the system based on QR-factorization of the matrix $\phi^T A^T$. The QR-factorization is preferred for systems within a range of small to medium sized, however, for most practical cone linear programming problems, A is a large sparse matrix with considerable more columns than rows. For this kind of matrices we have much more efficient algorithms like the conjugate gradient method, conjugate squares and others.

The usual approach is to build the matrix $A\Pi A^T$ and compute its Cholesky factorization. In the present case we compute $A\Pi A^T$ using the NT-scaling explained in the section before, where $\Pi = P(d)^2$.

4.3.4 Solving the Normal Equations

Once we have the system $A\Pi A^T$, it remains to solve Δy from the normal equations $A\Pi A^T \Delta y = -Ar$ or more specifically $A\Pi A^T \Delta y = b$. Now, assuming that Π is semidefinite positive, as is the case with the NT scaling, then the basic approach is to compute the $L\Theta L^T$ Cholesky factorization of $A\Pi A^T$. So, we compute a lower triangular matrix L with all-1 diagonal entries $l_{ii} = 1, \quad i = 1, \cdots, m$ and a positive (semi-)definite diagonal matrix Θ such that $A\Pi A^T = L\Theta L^T$.

After the Cholesky factors L and Θ have been computed, one obtains $L^{-1}b$, $\theta^{-1}(L^{-1}b)$ and $\Delta y = L^{-T}(\Theta^{-1}L^{-1}b)$ by a forward solve, element-wise division, and backward solve, respectively.

The forward and backward solve procedures may lead to a serious loss of numerical accuracy if $max_{i,j} \mid l_{ij} \mid$ is large, say more than 1000. Causes of this happening (L has large entries) is *poor scaling of the problem*, and, ill-conditioning in $A\Pi A^T$ when the optimum is approached.

In principle, equation (4.8) can be solved using the pre-conditioned conjugate gradient (P-CG) method in at most $1 + 2 \times nskip$ iterations in the case of skipping pivots, or $1 + nadd$ iterations in the case of adding quantities on the diagonal, where *nskip* is the number of skipped pivots and *nadd* denotes the number of additions to diagonal entries. In any case it is not recommended to proceed the conjugate gradient process for more than say 25 iterations.

[2]For a description on exploiting sparsity for the actual problem the reader is referred to section 6.1 from [7]

4.3.5 Initialization, Infeasibility and Embedding

The interior point method is initialized from a *cold start*, which means that no initial starting point is known. The initial point is a random vector within the interior cone $\mathcal{K}$, which may or may not satisfy the linear feasibility constraints. The interior point method should generate either an approximate solution pair, or an approximate *Farkas-type* dual solution to certify that no feasible solution pair exists.

Recall:

$$\begin{array}{lcl} b - Ax & = & 0 \\ A^T y + z - c & = & 0 \\ C^T x - b^T y & \leq & 0 \end{array} \tag{4.14}$$

and,

$$x \in \mathcal{K}, y \in \Re^m, z \in \mathcal{K}^* \tag{4.15}$$

are the optimality conditions for (4.1) and (4.4). On the other hand, the Farkas-type conditions to certify that there cannot exist (x, y, z) satisfying the equations just mentioned above are :

$$\begin{array}{lcl} Ax & = & 0 \\ A^T y + z - c & = & 0 \\ C^T x - b^T y + 1 & = & 0 \end{array} \tag{4.16}$$

together with

$$x \in \mathcal{K}, y \in \Re^m, z \in \mathcal{K}^*$$

for recent surveys on conic duality.

The *cold started* interior point method is initialized from a triple $(x^{(0)}, y^{(0)}, z^{(0)})$ satisfying $\lambda(P(x^{(0)})^{1/2} z^{(0)}) \in \mathcal{N}$. One also defines

$$y_0^{(0)} = \frac{\mathcal{V}(\mathcal{K}) + 1}{\mathcal{V}(\mathcal{K})};$$

and then for a given $(x^{(0)}, y_0^{(0)}, y^{(0)}, z^{(0)})$, the initial primal and dual residuals are defined as

$$r_p = \frac{1}{y_0^{(0)}}(b - Ax^{(0)}), \qquad r_d = \frac{1}{y_0^{(0)}}(A^T y^{(0)} + z^{(0)} - c)$$

after this initialization the self-dual embedding approach is used.

4.3.6 Self-Dual Embedding Technique

In the self-dual technique of ye et al [14], a slack variable z_0 is added to (3.16), and initialized at

$$z_0^{(0)} = \frac{(x^{(0)})^T z^{(0)}}{\sqsubseteq(\mathcal{K})} \tag{4.17}$$

where

$$r_g = \frac{c^T x^{(0)} - b^T y^{(0)} + z_0^{(0)}}{y_0^{(0)}} \qquad x_0^{(0)} = 1$$

so finally the primal and dual problems (4.1) and (4.4) are embedded into a self-dual optimization problem

$$min\,\{y_0 \mid (x_0, x, y_0, y, z_0, z)\} \tag{4.18}$$

which satisfy

$$\begin{bmatrix} 0 & -A & b \\ A^T & 0 & -c \\ -b^T & c^T & 0 \end{bmatrix} \begin{bmatrix} y \\ x \\ x_0 \end{bmatrix} + \begin{bmatrix} 0 \\ z \\ z_0 \end{bmatrix} = yo \begin{bmatrix} r_p \\ r_d \\ r_g \end{bmatrix} \tag{4.19}$$

$$r_p^T y + r_d^T x + r_g x_0 = 1, \tag{4.20}$$

$$(x_0, x) \in \Re_+ \times \mathcal{K}, (y_0, y) \in \Re^{1+m}, (z_0, z) \in \Re_+ \times \mathcal{K}^* \tag{4.21}$$

now, multiplying both sides of (4.19) with $[y^T, x^T, x_0]$ yields

$$x^T z + x_0 z_0 = y_0(r_p^T + r_d^T x + r_g x_0) = y_0 \tag{4.22}$$

now, $(x_0^{(0)}, x^{(0)}, y_0^{(0)}, z_0^{(0)}, z^(0)$ satisfies (4.19) and (4.20) and (4.21) therefore it can be used to solve (4.18) using a feasible interior point method.

Given an interior feasible solution to (4.19)-(4.21), we get the normalized solution

$$(\hat{x}, \hat{y}, \hat{z}) = \frac{(x, y, z)}{x_0}, \tag{4.23}$$

as an approximate solution to (4.14)-(4.15). In particular we have

$$\begin{cases} b - A\hat{x} & = & (y_0/x_0)r_b \\ A^T\hat{y} + \hat{z} - c & = & (y_0/x_0)r_c \\ c^T\hat{x} - b^T\hat{y} & < & (y_0/x_0)r_g \end{cases}$$

When y_0/x_0 approaches zero, the residual to (4.14) approaches zero as well.

However, this is not always possible, since the original problem pair can be infeasible. Therefore, a normalized solution for the problem (4.16) must be found

$$(\bar{x}, \bar{y}, \bar{z}) = \frac{(x, y, z)}{z_0} \tag{4.24}$$

as an approximate solution to (4.16) and (4.15). In particular

$$\begin{cases} -A\bar{x} & = & (y_0 r_b - x_0 b)/z_0 \\ A^T\bar{y} + \bar{z} & = & (y_0 r_c + x_0 c)/Z_0 \\ c^T\bar{x} - b^T\bar{y} + 1 & < & y_0 r_g/z_0 \end{cases}$$

and if at the final iteration the residual of $(\bar{x}, \bar{y}, \bar{z})$ is smaller than the residual of $(\hat{x}, \hat{y}, \hat{z})$, the problem is report as infeasible, providing $\bar{x}$ and $\bar{y}$ as a certificate.

During the interior point process, it can be predicted whether the original problem pair is infeasible based on the (x_0, z_0) component of the first order predictor direction.

$$feas = \frac{\dot{x}_0^{(0)}}{x_0} - \frac{\dot{z}_0^{(0)}}{z_0}$$

it can be shown that if a complementary solution exists then $\dot{x}_0^{(0))}\ x_0 \longrightarrow 0$ and $\dot{z}_0^{(0))}\ z_0 \longrightarrow -1$, so $feas \longrightarrow 1$. Conversely, if the problem is strictly infeasible one can show that $feas \longrightarrow -1$.

This technique is implemented in **SeDuMi**, the software used in the current research, of course there are much other packages available but this one was chosen because of its simplicity[3].

Finally, it is to mention the this chapter as well as the software were taken from J.F.Sturm [7] which was the result of his PhD. studies at the University of Tilburg.

[3]you do not need to defined a dual problem due to the self-dual embedding technique or to provide any feasible starting point due to the cold start procedure

Chapter 5

Interpolation Methods

5.1 Introduction

Interpolations methods are an important part of this research because they provide a valuable tool for the designer as they help finding new data points based on the given optimization results that will show the optimal trade-off curve between the design parameters for some given data without running the simulations again. Here, we will briefly show the basics of some interpolations methods used in the research and the advantage and accuracy between them.

5.2 The Interpolating Polynomial

Given n points in the plane, $(x_k, y_k), k = 1, \cdots, n$, with distinct x_k, there is a unique polynomial in x of degree less than n such that its graph passes through all its points. This polynomial is called the interpolating polynomial because it exactly reproduces the given data.

$$P(x_k) = y_k, \quad k = 1, \cdots, n$$

The most compact representation of the interpolating polynomial is the lagrange form.

$$P(x) = \sum_k \left(\Pi_{j \neq k} \frac{x - x_j}{x_k - x_j} y_k \right) \tag{5.1}$$

where there are n terms in the sum and $n - 1$ terms in each product, so this expression defines a polynomial of degree at most $n-1$. If $P(x)$ is evaluated at $x = x_k$, all the products except the k_th are zero. Furthermore, the k_th product is equal to one, so the sum is equal to y_k and the interpolation conditions are satisfied [32].

When the polynomial is written with respect to a sole variable x, i.e.

$$x^3 + x^2 + 1$$

then, the polynomial is called monomial and it is said to be using the power form. The coefficients of an interpolating polynomial using its power form,

$$P(x) = c_1 x^{n-1} + c_2 x^{n-2} + \cdots + c_{n-1} x + c_n$$

can be computed by solving a system of simultaneous linear equations

$$\begin{pmatrix} x_1^{n-1} & x_1^{n-2} & \cdots & x_1 & 1 \\ x_2^{n-1} & x_2^{n-2} & \cdots & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & 1 \\ x_n^{n-1} & x_n^{n-2} & \cdots & x_n & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

5.3 Least-Squares method

If we have a set of data equal to the order of a polynomial plus one, we can fit exactly the given data with a polynomial of degree less than the data set. However, if we have a very large amount of data we can improve the precision of the fitted curve by using the method of least squares.

We suppose the simplest case in which we would like to find a linear regression to fit the data set, say (x_i, y_i, z_i), $i = 1, \cdots, N$ of some design parameters, where $z = f(x, y)$;

The input data can be viewed as a linear system of the form

$$\begin{aligned} a + bx_1 + cy_1 &= z_1 \\ a + bx_2 + cy_2 &= z_2 \\ \vdots \qquad\qquad\quad &\vdots \quad \vdots \\ a + bx_N + cy_N &= z_N \end{aligned} \tag{5.2}$$

which can be rewritten as:

$$\begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_N & y_N \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{pmatrix}. \tag{5.3}$$

The least-squares method is based on the deviation error of the real data and the given linear regression, say:

$$r_i = \sum_{i=1}^{N} [y_i - f(x, y)] \quad i = 1, \cdots, N$$

now, if we take the square of this deviation, say:

$$R = \sum_{i=1}^{N} (y_i - f(x, y))^2 \quad i = 1, \cdots, N$$

due that a, b, c are arbitrary parameters, we have to determine them in such a way that they minimize R. This can be achieved by taking partial derivatives of R and making them equal to zero.

$$\begin{aligned} \frac{\partial R}{\partial a} &= 2\sum_{i=1}^{N}(z_i - a - bx_i - cy_i) &&= 0 \\ \frac{\partial R}{\partial b} &= 2\sum_{i=1}^{N}(z_i x_i - ax_i - bx_i^2 - cy_i x_i) &&= 0 \\ \frac{\partial R}{\partial c} &= 2\sum_{i=1}^{N}(z_i y_i - ay_i - bx_i y_i - cy_i^2) &&= 0 \end{aligned} \tag{5.4}$$

which can be rewritten as:

$$\begin{bmatrix} N & \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i y_i \\ \sum_{i=1}^{N} y_i & \sum_{i=1}^{N} x_i y_i & \sum_{i=1}^{N} y_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} z_i \\ z_i x_i \\ z_i y_i \end{bmatrix} \tag{5.5}$$

Now, we only have to solve the given system by an appropriate method and we will find the given linear regression

$$a + b_i x_i + c_i y_i = z_i$$

Following the methodology described above, we could fit the data not only to a linear regression but to a given polynomial of order N, say for example

$$g(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_N x^N$$

again, the deviation of the given curve is

$$r_i = z_i - g(x_i), \quad i = 1, \cdots, N$$

and the total square of the deviations is

$$R = \sum_{i=1}^{N}(z_i - g(x_i))^2,$$

now, as before we make the derivatives equal to zero to find the explicit form [20].

$$\begin{bmatrix} N & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^N \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^N \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \sum x_i^N & \sum x_i^{N+1} & \sum x_i^{N+2} & \cdots & \sum x_i^{2N} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} \sum z_i \\ \sum x i z_i \\ \vdots \\ \sum x_i^N y_i \end{bmatrix}$$

Again the coefficients can be found out by solving the linear system by an appropriate method, i.e Gauss Jordan Method.

5.4 Cubic Splines

Really often we found the need to fit a large number of data to a unique smooth curve, but Newton's or Lagrange's interpolation with high order polynomials are not suitable for the job, due that the error of a unique polynomial tend to grow in a drastic manner when the order is high. Cubic splines on the other hand is designed for this purpose[20].

In this interpolation, we use a cubic polynomial between each interval of two consecutive points. A cubic polynomial has four coefficients, therefore it requires four conditions. Two of them come from the fact that the function must go through the points on the boundaries. The other two, from the fact that the function must have the condition that its first and second derivative are continue in each of the given points.

Cubic Splines are piecewise cubic polynomials, that are analogous in a certain to Hermite polynomials.

Recall again the data set (5.3) which we will interpolate (every point) with a third degree parametric polynomial of the type

$$g(s) = a + bs + cs^2 + es^3 \tag{5.6}$$

with $x_i \leq x \leq x_{i+1}$ and the parameter $0 \leq s \leq h_i$ and the interval defined as $h_i = x_{i+1} - xi$.

