Durham E-Theses

Modelling and analysis of crosstalk in scaled CMOS interconnects

Mahoney, Patrick Francis

How to cite:

Mahoney, Patrick Francis (1995) *Modelling and analysis of crosstalk in scaled CMOS interconnects*, Durham theses, Durham University. Available at Durham E-Theses Online: http://etheses.dur.ac.uk/5376/

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the full Durham E-Theses policy for further details.

Academic Support Office, The Palatine Centre, Durham University, Stockton Road, Durham, DH1 3LE
e-mail: e-theses.admin@durham.ac.uk Tel: +44 0191 334 6107
http://etheses.dur.ac.uk

Abstract

The development of a general coupled RLC interconnect model for simulating scaled bus structures in VLSI is presented. Several different methods for extracting submicron resistance, inductance and capacitance parameters are documented. Realistic scaling dimensions for deep submicron design rules are derived and used within the model. Deep submicron HSPICE device models are derived through the use of constant-voltage scaling theory on existing 0.75μm and 1.0μm models to create accurate interconnect bus drivers. This complete model is then used to analyse crosstalk noise and delay effects on multiple scaling levels to determine the dependence of crosstalk on scaling level. Using this data, layout techniques and processing methods are suggested to reduce crosstalk in systems

Modelling and Analysis of Crosstalk in Scaled CMOS Interconnects

Volume 1 of 1

Patrick Francis Mahoney

Thesis for qualification for degree of Master of Science
Durham University
School of Engineering and Computer Science

Sept. 1995

The copyright of this thesis rests with the author.
No quotation from it should be published without
his prior written consent and information derived
from it should be acknowledged.

8 JAN 1996

Table of Contents:

1. INTRODUCTION 6

1.1 History of Crosstalk Modelling 7

1.2 Organisation 9

1.3 Simulation vs. Measurement 11

1.4 Project Methodology 12

2. ELECTRICAL MODELLING OF INTERCONNECTIONS 14

2.1 Introduction 14

2.2 RC Line Model 14

2.3 RLC Line Model 19

2.4 Transmission Lines 21

3. CAPACITANCE PARAMETER CALCULATION 23

3.1 Introduction 23

3.2 Area and Perimeter Based Capacitance Calculation Techniques 23
3.2.1 The Lewis Technique for Coupled Interconnect Capacitance Calculation 26

3.3 Numerical Techniques for Capacitance Calculation 29

3.4 Matthaei's Method for Coupled Interconnect Capacitance Calculation 32
3.4.1 Introduction to Mattahei's method 32
3.4.2 Implementation of Matthaei's Method 35

3.5 Implementation of Matthaei's Method in the Absence of a Reference Plane 39

3.6 Comparison of Accuracy for Calculation Methods 40

4. INDUCTANCE AND RESISTANCE PARAMETER CALCULATION 44

4.1 Introduction 44

4.2 Inductance Parameter Calculation 45

4.3 Resistance Parameter Extraction 48
4.3.1 DC Line Resistance 48
4.3.2 AC Line Resistance 49
4.3.3 Dielectric Loss 51

5. INTERCONNECTION SCALING 52

5.1 Introduction 52

5.2 Ideal Scaling 53

5.3 Quasi-Ideal Scaling and Other Methods 55

5.4 Variation of Track Capacitance with Scaling 57
5.4.1 Variation of Capacitance with Width 58
5.4.2 Variation of Capacitance with Height 59
5.4.3 Variation of Capacitance with Thickness 60
5.4.4 Variation of Capacitance with Separation 62
5.4.5 Conclusions on Scaling of Interconnect Capacitance 63

5.5 Scaled Track Geometries 64

6. TRANSISTOR SCALING AND MODELLING 66

6.1 Introduction 66

6.2 Transistor Scaling 67

6.3 Transistor Scaling Theory 68

6.4 HSPICE Transistor Model 70
6.4.1 History of SPICE 70
6.4.2 Transistor Modelling Using HSPICE 72

6.5 Comparison of Scaled Transistor Models 75

7. CROSSTALK DELAY AND NOISE 79

7.1 Introduction 79

7.2 Introduction to Crosstalk Noise 80
7.2.1 Analysis of Crosstalk Noise on Parallel Lines in Scaled Bus Structures 83
7.2.2 Analysis of Crosstalk Noise at Sampling Points along a Distributed Line 87
7.2.3 Evaluation of the Contribution of Inductance to the Crosstalk Signal 89
7.2.4 Effect of Temperature on Crosstalk Noise 92
7.2.5 Analysis of Crosstalk Noise in Scaled Bus Structures 94

7.3 Introduction to Crosstalk Delay 99
7.3.1 Analysis of Crosstalk Delay in Scaled Bus Structures 102

7.4 Methods of Reducing Crosstalk 105

8. CONCLUSIONS 109

8.1 Electrical Modelling of Interconnections 109

8.2 Capacitance Parameter Calculation 109

8.3 Inductance and Resistance Parameter Calculation 110

8.4 Interconnection Scaling 110

8.5 Transistor Scaling and Modelling 110

8.6 Crosstalk Delay and Noise 110

APPENDIX H: IMPLEMENTATION OF MATTHAEI'S METHOD IN PASCAL

Declaration:

I declare that this work is completely my own, that it comes from no other outside source and that no portion of this work has previously been submitted for a degree in this or any other university.

Statement of Copyright

The copyright of this thesis rests with the author.
No quotation from it should be published without
his prior written consent and information derived
from it should be acknowledged.

List of Tables and Figures:

Fig. 2.1. A lumped RC model for interconnection. 15

Fig. 2.2. A distributed RC line model approximated as a cascade of π-sections. 16

Fig. 2.3. Transient Response of lumped and distributed RC models. 18

Fig. 2.4. Circuit used to model 0.35μm inverter driving an RLC line model. 20

Fig. 2.5. Transient Response of lumped and distributed RLC models. 21

Fig. 3.1. Comparison of field lines for parallel-plate and Schwartz-Chistoffel transformation methods of calculating capacitance. 25

Fig. 3.2. Even and odd mode field line configurations in coupled microstrip lines. 26

Fig. 3.3. Even and odd mode capacitance models. 27

Fig. 3.4. (a) A two dimensional metal strip that extends into and out of the paper. (b) An even charge basis function for a fixed charge along the length of the strip. (c) The corresponding odd charge basis function.[15] 34

Fig. 3.5. Pseudo-code description of Matthaei's method for capacitance parameter extraction. 39

Fig.3.6. A comparison of the results from multiple methods for extracting interconnect capacitance.41

Fig. 3.7. Comparison of total line capacitance using different techniques. 43

Fig. 4.1. Basic RLC line model. 44

Fig. 4.2. Skin depth as a function of frequency for aluminium at room temperature. 50

Fig. 5.1. Scaling of Global Interconnections. 54

Fig. 5.2. Explanation of the terms used in the graphs in this section. 58

Fig.5.3. Variation of Self and Mutual Capacitance with Interconnect Width. 59

Fig.5.4: Variation of Self and Mutual Capacitance with Height (substrate thickness). 60

Fig.5.5. Variation of self and mutual capacitance with thickness. 62

Fig.5.6. Variation of self and mutual capacitance with separation. 63

Fig. 5.7. Table showing the physical dimensions of track parameters at different scaling levels. 65

Fig.6.1. Influence of different MOS-device scaling models. 69

Fig.6.2. Comparison of HSPICE process models for the falling transition of a 0.50μm inverter . 74

Fig.6.3. Comparison of the transition times of scaled transistors. 76

Fig.6.4. Comparison of the switching speeds for a chain of scaled inverters. Graph A. shows a 0.35μm inverter chain, while graphs B. and C. shows the switching of 0.50μm and 0.75μm devices respectively. 77

Fig. 7.1 Simple example of crosstalk coupling on a 2-bit signal bus. 80

Fig. 7.2: A simplified representation of the circuit in fig. 7.1 while the signal line inverter is switching. 81

Fig. 7.3: Graph illustrating signal coupling between an active and an inactive line. 82

Fig. 7.4. Coupled crosstalk signal on each signal line in an 9 bit bus. 84

Fig. 7.5. Illustration of the nine bit bus. (a) the complete nine bit bus, and (b), with two conductors removed. 86

Fig.7.6. Comparison of the coupled voltage at various point along a line at different scaling levels. 88

Fig. 7.7. Comparison crosstalk signal noise for RC and RLC line models on a 0.35μm process. 90

Fig.7.8. Comparison of the output slope of an interconnect of RC and RLC line models 91

Fig.7.9 Comparison of coupled crosstalk voltage against simulation temperature 93

Fig.7.10. Comparison of crosstalk noise against scaling for a 1cm long 8-bit bus. 95

Fig. 7.11. Table summarising key features in the graph in figure 7.12. 96

Fig. 7.12. Comparison of crosstalk noise against scaling for a 100μm long 8-bit bus. 97

Fig. 7.13. Comparison of crosstalk noise against scaling for a 10μm long 8-bit bus. 98

Fig. 7.14 Illustration of crosstalk delay. 100

Fig. 7.15. Example of crosstalk delay showing difference in the slopes between the delayed and undelayed signals. 101

Fig.7.16. Comparison of crosstalk delay against scaling for a 1cm long 8-bit bus for three processes: (a.) 0.35μm, (b.) 0.50μm and (c.) 0.75μm. 104

Fig.7.17. Comparison of crosstalk delay against scaling for a 100μm long 8-bit bus for three processes: (a.) 0.35μm, (b.) 0.50μm and (c.) 0.75μm. 104

Fig.7.18. Summary of 50% crosstalk delay measurements and percentage margin that crosstalk delay is to slope for different interconnect scaling and length values. 105

Fig.7.19. Different layout techniques that can be used to reduce crosstalk effects: (a) normal 3-bit bus, (b) stripline configuration, (c) shielded interconnects, and (d) shielded stripline (pseudo-coaxial) configuration. 106

Fig.7.20. Comparison of layout techniques used to reduce crosstalk effects. 107

1. Introduction

For VLSI and ULSI chip design, interconnections have a greater importance than in earlier technologies. Today, a chip no longer contains a single circuit, but may contain a large part of an entire system - for example, an entire computer on a chip. For such complex designs, there is less flexibility available to the engineer than in the design of simple circuits and this results in interconnections that are more than just the nuisance level of added delay, additional power consumption and higher noise. The layout of interconnections can change the entire architecture and operation of a VLSI system . Thus, the designer requires accurate methods for modelling interconnections in order to optimise the design of a modern chip.

Additionally, the scaling of devices and interconnect dimensions in VLSI has resulted in interconnections playing a more pervasive and dominating role in circuit performance. As CMOS device dimensions have been reduced resulting in devices with faster transition times, the RC delay associated with the interconnections between these devices has increasingly begun to dominate overall circuit performance. In addition, analogue signal effects such as crosstalk are a concern as they can lead to delay and logic hazards. Dynamic circuitry which is useful in high performance circuit is particularly susceptible to crosstalk logic errors.

A poor understanding of crosstalk can lead to overly conservative design rules resulting in poor performance, or it can lead to sporadic logic errors which may only be triggered by certain logic combinations and may be difficult to detect. Thus an accurate simulation model is essential for efficient and reliable circuit designs. This paper presents a general method for the creation of scaleable model for coupled lossy transmission bus structures. This model is then used to derive general trends in crosstalk as designs are scaled in the submicron region.

1.1 History of Crosstalk Modelling

Crosstalk noise in high-frequency designs is not a recent effect. It has been well-documented in microwave and RF literature for decades. The results for microstrip lines documented in these papers do not directly apply to CMOS because they consider very thin "lossless" lines (negligible resistance) and older studies neglected the thickness of the tracks. Investigation into crosstalk for integrated circuit design began with several papers within a few years of each other describing crosstalk in ECL and TTL logic circuits[1,2,3]. In all of these papers, the coupling of signals between two parallel lossless conductors were considered and the solution to the problem of terminating lines was considered. The main limitation of these papers with respect to CMOS circuits is that they consider lines with a negligible resistance only. The resistance of the metal and polysilicon tracks in a CMOS system, however, is significant and this resistance plays a significant role in crosstalk coupling.

The effects of crosstalk on coupled "lossy" (significant resistance) interconnections in a thin-film package were studied by Isaac and Strakhov[4]. In this paper, a general analytical model for coupled transmission lines is presented, but little consideration is made as to the driving devices and the model is not directly applicable to CMOS structures due to the parameterisation scheme. An analysis of wafer-scale transmission lines using weakly-coupled slightly-lossy interconnections was made by Kim and McDonald[12] and simulations and measurements of interconnections on tape automated bonding packages was made by Su, Raid, Elshabini-Raid and Poulin[5]. Similar to the work by Isaac and Strakhov, the analysis in these papers only considers the general case of two parallel interconnections. Although all three of these papers provide a detailed analysis of delay and crosstalk in parallel transmission lines, they both require extensive computer calculation which is impractical in VLSI systems with thousands or millions of interconnections to be considered.

More recent work has focused on methods which are more computationally efficient to speed up crosstalk and delay analysis in VLSI and ULSI systems. One new technique developed by Pillage and Rohrer[6], referred to as the asymptotic waveform evaluation (AWE) technique, approximates the waveform response of general linear circuits using a lumped model. This technique was adapted for use in estimating the response of linear lossy coupled transmission lines by Tang and Nakhla[7]. One further adaptation was proposed by Xie and Nakhla to allow the method to deal with non-linear terminations. The advantage of the AWE technique over the more exacting analysis of the methods described in the previous paragraph, is that it is comparatively much faster, although it is also less accurate.

While the AWE technique is useful for approximate evaluation of large networks of interconnections at lower frequencies and exact transmission line solutions are useful for understanding the coupling mechanism in detail at high frequencies, a more general method of examining the crosstalk response of an interconnection network is through the use of existing circuit simulation software (such as SPICE) with accurate interconnect models. Although simulation with SPICE is more time consuming than other methods such the AWE technique, it is the most commonly used method of simulating complex models and has the most complete models for devices.

One SPICE model proposed by Tripathi and Rettig approximates the response of a multiple coupled system of lossless interconnects using a system of uncoupled lines and linear dependent current and voltages sources[8]. This model was later adapted by Tripathi and Bucolo[9] to model multi-level parallel and crossing lossy coupled interconnections and by Papaioannou, Dimopoulos and Avarisiotis[10] for simulation of off-chip interconnections. Other models, such as that described by Chang, Chang, Oh and Lee[11] use SPICE subcircuits and macros to create complex general models which can be adjusted for different conditions.

The most serious deficiency of all of these models, with the possible exception of the model suggested by Chang[11], is that they only use limited approximations for the devices driving the interconnect lines. Since the device characteristics have a serious impact on the crosstalk signal, these imprecise models for the devices can result in large inaccuracies. In addition, none of these methods have detailed the complete model including the modelling of the devices and the extraction of the interconnect characteristic parameters. The model used in this project uses an distributed RLC model and includes accurate models for the device drivers.

1.2 Organisation

This first chapter introduces the concepts used in the paper and establishes the importance of this investigation. Prior work in the subject areas of interconnect modelling, parameter extraction, scaling and crosstalk and delay modelling are also mentioned. Although each of the sections will later mention previous work in the field in more detail, looking at what has already been done is a logical way to begin to present the subject of this thesis.

Chapter two examines various line models that can be used to represent an interconnect line by starting with the simplest model and eventually expanding to more complex models that more accurately describe the behaviour of interconnects under different drivers and signal inputs. A comparison of the accuracy and complexity of each of these models is then presented with a detailed description of the interconnect model used.

The third chapter analyses different methods of extracting capacitance parameter information from a physical layout. It starts with an examination of the simplest methods and adds greater complexity while increasing the accuracy of the method. The method used to compute capacitance for the project is then presented and compares the accuracy and the simulation time of this method against other algorithms that have been detailed in academic literature.

The methods used to calculate resistance and inductance parameters are examined in the fourth chapter. A description of the method used to calculate inductance from the capacitance matrix derived through the use of one of the methods described in the third chapter is presented. An explanation of the method used to determine the resistance of a track from physical dimensions and a discussion and evaluation of the contribution of each of the components of the overall track resistance is made.

In the fifth chapter, the scaling of interconnect physical dimensions is considered. The theories of ideal and quasi-ideal scaling are explained and the problems with each method are presented. A series of graphs showing the variation of self and mutual capacitance with changes in physical dimensions are shown and conclusions are made regarding the dependence of the two components of capacitance on geometry. Finally a summary of the scaled interconnect dimensions used in simulations at each certain process are given.

The sixth chapter looks at modelling and scaling of the MOSFET devices used to drive the interconnections. The theory of CMOS device scaling is explained and the advantages and difficulties with scaling are analysed. The history and the capabilities of the circuit simulation package are described and the method used to create the models used in the simulation is illustrated. Finally, a comparison of the models under loaded and unloaded conditions is presented to evaluate the speed switching improvement with scaling.

The results of the crosstalk delay simulations are presented in the seventh chapter. An example of crosstalk noise is described and the dependencies of crosstalk on certain circuit parameters is established. Crosstalk noise on parallel tracks in a bus is analysed for changes in temperature, scaling, and length. The peak crosstalk voltages at different points along a distributed interconnect are compared and the concept of shielding interconnects with other tracks is introduced. The concept of crosstalk delay in a bus and

the problems that it may lead to in the design of synchronous circuits is presented and described through the use of an example. The results of simulations showing the variation of crosstalk delay with scaling are presented. Lastly, methods that can be used to reduce crosstalk delay are described and evaluated.

Conclusions are made on the each of the results described and a final summary is made of the project is made in chapter eight. An appendix follows which lists the PASCAL computer program used to determine the resistance, inductance and capacitive components from the line geometries

1.3 Simulation vs. Measurement

The primary goal of the project is to analyse the effect that scaling of interconnections and devices has on the immunity of circuits to crosstalk noise. The most obvious method of performing this task would be to construct multiple scaled structures using a variety of scaled devices, interconnection layouts, and input waveforms to determine worst case conditions and inputs, analyse the data gathered and present conclusions on the significance of the problem and strategies that can be used to overcome these obstacles.

A small fraction of the previous research examining the effect of crosstalk on silicon systems have used methods similar to this[5,12], but the majority have used circuit models that simulate the effect of crosstalk rather than physically measuring it. This is primarily for four reasons:

- Cost: The cost required to fabricate the devices with the number of variables that often need to be considered can make the cost of manufacturing the test structures prohibitive in terms of time and money. This becomes especially true when the systems under investigation are still in the developmental stage and have limited availability outside of the companies developing them.

- Accuracy: Accurate measurement of the exact region that is desired can be difficult using a physical construction. For example, to measure the crosstalk of an on-chip interconnect, signal noise from other sources such as from coupling at the pins and in the probe wires must be considered and compensated for.

- Speed: In terms of the total time required by a project, it is much faster to develop and construct a model of a complex integrated circuit using existing simulation software than it is to actually design and fabricate the physical device.

- Flexibility: Once a model is constructed, it is easier to adjust the physical parameters of that model (for example, the wiring resistance) in a simulation than to manufacture a new device. In addition, in a simulation varying a physical parameter such as the thickness of a wire at small intervals requires only a minor adjustment to the simulation whereas on a physical system this would require the manufacturing of multiple devices.

Based on these reasons a simulation method was chosen over construction of multiple systems and the physical measurements of circuit characteristics.

1.4 Project Methodology

The project was divided into sections which appear in this paper as chapters. Each of these sections had an exact goal and in this paper each chapter concludes with a summary of results obtained. Every section added built upon the results of the last to create the complete model which was then used in simulations to acquire the crosstalk results. The sections are:

- Determination of an accurate, computationally efficient model for an RLC line model.

- Determination of an accurate and efficient method for the calculation of the capacitance parameters for the RLC model.

- Evaluation of a method for the efficient calculation of the inductance and resistance parameters for the RLC model.

- Determination of scaling dimensions for the interconnections.

- Creation of accurate device models for each of the scaling levels.

- Simulation of the complete model at each of the scaling levels to evaluate crosstalk noise and delay

2. Electrical Modelling of Interconnections

2.1 Introduction

The simplest model for a wire or an integrated circuit track is a short circuit in which the wire has no resistance or capacitance and thus adds no delay to the circuit. In an integrated circuit, this model is only a reasonable estimate when the device resistance and capacitance completely dominate over the wire resistance and capacitance and when the switching time of the driving device is slow. Only in the earliest days of integrated circuit design was this the case.

When the device delay is much larger than the RC delay* of the line, but the capacitance of the interconnect line is significant in regards to power consumption and device switching delay, then the line may be modelled as a lumped capacitor. In this case, the finite resistance of the line must be negligible compared to the transistor resistance when the device is in saturation mode. This model was used in earlier process technologies, but more recent processing techniques have increased the interconnect resistance and capacitance and recently more accurate models of integrated circuit tracks have been required.

2.2 RC Line Model

The simplest line model of a IC interconnection that considers both the resistance and the capacitance of the line is a lumped resistor and capacitor which form a basic low-pass

---

* The term "RC delay" refers to the amount of time required to charge up the interconnect capacitance through the combined resistance of the driver and the interconnect.

filter. This model is as shown in Fig 2.1 where R and C are the line resistance and capacitance per unit length and l is the wire length.

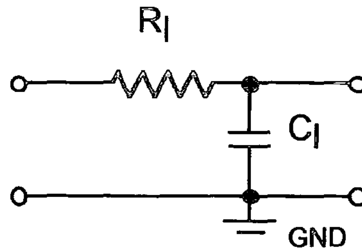

Fig. 2.1. A lumped RC model for interconnection.

This model illustrates the basic limitations of interconnections: they have a finite resistance and a finite capacitance. These two properties lead to a delay in signal propagation while the line capacitance is charged through the resistance. It also illustrates that an interconnect line is a low-pass filter. The size of the resistance and capacitance are both dependent on the length of the interconnect line, l. More specifically the resistance of the line is determined by the geometry of the line (length, width and thickness) as well as the resistance of the conducting material while the capacitance is determined by the geometry of the line, the dielectric material surrounding the wire, and the distance of the wire from other conductors.

In the lumped model, the resistance and capacitance are "lumped" at one node. More realistically, however, the capacitance and resistance aren't lumped at one specific point in the centre of line, but are distributed along the length of the conductor. In this case of a distributed RC model, this distribution of resistance and capacitance is represented by a cascade of the π-sections. This is shown in figure 2.2. These sections are referred to "π-sections" due their resemblance to the Greek letter pi.

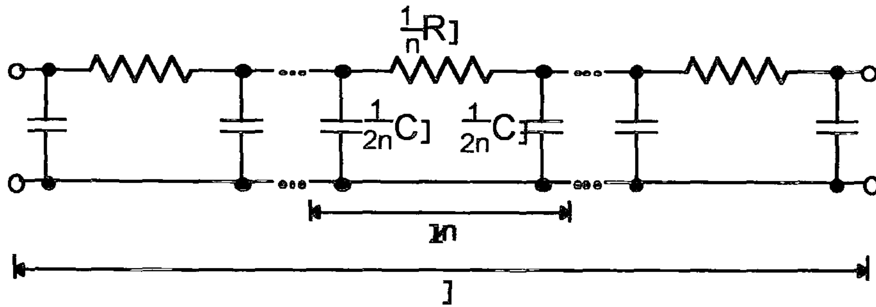

Fig. 2.2. A distributed RC line model approximated as a cascade of π-sections.

In this case, the number of subsections is given by n, and the value of each discrete resistor and capacitor are divided by the total number of sections. This creates a more accurate representation of the line in that the resistance and capacitance are interspersed. As would be expected, the more subsections there are in this model, the more realistic is the representation to that of a real wire. As the number of these subsections tends towards infinity when the total line length is fixed, the cascade becomes governed by the equations:

$$\frac{\partial I}{\partial z} = -C\frac{\partial V}{\partial t} \quad \text{and} \quad \frac{\partial V}{\partial z} = -RI \qquad (2.1)$$

These can be combined to produce the diffusion equations describing voltage and current signal propagation through an RC line:

$$\frac{\partial^2 V}{\partial z^2} = RC\frac{\partial V}{\partial t} \quad \text{and} \quad \frac{\partial^2 I}{\partial z^2} = RC\frac{\partial I}{\partial t} \qquad (2.2)$$

In equation 2.2, I and V are signal current and voltage, respectively, with respect to a ground plane and the co-ordinate z measures distance along the track. Solutions to the diffusion equation are well known from the theory of heat conduction and so the behaviour of such a circuit is well understood as long as R and C are voltage-independent.

Although this solution is useful in the initial stages of VLSI design, it is not compatible with circuit simulators. In order to model an interconnect accurately in these packages, a compromise between the number of sections to include and the amount of time required to simulate the circuit must be evaluated. In other words, a compromise between accuracy and complexity needs to be made. Sakurai analysed accuracy of distributed RC lines using multiple sections and showed that a three-section π model had a relative error of typically less than 3%[13].

Other circuits may be used in place of the π model described above. Other commonly used circuits are the L and the T ladder circuits named for the approximate shape of their unit blocks. Rajput proposed a non-linear form of cascaded circuit to describe a distributed RC line[14], but this circuit has several limitations in addition to being unnecessarily complex and the π, L and T configurations are more commonly used. Sakurai showed that the widely used L configuration is a poor approximation which may be as much as 30% in error from a true distributed line, even when as many as three stages have been added. The π and T circuits produce nearly identical results, but the π circuit is preferred because the T circuit contains two nodes per cascade while the π circuit has only one. Since the computational time of the HSPICE circuit simulation package is strongly dependent on the number of nodes in the circuit, the π configuration will give nearly identical results to the T, but can be calculated more efficiently.

The graph shown below in figure 2.3 illustrates the difference in response by that is found when multiple cascades are added in series. The circuits are simulated using the HSPICE circuit simulator (described in section 6.4). The circuit uses an HSPICE level 3 model of 1.0μm inverters which are driving a load of 1pF and 50Ω and the input of another 1.0μm inverter. There are three driving inverters in series at the input to create a realistic slope on the input of the line model from the much faster 0.5ns input to the circuit model. The devices in these inverters use 1μm minimum lengths and widths for

the n-channel devices and minimum length and 3μm wide p-channel devices. The voltage source for the inverters is 5V and the operating temperature is 20°C.

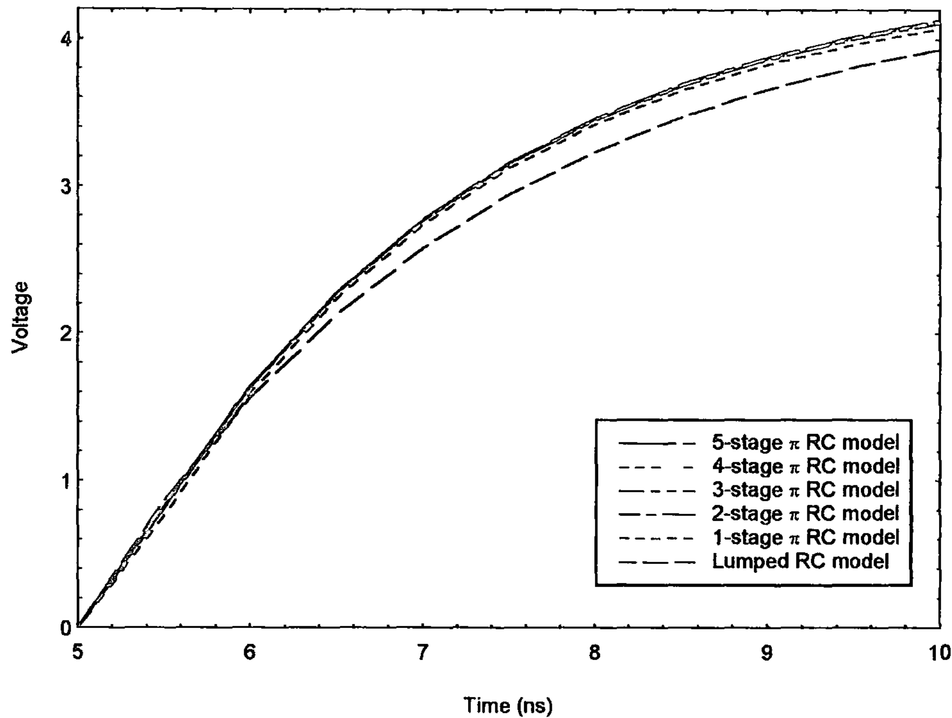

Fig. 2.3. Transient Response of lumped and distributed RC models.

From this graph it can be seen that, while significant variations are evident between the lumped and the one stage models, and some difference exists between the two stage model, the transient response of the other three models are nearly identical. In terms of HSPICE simulation time, however, for a system of eight interconnects the five stage π RC model requires more than one and half times the computation time and results in 32 more lines of code than the comparable three-stage π RC model of the same system.

The graph in figure 2.3 agrees well with results published by Sakurai as well as Mey[15]. Bakoglu[16], the author of "Circuits, Interconnections and Packaging for VLSI", commenting on the results obtained by Sakurai, noted that the accuracy of the models is improved as the number of cascades in the system is increased, but concluded that "the

accuracy of the [π and T] models is improved as n increases, but it saturates at about three or four ladder steps." where the value n mentioned is the number of cascaded circuits. Similarly, Goel[17], commenting on the results reported by Mey said, "In fact, there is negligible difference between the results for the 5-stage and 10-stage ladder networks". After simulations were performed on different systems using various values of load and line capacitances which resulted in results similar to those published, it was decided that a three-stage π network would be an adequate compromise between complexity and accuracy.

2.3 RLC Line Model

The inclusion of inductance into an RC line model has two primary effects on the propagation of the signal through the line. It can introduce the problems of ringing and overshoot which are not found in RC models. Ringing and overshoot can lead to logic errors and can result in slower transition times since the output will take longer to settle. Inductance also creates wave propagation and transmission line effects which are quite different from the diffusive propagation found in distributed RC lines. Transmission line effects can result in reflection noise and can introduce a fundamental limit to how fast a signal can travel down the line.

Inductance becomes important in interconnect simulation if the line is long and as a result has a large inductance, or if the transition times are sufficiently fast that $L dI/dt$ becomes significant. At this point, the current in the line cannot be increased indefinitely by reducing the source resistance of the driver due to the effect of the inductors which resist changes in current by generating a reverse electromotive force. This limits the amount of current in the line and a fundamental limit on waveform propagation is introduced based on the amount of time this limited current can charge up the capacitance in the line. The line is no longer equipotential, but now accommodates a travelling wave.