For this we take into account that in order to find the values for a, b, c, e we need to have four (4) initial conditions at the points:

$$s = 0, \quad \text{and} \quad s = h_i \tag{5.7}$$

so,

$$\begin{aligned} f_i &= a \\ f_{i+1} &= a + bh_i + ch_i^2 + eh_i^3 \end{aligned} \tag{5.8}$$

now, to find the other two conditions we find the second derivative of the parametric equation (5.6)

$$\begin{aligned} g'(s) &= b + 2cs + 3es^2 \\ g''(s) &= 2c + 6es \end{aligned} \tag{5.9}$$

and evaluating at $s = 0$ and $s = h_i$ we get the other two boundary points needed to evaluate the polynomial.

$$\begin{aligned} c &= \frac{g_i''}{2} i \\ e &= \frac{g_{i+1}'' - g_i''}{6h_i} \end{aligned} \tag{5.10}$$

allowing us to find also an expression for the constant b, say:

$$b = \frac{f_{i+1} - f_i}{h_i} - \frac{(g_{i+1}'' + 2g_i'')}{6} h_i \tag{5.11}$$

and from this we can rewrite the cubic polynomial like:

$$g(s) = f_i + [\frac{f_{i+1} - f_i}{h_i} - \frac{g''_{i+1} + 2g''_i}{6}h_i]s + \frac{g''_i}{2}s^2 + \frac{g''_{i+1} - g''_i}{6h_i}s^3$$

5.5 Hermite interpolation using harmonic averages

In a recent book by Cleve Moler [32], it is stated that the monotonic behavior of interpolating functions can be guaranteed when Hermite interpolation is used with harmonic averages for the derivatives. Unfortunately, the book does not contain a proof of this statement and email contact with Moler revealed that he does not know where it is proved, or whether it has been proved. In this apart, we present a proof of the statement[1].

Proof. Let the interval under consideration be subdivided using equally spaced nodes $x_0, x_1, \cdots, x_N$. Using Hermite interpolation with cubic polynomials, we find the following interpolation polynomial on the interval $[x_k, x_{k+1}]$

$$p(x) = \phi(\frac{x - x_k}{h})f_{k+1} + \phi(\frac{x_{k+1} - x}{h})f_k + h\psi(\frac{x - x_k}{h})f'_{k+1} - h\psi(\frac{x_{k+1} - x}{h})f'_k \qquad (5.12)$$

where h is the mesh spacing, f_k and f_{k+1} are the given function values at x_k and x_{k+1}, respectively, and f'_k, f'_{k+1} are approximations to the derivatives at these nodes. The functions ϕ and ψ are given by:

$$\begin{aligned} \phi(t) &= 3t^2 - 2t^3 \\ \psi(t) &= t^3 - t^2 \end{aligned}$$

If the sequence f_{k-1}, f_k, f_{k+1} is not monotonic, the algorithm requires that we define $f'_k = 0$. Similarly, if the sequence f_k, f_{k+1}, f_{k+2} is not monotonic, we must define $f'_{k+1} = 0$ in order to guarantee monotonicity of the interpolant. Of course, it is possible that only one of the two sequences is not monotonic. In that case, harmonic averaging must be used for the other sequence. This case will be consider later, first we will assume that the entire sequence $f_{k-1}, f_k, f_{k+1}, f_{k+2}$ is monotonic, and without loss of generality we will assume that the sequence is monotonically increasing. In this case, derivatives are given by harmonic averages, so we have:

$$\begin{aligned} f'_k &= \frac{1}{\frac{1}{2}(\frac{h}{f_{k+1}-f_k} + \frac{h}{f_k - f_{k-1}})} \\ f'_{k+1} &= \frac{1}{\frac{1}{2}(\frac{h}{f_{k+1}-f_k} + \frac{h}{f_{k+2} - f_{k+1}})} \end{aligned} \qquad (5.13)$$

Clearly, we have the following inequalities:

$$\begin{aligned} 0 < f'_k &< \frac{2(f_{k+1}) - f_k}{h} \\ 0 < f'_{k+1} &< \frac{2(f_{k+1}) - f_k}{h} \end{aligned} \qquad (5.14)$$

[1] Proof reproduced by request of W. Schilders

Writing

$$x = x_k + \theta h,$$

the derivative of the interpolating polynomial is given by:

$$p'(x) = 6\theta(1-\theta)\frac{f_{k+1}-f_k}{h} + \theta(3\theta-2)f'_{k+1} + (1-\theta)(1-3\theta)f'_k$$

Case 1: $0 \le \theta \le \frac{1}{3}$
In this case, $(1-\theta)(1-3\theta) \ge 0$ and $\theta(3\theta-2) \le 0$. hence, making use of (5.14)

$$p'(x) \ge 6\theta(1-\theta)\frac{f_{k+1}-f_k}{h} + \theta(3\theta-2)\frac{2(f_{k+1}-f_k)}{h} = 2\theta\frac{f_{k+1}-f_k}{h} \ge 0$$

Case 2: $\frac{1}{3} \le \theta \le \frac{2}{3}$
Then $(1-\theta)(1-3\theta) \le 0$ and $\theta(3\theta-2) \le 0$. So that

$$\begin{aligned} p'(x) &\ge 6\theta(1-\theta)\tfrac{f_{k+1}-f_k}{h} + \theta(3\theta-2)\tfrac{2(f_{k+1}-f_k)}{h} + (1-\theta)(1-3\theta)\tfrac{2(f_{k+1}-f_k)}{h} \\ &= (6\theta^2-6\theta+2)\tfrac{f_{k+1}-f_k}{h} \ge \tfrac{1}{2}\tfrac{f_{k+1}-f_k}{h} \ge 0 \end{aligned}$$

Case 3: $\frac{2}{3} \le \theta \le 1$
In this case, $(1-\theta)(1-3\theta) \le 0$ and $\theta(3\theta-2) \ge 0$. Thus

$$p'(x) \ge 6\theta(1-\theta)\frac{f_{k+1}-f_k}{h} + (1-\theta)(1-3\theta)\frac{2(f_{k+1}-f_k)}{h} = 2(1-\theta)\frac{f_{k+1}-f_k}{h} \ge 0$$

The conclusion is that

$$p'(x) \ge 0 \quad \forall\theta \in [0,1]$$

We now consider the remaining case where the sequence $f_{k-1}, f_k, f_{k+1}, f_{k+2}$ is not monotonic. If neither of the subsequences f_{k-1}, f_k, f_{k+1} and f_k, f_{k+1}, f_{k+2} is monotonic, then the algorithm must define $f'_k = f'_{k+1} = 0$, in which case the derivative of the interpolating function is given by

$$p'(x) = 6\theta(1-\theta)\frac{f_{k+1}-f_k}{h},$$

which is of constant sign.
If the sequence f_{k-1}, f_k, f_{k+1} is not monotonic, but the sequence f_k, f_{k+1}, f_{k+2} is monotonic (we shall assume: increasing), then we define $f'_k = 0$, so that

$$p'(x) = 6\theta(1-\theta)\frac{f_{k+1}-f_k}{h} + \theta(3\theta-2)f'_{k+1}$$

Case a: $0 \le \theta \le \frac{2}{3}$
In this case, $\theta(3\theta-2) \le 0$, whence

$$p'(x) \ge 6\theta(1-\theta)\frac{f_{k+1}-f_k}{h} + \theta(3\theta-2)\frac{2(f_{k+1}-f_k)}{h} = 2\theta\frac{f_{k+1}-f_k}{h} \ge 0$$

Case b: $\frac{2}{3} \leq \theta \leq 1$
Then $\theta(3\theta - 2) \geq 0$, so that

$$p'(x) \geq 6\theta(1-\theta)\frac{f_{k+1} - f_k}{h} \geq 0$$

In the above, all possible cases have been considered, and whence we conclude that $p'(x) \geq 0$ on the entire interval $[x_k, x_{k+1}]$. This concludes the proof of the monotonicity of the Hermite interpolant with harmonic averages for the derivative values.

□

Chapter 6

Wire Sizing, area and power optimization

6.1 Introduction

We collect the results of our simulations in this chapter. We start by discussing the circuit topology. Then we present optimizations on area and power dissipation, including two tests with benchmark problems from [3]. Our main contribution is the optimization of the power dissipation with real life technology parameters. Furthermore, we present Pstar simulations to show the accuracy of our results and we finalize the chapter with some theory and simulation results on optimal repeater insertion.

6.2 Uniform Wire Sizing

We define Uniform Wire Sizing (UWS) as wire sizing optimization without tapering (all the wire segments have the same constant width). We observe from figure(6.1) that UWS has uniform width w for the wires and also uniform repeater size d. Non uniform wire sizing or wire tapering (shortly abbreviated by NUWS) is represented in figure(6.2), where clearly the width of the element varies for the total length of the wire and the size of the repeater can be equal or different, in figure(6.2) they are the same.

In this section, we describe briefly why uniform wire sizing is our first study case and then we use it to optimize area, power dissipation and delay.

Alpert [22] has shown that the improvement in delay from wire tapering (NUWSR) compared with wire sizing (UWS) is at most 4% when full buffer insertion is applied for the Elmore delay model. Based on this fact, we choose uniform wire sizing under the dominant time constant model which is based on the main characteristics of the capacitances and conductances matrices of the circuit [3], as discussed in Chapter 3.

The basic idea behind the optimization is to split the wire into $k+1$ sections, each section

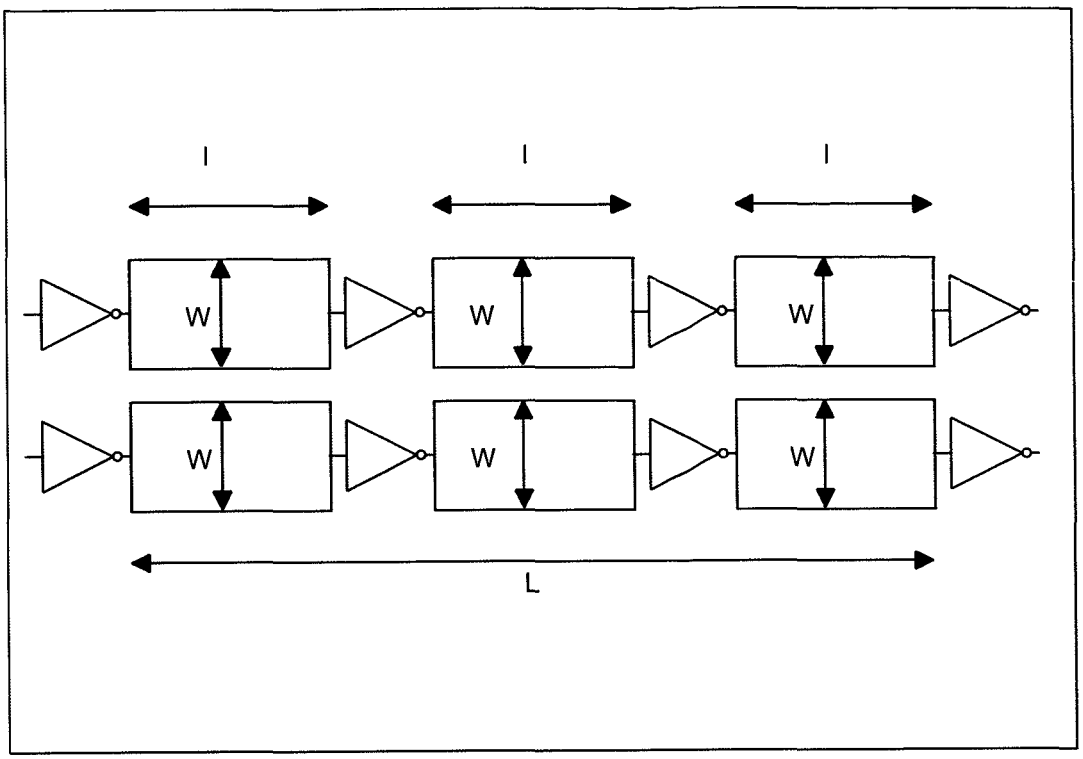


Figure 6.1: Uniform Wire Sizing approach.

into M πRC elements, where k is an even number of repeaters and n is the number of nodes into each one of the sections and $M = n - 1$. Note that the dominant time constant is a quasiconvex function of the design parameters[1] width, space between the wires and repeater sizes.

Each one of the mentioned sections has a correspondent dominant time constant T_i, $i = 1, \cdots, k+1$. Remember that the time dominant constant is a quasiconvex function of the parameters and note that the sum of $k+1$ quasiconvex functions is not a quasiconvex function. $T_1^{\text{dom}} + T_2^{\text{dom}} + \cdots + T_{k+1}^{\text{dom}} \leq T_{\max}$, so to use the dominant time constant in our optimization problem we need a reformulation that meets convexity, for this, we reformulate the problem in a convex way requiring that each $T_i \leq T_{\max}/(k+1)$, $i = 1, \cdots, k+1$, like:

$$\begin{array}{lcl} T_1^{\text{dom}} & \leq & T_{\max}/(k+1), \\ T_2^{\text{dom}} & \leq & T_{\max}/(k+1), \\ \vdots & \vdots & \vdots \\ T_{k+1}^{\text{dom}} & \leq & T_{\max}/(k+1). \end{array} \tag{6.1}$$

[1]See chapters 2&3 for a treatment on quasiconvex functions and the dominant time constant respectively

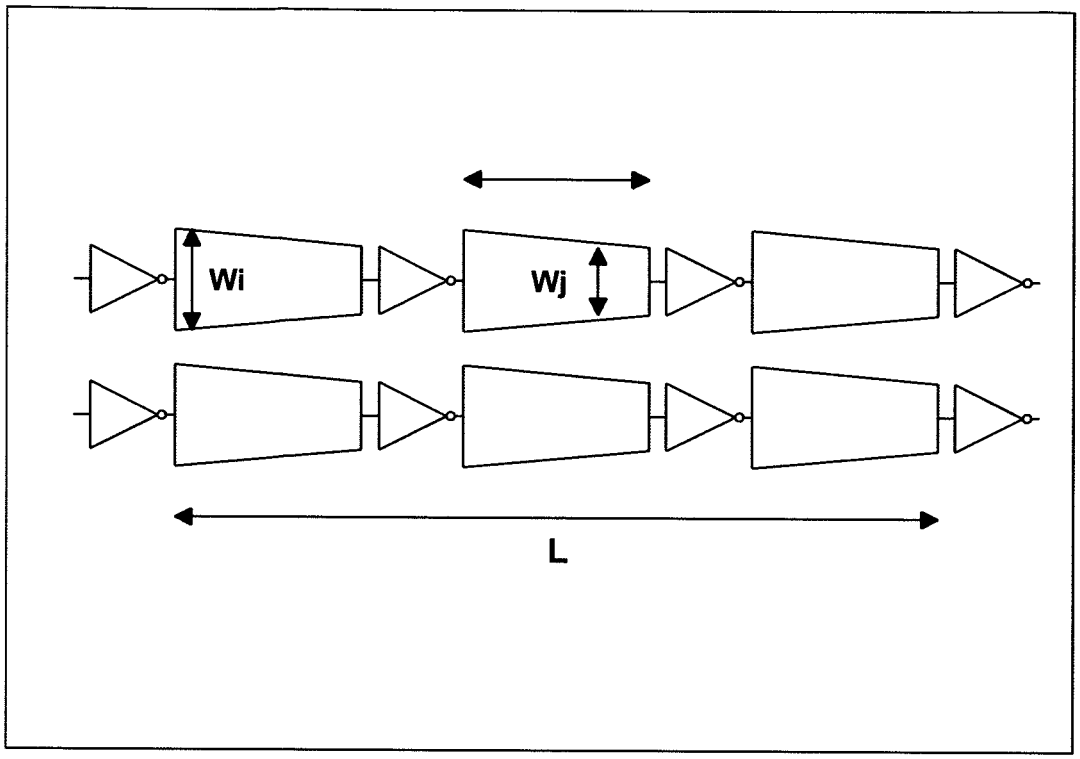


Figure 6.2: Non Uniform Wire Sizing approach.

Equation(6.1) allows to express the constraints on the dominant time constant in our optimization as a convex function.

6.3 Area Optimization(UWS)

6.3.1 Problem Definition

We want to optimize the area of a bus consisting of N wires, $k+1$ sections per wire, n nodes per section, $M = (n-1)\ \pi RC$ elements per section. We assume uniform width w for all the wires, uniform space s between the wires, same size d for all repeaters in the system and uniform repeater insertion, using the dominant time constant approach (see figure(6.3)). Figure(6.4) shows an example of the initial topology to investigate.

This specific problem can be cast as an area optimization problem subject to a constraint on the dominant time constant[2] and some other linear and semidefinite constraints:

- The size d of the repeaters has a minimum and maximum value

[2]which is a quasiconvex function of the design parameters[3]

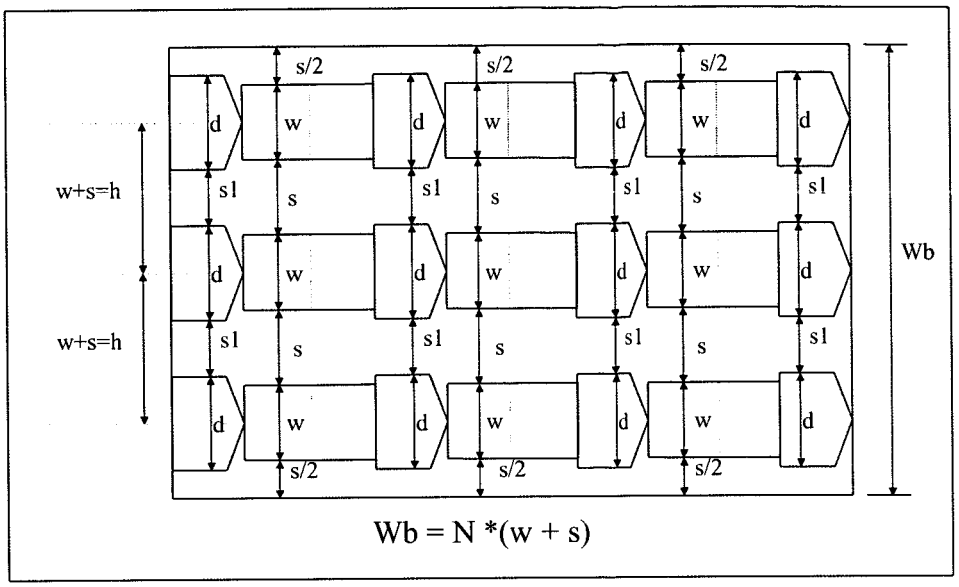


Figure 6.3: Topology of the problem to investigate.

- The space s between the wires has a minimum and maximum value
- The width w of each πRC element has a minimum an maximum value
- The bus width W_B has a length minimum and maximum value

Which leads to the following formulation

$$\begin{array}{ll} \text{minimize} & N\ k[\ h\ d + M\ l\ (\ w\ +\ s)] \\ \text{subject to} & (T_{max}/k + 1)G(w,d) - C(w,s,d) \geq 0 \\ & s \geq s_{min} \\ & s_{max} \geq s \\ & w \geq w_{min} \\ & w_{max} \geq w \\ & d_{max} \geq d \\ & d \geq d_{min} \\ & W_B \geq N * (w + s) \end{array} \tag{6.2}$$

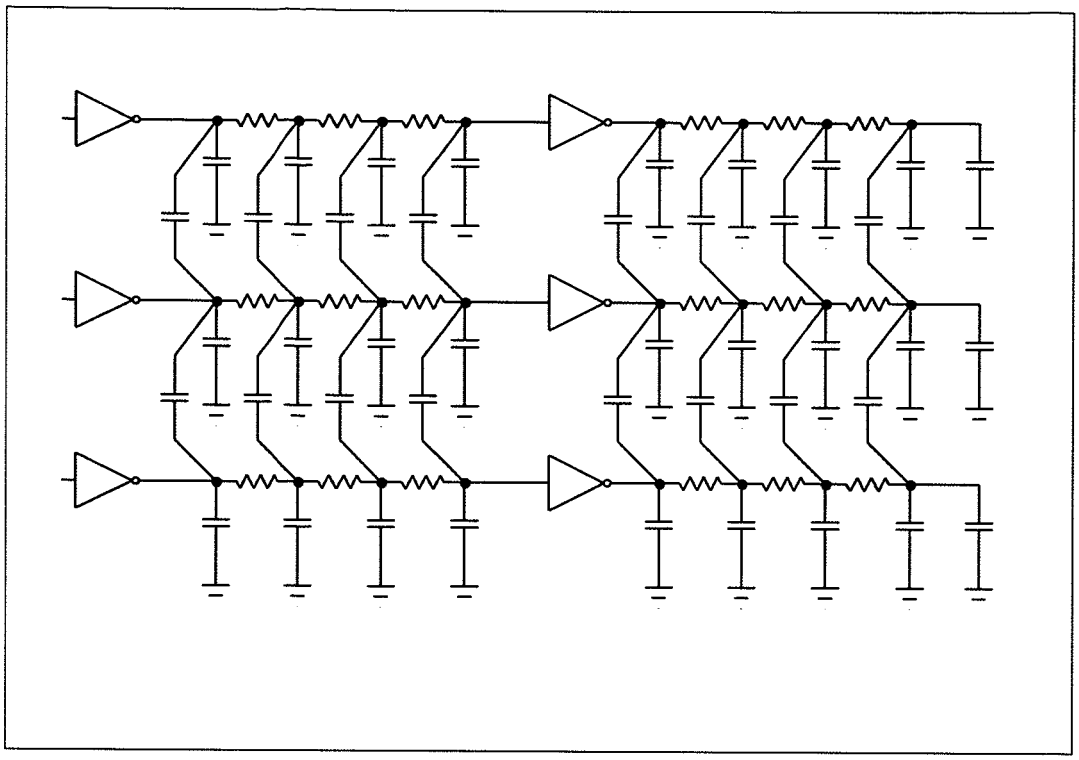


Figure 6.4: Interconnect model with 3 wires, 1 repeater per wire and 3 πRC elements per wire segment.

where

h	area of the repeater
$k+1$	number of sections
M	number of πRC elements per section
N	number of wires
s	space between the wires
w	width of the πRC element
d	size of the drivers
l	length of the πRC element
s_{min}	minimum space allowed between two wires
s_{max}	maximum space allowed between two wires
w_{min}	minimum width allowed for the πRC element
w_{max}	maximum width allowed for the πRC element
d_{min}	minimum size allowed for the drivers
d_{max}	maximum size allowed for the drivers
W_B	maximum bus area length

We observe that the objective function in equation(6.2) is not linear and therefore not a

semidefinite program because the capacitance matrix contains terms that are inversely proportional to the variable s (the spacing between the wires). However, we can reformulate the given problem by introducing the new variable $t = \frac{1}{s}$ we can then either use the formulation of equation(6.2) to find the area of only one wire (that is eliminating the space dependent term) or reformulate the problem by including a new variable $q = w + s$, such that the objective is linear. This formulation was applied by Vanderberghe [3][3]:

$$\begin{array}{ll} \text{minimize} & N\,q \\ \text{subject to} & (T_{max}/k+1)G(w,d) - C(w,t,d) \geq 0 \\ & \frac{1}{s_{min}} \geq t \\ & t \geq \frac{1}{s_{max}} \\ & w \geq w_{min} \\ & w_{max} \geq w \\ & d_{max} \geq d \\ & d \geq d_{min} \\ & (q-w)t \geq 1 \\ & \begin{pmatrix} t & 1 \\ 1 & \frac{W_B}{N} - w \end{pmatrix} \geq 0 \end{array} \tag{6.3}$$

which is a convex problem in the parameters w, t, d, q.

Because the product of parameters $(q - w)t \geq 1$ is not linear we have to find a way to reformulate this term. To this purpose, we can use the following general equivalence relations

$$\begin{aligned} x \geq 0 \quad xy \geq 1 \quad &\Leftrightarrow \quad x \geq 0, \quad \left\| \begin{bmatrix} 2 \\ x-y \end{bmatrix} \right\| \leq x+y \\ &\Leftrightarrow \quad x \geq 0, \quad \begin{bmatrix} x+y & 0 & 2 \\ 0 & x+y & x-y \\ 2 & x-y & x+y \end{bmatrix} \geq 0 \end{aligned} \tag{6.4}$$

Substituting $q-w = x$ and $t = y$, we can apply (6.4) to equation(6.3) to obtain the following optimization problem:

[3]This problem was regarded to compare our method against Vandenberghe's work. It should not be considered as a real-life optimization problem

$$
\begin{array}{ll}
\text{minimize} & N\,q \\
\text{subject to} & (T_{max}/k+1)G(w,d) - C(w,t,d) \geq 0 \\
& \frac{1}{s_{min}} \geq t \\
& t \geq \frac{1}{s_{max}} \\
& w \geq w_{min} \\
& w_{max} \geq w \\
& d_{max} \geq d \\
& d \geq d_{min} \\
& \begin{bmatrix} q-w+t & 0 & 2 \\ 0 & q-w+t & q-w-t \\ 2 & q-w-t & q-w+t \end{bmatrix} \geq 0 \\
& \begin{pmatrix} t & 1 \\ 1 & \frac{W_B}{N} - w \end{pmatrix} \geq 0
\end{array}
\tag{6.5}
$$

In this approach we are investigating only one out of the $k+1$ sections per wire. This is allowed because of the symmetry of the problem (the width of the πRC elements, the space and the repeater sizes are equal). Therefore, we study the behavior of one section and then extend the results automatically to all the others.

6.3.2 Proposed tests

We have run the following tests to compare our method with other approaches.

1. Comparison against Vanderberghe [3] results, specifically:
   (a) Example2.m of [3] -Combined sizing of drivers, repeaters, and wires-
   (b) Example5.m of [3] -Combined wire sizing and spacing-
2. Test using parameters from CMOS12 technology
3. Pstar simulations

6.3.3 Example2.m: Combined sizing of drivers, repeaters and wire.

In exercise2.m [3] the combined sizing of drivers, repeaters and wires is investigated using the dominant time constant approach. This particular example uses M πRC elements with capacitances and conductances proportional to the element widths. The input capacitance of the repeater is affine in d: $C_0 + cd$ and the output conductance is linear in d : gd, see figure(6.5)

We modified the script "*example2.m*" to use one πRC element[4]. The example is designed for wire sizing using tapering (different values for the width w), different drivers sizes and one wire[3](no coupling capacitances).

[4]Element and segment will be indistinctly used through this document

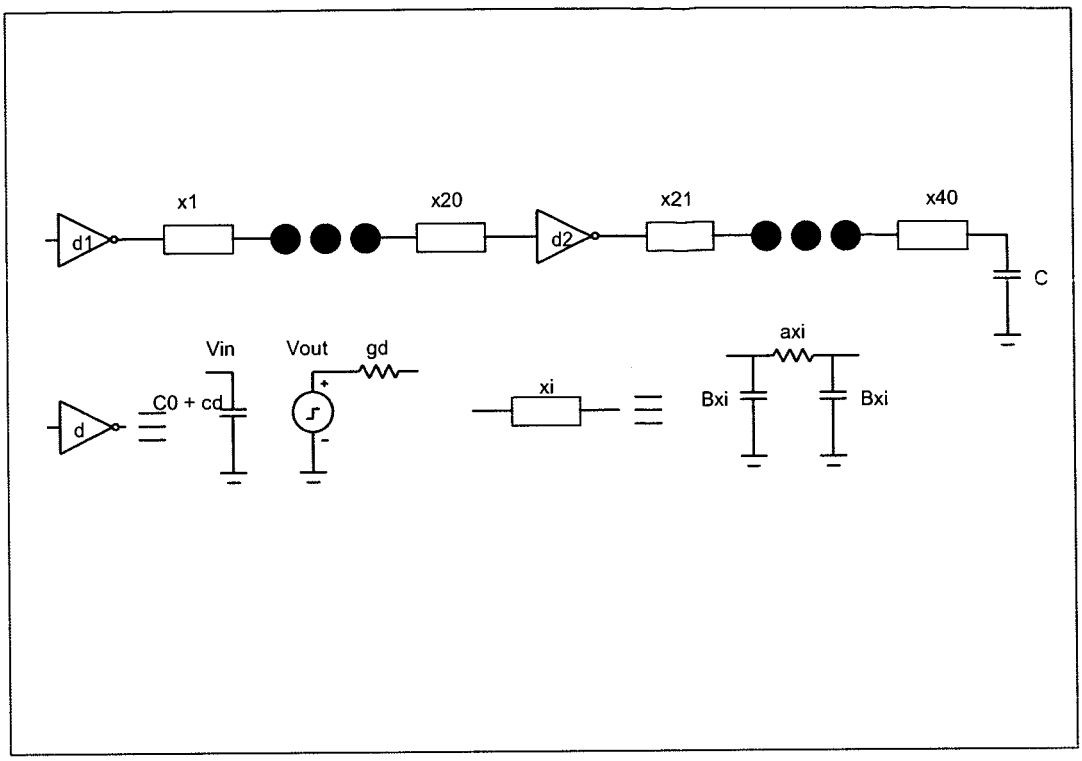


Figure 6.5: Example 2 from Boyd and Vanderberghe [3].