This explanation of transmission line effects can also be shown quantitatively using equation (2.1) modified to include inductance:

$$\frac{\partial I}{\partial z} = -C\frac{\partial V}{\partial t} \quad \text{and} \quad \frac{\partial V}{\partial z} = -RI - L\frac{\partial I}{\partial t} \qquad (2.3)$$

which combine to give:

$$\frac{\partial^2 V}{\partial z^2} = RC\frac{\partial V}{\partial t} + LC\frac{\partial^2 V}{\partial t^2} \qquad (2.4)$$

and similarly for current:

$$\frac{\partial^2 I}{\partial z^2} = RC\frac{\partial I}{\partial t} + LC\frac{\partial^2 I}{\partial t^2} \qquad (2.5)$$

The second derivative in time indicates that the RLC line supports the propagation of a wave rather than simple diffusion with the coefficient of the second derivative determining the velocity of propagation. Similar to (2.2), exact solutions exist for these equations, but for use in circuit simulation packages an approximation must be made in terms of the number of sections to be included in the model.

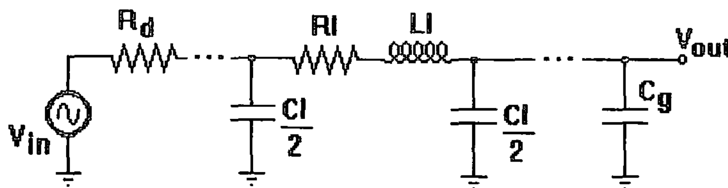

Fig. 2.4. Circuit used to model 0.35µm inverter driving an RLC line model.

In order to evaluate the number of sections required to accurately simulate an RLC model, a model of 0.35µm inverter modelled as an AC voltage source in series with a resistor, R_d, as shown in figure 2.4, was used. This simplified model was used because HSPICE models for 0.5µm and 0.35µm transistors were not completed at the time these

simulations were performed. An rising input slope of 0.5ns was used as the switching delay of the inverter and 8.35kΩ was used as the device resistance of a minimum width and length NMOS transistor. A capacitor with a value of 20fF was used to represent the gate capacitance. The interconnect being modelled was 1cm long, 2μm wide, 0.5μm thick and the dielectric thickness (height) was 1μm.

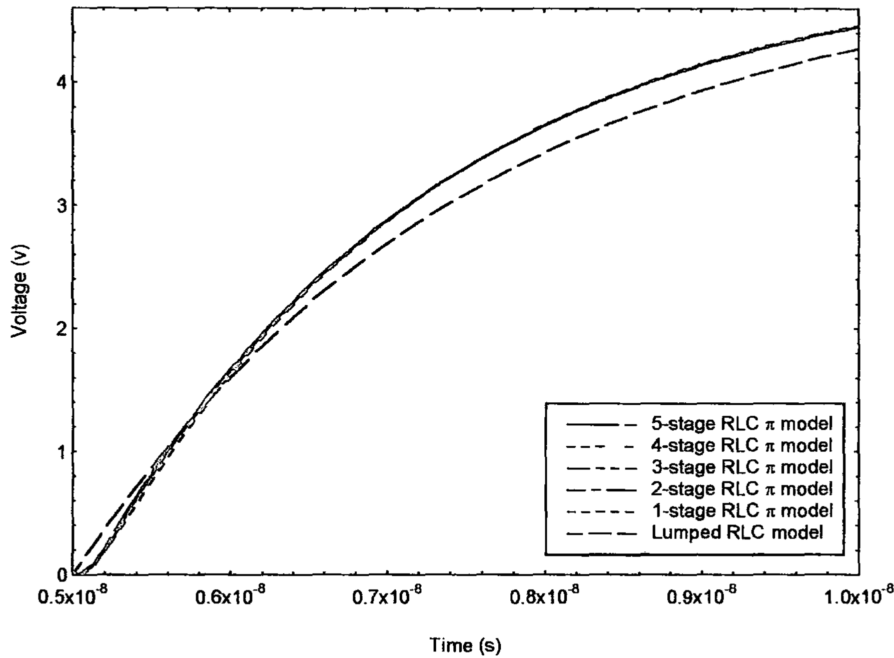

Fig. 2.5. Transient Response of lumped and distributed RLC models.

Essentially this graph shows similar results to those shown in figure 2.4. It can be seen that the three stage RLC interconnect model produces similar results to the more complex four and five stage models while requiring less computational time to simulate. The most significant difference between the RC and RLC models is that the differences between the models are less apparent. Using the same logic as was described in choosing the number of stages of the RC model to use in simulations it was concluded that 3 stages provided reasonable accuracy. A comparison between the contribution of inductance to the output waveform of a coupled crosstalk will be evaluated in section 7.4 using a more accurate model.

2.4 Transmission Lines

The label transmission line could be applied to any line which transfers electromagnetic energy between two points in a circuit or communication system. In this section, the term transmission line specifically refers to the lossy transmission line element which is provided for use in modelling high frequency signals within HSPICE (version H92). The transmission line model included within the HSPICE package conveniently solves many of the problems discussed in the previous two sections as well as all of the difficulties associated with parameter extraction that will be covered in chapters 3 and 4. It reduces the problem of accuracy by automatically calculating the number of circuit sections that should be included. It is capable of calculating the R,L,C, and G circuit parameters and will either calculate them based on the physical dimensions of the interconnect, or will allow them to the pre-determined and inputted. It is also capable of supporting three levels of dielectric and can simulate a stripline ground plane configuration. Finally, the model provided includes the ability to automatically simulate crosstalk between conductors. In essence, the model provided within the program seems to provide the ideal tool for interconnect simulation.

Unfortunately, several restrictions are imposed by HSPICE and these limitations greatly reduce the usefulness of this element for analysing crosstalk. One major limitation of this model is that it only supports a maximum of five conductors. Additionally, these conductors can't be stacked vertically and must be made of the same material. There is also an error built into the calculation of the number of stages which can result in more ringing in the simulation than is actually present. The transmission line element calculation takes longer than the three-stage π RLC model described in the previous section. Finally, this included element is only available within the latest version of HSPICE. This reduces the value of any developed method for analysing crosstalk to those who have access to the HSPICE package (Inmos presently use a circuit simulator called ST-SPICE which doesn't currently support this HSPICE element).

3. Capacitance Parameter Calculation

3.1 Introduction

Any two isolated conductors form a capacitor with a finite value of capacitance between them. The value of the capacitance is the constant of proportionality between charge and voltage on each of the two conductors. Capacitors are essential to nearly every circuit application in electronics. They are used for waveform generation, filtering, in the blocking and bypass of signals, and as integrators and differentiators. Although in CMOS VLSI circuits the gate capacitance creates the inversion layer necessary to switch the devices on and off, unwanted capacitance can also be the cause of several problems.

Capacitive coupling between tracks can lead to RC delays between devices, increased power dissipation and logic errors caused by crosstalk. As minimum feature sizes decrease and chip dimensions increase, these problems will become more prominent and will require the development of accurate tools for estimating capacitance in order to anticipate its effects. Research in the development of methods of capacitance parameter extraction in interconnects has led to two schools of thought. The first applies analytical formulae for simplified cases to obtain fast calculations at the expense of limited accuracy and minimal flexibility. The second school uses more rigorous mathematical techniques such as finite-element and boundary-element methods that are more flexible about conductor layout and are capable of more accurate calculations at the cost of extensive computation time and memory requirements. Other methods lie between these two extremes but all of them strike a compromise between accuracy, flexibility and computational resources.

3.2 Area and Perimeter Based Capacitance Calculation Techniques

The most basic of the analytical methods considers each conductor to be an isolated parallel-plate capacitor in which distortion of the electric field due to edge effects is ignored and in which the bottom plate of the capacitor is an infinite plane. For this method to be reasonably accurate, the following must be true:

1. The conductor must be of negligible thickness. In other words, the width of the conductor must be much greater than the thickness. (W >> t)

2. The ratio of the separation of the conductor from other conductors versus the height of the conductor over the reference plane must be much greater than the width. (S/H >> W)

3. The ratio of the width of the conductor over its height must be much greater than one. (W/H >>1)

To summarise these three conditions in one rule, the conductor must be thin, much closer to the reference plane than it is to other conductors and must be have a width much greater than its height. If any of these three conditions are not met then fringing effects will become significant and the result will be increasingly inaccurate depending on how badly these conditions are broken.

A better approximation of a interconnect using an analytical formula is to use the Schwartz-Christoffel[18] transformation on the conductor which effectively transforms the edges of the base of the conductor into two point-sized wires†. This technique takes into account the fringing fields from the top of the conductor as well as the distortion of the electric field at either end due to edge effects. This formula, however, underestimates

---

† A point-sized wire in two dimensions is formed by the intersection of an infinitely thin wire with a plane.

the capacitance of a track because it fails to take into account the thickness of the conductor.

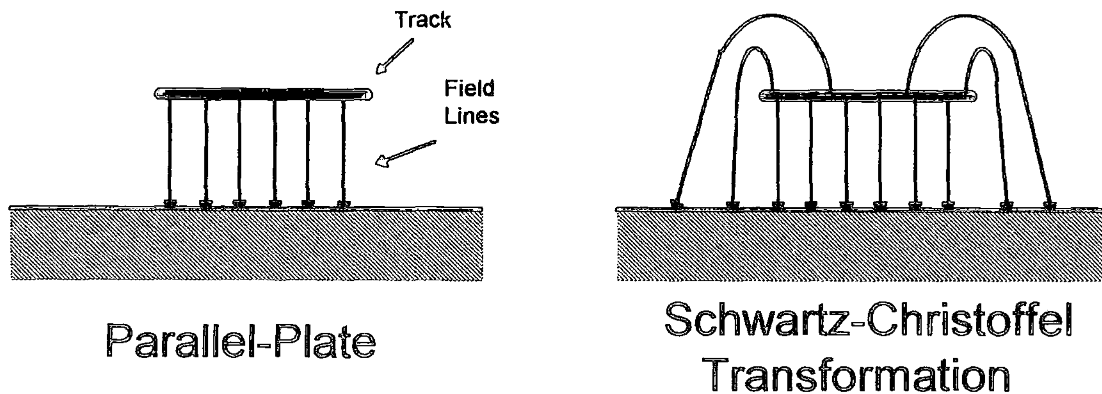

Fig. 3.1. Comparison of field lines for parallel-plate and Schwartz-Chistoffel transformation methods of calculating capacitance.

When research into capacitance calculation of microstrips first began, it was focused entirely on the microstrips used in microwave circuits. In older microwave circuits, the width was in fact much greater than the thickness and the assumption that the conductors be of negligible thickness was justified. In certain circumstances, when the thickness was comparable to the width, a method called effective width was used to compensate for the fringing field from the top and sides of the conductor[19]. By adjusting the effective width of the conductor for certain values of thickness, the requirement of negligible thickness in both the parallel-plate and Schwartz-Christoffel methods can be effectively ignored with little or no loss in accuracy.

While the parallel-plate formula and the use of effective width historically form the basis of many of the analytical techniques that were developed for use in microwave circuits, more recent methods have been developed specifically for microelectronics which don't have limitations on thickness, width or separation and can calculate the coupling

capacitance between neighbouring conductors as well. These methods consider the simplest case of two symmetric conductors and apply these results to more general cases. Although there are several methods which are generally employed, the simplest and most convenient is to describe the wave propagation of a signal along a coupled pair of symmetric lines as the sum of an even and odd mode of propagation.

3.2.1 The Lewis Technique for Coupled Interconnect Capacitance Calculation

One treatment which used even and odd modes of propagation and which is directly applicable to microelectronic structures was described in a paper by E.T.Lewis[20] which builds on earlier formulas used in microwave circuits developed by K.C.Gupta[21]. The method developed by Lewis is based on the analysis of the propagation modes supported by two adjacent coupled microstrips. If these two lines are symmetric then the supported modes can be reduced to an even and an odd mode corresponding to an even and odd symmetry of field lines as shown below in figure 3.2. In a typical VLSI chip all of the interconnects usually have the same dielectric completely surrounding the interconnects and this simplifies the analysis of the system compared to typical microwave circuits which have two or more layered dielectrics.

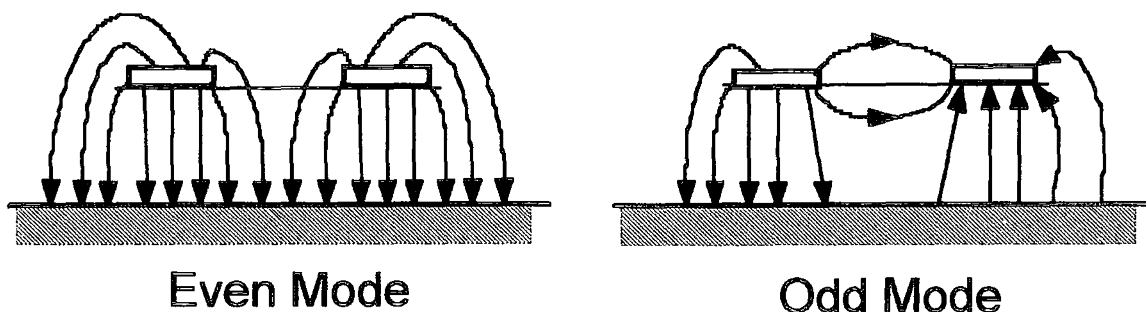

Fig. 3.2. Even and odd mode field line configurations in coupled microstrip lines.

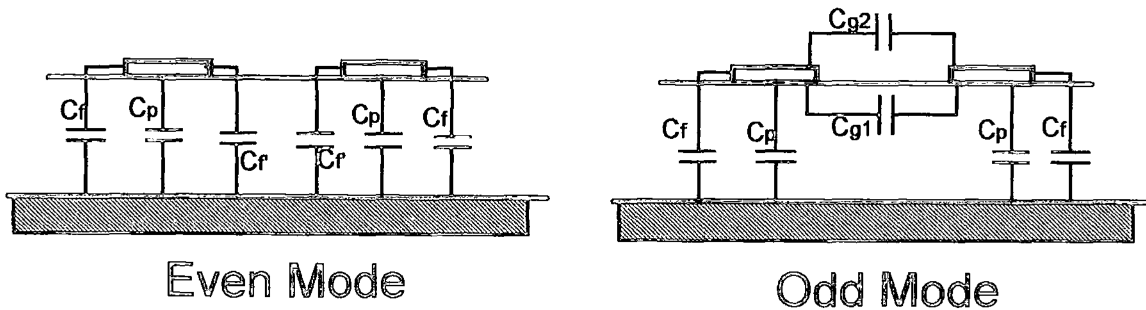

Fig. 3.3. Even and odd mode capacitance models.

Figure 3.2 shown above illustrates the field lines configurations for the even and odd modes of propagation. In the even mode, the lines have the same voltage (+V or -V) and thus the field lines are between the conductors and the plane of metallisation (illustrated by the cross-hatched lines). In an odd mode, the conductors each have different voltages so the field lines are split between the ground plane and the other conductor. This can also be seen by examining the capacitance models describing these two modes which are shown in figure 3.3.