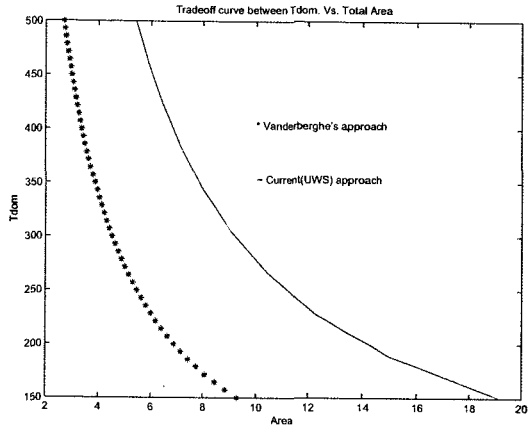
The purpose of the comparison against our model is to find differences between UWS and NUWS and to test our first academical problem before we proceed with a real life problem in the next section, namely, the optimization of power optimization.

The numerical values in the calculations are those used by Vanderberghe in his example. Also, a modification to the matrices in our script was necessary because the input and the load capacitance of the repeater are affine in $d \longrightarrow (C_0 + c)d$. Furthermore, we eliminated coupling and fringing capacitance in the script to make the problem similar.

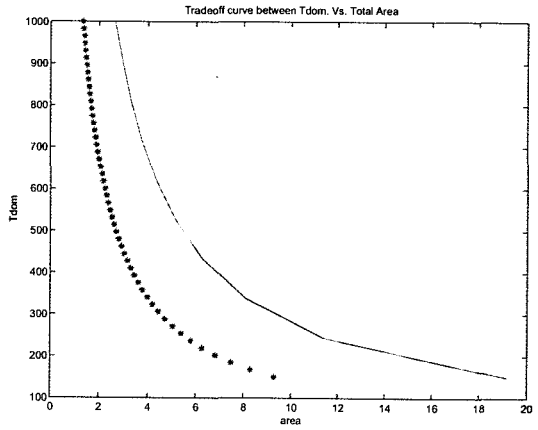
The parameter values where taken from section 5.2 of [3]:

$$g = 1, \quad C_0, = 1, \quad c = 3, \quad \alpha = 5, \quad \beta = 0.1, \quad C = 50, \quad l_i = 1, \quad L = 10, \quad n = 2, \quad w_i \geq 2.$$

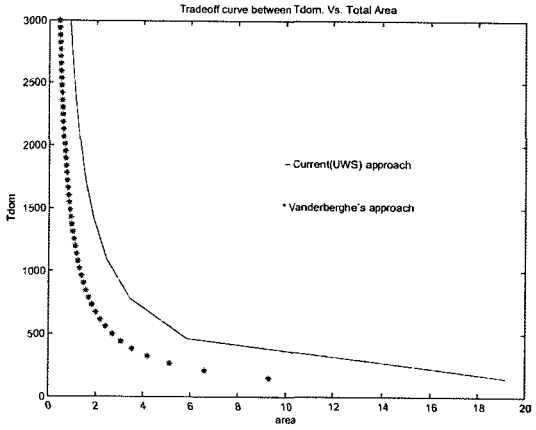
On figure(6.6(a)) we see that Vanderberghe's approach produces smaller area results. This can be explained by noting that the area is reduced if wire sizing with tapering is used. In our current approach we use uniform wire sizing instead. The reader can also observe that the curves get closer to each other as the delay increases (figures(6.6(b),6.6(c),6.6(d))).



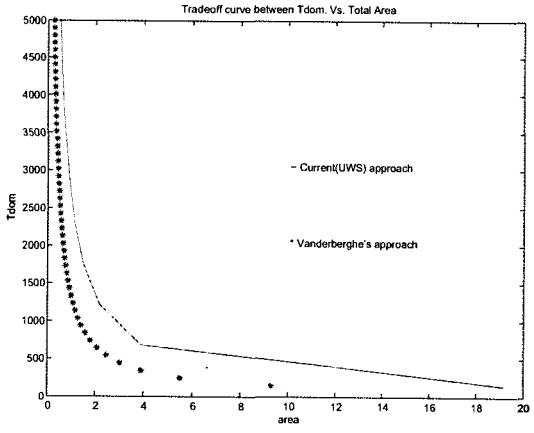
(a) Tdom(500 units) Vs. Area(μm^2)



(b) Tdom(1000 units) Vs. Area(μm^2)

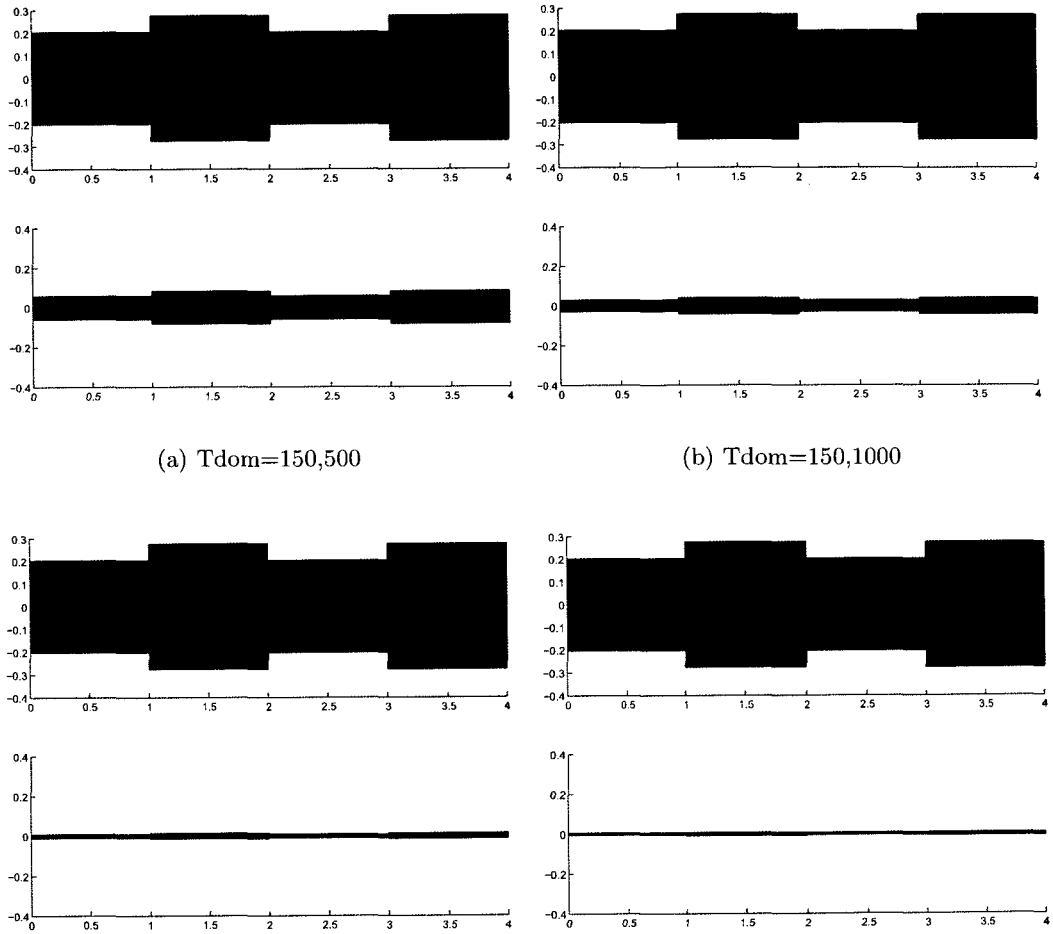


(c) Tdom(3000 units) Vs. Area(μm^2)



(d) Tdom(5000 units) Vs. Area(μm^2)

Figure 6.6: Different values for Tdom.



(a) Tdom=150,500

(b) Tdom=150,1000

(c) Tdom=150, 3000

(d) Tdom=150,5000

Figure 6.7: Wire Topology.

Figure(6.6) shows that the trade-off curves shown are convex. For very fast propagation,i.e, for very small values of Tdom, the optimal area from UWS is about two times as large as from Vandenberghe's approach. Note, however, that the number of repeaters has not been optimized in this example (only one repeater was used). For slow propagation, i.e. for large values of Tdom, the differences between the two approaches decrease. In the case the area approaches its lower limit which is imposed by the lower bound of the width.

Finally, figure(6.7) presents four solutions of the semidefinite problem: figure(6.3.3) presents the solutions for $Tdom = 150$ and Tdom $= 500$, figure(6.7(b)) presents the solutions for Tdom $= 150$ and Tdom $= 1000$, figure(6.7(c)) presents the solution for Tdom $= 150$ and Tdom $= 3000$, figure(6.7(d)) presents the solutions for Tdom $= 150$ and Tdom $= 5000$.

6.3.4 Example5.m: Combined wire sizing and spacing.

We compare Vanderberghe's example 5 [3] with our research because it shows what is called *"an important advantage of dominant time constant minimization over techniques based on Elmore delay: the ability to take into account non-grounded capacitors"*[3]. This example includes bottom and coupling capacitances which stays a step before our final objective which includes also fringing capacitances and repeater sizing. The problem is therefore to determine the optimal sizes of interconnect wires and the optimal distances between them(figure(6.8)).

The wires are connected to a voltage source with output conductance G and to a capacitive load, the problem consists of three wires and six nodes. In our work we have the same topology and the same values used in the example by Vandenberghe with the exception that our approach is designed for uniform wire sizing with equal repeater sizes. This allow us to see differences between the two approaches.

The parameter values[5] were taken from [3]:

$$G = 100, \quad C_1 = C_2 = C_3 = 10 \quad \alpha = 1, \quad \beta = 0.5, \gamma = 2$$

The distance between the wires is greater than $s_{ij} > 1$ and the wire widths are less than $w_{ij} < 2$.

Figure(6.9) shows the optimal value of Tdom as a function of W_B for the UWS and NUWS approaches. Clearly, we see that the NUWS approach gives smaller optimal bus widths than the UWS. Once more,this can be explained by noting that NUWS has an advantage over UWS when no full buffer insertion is applied. We'll see more details and comparisons when we board CMOS12 optimization in a further section.

[5]This exercise uses arbitrary units

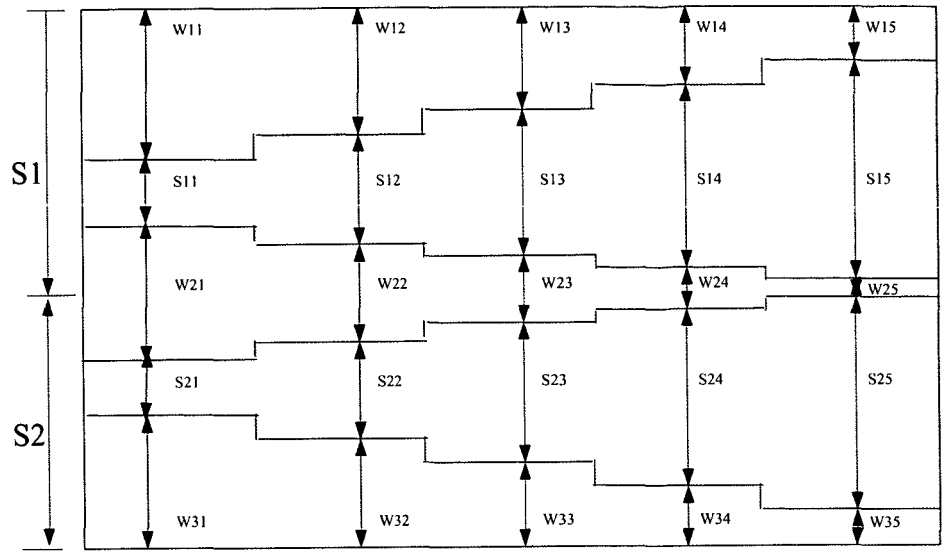


Figure 6.8: Example 5 from Boyd and Vanderberghe [3].

Figures(6.9(a), 6.9(b), 6.9(c), 6.9(d)) show different ranges for Tdom to try to establish a behavior for Tdom $\rightarrow \infty$. Different from figure(6.6), the curves in figure(6.9) do not converge to a common value as Tdom becomes large.

6.3.5 CMOS12 technology simulations

The data provided by Philips research for the CMOS12 technology was taken from the design manual, gcorelib [25].

We introduce real life parameters to provide a trade-off curve between area and Tdom that may be useful for the designer in his quest for area optimization. The introduction of CMOS12 parameters constitutes the second modification to Vanderberghe's approach [3]. One of the objectives of this thesis is to focus on the CMOS12 technology.