The general method described by Lewis consists of dividing the total capacitance of each conductor into its component parts: the coupled capacitance between the two conductors and the self capacitance between each conductor and the ground plane. These two capacitances can then be divided further by considering the field components of the two propagation modes. As is shown in figure 3.3, the total capacitance of the even mode, C_{et} is:

$$C_{et} = C_p + C_f + C_{f'} \qquad (3.1)$$

and it can be noted that there is no mutual capacitive component of the even mode. For the odd mode, the total capacitance, C_{ot}, is:

$$C_{ot} = C_S + C_m \qquad (3.2)$$

$$\text{where } C_S = C_p + C_f \quad \text{and} \quad C_m = C_{g1} + C_{g2}$$

Once the total capacitance has been split into its component elements, the actual values must be determined. C_p is simply the parallel plate capacitance between the strip and the ground plane. C_f is the outer fringe capacitance and can be found by working out the Schwartz-Christoffel transform of one of the strips, subtracting the parallel-plate component and then dividing by two (to obtain only one half of the fringing component). The modified inner fringe capacitance C_f' can be found by dividing C_f by the ratio of the height over the separation. The final two components, C_{g1} and C_{g2}, describe the mutual capacitance between the two tracks and their calculation requires two complex formulas which are given in the book by Gupta[4].

While the Lewis method for capacitance calculation was only rarely used in the course of the research, the method behind it is simple to understand and illustrates a technique that is easy to use and requires minimal computing power. It also provided a good approximation used to check whether, in the early programming stages, the more accurate method described later in section 3.4 was working correctly.

It was recognised early on that the Lewis method had several limitations that made it impractical for accurate calculations of realistic interconnect models. In order to satisfy the requirement that the conductors be symmetrical, all of the conductors in the calculation must be completely identical. In realistic bus structures, however, the power, the signal and the clock lines are typically made to different separations and widths to satisfy restrictions imposed by delay, crosstalk and electromigration problems. Further, most modern microelectronic integrated circuits utilise multiple levels of metallisation and commonly employ different interconnect dimensions on each level. Not only does the Lewis method require that all conductors be of the same dimensions, but it also

requires that all conductors should be in the same y-plane. This restricts this technique to single level structures.

One additional limitation that is imposed by the method applies to systems with more than two conductors. In this case, the conductor width must be less than or equal to twice the height of the conductors above the ground plane ($w \leq 2h$). While this is a reasonable restriction in most VLSI systems, it can result in large errors in the calculation of inductance (see section 4.1) and in alternate materials such as GaAs and SOI‡[22]. These limitations in combination with the inherent inaccuracies in this method made it obvious that a more rigorous approach to capacitance calculation was required.

3.3 Numerical Techniques for Capacitance Calculation

As mentioned in the introduction to this chapter, rigorous numerical techniques for the calculation of capacitance are characterised by high accuracy, flexibility on conductor geometry, complex computational algorithms, long computation times and large requirements on computing power and memory. There are three principle numerical techniques used to calculate capacitance in an arbitrary multiconductor system, namely, the finite element method (FEM)[23,24], the boundary element method (BEM)[25,26,27] and the partial element equivalent circuit technique (PEEC)[28,29] The FEM, when used to calculate capacitance, works by partitioning the entire region of interest into a mesh of elements to determine the potential distribution by modelling the electric field. Capacitances are then derived from this potential distribution by the application of either an electric field on the conductors or through the use of a potential energy technique to find the charge on each conductor. This technique allows the modelling of non-homogeneous conductors, curved geometries and very large conductor systems (since the resulting element matrix is sparse).

---

‡ In GaAs and SOI (Silicon On Insulator) processes, the effective ground plane of the substrate is farther from the the conductors than in silicon processes due to the high resistance of the substrate material.

The BEM, more commonly known as the Green's function technique, is the most frequently used numerical method for computing the inductive and capacitive matrices for multiconductor systems. It works similarly to the FEM, but models the charges on the conductor rather than the electric field. It's primary advantage is its computational efficiency. Additionally, only conductor surfaces and dielectric interfaces need be discretised so that open regions do not contain elements and artificial boundaries do not need to be introduced. The BEM replaces all conductor surfaces and dielectric interfaces with the charge distribution that exists in free space to produce a potential distribution equivalent to the original system. Using the principle of superposition, the electric field of a point in space is due to the cumulative effects of all charges present in the system. A Green's function is then defined that gives the potential at any point based on the distribution of total charge in the system and the potential of the conductor used as a reference (typically an infinite ground plane). The main limitation to the BEM is that the segmentation of the dielectric interfaces can result in extremely large and dense matrices. There are several disadvantages of this method which include complex mesh generation leading to elaborate data preparation and the necessary imposition of boundaries for open region problems which can lead to inaccuracies.

The PEEC is a technique in which all of the conductors in the system are broken down into rectangular cells and the equivalent electrical circuit is determined by computing the "partial" capacitances and inductances within these subsections of the main conductor. The resulting equivalent circuit consists of a three-dimensional mesh of inductors and capacitors. The primary advantage of the PEEC technique is that it is inherently three-dimensional due to the method used to construct the mesh and as such is more efficient in terms of computation time and memory requirements at calculating parameters in three dimensional systems. This is not usually an advantage in many practical structures as they can be modelled by assuming considering a cross-section and assuming that all parameters in the third dimension are uniform. One obvious restriction with this method

is that it requires that all conductors are approximated by rectangular cells, but this is not a serious limitation with most practical interconnect structures.

Comparing each of these three numerical methods a few points can be made:

- For most practical structures which are uniform in the third dimension, there is no computational advantage to be gained by using the PEEC technique over the BEM and FEM methods.

- Compared with the FEM, data preparation is simpler with the BEM and the PEEC technique.

- Open areas are automatically catered for in the BEM and PEEC technique whereas the FEM requires a truncation of the problem space for calculation which can introduce a significant error.

- In simple systems involving few dielectric layers and containing a minimal number of conductors, the BEM provides the same accuracy as the FEM with a smaller mesh. As the problem complexity increases, however, the required computer resources needed by the BEM increases very rapidly, and a geometry dependent cutoff point will be reached above which the FEM becomes more efficient.

All of the numerical techniques described result in extremely accurate calculations of capacitance with varying degrees in geometric flexibility dependent on the method in use. They are all very computationally intensive, however, and generally require the inversion of extremely large matrices - a process that is demanding on both memory requirements and computational resources. In the course of the project I anticipated using different conductor configurations which would each involve multiple calculations.

Investigation into these techniques found that a FEM implementation using a fast Sun 4 computer resulted in typical run-times of 6 minutes for simple three conductor systems using a relatively coarse mesh and in excess of 30 minutes for more complex eight conductor systems using finer meshes[30,31,15] These run-times do not include the time spent creating the mesh used to calculate the inductance and capacitance parameters which would commonly be a significantly larger amount of time. Ideally, a method was sought which would run faster than this on smaller machines such as the more common and (compared to the Sun 4) more inexpensive 386 and 486 desktop personal computers and which required much smaller memory considerations. The basic requirements for this new method were: computationally efficient with typical run times of less than a minute for three conductor systems on a 386 or 486 compatible desktop computer, modest memory requirements for small systems (matrix fits into 8Mb of RAM), reasonable accuracy (within a 5% tolerance), and that it allows multiple layers and different conductor dimensions.

3.4 Matthaei's Method for Coupled Interconnect Capacitance Calculation

3.4.1 Introduction to Mattahei's method

An alternative technique which can be used to determine distributed capacitance parameters is described in a paper by G.L.Matthaei[32]. The method was investigated and as it was found to match the requirements for this project, it was adapted for our capacitance calculations. This method is essentially a combination of the BEM and the even-odd mode of capacitance calculation described in section 3.2.1. In essence it works very much like the BEM described in the previous section, but rather than determine the charge distribution through the solution of the Green's function, it uses two charge basis functions, each associated with an entire side of a conductor. Since there are two functions used per side of rectangular conductor, the calculation requires a square matrix

of eight times the number of conductors - a significant reduction over the BEM. Since the matrix is smaller the memory requirements are economical and computation time is low.

The two charge basis functions used are similar to the even and odd mode functions described in section 3.2.1, but in this case they are describing the charge distribution within the conductor rather than electric field or equivalent capacitances. Figure 3.4a shows a one dimensional conductor of width w and figure 3.4b shows the first of the two charge basis functions - the even-symmetric function. If a charge of q was placed along the length of the conductor with an equal and opposite charge existing at infinity, this is the charge distribution that would occur on an infinitely thin conductor . Using Green's function, it can be shown[33] that the potential arising from this charge distribution at any point z in the complex plane is:

$$\phi_e(z,b,w) = q\left(\frac{1}{\varepsilon_0} H_e(z,b,w)\right) \qquad (3.3)$$

where

$$H_e(z,b,w) = \left(-\frac{1}{2\pi}\right) \mathrm{Im}\left[\arcsin\left(\frac{2(z-b)}{w}\right)\right] \qquad (3.4)$$

In these equations, ε_0 is the dielectric constant of the medium, b is the position of the centre of the conductor and w is the width of the conductor as indicated in the figure.

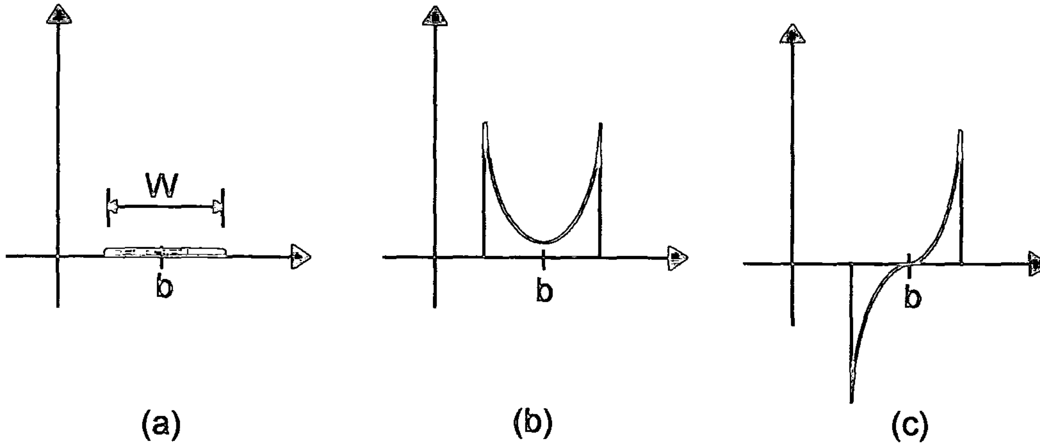

Fig. 3.4. (a) A two dimensional metal strip that extends into and out of the paper. (b) An even charge basis function for a fixed charge along the length of the strip. (c) The corresponding odd charge basis function.[15]

The corresponding odd-symmetric charge distribution is shown is figure 3.4c. This is the charge distribution along the conductor shown in figure 3.4a if it were placed in a uniform electric field along the x-axis direction in the figure. Again, using Green's function, it can be shown that the potential at any point z in the complex plane caused by this charge distribution is given by:

$$\phi_o(z,b,w) = g\left(\frac{1}{\varepsilon_0} H_o(z,b,w)\right) \qquad (3.5)$$

where

$$H_o(z,b,w) = \mathrm{Re}\left[(z-b) - \mathrm{sign}(\mathrm{Re}(z-b))\sqrt{(z-b)^2 - \left(\tfrac{w}{2}\right)^2}\,\right] \qquad (3.6)$$

The constant g in equation (3.5) is derived as a result of the Green's function conversion and it s value is not required as it drops out in later equations due to the fact that the odd-symmetric distribution has a net charge of zero.

Matthaei's method uses a superposition of these two charge distributions to approximate the actual charge distribution on the face of each conductor. The use of both of these charge distributions is seen to have a weakness in two dimensional conductors in that both of these distributions contain a stronger charge singularity at the edges than is actually present. In the one dimensional case shown above these equations are correct, but in the two dimensional case they overestimate the charge. In order to account for this effect the sampling points, referred to as match points, used to determine the potential must be kept away from the corners of the conductor.

In order determine the potential at any point z caused by the vertical faces of a rectangular conductor, two more potentials require definition. Referring to the functions described in equations (3.4) and (3.6), these can be given as:

$$U_e(z_k, b_m, w_m) = H_e(-jz_k, -jb_m, w_m) \tag{3.7}$$

and similarly,

$$U_o(z_k, b_m, w_m) = H_o(-jz_k, -jb_m, w_m) \tag{3.8}$$

where z_k is the point in the complex plane in which the potential is being determined, b_m defines the centre of the charge distribution on the mth conductor face and w_m defines the width of that conductor face and j is used to represent an imaginary number.

3.4.2 Implementation of Matthaei's Method

For these calculations, the conductors are described in terms of their number and dimensions. These parameters are then converted into points b_m on the complex plane which denote the position of the centre of each conductor and a width value w_m which gives the width of each conductor face (where m denotes the mth conductor face). For every dielectric interface an "image" charge must be calculated to account for the mirroring effect of the interface. This image conductor has a charge with an amplitude

that is scaled by a value K ($K \leq 1$) which is determined by the ratio of the two dielectric constants. This image conductor is accounted for by taking the complex conjugate of each of the b_m points to obtain a conductor that is symmetric across the x-axis. The charge on this mirrored conductor is then scaled by the factor K.

Including the image charge, the potential from a horizontal even-symmetric basis function for a conductor that is centred at b_m is given by:

$$P^{e}_{km} = [H_e(z_k, b_m, w_m) + KH_e(z_k, \overline{b_m}, w_m)] \qquad (3.9)$$

while for a vertically oriented even-symmetric basis function the potential is:

$$P^{e}_{km} = [U_e(z_k, b_m, w_m) + KU_e(\overline{z_k}, b_m, w_m)] \qquad (3.10)$$

Similarly, the potential from a horizontal odd-symmetric charge distribution is:

$$P^{o}_{km} = [H_o(z_k, b_m, w_m) + KH_o(z_k, \overline{b_m}, w_m)] \qquad (3.11)$$

while that from a vertical conductor face is:

$$P^{o}_{km} = [U_o(z_k, b_m, w_m) + KU_o(\overline{z_k}, b_m, w_m)] \qquad (3.12)$$

To find the potential on the conductor faces, two points on each face are selected. These points, z_k, are termed "match points" and can be chosen to be anywhere along the face determined by the parameters rh and rv. The match points are determined for a horizontal conductor by:

$$z_{2m-1} = b_m - \frac{(rh)w_m}{2} \qquad \text{and} \qquad z_{2m} = b_m + \frac{(rh)w_m}{2} \qquad (3.13a)$$

and likewise for a vertical conductor:

$$z_{2m-1} = b_m - \frac{j(rv)w_m}{2} \quad \text{and} \quad z_{2m} = b_m + \frac{j(rv)w_m}{2} \qquad (3.13b)$$

As was mentioned previously, choosing match points close to the edge of the conductor can lead to inaccurately high values of calculated capacitance due to overestimation of the charge by the distribution curves. In this paper, Matthaei suggested that the values of 0.67 and 0.8 for rh and rv respectively lead to accurate results in systems with a ground plane Personal investigation, however, found that the results were not particularly sensitive to the precise value of rh and rv as long as both were between the values of 0.5 and 0.9.

Once the conductors have been plotted on the complex plane and the match points have been chosen, two matrices of calculated P^e and P^o values are formed according to the equation:

$$\sum_{k=1}^{8N}\sum_{m=1}^{4N} P^e_{km}Q_m + P^o_{km}G_m = v_k \qquad (3.14)$$

P^e and P^o are both matrices with 8N rows and 4N columns in which each rows represents a match point on a conductor face and each of the columns the centre point of each face. In this case the value for G_m can be ignored since, as was mentioned previously, the odd-symmetric charge basis function has a zero net charge. As we are trying to solve for Q_m, the P^e and P^o matrixes must be inverted and the values for v_k determined.

The actual computation is performed by setting v_k to a constant on each conductor in turn while fixing the potentials on other conductors to zero. This is repeated on each conductor in turn so as to compute the total charge induced on each of the N strips. Then

the charge on every conductor face of each track is summed to obtain the total charge on that conductor. Finally, the capacitance calculated using the formula:

$$C_{np} = \frac{q_{tn}}{V_p}\bigg|_{\text{All } V_n=0 \text{ except for } n=p} \tag{3.15}$$

where C_{np} is the capacitance between conductor n and conductor p, and q_{tn} is the total charge on conductor n. So, in order to compute the N^2 capacitance coefficients that are required for N conductors, equation (3.14) must be solved N times, each time with a voltage of V_p (1V is used for convenience) applied to a different conductor with all of the other conductors grounded. The solution to equation 3.15 results in a capacitive matrix for the system of conductors.

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{23} & C_{33} \end{bmatrix} \tag{3.16}$$

This matrix describes the capacitance between any two conductors in a system of 3 conductors over a ground plane. The centre diagonal, C_{ii}, contains the self capacitance between the conductor and the ground plane, while the other matrix elements, C_{ij} (where $i \neq j$) are the mutual capacitances between two conductors in the system. Thus, C_{12}, refers to the mutual capacitance between the first and second conductors. This notation will also be used throughout to describe two-dimensional inductive matrices and one-dimensional resistive matrices.

Shown below in figure 3.5 is the core pseudo-code describing in detail the algorithm used to calculate capacitance. The full program used to implement the method described is listed in Appendix A.

```
Input( )                        ; input description of conductors
Convert( )                      ; converts description to complex coordinates
Calculate_PePo_Matrix( )        ; calculates Pe and Po matrices for all points
Capacitance_Calculation( ) {         ; converts the Pe Po matrices into capacitance
   Invert_PePo_Matrix( )        ; inverts the Pe Po matrix
   For Current_Conductor = 1 to N {     ; cycles through conductors to set voltages
      For m = 1 to 4*N {        ; cycles through each of the centre coordinates
         For k = 1 to 8*n {     ; cycles through each of the match points
            If k is a point on Current_Conductor then V = Vp
              else V = 0        ; sets the voltages on each conductor
            Qt[m] = Qt[m] + V*PePo[m,k]
         }
      }
   For b = 1 to N {             ; sums the charge on each of the four sides
      C[b,Current_Conductor] = ε*(sums charge on each of the sides of conductor b)
   }
Output_Results( )               ; outputs the results
```

Fig. 3.5. Pseudo-code description of Matthaei's method for capacitance parameter extraction.

3.5 Implementation of Matthaei's Method in the Absence of a Reference Plane

The method as described above is sufficient for typical conductor geometries. At submicron geometries, however, it was discovered that the way the method uses image charges to represent the dielectric interface of the ground plane leads to a significant source of error when the mutual capacitance is larger than the self capacitance. The method as described above uses charges on the ground plane to balance the charges on the conductors. The net charge on the conductors in the system is zero due to the charge on the ground plane which is represented by the image charge. In the case of a metalised

ground plane, the scaling value K described in the previous section is equal to -1. If K=-1 then the charge basis functions describing the charge on given conductor associate an equal and opposite value of charge to the ground plane. In effect, this means that even in systems where the ground plane is infinitely far away from the conductor system, the mutual capacitance can never be larger than the self capacitance. This inaccuracy is not an issue in a single conductor system, but becomes a problem in multiple conductor systems when the distance between the wires is greater than the distance to the ground plane. In this project this is not an issue since this case never occurs, but as the minimum feature sizes are reduced with further scaling, this limitation in Matthaei's method can result in inaccuracies in capacitance estimation.

3.6 Comparison of Accuracy for Calculation Methods

Two different comparison methods were used to verify the accuracy of the capacitance estimation method. Initially, the method was compared against the published results in the paper in which the method was described in order to ensure that the method worked as expected. In the paper, multiple examples are used to illustrate the speed and accuracy of the method when compared with a finite element method. Using the results documented in these examples, the computer program developed from the paper was checked against two of the examples within the paper and the differences between the published and computed results were negligible.

After verifying that the computer program operated as expected, the results were then compared to values obtained from other methods. These other methods varied from the simple parallel-plate formula which was used as a base-line comparison, to the several different published methods. In all of these comparisons, the estimation techniques used resulted in approximations of the actual capacitance of the line. Thus a strict evaluation of the accuracy of Matthaei's method was not possible, since the accuracy of the techniques which were used for comparison was unknown. A numerical technique would have been used as a direct comparison, but none was readily available and creating

a program from a published method would have required too much time, so a comparison against closed-end methods was used as an alternative.

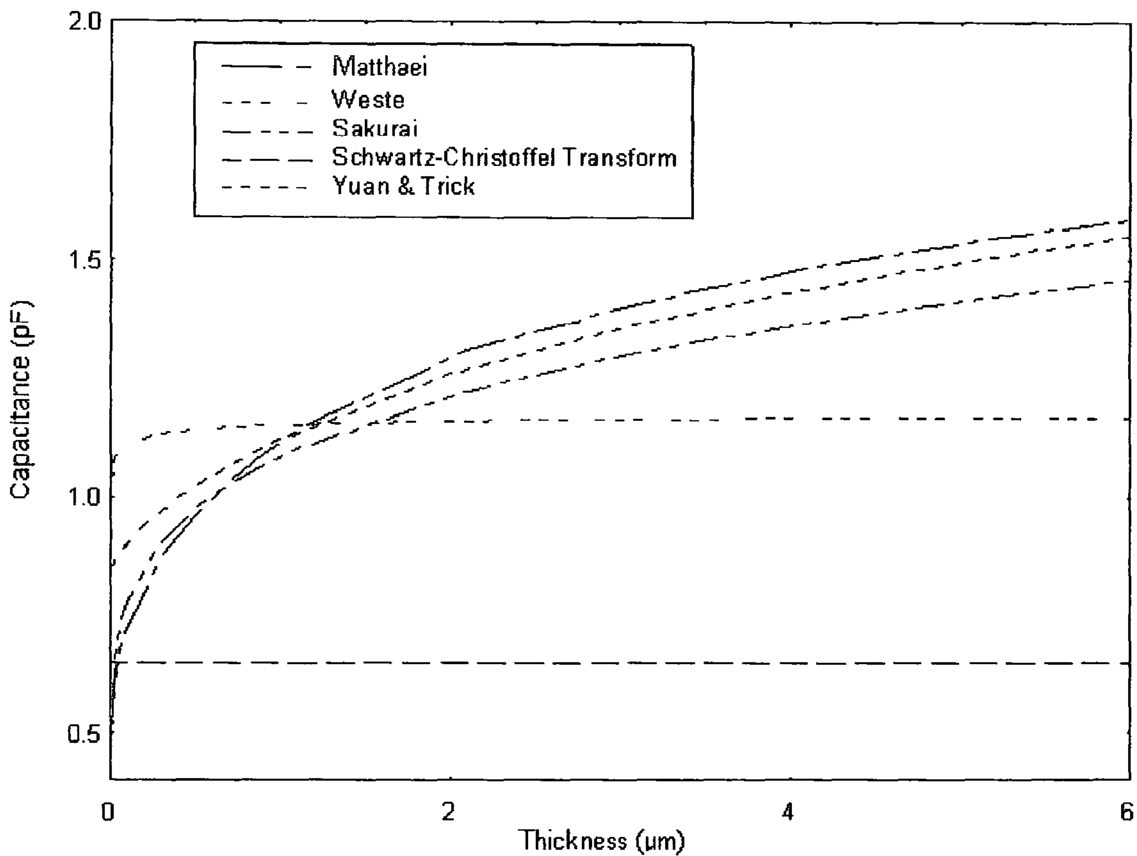

Fig.3.6. A comparison of the results from multiple methods for extracting interconnect capacitance.

A comparison made between several different techniques for determining the interconnect capacitance of a one track system is shown above in figure 3.6. The methods used to compare against the Matthaei method are the Schwartz-Christoffel transformation[18], the Yuan and Trick method[34], the Weste method [35] and the Sakurai method[36]. This comparison is limited to single conductor systems because, of all of these methods, the Yuan and Trick (Y&T) method was documented as having the highest accuracy, but it can only be applied to single conductor systems. The Schwartz-Christoffel transformation (SCF) is used as a reference baseline. Since the SCF only

considers the width, height and separation of the conductor, and not the thickness, it is shown to remain a constant value of capacitance as thickness is varied.

Since the SCF accurately describes the capacitance of an infinitely thin conductor, none of the other methods should estimate a value for capacitance that is less than this value. As the value for thickness is reduced towards zero, the capacitance value should approach the capacitance value calculated by the SCF method, but not drop beneath it. In this case, the Matthaei method is shown to underestimate capacitance when the conductor is extremely thin. The other techniques, such as the Weste and the Y&T methods, overestimate the capacitance when the conductor is infinitely thin. At higher thickness, the Y&T method is documented to be the most accurate. In this case, it can be seen that the Matthaei technique is extremely close to it as thickness is increased. In the paper documenting the method presented by Yuan and Trick, they describe that this method typically underestimates the capacitance when the thickness is between 1μm and 10μm (all other dimensions are 1μm). When compared with another more accurate numerical method[37] this underestimation is approximately 5%. This indicates that a more accurate value for the capacitance of a track when the conductor is reasonably thick is much closer to the value given by the Matthaei method.

Another comparison between methods examining the total capacitance of a one conductor system is made between the results published in a paper describing a method proposed by Chang[38] and the methods described previously. In this case, the numerical method proposed by Weeks was cited by Chang as reference to the accuracy of his proposed method. The values for the track dimensions come from the paper by Chang. I

W/H	T/H	Chang	Weeks	Weste	Sakurai	Matthaei
1.12	0.318	3.55	3.59	3.76	3.06	3.18
2.01	0.485	4.76	4.79	4.57	4.16	4.24
2.52	0.318	5.23	5.24	5.00	4.49	4.62

2.72	0.802	5.79	5.82	5.22	5.13	5.14
3.18	0.936	6.37	6.39	5.64	5.68	5.64
3.63	0.802	6.81	6.82	6.03	6.06	6.00
4.24	0.936	7.54	7.54	6.58	6.76	6.63
5.44	1.200	8.96	8.98	7.65	8.12	7.84
6.36	0.802	9.76	9.75	8.45	8.84	8.48
7.42	0.936	10.96	10.94	9.39	10.00	9.49
8.21	1.805	12.12	12.14	10.12	11.19	10.53
9.85	1.805	13.84	13.83	11.57	12.86	11.98
11.90	1.453	15.86	15.85	13.38	14.81	13.66
14.78	1.805	18.95	18.97	15.94	17.88	16.34
22.22	4.070	27.07	27.08	22.55	26.01	23.68
58.25	7.113	63.94	63.94	54.46	63.15	57.94

Fig. 3.7. Comparison of total line capacitance using different techniques.

It can be seen from the table that Matthaei's method underestimates the value given by the Weeks method consistently between 8% and 13%. Since all three of the methods that were calculated underestimate the values cited by Chang with reasonable consistency, there are three possible explanations. The first is that Matthaei's, Sakurai's and Weste's methods all consistently underestimate by a small percentage. The second is that Weeks's and Chang's technique both consistently overestimate. The third and most likely explanation is that there is some undocumented information such as the dielectric constant that is different in the estimations performed by Chang and the estimations using these three methods. The fact that the underestimation is consistent and within a small percentage indicates that that the third explanation is most likely.

From these two graphs, it can be concluded that Matthaei's method has been verified to work correctly at different values of physical track dimensions. It has also been shown that this method for capacitance determination has comparable accuracy to other widely methods such as that proposed by Sakurai.

4. Inductance and Resistance Parameter Calculation

4.1 Introduction

Although these resistance and inductance are extremely important in traditional electronics, in microelectronic metal lines both have often been ignored in the past. In 3μm and 5μm CMOS technologies, the capacitance of the lines and the resistance of the devices determined the speed of the device. Although the resistance of metal lines for these technologies was significant, the resistance of the devices in series with the metal line resistance was much larger and dominated over the smaller term. As effective channel lengths have been reduced through improvements in technology and the cross-sectional area of wires have correspondingly been reduced, the resistance of the devices has become comparable to the resistance of the interconnections between them.

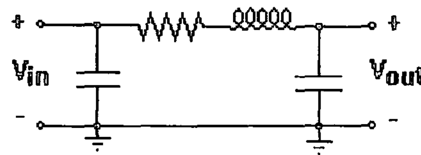

Fig. 4.1. Basic RLC line model.

A relationship between resistance, inductance and the device switching speed exists due to the fact that in a interconnection model, the resistance of the line is in series with the inductance as shown in the RLC π model shown in figure 4.1. The capacitance is distributed at both ends and is in parallel. Since line resistance is in series with inductance, the transfer function for a line model containing both resistance and inductance will always include the combination $R + j\omega L$ where R is the line resistance, ω is the frequency and L is the line inductance. In older technologies the frequency,

determined by the switching speed, and the inductance were both much smaller than the series resistance of the line and the devices.

On newer processes, the switching speed is much faster, the device resistance is smaller due to the reduced channel length and the inductance, similar to the capacitance, is slightly higher. In this case, resistance does not completely dominate over the inductance. This is referred to in literature as a lossy transmission line. In microwave and radio-frequency circuits, the frequency-inductance parameter dominates over the resistance completely and the model is called a lossless transmission line.

This section first looks the calculation of inductance using a variation of Matthaei's method for capacitance calculation described in the previous chapter. Then the calculation of resistance is examined and an evaluation of the contribution of frequency-dependent resistive effects is made. Finally, an assessment of the contribution of inductance to signal transmission and coupling in typical submicron wire is made.

4.2 Inductance Parameter Calculation

The idea of an inductor as a coil of wire which is taught in introductory electronics classes is only a starting place to begin to understand how inductance exists in microelectronic circuits. In reality, inductance exists in any part of a circuit element that carries current. Since the amount of inductance between two objects is defined in a manner similar to the amount of capacitance, these elements include conductors as large as a PCB track and those as small as interconnect lines on integrated circuits.

Similar to capacitors, inductors play a vital role in modern electronics but, unlike capacitors, inductance in microelectronic circuits is rarely a desired characteristic. It is largely responsible for a source of signal noise called simultaneous switching noise, also referred to as grounded bounce, and it can add to coupling of signals between adjacent lines. In high speed packaging it is the primary cause of both ringing and reflection. All

of the problems associated with inductance in microelectronics appear only in extremely high speed circuits, however, and it will be shown later that inductance is only a minor problem in the fastest of modern silicon CMOS microelectronic circuits. Inductance only becomes an important parameter if the length of the line is long and as a result has a large inductance or if the transition time is fast and the inductive voltage drop becomes large. In either of these two cases, transmission line effects begin to dominate and the distributed inductance of the line must be considered.

The compromise between flexibility and simplicity, and the accuracy of the simulation has been a primary concern throughout the course of the project. Since the aim of the project was to evaluate the effects of crosstalk noise in sub-micron geometries, it was decided that inductive effects must be considered in order to account for the possibility that scaled 0.35μm and 0.50μm transistors could operate at such high speeds that inductance would form a significant contribution to the crosstalk voltage coupled.