Table (6.1) gives the parameter values applied in this real life example. the input capacitance c_{in} and the output conductance g, both per unit width, were computed using a linear fit of c_{in} and g respectively as a function of d. The data was taken from the gcorelib

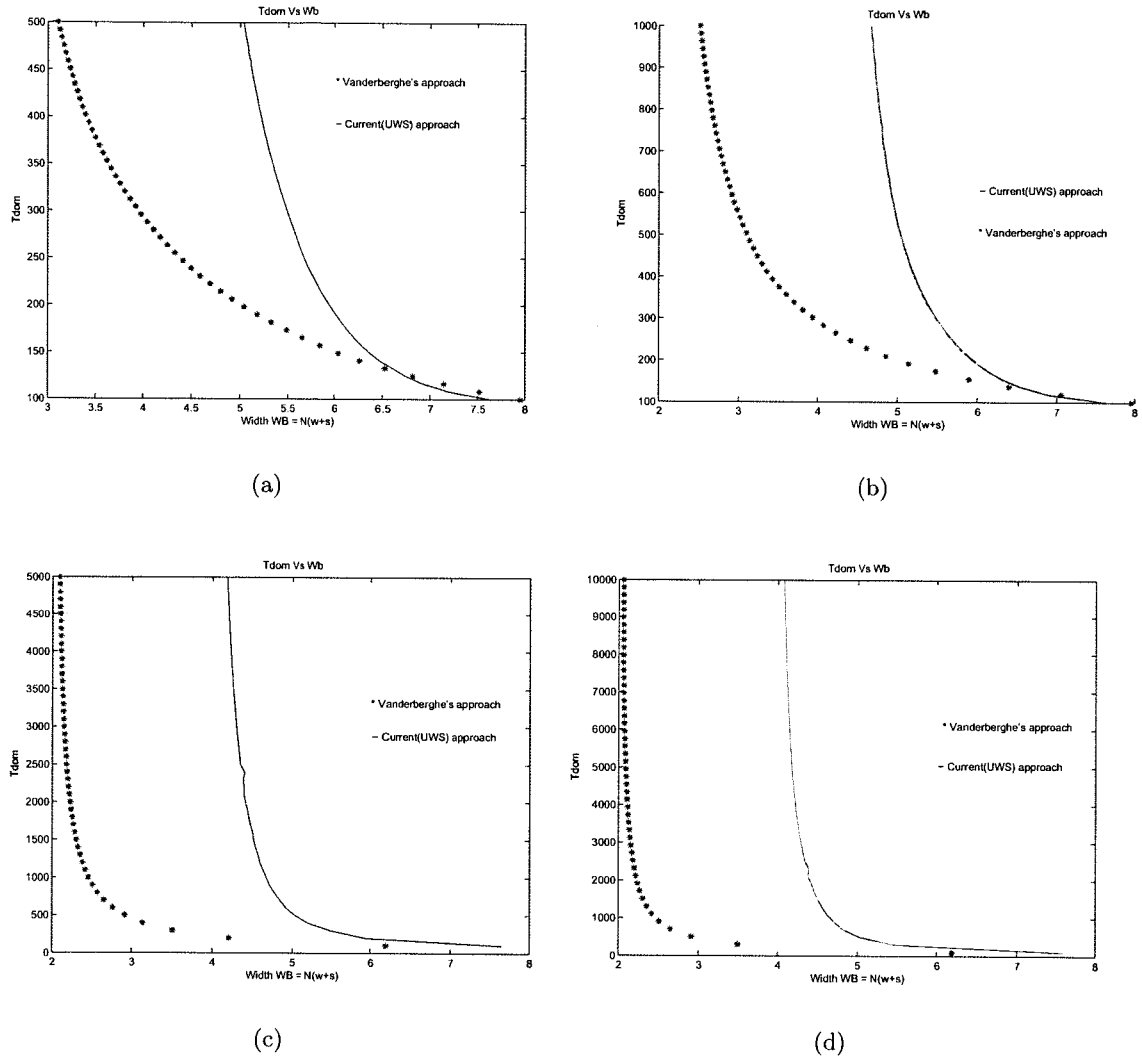


Figure 6.9: Comparison against Vanderberghe's example No. 5.

k	$=$	1	$->$	Buffers in the system
n	$=$	6	$->$	Nodes per section per wire
M	$=$	$n-1$	$->$	πRC elements or segments
N	$=$	3	$->$	Wires
C_L	$=$	5×10^{-15}	$->$	Load capacitance (constant) in $\frac{F}{\mu m}$
c_b	$=$	14.45×10^{-18}	$->$	Unit area parallel plate capacitance in $\frac{F}{\mu m}$
c_{eb}^{∞}	$=$	36.0×10^{-18}	$->$	Unit length edge-to-bottom capacitance for infinite spacing, in $\frac{F}{\mu m}$
$c_{\triangle e}$	$=$	16.31×10^{-18}	$->$	Unit length spacing-dependent capacitance, in $\frac{F}{\mu m}$
c_{ec}	$=$	24.1×10^{-18}	$->$	Unit length coupling capacitance, in $\frac{F}{\mu m}$
r_{sheet}	$=$	57×10^{-3}	$->$	Sheet Resistance, in $\frac{\Omega}{Square}$
c_{in}	$=$	5.3×10^{-9}	$->$	Input repeater capacitance per unit width, in $\frac{F}{\mu m}$
c_{out}	$=$	5×10^{-15}	$->$	Output repeater capacitance per unit width, in $\frac{F}{\mu m}$
g	$=$	450×10^{-6}	$->$	Repeater conductance per unit width, in $\frac{1}{\Omega \mu m}$
L	$=$	20000	$->$	Length of the bus, in μm
l	$=$	$L/((k+1)*M)$	$->$	length of each πRC element, in μm
h	$=$	$\frac{1.64*4.92}{1.17}$	$->$	Width factor for the buffers, in μm

Table 6.1: Values for the real-life optimization example.

library [25]. The data used in the linear fit is presented in table(6.2)
Figure(6.10) shows the linear fit for the output conductance and the input capacitance. The result is a rect with the driver size d for the independent term and dependent for the capacitance and the conductance *i.e.* $G_0 = 450 \times 10^{-6} d + 2.8 \times 10^{-7}$.

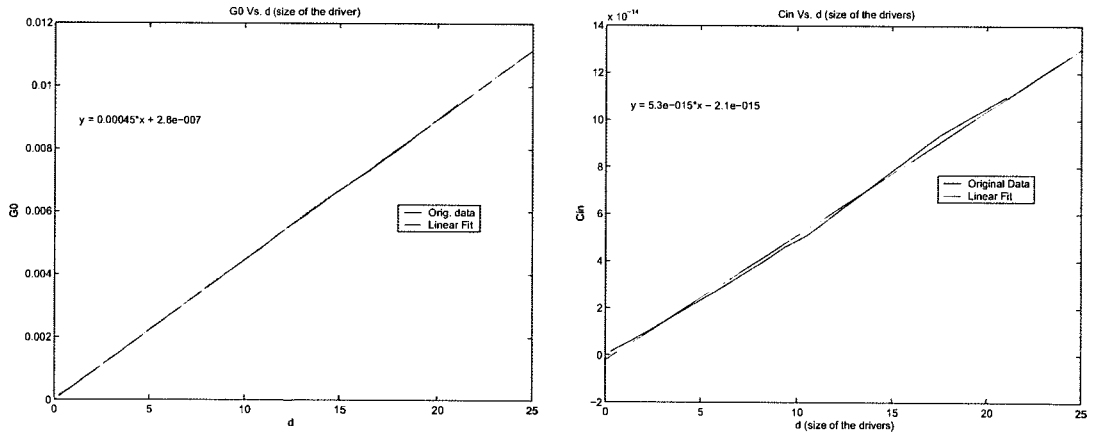
Consider an example with three wires $N = 3$, one repeater($k = 1$), six nodes($n = 6$) and $M = 5$ πRC elements. Figure(6.11(a)) shows the dominant time constant as a function of the area, figure(6.11(b)) shows Tdom as a function of the bus area, (figure(6.11(c))) shows Tdom as a function of the dissipation power and (figure(6.11(d))) shows two solutions for the problem(T = Tdom(max) and T = Tdom(min)).

Notice that the curves are convex and that each point represents a solution of the optimization problem for a given value of Tdom. Figure(6.11(d)) shows wires representing solutions for the minimum and the maximum dominant time constant for the circuit, respectively.

The power dissipation can be separated into switching power and short-circuit power (see section 2.1.2), figure(6.12(a)) shows the switching power, figure(6.12(b)) the short circuit power, figure(6.12(c)) the total power and figure(6.12(d)) the total area. Observe that every point of the graphic is a solution of the optimization problem. In particular, we have a

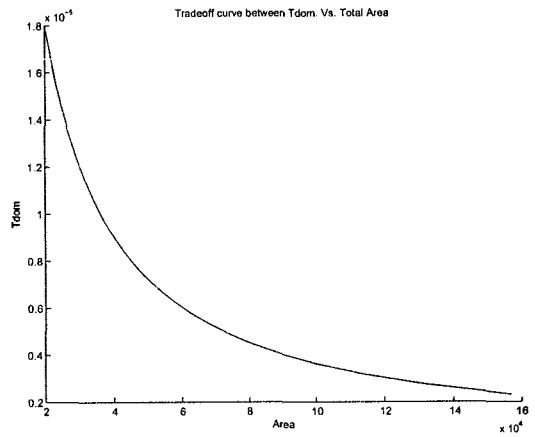
d	$G_{driver} \times 10^{-4}$	$C_{in} \times 10^{-15}$
0.26	1.35	1.40
0.58	2.71	2.72
1.17	5.15	5.27
2.34	10.39	10.60
3.51	15.62	16.30
4.68	20.87	21.80
5.85	26.11	27.40
7.02	31.34	33.50
8.19	36.63	39.40
9.36	41.84	46.00
10.53	46.94	51.00
14.04	62.89	72.60
17.55	78.12	93.80
21.06	94.34	110.00

Table 6.2: Values for input capacitance and load conductance.

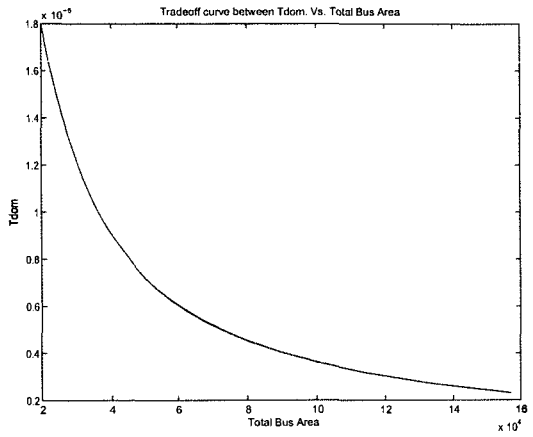


(a) $G_{driver} = 450 \times 10^{-6}d + 2.8 \times 10^{-7}$ (b) $C_{in} = 5.3 \times 10^{-15}d - 2.1 \times 10^{-15}$

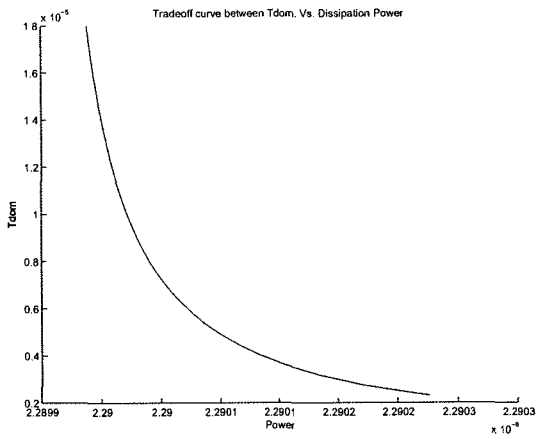
Figure 6.10: Linear fit of driver dependent conductance and capacitance.



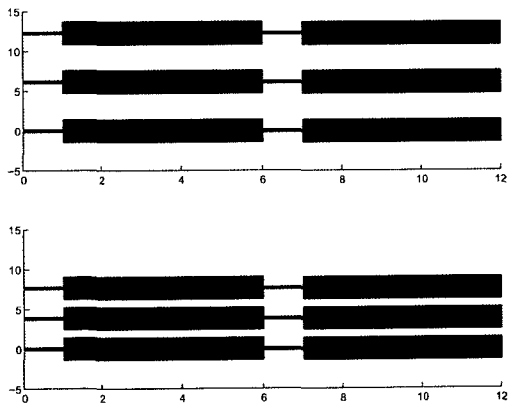
(a) Tdom Vs. Total Area



(b) Tdom Vs. Bus Area



(c) Tdom Vs. Dissipation Power



(d) Wire Sizing

Figure 6.11: CMOS12 trade-off curves.

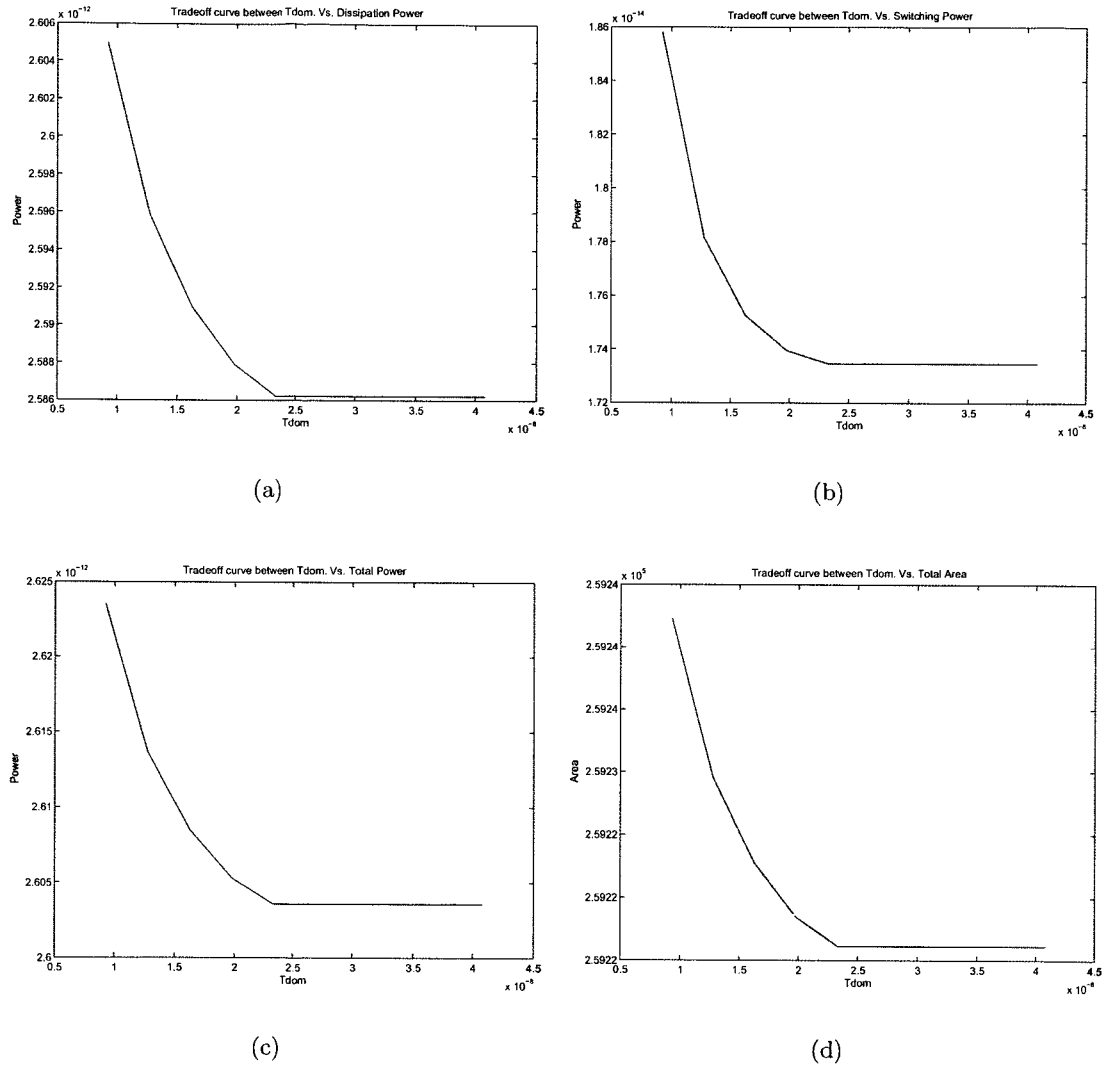
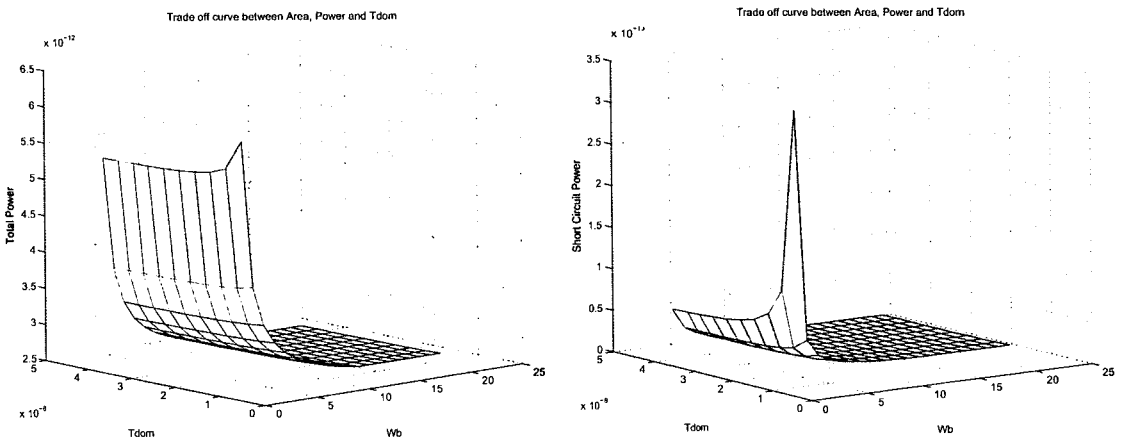