Although many methods of directly calculating the inductance matrix for a system exist, it is possible to calculate the inductance matrix for multiple interconnect lines using the capacitance matrix. Since the capacitance matrix can be calculated using methods described in the previous chapter, this method was employed in the inductance calculations for purposes of simplicity and computational speed. This calculation is based on the transmission line equations.

Although the transmission line equations have been already derived in chapter 2 (equations (2.3-2.6), they are redefined here using vector notation and ignoring the effects of resistance and conductance for simplicity. If [V] is a vector of line voltages, [I] is a vector of line currents and the inductance and capacitance matrices are given by [L] and [C] respectively, then the transmission line equations described voltage and current propagation through the line are:

$$-\frac{d}{dz}[V] = j\omega[L][I] \qquad (4.1a)$$

$$-\frac{d}{dz}[I] = j\omega[C][V] \qquad (4.1b)$$

These two equations can be combined to give:

$$\frac{d^2}{dz^2}[V] = -\omega^2[C][L][V] \qquad (4.2)$$

If we assume that all modes have the same propagation velocity, v, and that neither [C] nor [L] are frequency dependent, then the computation of inductance using the capacitance matrix can be found from (4.2) using static voltages as:

$$[L] = \frac{1}{\upsilon^2}[C]^{-1} \qquad (4.3)$$

For silicon systems, the assumption that all modes have the same propagation velocity is not realistic since the silicon substrate and the dielectric have different relative dielectric constants. Different propagation modes, for example the even and odd modes of propagation described in chapter 3.2.1, will have different field distributions through the dielectric layers. These different field distributions through the layered dielectrics lead to different effective dielectric constants and differing propagation velocities. In this case, equation (4.3) can modified to account for the variation in dielectric constants. The solution is to find the inductance matrix as the reciprocal of a different capacitive matrix. In this new matrix, $[C_0]$, the capacitance is calculated by considering the system to be in free space with a relative dielectric constant of unity and propagation velocity of the speed of light in free space, c. Then the inductance matrix for the layered dielectric is estimated using:

$$[L] = \frac{1}{c^2}[C_0]^{-1} \qquad (4.4)$$

The dielectric materials can be ignored in silicon-based microelectronic systems, because there is a negligible difference in magnetic permeability between silicon and free space[39].

4.3 Resistance Parameter Extraction

Resistance in a wire, to a first order approximation, is inversely proportional to the cross-sectional area of an interconnect. As interconnects are scaled into the submicron region, the conductor width (and to a lesser extent, the conductor thickness) are scaled down in proportion with the transistors in order to improve packing density. This results in a corresponding increase in the series resistance of the wire. The first order contribution from line resistance is an attenuation of the waveform. In other words, a reduction in the amplitude and bandwidth of the propagating signal. Second order effects include frequency dependence of the characteristic impedance and the speed of propagation. Line resistance is caused by a combination of three types of loss: the DC resistive component, the AC resistive component caused by skin depth and the dielectric loss of the insulator.

4.3.1 DC Line Resistance

The DC resistive component currently dominates over the other components in current CMOS technology and is expressed by:

$$R_s = \frac{\rho l}{wt} \qquad (4.2)$$

where R_s is the DC resistance of the interconnect, ρ is the resistivity of the conductor material, and l, w, and t are the length, width and thickness of the conductor respectively. Thus to keep the resistance of scaled interconnects as low as possible, the conductor

width and thickness need to be kept as large as possible, the wire material needs to have a low resistivity and the length of the interconnect needs to be minimised.

4.3.2 AC Line Resistance

A direct (DC) current is distributed evenly throughout the cross-section of the conductor through which it flows. In the case of an alternating current, however, the component of the electric field along the conductor which drives the current flow does not penetrate completely into the depth of the conductor. An incident electromagnetic field generates currents on the surface of a conductor in the direction of the electric component of the field. If a given conductor is perfect, the current is confined to an infinitesimally thin layer at the surface and the electric field induced by this current cancels the incident field so that there is no electric field present within the conductor. In reality, all conductors have a finite resistance which causes the field and the current to penetrate into the conductor and this gives rise to a resistive loss.

The higher the conductivity and the frequency are the thinner the resulting penetration within the conductor. Therefore the effective resistance of an interconnect for high-frequencies does not decrease by making it thicker than a critical value. This value is called the skin depth and is expressed as:

$$\delta = \sqrt{\frac{\rho}{\pi \mu f}} \qquad (4.3)$$

where δ is the skin depth, f is the sine wave frequency of the alternating current and μ and ρ are the permeability and conductivity of the material. The fact that the skin depth for more conductive materials is thinner may seem to imply that a less conductive material is more desirable as a high-frequency conductor since a smaller cross-sectional area will result in a larger resistance. In fact, because the skin depth is proportional to

the square root of the resistivity, the product of the conductivity and the effective cross-sectional area is a factor of $\sqrt{\Delta\rho}$ larger for better conductors.

It must be noted that a real signal has a spectrum of sine wave components which extends up to its bandwidth and that each of these frequency components will see a slightly different resistance, equation (4.3), however, is a good first order approximation of the actual skin depth at a given frequency.

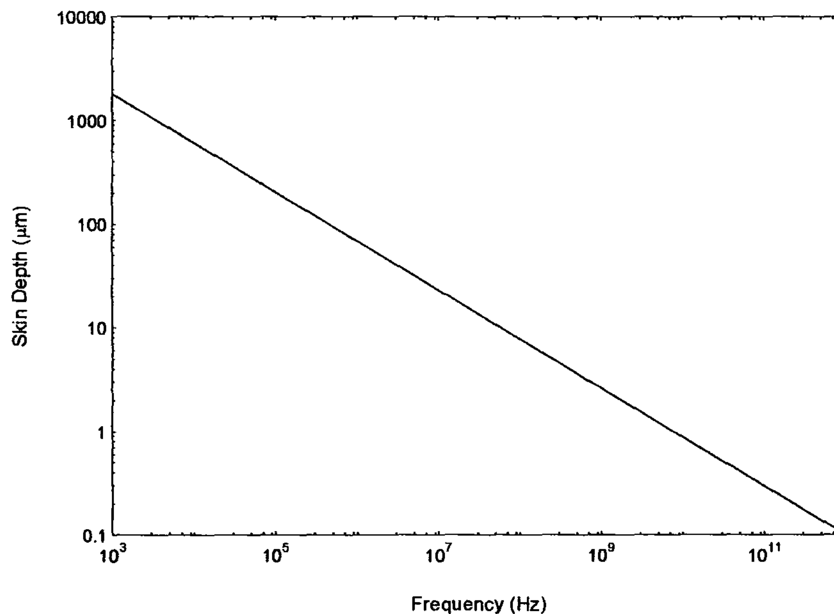

Fig. 4.2. Skin depth as a function of frequency for aluminium at room temperature.

From figure 4.2 is can be seen that the skin depth at a frequency of 1GHz is still more than 2.5μm. It is only around 10GHz that the skin depth becomes comparable to the thickness of the conductor. The effect of the transient response of transmission lines with frequency dependent skin depth losses and the significance of the skin depth considerations has been examined by several authors in literature[40,41]. While looking at the problem of crosstalk in WSI (Wafer Scale Integration) and in reference to the importance inclusion of skin depth into the model, Kim and McDonald concluded, "...the results of skin effect loss can be ignored throughout the frequency of interest (1MHz -

4GHz). However, for GaAs or advanced bipolar packaging applications where performance at higher frequencies (5-20GHz) is of interest, our method can be easily extended to consider the skin effect."[42] In order to avoid complexity in the calculation of the model and to shorten computation times by avoiding use of frequency dependent components, it was decided that skin depth would not be included in the calculation of the resistance parameter.

4.3.3 Dielectric Loss

Dielectric loss is a form of resistance caused by the material that forms the dielectric between the conductor and the ground plane, or return path. As was mentioned briefly in chapter two, usually dielectric loss is modelled as a parameter, G, in an RLCG model where G is a resistor in parallel with the capacitance. It is typically a very high frequency effect (in excess of 10GHz), and is commonly ignored for Si-SiO_2 systems. Dielectric losses were considered to evaluate their contribution at the sub-nanosecond switching speeds expected by submicron devices.

Dielectric loss is caused by two separate physical mechanisms: DC conduction through the dielectric and high frequency dipole relaxation. There is a finite amount of current that leaks through the dielectric and forms a conductive loss. From the small amount of information available, DC conduction through the dielectric is negligible for silicon systems due to the high resistance properties of silicon dioxide.

At high frequencies, typically over 10GHz depending on the technology, dipole relaxation begins to dominate the conduction current and causes it to become frequency dependent. Due to the difficulties involved in incorporating a frequency dependent resistive component in an HSPICE model as well as the high frequencies that are necessary for dipole relaxation to contribute significantly to the overall line resistance, it was decided that dipole relaxation would be ignored.

5. Interconnection Scaling

5.1 Introduction

The term "scaling" refers to the reduction of features and device dimensions in either two or three dimensions on integrated circuit designs in order to benefit from advances in silicon process technology. There are two primary motivations behind scaling of integrated circuits: the reduction in area and an increase in device switching speed. The reduction in area achieved by scaling allows more complex chip designs to be implemented in a smaller area than would be needed at previous technology levels. Since the circuit proportions are being scaled in two dimensions, there is a square law improvement in packing area which in turn improves yield and lowers cost. The improvement in switching speed enables better overall circuit performance which results in faster IC's.

Previously the performance of an integrated circuit was only limited by device performance and simply scaling these devices down automatically resulted in much faster circuit designs. More recently, as minimum feature size has continued to scale down to submicron proportions track line widths and spacings have, of necessity, followed suit in order to take full advantage of the scaling process. As a result, interconnect performance has become a serious limiting factor to the improvement of overall circuit performance. The parasitic resistance and, under certain conditions, parasitic capacitance of the lines increases and thus their corresponding RC delay rises. This delay has begun to dominate over gate delay for long interconnects and this has tended to limit the length of global routing[13,43] Crosstalk between lines limits the scaling of interconnect separations and thickness, while current density and electromigration problems limit the cross-sectional area of the tracks.

RC Delay	$S^2 S_C^2$	$S S_C^2$	$S S_C^2$
Current Density	S	$S^{1/2}$	$S^{1/4}$

Fig. 5.1. Scaling of Global Interconnections.

The table above assumes a given circuit in which the device dimensions are changed by a factor S and the overall circuit size is scaled by a factor S_c. In this case due to the increase in chip size the track length of global interconnections (such as busses, clocking wires, and the buffers from pin inputs) is assumed to be scaled by a factor Sc.[16] From table 5.1, it can be seen that since the cross-sectional area of the track is reduced by a factor S^2, the resistance of global interconnections scales by S^2S_C. It should be noted that the capacitance of the interconnection is constant for ideal scaling since all of the dimensions are scaled and that increases in global interconnection capacitance are due to the increased length of the tracks themselves.

The gate delay is given by the product of the transistor resistance and the gate capacitance (R_{tr}*C_G) and it can be seen that, although the gate delay decreases by a factor of 1/S in ideal scaling, the RC wiring delay increases by a factor of $S^2S_C^2$. Therefore, since the scaling of the length of global wiring is directly proportional to the chip size scaling factor and since the scaling factor and the chip size scaling factor are often approximately the same, it can be seen that global wiring delay increases by a rate of S^4 while device delay decreases by a rate of 1/S. Using older 3.0μm and 1.5μm process technologies, the gate capacitance dominated over the interconnect capacitance and the relative RC delay of interconnections as compared to the switching delay of the devices themselves was negligible. As faster devices have been manufactured using more advanced processing techniques, a transition has gradually occurred in which the reverse is now often true. The switching times of the gates, which on advanced processes can be much less than 200ps, is an order of magnitude or more smaller than the switching delay for global interconnections which can be in excess of 2ns. The problem of routing global interconnections in such a way that integrated circuit speeds continue to improve at a rate

5.2 Ideal Scaling

The most straightforward approach to satisfying the process and layout constraints was proposed by Dennard[44] in 1974. He suggested scaling all the horizontal and vertical dimensions and voltages by the same factor as is used for the transistors, and his method is now referred to as "ideal scaling", although it will be shown that this technique produces far from ideal results.

Using the method of ideal scaling, the devices and interconnections all require the same relative accuracy from lithography, pattern etching and material deposition techniques and the aspect ratios of tracks and steps do not change sizes as they are reduced. The effects of several different scaling methods on local and global interconnections are listed on the next page in table 5.1. In this table, S represents the scaling factor for the device dimensions ($S > 1$) and S_C is the scaling factor for the chip size ($S_C > 1$). This chip size scaling factor accounts for the increase in die size from one generation of ICs to the next due to increases in complexity and functionality.

Parameter	Ideal Scaling	Quasi-Ideal Scaling	Constant-R Scaling
Thickness	1/S	$1/S^{1/2}$	$1/S^{1/2}$
Width	1/S	1/S	$1/S^{1/2}$
Oxide Thickness	1/S	$1/S^{1/2}$	$1/S^{1/2}$
Separation	1/S	1/S	$1/S^{1/2}$
Transistor Resistance (R_{tr})	1	1	1
Gate Capacitance (C_G)	1/S	1/S	1/S
Resistance	$S^2 S_C$	$S^{3/2} S_C$	$S\ S_C$
Self-capacitance	S_C	$S^{-1/2} S_C$	S_C
Coupling Capacitance	S_C	$S^{1/2} S_C$	S_C

comparable to the increase in device speeds is one of the primary reasons for the use of other scaling methods.

5.3 Quasi-Ideal Scaling and Other Methods

Ideal scaling can increase the resistance of global wires by S^2S_C and this in turn leads to large RC delays and high current densities in the wire. Another method termed "quasi-ideal" scaling reduces the impact of these problems while retaining the improvements in the packing density. This is performed by reducing the vertical dimensions by a smaller scaling factor than that used to reduce the horizontal dimensions. With minor changes varying on the manufacturer, it is the method predominately followed within the semiconductor industry today.

In quasi-ideal scaling, all of the vertical dimensions are scaled by the square root of the horizontal scaling factor (S). Since the horizontal dimensions are reduced by the same factor, 1/S, as the devices, the packing density of the circuit is improved by S^2. Likewise there is an improvement in track resistance, current density and interconnect RC delay. If we assume again that the average interconnect length for global interconnects scales by the same value as the scaling value for the chip size (Sc) and that the scaling value for the chip is the same as that of the devices ($S = S_C$), then the wire RC delay scales by S^3. Using the same assumption, the wire resistance will scale by a factor of $S^{5/2}$. The current density is also improved over ideal scaling and scales by a value of $S^{1/2}$.

Although the resistance of the track is reduced by a smaller factor than in ideal scaling, it can be seen that the capacitance of the track increases by a factor $S^{-1/2}S_C$. This increase in both the self and mutual capacitive components is the primary disadvantage to quasi-ideal scaling and directly contributes to increased problems with crosstalk noise and increased interconnect RC delay.

All of these problems are due to the fact that the tracks reduce in thickness and height at a slower rate than they do in separation and width. In other words, that the ratio of T/H is scaled by a factor which is smaller than the ratio of W/S. Using submicron processes both of these trends lead to interconnects that are tall and thin, spaced more closely together and which are higher above the substrate. This in turn increases the mutual capacitance to self capacitance ratio of the wires (Cm/Cs) by a factor of S. Thus, the mutual capacitance rises by a factor of S faster than the self capacitance. The primary effect of this increased capacitance ratio is increased problems with signal crosstalk.

Despite these problems quasi-ideal scaling has been used in the past to scale circuits because it allows the full improvement in packing density to be realised, it has better resistance to the problem of electromigration, and the resistance of the tracks is reduced when compared with ideal scaling. It has been envisioned that eventually the crosstalk noise within the wires will become such a problem that further scaling using quasi-ideal scaling rules will no longer be practical.

Another alternative which will be mentioned briefly for comparison is to scale the interconnect dimensions more slowly that the device dimensions, this method is commonly referred to as constant-resistance or constant-R scaling. Since the interconnections scale more slowly, this method degrades the packing density of the chip and thus it is not commonly practised. Compared to the other two scaling techniques, it offers improvements in terms of interconnection performance at the cost of lower packing density and is employed where speed is of greater importance than production costs - a niche market shared predominantly with the more costly GaAs IC's.

One other attractive alternative is to use a combination of the quasi-ideal scaling and constant-R scaling on different interconnect levels. On the lowest level where routing efficiency is of highest importance and tracks are typically short, local interconnects do not cause serious delay and resistance problems are not significant. On higher levels used for global interconnect routing, clocking and power distribution where routing

efficiency is reduced, constant-R scaling, or another scheme which allows lower resistance tracks to be used.

5.4 Variation of Track Capacitance with Scaling

For routing in submicron VLSI, the track width is typically increased for long lines to improve the track resistance and reduced for short lines to improve packing density. There is, however, always a minimum track width which is defined by the process technology used. In other words, the minimum track width is defined by the resolution of the process. A similar argument applies to minimum track separation, which is increased in long tracks to reduce coupling effects. The conductor thickness and the dielectric thickness are always defined by the process used and cannot be changed in layout to improve delay or crosstalk problems. Thus, when examining track capacitance and resistance for a specific process it is possible only to consider changes in separation and width, while leaving the thickness of both the dielectric and the conductor constant.

The thickness and height are variable, however, when scaling is considered. As is discussed in the section on interconnect scaling (section 5.1), the thickness are height are typically reduced, or scaled, at a lower rate than the separation and the width. These four dimensions can each be grouped into pairs based on the capacitive components that they primarily affect. To a first order approximation, the width and height of a conductor determine the self-capacitance of a track while the thickness and separation determine the mutual capacitance.. It will be seen that this approximation remains true when the ratios of width to height and thickness to separation are greater than one (W/H>1 and T/S>1). When the reverse is true (W/H<1 and T/S<1)second order effects begin to dominate and this approximation is no longer accurate.

In all of the graphs, the capacitance parameters were extracted using Matthaei's method for capacitance parameter extraction method as described in chapter 3.5. All of the dimensions are fixed at constant values except for the specific dimension being varied.

A diagram showing the layout of the tracks and explaining the terms used is shown in figure 5.2. Although the term dielectric thickness is commonly used in literature to refer to the distance through the dielectric layer between the conductor and the substrate, this distance is termed as height, h, to differentiate it from the conductor thickness, t. For all of the graphs, self capacitance refers to the capacitance between the centre track and the ground plane below it, while mutual capacitance refers to the capacitance between the centre track and its two neighbouring tracks. The extraction method assumes the substrate to be infinite and the relative dielectric constant, ε_r is set to 3.9.

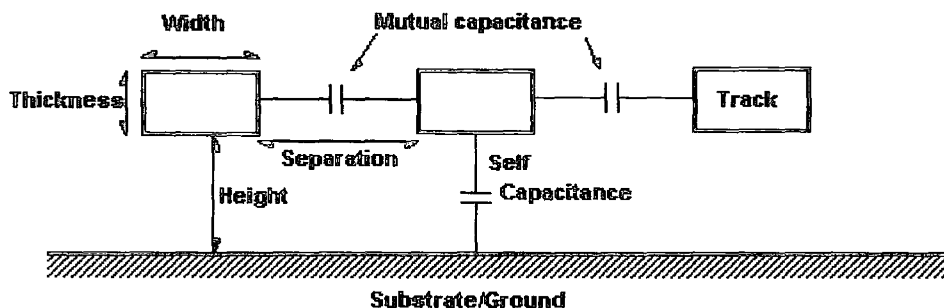

Fig. 5.2. Explanation of the terms used in the graphs in this section.

5.4.1 Variation of Capacitance with Width

Graph 5.1 shows the variation of the self and mutual capacitive components of the centre track of three parallel interconnects when the width of all three tracks is varied between 0 and 3μm while the other parameters, such as substrate thickness, track thickness, and track separation, are all held constant at 1μm. From the graph it can been seen that both self and mutual capacitance increase with increasing width. As would be expected the self capacitance increases with increasing width following the basic parallel-plate capacitance equation:

$$C_{pp} = \frac{\varepsilon_0 wl}{h} \qquad (5.1)$$

The small increase in mutual capacitance observed is caused by the increased surface area of the region on top of the track.

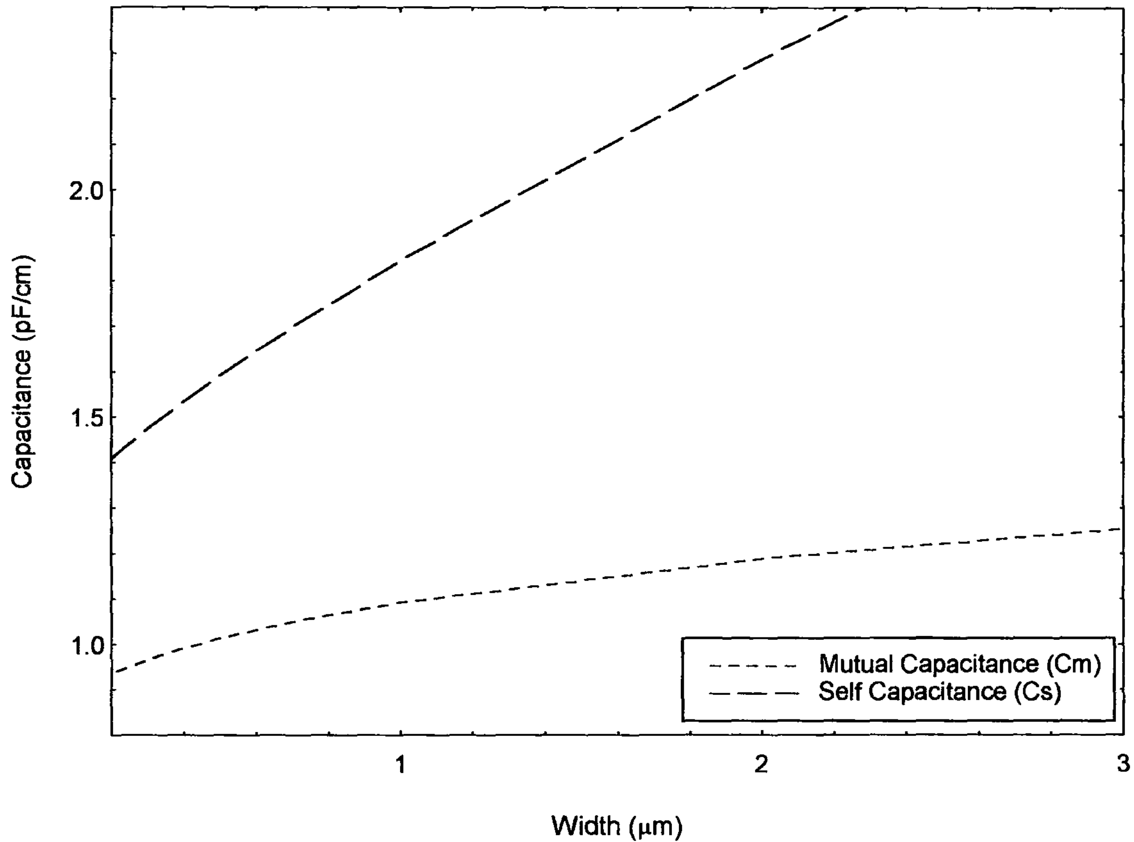

Fig.5.3. Variation of Self and Mutual Capacitance with Interconnect Width.

5.4.2 Variation of Capacitance with Height

Graph 5.4 shows the variation of capacitance when the height (substrate thickness) of all three tracks is varied between 0 and 3μm while the other parameters are held constant at 1μm. From the graph it can been seen that the mutual capacitance is relatively

unaffected by changes in the height of the conductor above the ground plane, while the self capacitance is strongly dependent on height. Considering the parallel-plate capacitance equation from (5.1), this dependence is due to the inversely proportional dependence of capacitance on distance between conductors and the substrate, h. It is noted that the small increase in mutual capacitance as the height is increased is caused by the fringing of the electric field from the increasingly more remote ground plane to the closer neighbouring tracks.

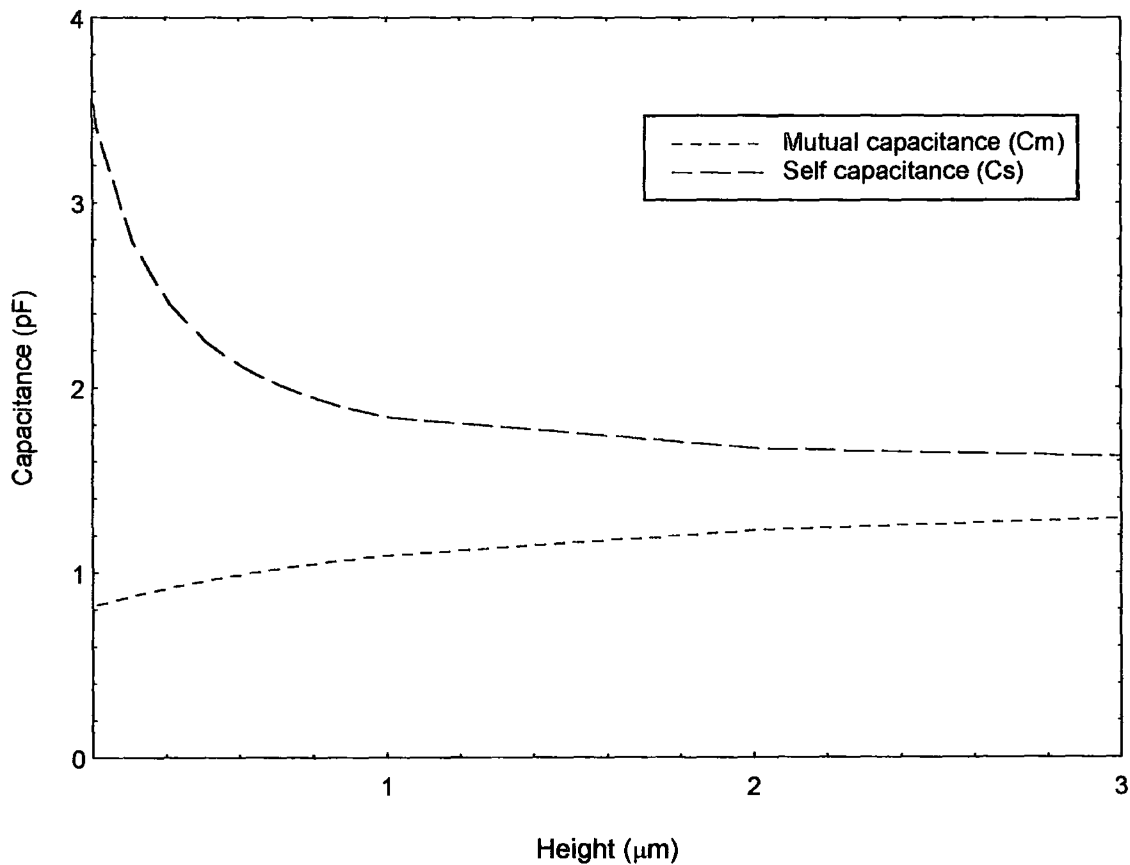

Fig.5.4: Variation of Self and Mutual Capacitance with Height (substrate thickness).

5.4.3 Variation of Capacitance with Thickness

Graph 5.5 shows the variation of the self and mutual capacitances of three interconnects when the thickness of all three tracks is varied between 0 and 4μm while the other dimensions are held constant at 1μm. The graph illustrates the large dependence of mutual capacitance on interconnect thickness as well as the relative independence of self capacitance to changes in thickness. As the thickness is increased, the mutual capacitance rises appreciably, while the self capacitance remains relatively constant. This trend can be seen if we consider the capacitance between the tracks to be comparable to a vertically-aligned parallel-plate capacitor. In this case the distance between the conductors, h in equation (5.1), is the separation of the tracks, while the width of the capacitor plate, w in (5.1), is the thickness of the tracks. The small rise in self capacitance when the thickness is reduced can be explained by the reduced surface area of the neighbouring tracks resulting in fringing of the electric field to the larger surface area of the ground plane.

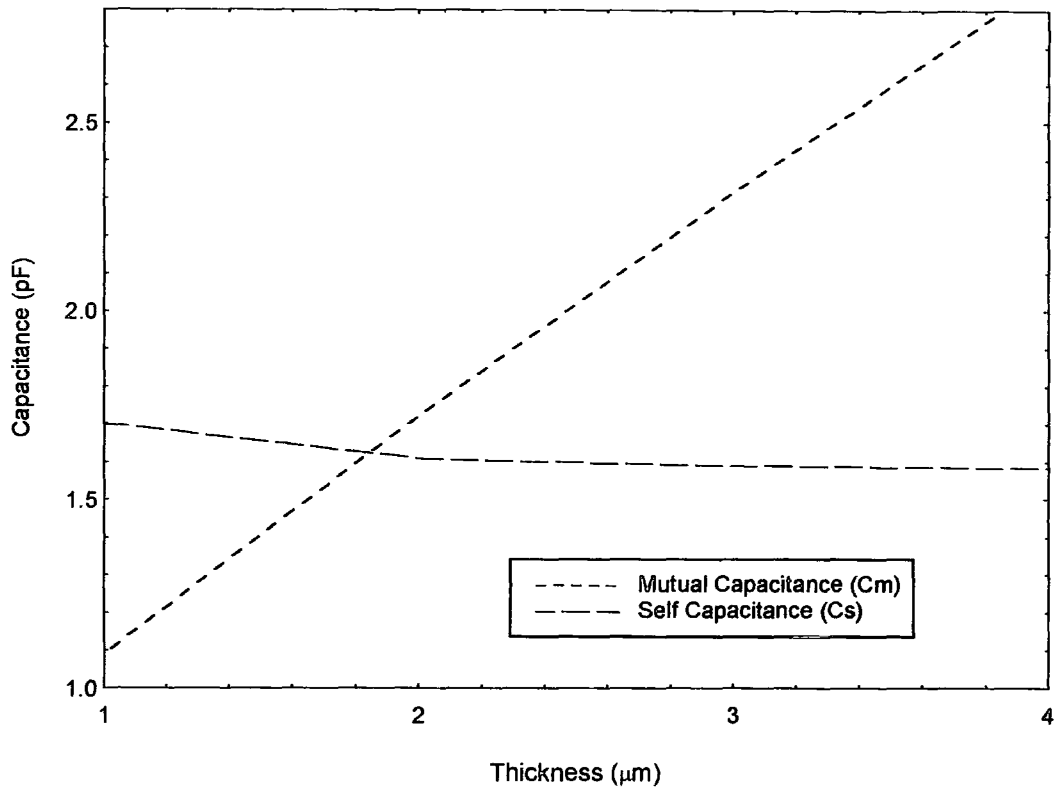

Fig.5.5. Variation of self and mutual capacitance with thickness.