Figure 6.12: Area Optimization for $n = 6$ nodes , $N = 3$ wires , $k = 1$ repeaters . The power dissipation is given in Joules per cycle and Tdom is given in seconds.

monotonically decreasing bijective function $f(T^i_{dom}) < f(T^{i+1}_{Tdom})$ where $i < i+1$.

Figure(6.13) shows the total power and the time dominant constant as a function of the total width of the bus W_B as defined by the equation (6.5).



(a) Total Power vs. Tdom vs. Wb

(b) Short Circuit Power vs. Tdom vs. Wb

Figure 6.13: Dominant time constant vs. power vs. area width.

We see from figure(6.13(a)) that for a small value of W_B (that is small width and space between the wires) and a small Tdom value the power reaches its maximum. This result agrees completely with the theory because when the space between the wires is small then the coupling increases. Also the switching power increases because the total capacitance that is observed by the repeaters increases if the wire spacing is reduced. Because of this, the rise time *trise* in equation(3.27) increases.

On the other hand, we can determine an optimal value for the size of the repeater d by bounding the width of the bus W_B (we specify bounds for the width of the wire w and the space between them s) and by choosing an expected Tdom in the simulation.

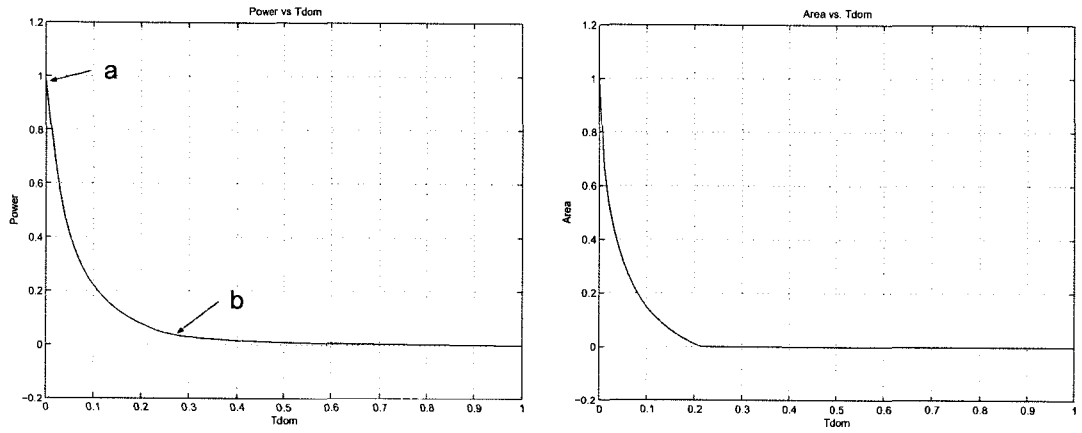
Example:

Define a simulation with $N = 3, k = 1, n = 6, M = 5, w_{max} = 2, s_{max} = 1, W_b = 3(2+1) = 9$ to find the following results:

```
Wb -> 9.000000
Tdom-> 1.38e-009 w-> 1.95 t-> 1 d-> 21.0  power-> 8.64e-012 area -> 178160.138647
Tdom-> 3.69e-009 w-> 0.52 t-> 1 d->  8.2  power-> 5.69e-012 area ->  91613.847982
```

```
Tdom-> 6.01e-009 w-> 0.32 t-> 1 d->  4.6  power-> 5.19e-012 area ->  79393.348422
Tdom-> 8.32e-009 w-> 0.32 t-> 1 d->  2.3  power-> 5.05e-012 area ->  79296.361883
Tdom-> 1.06e-008 w-> 0.32 t-> 1 d->  1.5  power-> 5.01e-012 area ->  79264.931574
Tdom-> 1.29e-008 w-> 0.32 t-> 1 d->  1.1  power-> 4.99e-012 area ->  79249.118744
Tdom-> 1.52e-008 w-> 0.32 t-> 1 d->  0.95 power-> 4.98e-012 area ->  79239.549300
Tdom-> 1.75e-008 w-> 0.32 t-> 1 d->  0.80 power-> 4.97e-012 area ->  79233.120448
Tdom-> 1.98e-008 w-> 0.32 t-> 1 d->  0.68 power-> 4.97e-012 area ->  79228.498608
Tdom-> 2.21e-008 w-> 0.32 t-> 1 d->  0.60 power-> 4.96e-012 area ->  79225.013455
```

Figure(6.14) depicts the results.



(a) Total Power vs. Tdom

(b) Total Power vs. Tdom (100 intervals)

Figure 6.14: Normalized power, area vs. Tdom ($W_b = 9$).

By sacrificing a small percentage 10% of the total possible delay, we greatly reduce power 77% (see rows number 1 and 2 of the data shown above). Observe that according to the data, the repeater width ($d = 8.2$) to obtain a power of 5.69×10^{-12} agrees most closely to the size of the "IVX7" repeater (see table(6.3)).

Names	$IVX0$	$IVX05$	$IVX1$	$IVX2$	$IVX3$	$IVX4$	$IVX5$
Sizes	0.26	0.58	1.17	2.34	3.51	4.68	5.85
Names	$IVX6$	$IVX7$	$IVX8$	$IVX9$	$IVX12$	$IVX15$	$IVX18$
Sizes	7.02	8.19	9.36	10.53	14.04	17.55	21.06

Table 6.3: Driver size.

We show solutions marked as (a) and (b) in figure(6.15). This result allows the designer to play with the driver sizes and the delay of the topology according to the requirements defined for its circuitry.

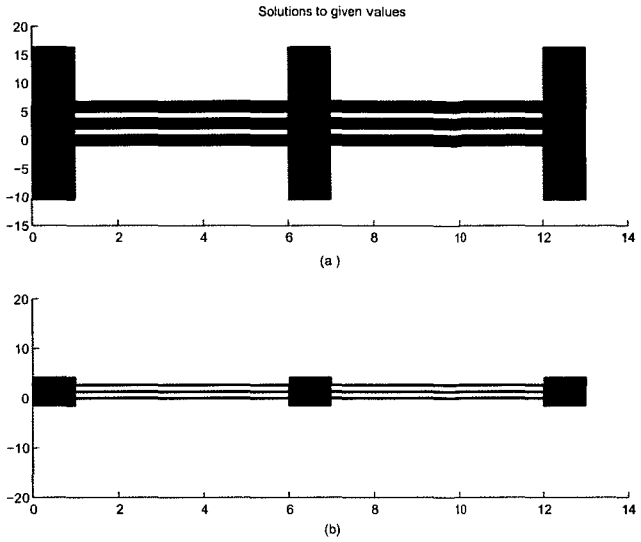


Figure 6.15: Solutions shown in figure(6.14).

To obtain new data for the power dissipation without running the simulation again, we use an interpolation method (see chapter(5)). The procedure is the following: First we run the simulation, then we use the resulting values for Tdom, W_B and the power dissipation to generate new values for the variables by using interpolation, see figure(6.16).

In our case we use cubic spline interpolation which uses a cubic polynomial to fit every pair of points (chapter(5)). This allows a transparent use of the results of the investigation for the designer without needing technical details of the simulation itself which permits mobility and efficiency for finding new data values

To measure the accuracy of the results we run the simulation with points in between the points used to construct the interpolation curve. The results from simulation and interpolation are then compared. Figure(6.16) shows the curves for the original data and for the interpolated data. The figures match well with an error of $\max(\text{abs}(z - zfit)) = 3.2561 \times 10^{-5}$, where z is the original data and *zfit* the interpolated data. The accuracy of the solutions is increased if the number of points is also increased. See for example table(6.4)

We observe that the error decreases when the number of the data points increases but because splines interpolation uses a third degree polynomial[6] between every two points, we

[6]See Chapter 5 for details on interpolation methods

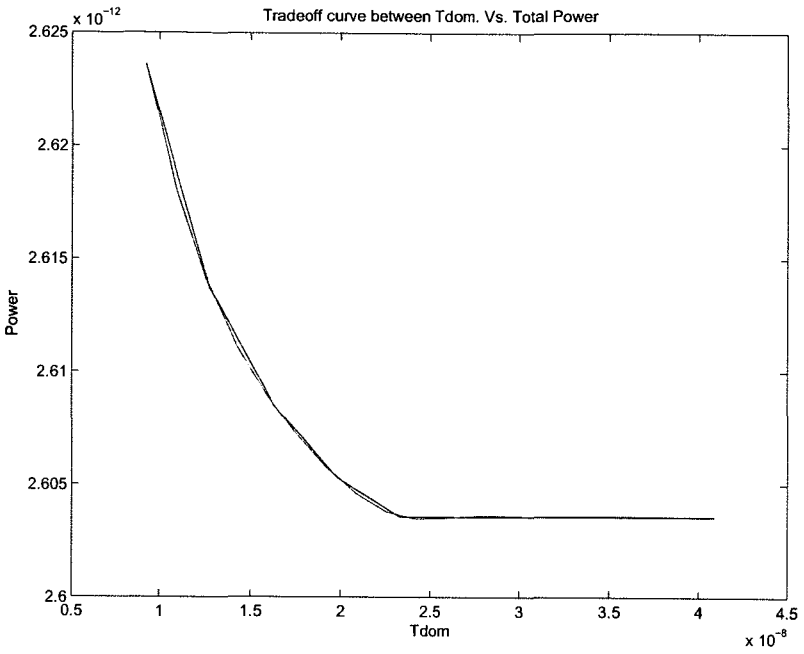


Figure 6.16: Cubic splines interpolation of the power.

see that even when the number of the sample is small, the error between the sample and the original data is of the order $\mathcal{O}(\times 10^{-5})$ which is very good indeed.

6.3.6 Pstar Simulations

In order to verify the accuracy of the optimization results with the dominant time constant approach, we performed circuit simulations with the PSTAR circuit simulator [24]. To this purpose, a script is used that writes to a file the circuit topology that was created within our main file in Matlab. This file allows us to run the Pstar simulations and to calculate the power dissipation for further comparison with our script.

No. data	max(error)
10	$6.508e-005$
15	$6.4219e-005$
20	$3.2561e-005$
30	$8.2709e-007$

Table 6.4: Interpolation error as a function of the number of data points

We ran a Pstar simulation for a fixed Tdom $= 4.0859 \times 10^{-8}s$ and a linear space for the width of the bus W_B between $1.92\mu m$ and $21\mu m$. The result is compared with our semidefinite optimization(Figure(6.17)).

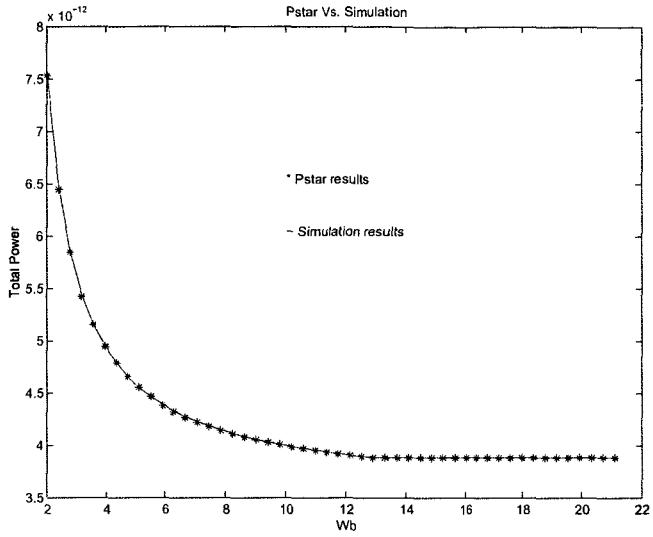


Figure 6.17: Comparison of our SDP approach and power dissipation computed from Pstar simulation.

We observe in figure (6.17) that the results agree very well. The margin error was less than 1%. This result shows that our model is sufficiently accurate.

6.4 Power optimization

6.4.1 Problem definition

The voltage source delivers an energy of $\mathbf{v}^T C \mathbf{v}$ during a transition from initial voltage 0 to final voltage $\bar{v}$, half of this energy is dissipated in the resistors and the other half is stored in the capacitors, this process is called charging. During the next transition from voltage $\bar{v}$ to 0 the reverse process takes place, the capacitors are discharged and the energy is dissipated in the resistors.
For a fixed clock rate and a fixed probability of transition, the average dissipated power is proportional to:

$$\bar{v}C(x)\bar{v} = \Sigma_{i=1}^{m} x_k(\bar{v}^T C_i \bar{v})$$

which is a linear function of the design parameters $x = (w, t, d)$. We can optimize the power dissipation subject to constraints on the dominant time constant and parameter bounds by

solving the following SDP:

$$\begin{array}{ll} \text{minimize} & \mathbf{v}^T * C(w,t,d) * \mathbf{v} \\ \text{subject to} & (T_{max}/(k+1))G(w,d) - C(w,t,d) \geq 0 \\ & \frac{1}{s_{min}} \geq t \\ & t \geq \frac{1}{s_{max}} \\ & w \geq w_{min} \\ & w_{max} \geq w \\ & d_{max} \geq d \\ & d \geq d_{min} \\ & \begin{pmatrix} t & 1 \\ 1 & \frac{W_B}{N} - w \end{pmatrix} \geq 0 \end{array} \tag{6.6}$$

where the last constraint of (6.6) is a bound on the bus width $W_B \leq N(w+s)$, and $\mathbf{v}$ is the vector of node voltages.

The expression for the bound on area in equation (6.6) can be replaced by an equivalent constraint by using the approach of[3]. This constraint is expressed as (see [3] for details) or equation(6.4):

$$t(\tfrac{W_B}{N} - w) \geq 1$$

$$\begin{pmatrix} t + \frac{W_B}{N} - w & 0 & 2 \\ 0 & t + \frac{W_B}{N} - w & t - \frac{W_B}{N} + w \\ 2 & t - \frac{W_B}{N} + w & t + \frac{W_B}{N} - w \end{pmatrix} \geq 0$$

This constraint is of the hyperbolic type and also semidefinite positive and can be represented as a linear matrix inequality.

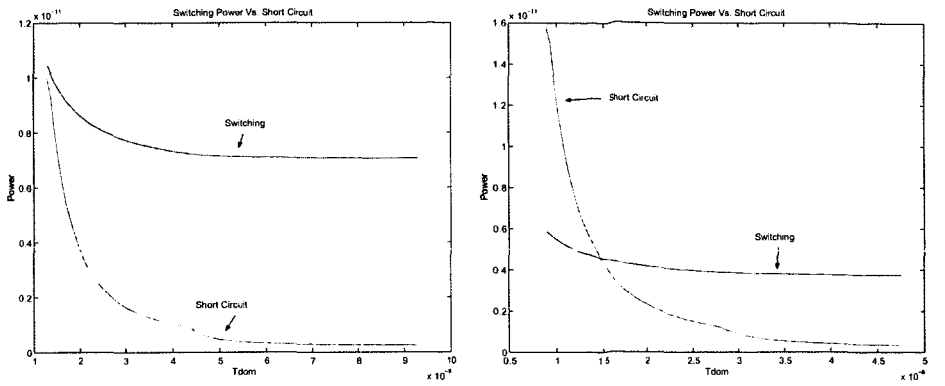
We see that the expression of the power dissipation only includes the switching power. For the short-circuit power a suboptimal value is computed from the design parameter values that are obtained from the optimization of the switching power, as was discussed in section(3.7).

The switching power is linear with respect to the parameters w, t and d. We use the optimal values to compute short-circuit power, which is a suboptimal value. Short-circuit power is not included in the optimization because it is a nonlinear function of the design parameters.

6.4.2 Short Circuit Power Vs. Switching power

We start by showing the differences between switching power and short circuit power: as discussed in chapter(3), the short circuit power in most practical situations is about 10% of the switching power.

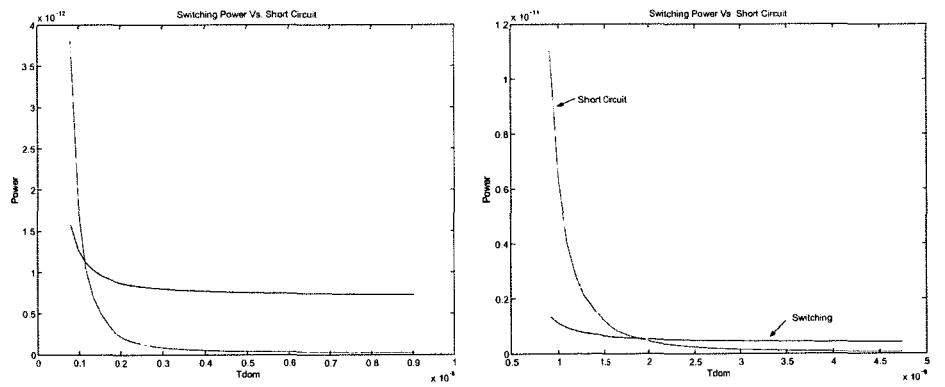
Figure(6.18(a)) shows that the switching power is dominant for $k = 10$ repeaters. When



(a) 10 repeaters (b) 20 repeaters

Figure 6.18: Short circuit power Vs. Switching power, $L = 200mm$.

the number of repeaters is increased to $k = 20$ (figure(6.18(b))), the short circuit power becomes dominant for very fast lines. The length of the bus has been increased to $200mm$, which shows that for longer lines switching power becomes dominant Figure(6.19(a)) and



(a) 10 repeaters (b) 20 repeaters

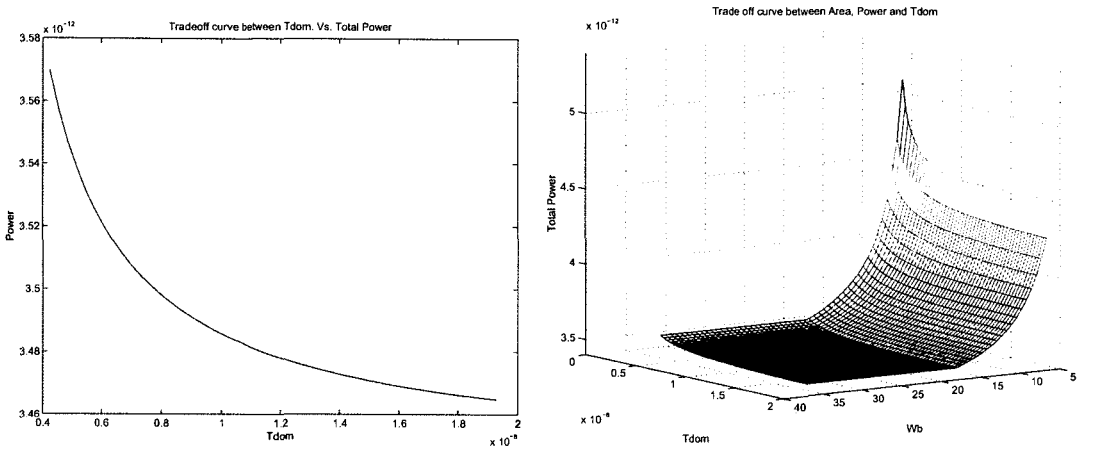
Figure 6.19: Short circuit power Vs. Switching power, $L = 20mm$.

figure(6.19(b)) show the results for $k = 10$ and $k = 20$ repeaters, respectively. This confirms that the short circuit power is dominant for very fast circuits and short lines ($20mm$).

6.4.3 Finding the optimal repeater size for a desired performance

One of the most important results of this research is that the approach allows the designer to identify the optimal driver size for a given set of parameters with some desired performance.

i.e. Consider a topology four wires ($N = 4$) with $n = 10$ nodes and $M = 9$ πRC elements, with $k = 2$ repeaters and a length of $L = 20mm$. The minimum spacing between the wires was set to $1\mu m$ to avoid Miller's effect on the lines. The maximum width for the wires was set to $5\mu m$.



(a) Power 2d

(b) Power 3d

Figure 6.20: Power in 2D and 3D.

Figure(6.20(a)) shows a 2D plot of the total power dissipation, including both switching and short circuit power as a function of the dominant time constant. From this figure we observe that the delay is a convex function of the power. Short signal delay correspond to high performance dissipation.

Again, as in section(6.3.5), we first determine the optimal repeater size d. To this purpose, we run the optimization for various values of Tdom:

```
Wb -> 36.000000
Tdom-> 1.03e-009 w-> 8.92e-001 d-> 1.45e+001 power-> 5.785431e-012
Tdom-> 1.42e-009 w-> 5.87e-001 d-> 9.61e+000 power-> 4.678380e-012
Tdom-> 1.81e-009 w-> 4.38e-001 d-> 7.17e+000 power-> 4.205846e-012
Tdom-> 2.20e-009 w-> 3.49e-001 d-> 5.71e+000 power-> 3.947200e-012
Tdom-> 2.59e-009 w-> 3.20e-001 d-> 4.31e+000 power-> 3.774120e-012
```

```
Tdom-> 2.97e-009 w-> 3.20e-001 d-> 3.29e+000 power-> 3.682266e-012
Tdom-> 3.36e-009 w-> 3.20e-001 d-> 2.67e+000 power-> 3.631177e-012
Tdom-> 3.75e-009 w-> 3.20e-001 d-> 2.25e+000 power-> 3.598455e-012
```

From the optimization results we conclude that a bound of $1.81 \times 10^{-9}s$ for Tdom gives sufficient reduction in power dissipation. Notice that in this example we used large upper bounds on the wire width w ($w_{max} = 5\mu m$) and on the wire spacing s ($s_{max} = 4\mu m$). This configuration corresponds to relatively low power dissipation, see figure(6.20(b)) for $W_B = 36\mu m$. However, the dominant time constant is less than $2ns$.

The solution for $Tdom = 1.81 \times 10^{-9}s$ gives an optimal value for d of $7.17\mu m$. Using table(6.3) shows that this value corresponds most closely to the "IVX6" repeater, with $d = 7.02\mu m$.

6.4.4 Finding the optimal wire size for some fixed repeater size

From the previous section before we obtained as a result that the repeater "*IVX6*" was optimal for some desired performance ($Tdom < 2ns$). Using the corresponding repeaters size $d = 7.02\mu m$, the optimization was run again with the following results:

```
Wb -> 36.000000
Tdom-> 1.412623e-009 w-> 1.402899  d-> 7.020000 power-> 5.242596e-012
Tdom-> 1.412635e-009 w-> 1.394097  d-> 7.020000 power-> 5.232853e-012
Tdom-> 1.976566e-009 w-> 0.371871  d-> 7.020000 power-> 4.118836e-012
Tdom-> 1.982424e-009 w-> 0.369683  d-> 7.020000 power-> 4.116594e-012
Tdom-> 1.988283e-009 w-> 0.367523  d-> 7.020000 power-> 4.114383e-012
Tdom-> 1.994141e-009 w-> 0.365391  d-> 7.020000 power-> 4.112203e-012
```

These results show that a wire width $w = 0.365391\mu m$ gives a minimal value of the power dissipation, while Tdom is less than $2ns$.

Furthermore, we can set any of the parameter in the simulation to a fix value and this allows to define a subproblem for one or several of the parameters that we are regarding. This example shows the flexibility that the designer has to play with the tool according to his experience and current design limitations.

6.5 Repeater Insertion

Work on repeater insertion has been undertaken by Bakoglu [9] and more recently Garcea et al. [17], who proposed an approach for finding the optimal values for the repeater size and the critical distance between the repeaters in the interconnect line. In this approach (figure(6.21)) the wire width is not optimized. Both the resistance and the conductance of the wire have a constant value per unit length. We use the resulting equations to approximate the number of repeaters we should include into our script in order to work with optimal repeater insertion.

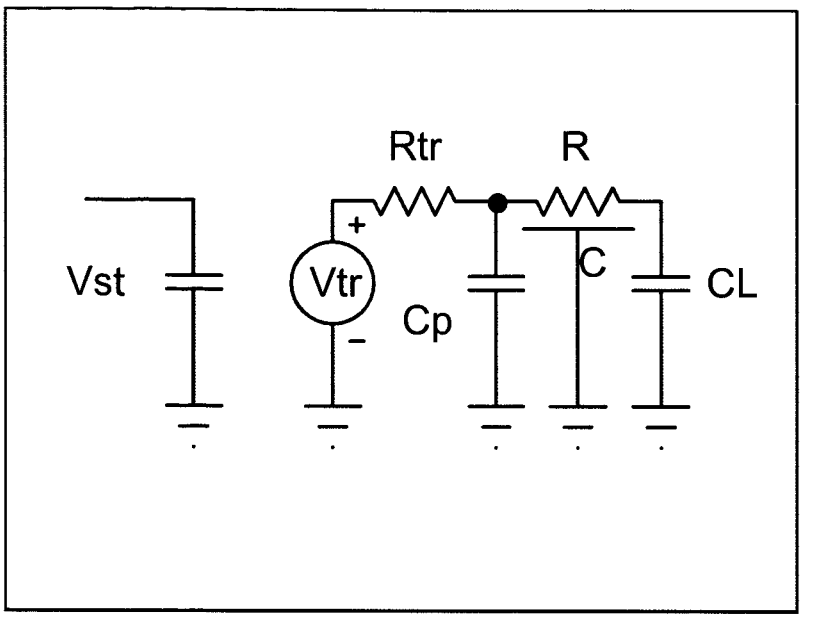


Figure 6.21: Garcea's model for an interconnect line

The delay of the circuit model of figure(6.21) is given by:

$$\begin{aligned} T &= (k+1)\left[bR_{tr}\left(C_p + C + C_L\right) + R\left(aC + bC_L\right)\right] \\ &= (k+1)\left[br_0(c_0 + c_p) + b(\tfrac{r_0 c}{d} + rc_0 d)l + arcl^2\right] \end{aligned} \tag{6.7}$$

Where

$$\begin{aligned} R_{tr} &= \tfrac{r_0}{d} \\ R &= r\,l \\ C_p &= c_p\,d \\ C_L &= c_0\,d \\ C &= c\,l \end{aligned}$$

k and l denote the number of repeaters and the length of a wire element between two repeaters, respectively; a, b depend on the switching model that is used.

Taking the partial derivative of Equation(6.7) with respect to the length l and the repeater size d and setting them to zero[17] the following relations for the critical length and width of the wire are found:

$$\begin{aligned} l_{crit} &= \sqrt{\tfrac{b\tau_0}{a\tau_i}} \\ d_{opt} &= \sqrt{\tfrac{r_0 c}{rc_0}} \end{aligned} \tag{6.8}$$

$$
\begin{aligned}
c_0 &= c_{in} \\
c_p &= c_{out} \\
r_0 &= 1/g \\
r &= R_s \\
c &= c_b \\
a &= 0.39 \\
b &= 0.69
\end{aligned}
$$

Table 6.5: Equivalence of parameters (Garcea's approach Vs. Ours)

Where

$$
\begin{aligned}
\tau_0 &= r_0(c_0 + c_p) \\
\tau_i &= rc
\end{aligned}
$$

Equations (6.8) provide the optimal length and the optimal driver size for a given technology. In our case we are interested in evaluating the CMOS12 technology parameters [26]. We evaluate(6.8) using parameter values that make Garcea's approach and ours equivalent. Table(6.5) lists the parameter values that we use in our model for this purpose.

Plugging in the values for the CMOS12 technology we find the following results for a total wire length of $L = (k+1) * l = 20$ mm,

$$
\begin{aligned}
l_{crit} &= 7.01180\ mm \\
d_{opt} &= 10.30980\ \mu m \\
k &= L/l_{crit} - 1 = 1.85232
\end{aligned}
$$

Next, we round $k = 1.85232$ to an integer ($k = 2$) number of repeaters and use this value in our script to run our simulations.

```
Wb -> 40.000000
Tdom-> 8.07e-010 w-> 1.36 d-> 21.00 power-> 10.10e-012
Tdom-> 1.31e-009 w-> 0.67 d-> 10.90 power->  6.40e-012
Tdom-> 1.32e-009 w-> 0.67 d-> 10.75 power->  6.36e-012
Tdom-> 1.34e-009 w-> 0.66 d-> 10.60 power->  6.32e-012
Tdom-> 1.35e-009 w-> 0.65 d-> 10.46 power->  6.27e-012
Tdom-> 1.36e-009 w-> 0.64 d-> 10.32 power->  6.23e-012  **1
Tdom-> 1.45e-009 w-> 0.59 d->  9.56 power->  6.02e-012
Tdom-> 1.65e-009 w-> 0.50 d->  8.15 power->  5.64e-012
Tdom-> 3.89e-008 w-> 0.32 d->  0.32 power->  4.36e-012
```

Figure (6.22) shows the resulting minimum power dissipation as a function of the dominant time constant for $k = 3$. The delay computed with Equation(6.7), with $l = l_{crit}$ and $d = d_{opt}$, equals 1.36 ns. Setting T_{dom} to this value results in a optimal power dissipation of 6.23 PJ per cycle. This solution has been marked on the curve in Fig. (6.22), together

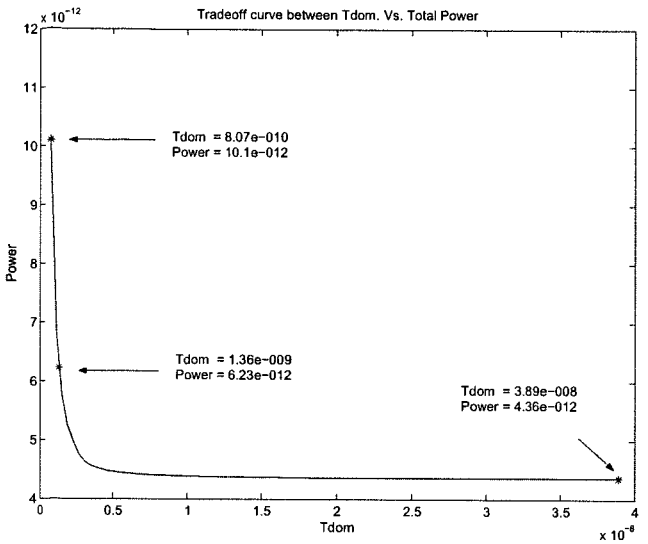


Figure 6.22: Total Power

with the solutions for the smallest and largest possible values of T_{dom} for the given bus width. Compared with the solution for minimum delay, the solution for $T_{\text{dom}} = 1.36$ ns corresponds to an increase of 59% in delay and a reduction of power dissipation of 68%.

The work reported in [17] helps to make a first estimation of the optimal number of repeaters that is to be inserted. A detailed comparison of their optimization approach and ours will be done in the future.

Chapter 7

Conclusions

A circuit optimization approach has been presented based on semidefinite programming. The approach is based on the time dominant constant, which is a quasiconvex function of the design parameters. In semidefinite programming a linear function is optimized subject to linear matrix inequality (LMI) constraints. Such an LMI constraint requires that a matrix, which is affine with the optimization parameters, is positive semidefinite. Scalar linear inequality constraints form a subclass of LMI constraints. We have formulated several primal semidefinite problems for the optimization of power dissipation and area with constraints on the width of the bus as well as with linear constraints on the optimization parameters. The problems were reformulated in order to fulfill the requirements of convexity and semidefinite positiveness.

The current work is based on the research of Vandenberghe and Boyd [3]. Their work has been extended by using real life parameters corresponding to the CMOS12 technology. Also, a suboptimal approximation for the short circuit power dissipation was included. Furthermore, we developed an interpolation model that can be used to find trade-offs between power dissipation, time dominant constant and maximum bus width. This model allows to draw a 3D graph relating the mentioned parameters. In Vandenberghe's and Boyd's work models were presented for the πRC wire elements and for the repeaters [3]. The repeater model has been made more realistic in this work. Vandenberghe and Boyd used a repeater with constant input capacitance. In our approach the input capacitance, the output capacitance, and the load conductance are all dependent on the repeater size d. This allows the computation of the power dissipation to be more accurate, which was one of the objectives of the research.

Several scripts have been implemented in Matlab in order to control the full and precise formulation of the problem. For example: we have a script for creating the reduced node incidence matrix, one to calculate the conductance and capacitance matrices, one that computes the linear, hyperbolic and semidefinite positive constraints and that calls SeDuMi, one to generate the graphics, one to create a Pstar circuit description file and many other small utilities including scripts to save graphics and interpolating functions.

The Self Dual Minimization/Optimization over self-dual homogeneous cones (SeDuMi) software [6] is the optimization tool used throughout this work. This tool, created by Sturm, has proved itself as it allows one to solve a great variety of problems, including semidefinite problems, by just defining the primal objective and its respective constraints. In contrast with other software, there is no need to specify the dual problem, which sometimes may be harsh. Furthermore, there is no need to define an initial starting point. Instead, the software starts from a random point and the algorithm then finds a feasible point by itself. These characteristics make SeDuMi a valuable tool for solving our optimization problem. Specifically, we have used SeDuMi as a semidefinite positive programming tool and we also took advantage of its capability of including linear and semidefinite constraints.

Thanks to SeDuMi and the semidefinite problem definition, it was possible to determine the minimum power dissipation and the optimum values for our optimization parameters. For example, for a topology with three parallel wires, we can reduce the power dissipation with 77% at the cost of a relatively small increase of 10% of the total delay. To this purpose, first the optimal repeater size is determined, then a matching repeater is selected from a library. This approach proofs to be valuable for a chip designer in his quest for low power, high performance topologies.

The method also allows the designer to construct suboptimal problems by fixing a given optimization variable and finding the optimal results for the other variables. In the previous paragraph we discussed that the repeater was selected which is closest to the optimum value for the repeater size d with respect to power dissipation. As a next step, the wire width could be optimized by setting the repeater size to a fixed value and running the simulation script again. This exercise can be repeated for different parameters, such as the spacing or the wire width, depending on the specific needs of the designer.

An important result of this work is that our approach allows the designer to investigate a low power, high performance topology by interchanging the optimization parameters. This constitutes an important result of our research. Also, the researcher can make use of the script to find a cubic splines interpolation for the trade-off curve between power dissipation, Tdom and bus width to find values in between those found by simulation. This can be regarded as an extra feature for the designer.

In addition, we proved the accuracy of our results through comparison against Pstar, which is a Spice-like full-numerical circuit simulator [24]. First, we created a circuit description of our model that could be run by Pstar. The results are then compared against the results from our model. The error is less than 1%, which shows that our model is valid and accurate.

Further, we investigated an approach called uniform wire sizing (UWS) where we assumed uniform width w for all wires, uniform spacing s between the wires and the same size for all the repeaters d. This is in contrast with the method of wire tapering, also called non-uniform wire sizing (NUWS), were the width of the element varies over the length of the

wire. Alpert[22] has proved that the benefits from NUWS when full repeater insertion is applied are minimal. However, we observed a large difference in the results from the UWS and NUWS approaches when repeater insertion was not optimized. We did an exercise with the equations provided by Garcea [17] for repeater insertion and we observed an increase in the quality of the results. This suggest that the time dominant constant approach can be further improved by including the optimization of the repeater insertion.

Chapter 8

Future Work

We propose for future work a benchmarking against the work by Garcea et al. [17]. Specifically, it would be useful to derive an Elmore delay model that includes the wire width dependent term for the πRC element, the division of the bottom capacitance into fringing and bottom capacitances, and also includes coupling capacitance. Such a model could be analyzed analytically and compared against the model applied in [17].

It is also proposed to do an initial comparison against Bakoglu's work [9]. This comparison should include the equations that provide the optimal number of repeaters that are inserted in the given topology.

Finally, we would like to propose to take into account the inductance effect in the optimization method. Inductance is becoming more important with faster onchip rise times and longer wire lengths. For very-deep submicron (VDSM) designs inductance cannot be neglected and the distributed RC model is no longer accurate for this kind of designs. An RC model as compared to an RLC model creates errors of up to 30% in the total propagation delay of a repeater system [33]. Therefore, it becomes necessary to find new approaches to delay analysis and optimization for the next generation of chip design.

Bibliography

[1] T. Heijmen and J. van Staalduinen; Interconnect Optimization by Semidefinite Programming. Proc. SCEEE 2002, pp. 206-213, 2002.

[2] L. Vandenberghe, S. Boyd, and A. El Gamal; Optimizing dominant time constant in RC circuits. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, Vol. 17 No. 2, pp. 110-124 February 1998.

[3] L. Vandenberghe, S. Boyd, and A. El Gamal; Optimizing dominant time constant in RC circuits. Information Systems Laboratory, Electrical Engineering Department, Stanford University, Stanford CA 94305.

[4] L. Vandenberghe, S. Boyd; Semidefinite Programming. Siam Review, vol 38, pp. 49-95, 1996.

[5] L. Vandenberghe, S. Boyd; Convex Optimization. Cambridge University Press, December 2002.

[6] J.F. Sturm; Using Sedumi 1.02, A matlab toolbox for optimization over symmetric cones (Updated for version 1.05). Department of Econometrics, Tilburg University, Tilburg, The Netherlands. http://fewca.kub.nl/ sturm, October 2001.

[7] J.F. Sturm; Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. ISSN 0924-7815 PhD Thesis. August 2002.

[8] S. Zhang; A new self-dual embedding method for convex programming. Department of systems and engineering management,The Chinese University of Hong Kong, Shatin, Hong Kong. October, 2001.

[9] H.B. Bakoglu; Circuits, interconnections and packaging for VLSI. Addison-Wesley VLSI Systems Series, 1990.

[10] L.O. Chua, C.A. Desoer, E.S. Kuh, Linear and Nonlinear Circuits,New York Mc GRaw Hill, 1997.

[11] E.N. Pronotarios and O. Wing, Delay and rise time of arbitrarily tapered RC-transmission lines. in *IEEE* International convention record, part 7, pages 1-7, 1965.

[12] J.P. Fishburn, Shaping a VLSI wire to minimize elmore delay, IEEE computer society, 1994.

[13] J.F. Sturm. Similarity and other spectral relations for symmetric cones. Linear Algebra and its applications, pp. 135-154,2000.

[14] Y. Ye, M.J. Todd, and S. Mizuno, An $\mathcal{O}(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm, Mathematics of Operations Research 19,53-67, 1994.

[15] Y. Nesterov and M.J. Todd. Self-scaled barriers and interior-point methods for convex programming. Mathematics of Operations research, pp. 1-42, 1997.

[16] J.F. Sturm and S. Zhang. Symmetric primal-dual path following algorithms for semidefinite programming. Applied Numerical Mathematics, 29:301-315, 1999.

[17] G.S. Garcea, N.P. van der Meijs, R.H.J.M. Otten; Analytic model for Area-constrained optimal repeater insertion, IEEE workshop on Signal Propagation, pp. 127-130, Siena, IT, 2003.

[18] C. Moler, Numerical Computing with Matlab, http://www.mathworks.com/moler/chapters

[19] K. Banerjee, M. Amit, Power dissipation issues in interconnect performance optimization for sub-180 nm designs. 2002 Symposium on VLSI Circuits digest of technical papers.

[20] Shoichiro Nakamura, Métodos numéricos aplicados con software, 1992, Prentice-Hall Hispanoamerica S.A.

[21] W. Schilders, Hermite interpolation using harmonic averages, unpublished.

[22] C.J. Alpert, A. Devgan, J.P. Fishburn, and S.T. Quay; Interconnect Synthesis Without Wire Tapering. IEEE Transactions on computer-aided design of integrated circuits and systems, Vol. 20, No. 1, January 2001.

[23] H. Veendrick, Deep-submicron CMOS ICs: from basics to ASICS, Deventer: Kluwer, 1998.

[24] Philips ED&T/AS, Pstar User Guide for Pstar 4.2, first edition, October 2001.

[25] Philips semiconductors LTG, PcCMOS12 gcorelib, release 1.1.

[26] TSMC 0.13μm LOGIC 1p8m SALICIDE 20V/2.5 V, 1.2V /2.5V, 1.2V / 2.5V, 1.0V/3.3V, 1.2V/3.3V DESIGN RULE, Document No. TA-10B2-4001 (T-013-LO-DR-001), Rev. 1.5

[27] T. Sakuray, Approximation of Wiring Delay in MOSFET LSI, 1983 IEEE.

[28] W.C. Elmore. The transient response of damped linear systems wqith particular regard to wideband amplifiers. Journal of Applied Physics, PP.19:55-63, 1948.

[29] J. Rubinstein, P. Penfield, and M.A. Horowitz. Signal delay in RC tree networks. IEEE Transactions on Computer-Aided Design, PP. 202-211, 1983.

[30] J.P. Fishburn and A.E. Dunlop. TILOS: A posynomial programming approach to transistor sizing. Proc. ICCAD'85, PP. 326-328, 1985.

[31] L. Vandenberghe, S. Boyd; Software for Semidefinite Programming, User's Guide, Stanford University, USA, 1998.

[32] C.Moler; Numerical Computing with MATLAB; SIAM, June 2004.

[33] Y.I. Ismail and E.G. Friedman, Effects of inductance on the propagation delay and repeater insertion in VLSI circuits, Department of electrical engineering, University of Rochester, Rochester, New York.