5.4.4 Variation of Capacitance with Separation

The graph in figure 5.6 shows the variation of capacitance when the separation between each of the three tracks is varied between 0 and 3μm while the other parameters are held constant at 1μm. Since to graph 5.5, there is a strong dependency of mutual capacitance to separation between the tracks while there is a much smaller dependency between self capacitance and separation. This can also be explained by considering the example of a vertically-aligned parallel-plate capacitor and equation (5.1). If the distance between the conductors, h in (5.1), is the separation, the capacitance between those conductors will be inversely proportional to the distance between the conductors, as shown in figure 5.6. The smaller increase in self-capacitance as separation increases is caused by the error in Matthaei's method which is discussed in detail in section 3.5.

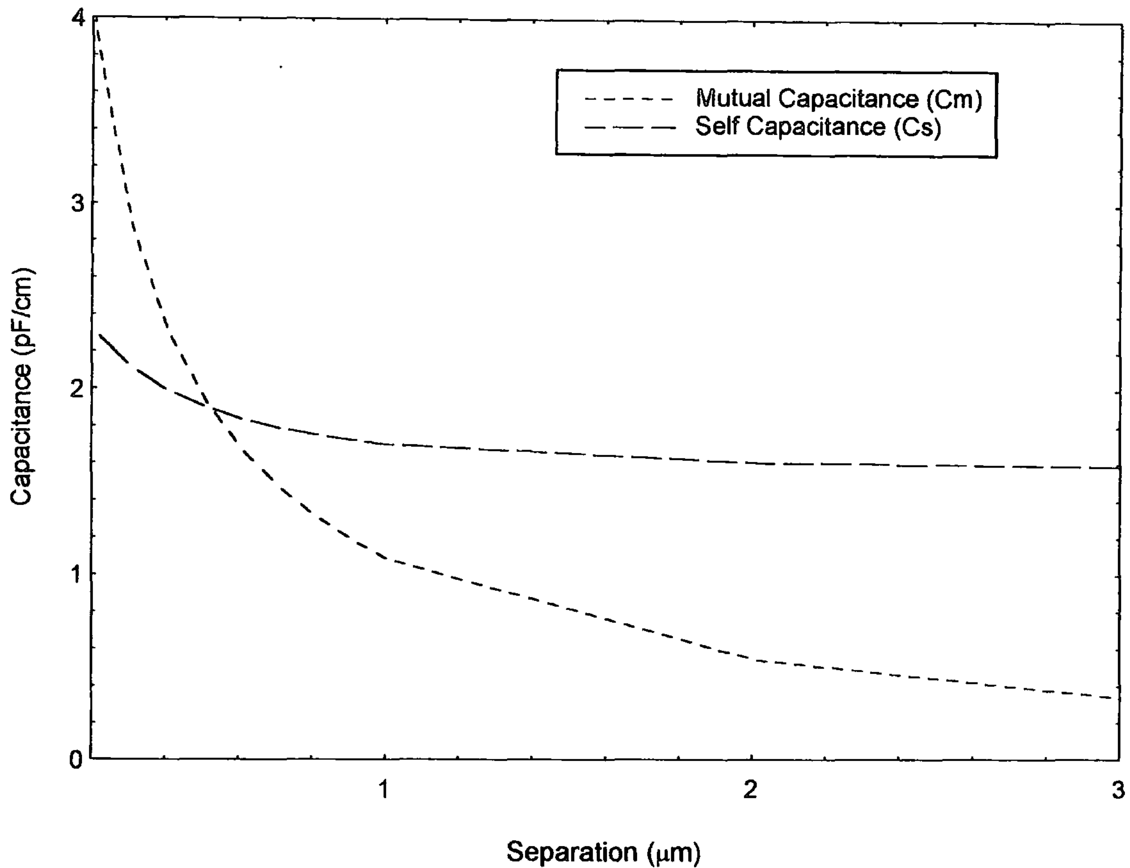

Fig.5.6. Variation of self and mutual capacitance with separation.

5.4.5 Conclusions on Scaling of Interconnect Capacitance

From the graphs shown and discussed in this section, one important trend appears that was mentioned briefly at the beginning of the chapter: mutual capacitance is strongly dependent on thickness and separation while self capacitance is strongly dependent on width and height. Although these trends are fairly obvious from a consideration of even the simplest capacitance determination models, they have a serious impact on scaling and impose limitations on each of the scaling methods.

There are many trade-offs between manufacturing cost, which is dictated by the process complexity and the size of the die of the chip, and the performance of the chip in terms such as performance and power dissipation. Minimising the size of the die by reducing the pitch of the tracks can result in lower performance due to high RC delays. Using multiple metal layers to increase performance while reducing die size requires more a complex process. Increasing the thickness of the tracks results in lower yields due to processing difficulties and increased crosstalk noise. This simplified summary illustrates some of the large-scale decisions involved in designing and optimising very-large scale integrating circuits. In the next section, theoretical and actual process parameters are presented indicating the way industry has dealt with these trade-offs.

5.5 Scaled Track Geometries

Using literature and industry sources, a table of general minimum metal layer 1 process parameters and track design rules was compiled for determining realistic worst-case design conditions for the interconnect models used in the crosstalk simulations described in chapter 7. The source of the 1.5µm, 1.0µm, and 0.75µm parameters was from design rules specification papers distributed by Eurochip for these respective processes[45]. The 0.50µm metal 1 geometries were derived primarily from a paper describing the design of the DEC Alpha microprocessor[46] as well as from sources in literature[47]. The source of the 0.35µm design parameters was from a combination of literature and extrapolation from values given for previous process geometries.

Process Technology	Width	Thickness	Height	Separation
1.5μm	2.4	0.9	0.9	2.4
1.0μm	1.5	0.8	0.8	1.5
0.75μm	1.0	0.7	0.7	1.0
0.50μm	0.8	0.6	0.6	0.8
0.35μm	0.4	0.4	0.4	0.4

Fig. 5.7. Table showing the physical dimensions of track parameters at different scaling levels.

Elements of the quasi-ideal scaling rules described in the previous section are apparent in the table in figure 5.7 in that the thickness and dielectric thickness (height) are scaled much more slowly than the separation and width. It should be noted, that these geometric parameters are used to illustrate general trends in scaling and to describe worst-case crosstalk and RC delay conditions should quasi-ideal scaling rules be used in submicron integrated devices. In reality, the use of these “design rules” on all of the metal layers in a process would lead to large RC delays and significant crosstalk noise problems in long lines. For more realistic processes, these separations and widths would be increased beyond the minimum measurements described in the table and packing density and performance would be maximised through the use of multiple metal layers with varying tracks dimensions.

6. Transistor Scaling and Modelling

6.1 Introduction

The resistance and capacitance parameters and the switching delay of a CMOS transistor driving (or being driven by) an interconnect line determine the crosstalk waveform duration, amplitude and overall shape. Since these parameters change dramatically as the devices are scaled down into submicron levels, device models that precisely represent the true device charateristics are essential for accurate simulations of crosstalk response. The problem of accurately modelling MOSFET transistors is an well documented subject and is its own complete field of study well beyond the scope of this research project.

A first order model of a MOSFET transistor requires an ideal voltage source to approximate the output slope of the signal, an output resistance and a gate capacitance. The values for each of these three components for an arbitrary process method can be relatively easily calculated using scaling theory. A line model representing both the input and output devices was shown in figure 2.4 as was the transistor model used for early simulations in this project. The difficulty with this simplistic model is similar to the problem with the lumped model for approximating line effects. While it is simple to implement and results in faster simulation times than other more complex models, the results of more complex simulations have higher accuracy.

The SPICE simulation software package was used to simulate the transistor models. Originally, this software tool was developed primarily to simulate bipolar transistor second-order effects. Over the years, several accurate MOSFET transistors have been incorporated into it as well with corresponding degrees of accuracy and complexity. This chapter examines the method used to develop and evaluate an accurate HSPICE model for scaled submicron transistors to be later incorporated as drivers of a general line model.

The general method used to generate the simulation parameters required by the HSPICE package is to physically construct the device and use specialised tools to measure specific device parameters combined with the physical parameters of the device itself. Although for the older technologies it would have been possible to construct the actual devices themselves and extract the required parameters, it would have been extremely difficult to gain access to the facilities required to construct the submicron devices that are examined in this project.

An alternative method for determining the HSPICE device parameters is to acquire them from an outside source. While the parameters for less detailed models can often be compiled from industry trade journals and research journals, and older models commonly are available to the public, the most accurate models for current and future technologies have been developed by the larger semiconductor companies for their own simulation and testing. These internal models are generally inaccessible to the public. Fortunately, both Inmos[48] and Eurochip[45] generously allowed their HSPICE models for 1um, 0.75um and 0.50um transistors to be used within this research project.

6.2 Transistor Scaling

Scaling theory is necessary to develop a model with parameters extrapolated from simulations and measurements of existing models and devices. Models for older manufacturing processes can be found in industry trade magazines and in academic literature. Accurate simulation models for more advanced processes, however, are nearly always confidential. At the time that research into this project began, the 0.50μm was just coming into use within the industry and the 0.35μm process was still in development.

Models for the older 3.0μm and 2.0μm processes were readily available through the Internet archives and technical literature, but models for the newer 1.0μm and 0.75μm processes were still unavailable to the public. Using two existing models obtained

through Eurochip[45] under an agreement with Durham University and constant-voltage scaling rules, several different 0.50um and 0.35um HSPICE model were developed. These models were then examined to look at switching characteristics and source resistance. The models most apparently accurate were then used to simulate the interconnect drivers.

6.3 Transistor Scaling Theory

The concept behind first-order "constant electric field" MOS scaling theory was first proposed by Dennard et. al. In this paper it was proposed that the basic operational characteristics of an MOS device can be maintained and the integrity of and operation of a scaled circuit can be preserved if the all critical parameters of a device are scaled in accordance with certain scaling criterion. With constant electric field scaling a dimensionless scaling factor, α, is applied to all three dimensions, the device voltages and the dopant concentration densities.

If all of the dimensions and the voltages are reduced by α and the doping concentration is increased by α, the junction depletion is reduced approximately by the same factor α. Since the voltages are scaled, the electric field within the device remains constant. This has the desirable effect of maintaining large geometry device behaviour in small geometry devices by keeping many of the non-linear parameters relatively unchanged.

Parameter	Scaling Model		
	Constant field	Constant voltage	Lateral
Gate oxide thickness (t_{ox})	$1/\alpha$	$1/\alpha$	1
Device width (W)	$1/\alpha$	$1/\alpha$	1
Device length (L)	$1/\alpha$	$1/\alpha$	$1/\alpha$

Substrate doping concentration (N_A)	α	α	1
Supply voltage (V_{dd})	$1/\alpha$	1	1
Current (I)	$1/\alpha$	α	α
Transconductance (g_m)	1	α	α
Junction depth (X_j)	$1/\alpha$	$1/\alpha$	1
Load capacitance (C_g)	$1/\alpha$	$1/\alpha$	$1/\alpha$
Electric field across gate oxide (E)	1	α	1
Depletion layer thickness (d)	$1/\alpha$	$1/\alpha$	1
Gate delay (VC/I)	$1/\alpha$	$1/\alpha^2$	$1/\alpha^2$
DC power dissipation (P_S)	$1/\alpha^2$	α	α
Dynamic power dissipation (P_d)	$1/\alpha^2$	α	α
Power density	1	α^3	α^2
Power-delay product	$1/\alpha^3$	$1/\alpha$	$1/\alpha$
Contact resistance	α^2	α^2	α^2
Device area	$1/\alpha^2$	$1/\alpha^2$	$1/\alpha$

Fig.6.1. Influence of different MOS-device scaling models.

As can be seen in figure 6.1, the benefits of constant electric field scaling are increased current drive, switching delay, and power dissipation as well as increased packing density. There are, however, many problems associated with constant field scaling. The most serious difficulty is that the operating voltage of the circuit is dependent on the process technology used to make the integrated circuit. Since this would be totally unacceptable to systems and board-level designers, a constant voltage scaling law which keeps circuit supply and logic swing voltages compatible with other logic families has been generally used in the industry. Aside from the primary advantage of simplifying electronics design, the device switching speed is also improved over ideal scaling rules.

The problems with constant-voltage scaling can be separated into three areas: design and manufacturing difficulties, power dissipation problems and reliability issues. Design and manufacturing difficulties include fabrication difficulties, current drive considerations, increased RC delays in signal path and crosstalk noise. Since power dissipation in constant-voltage increases by a factor α, as device dimensions have continued to scale the problem of reducing the heat dissipated per unit area has greatly increased. Initially, this issue was solved through the use of various packaging designs specially designed to dissipate heat. In modern VLSI IC packages, large heat sinks, fans for individual devices and efficient packaging techniques reduce the problems with power dissipation. Other reliability issues include hot electron effects, increased electromigration effects from reduced cross-sectional track dimensions, and reduced tolerance to electrostatic discharge (ESD) due to oxide breakdown. Electromigration is also exacerbated by increases in heat and thus is dependent on power dissipation.

6.4 HSPICE Transistor Model

6.4.1 History of SPICE

In 1968, a junior faculty member at the University of California at Berkeley named Ronald Rohrer and a dozen of his students developed a non-linear circuit simulator which was christened CANCER (Computer Analysis of Non-linear Circuit Excluding Radiation) by a student named Lawrence Nagel. CANCER was capable of DC operating point, DC sweep, AC frequency sweep analysis and transient sweep, all of the same analysis types which are today in SPICE. CANCER supported five basic components: resistors, capacitors, inductors and two types of non-linear devices: junction diodes and bipolar junction transistors and allowed up to 400 components and 100 circuit nodes to be simulated in one circuit.

Throughout 1970 and 1971, Nagel continued to improve the CANCER program and in 1971, this improved version, named SPICE1, was released. The university of Berkeley distributed this program with few restrictions into the public domain where it quickly became an industry standard simulation tool. SPICE offered several improvements over CANCER including macros, improved transistor modelling and better support for larger circuits. Throughout the 1970's, the growth of the integrated circuit industry supported further improvements to SPICE and this led to algorithms optimised for integrated circuit design work.

The next major release of SPICE came in 1975 with the introduction of SPICE2 which improved upon the accuracy and the speed of transient analysis of simulations by developing two dynamic timestep control algorithms and a multi-order implicit integration scheme. It also improved the formulation of voltage-defined elements such as inductors, and voltage and current sources. Support for new device models was also added to keep pace with device technology changes. In 1983, SPICE2 version G.6 was released and this is the version upon which the majority of SPICE compatible simulation packages are based.

The final release of SPICE, SPICE3, was to be a superset of the original SPICE and CANCER programs with the addition of newer features such as pole-zero analysis, and voltage and current controlled switches, as well as the improved device models that each successive generation had previously brought. Unfortunately much of the driving enthusiasm that was available in the early 1970's had disappeared, and the overwhelming task of converted the bulky Fortran program into a C program was left to undergraduate and graduate students. The first release, SPICE3A, was largely incompatible with previous versions of SPICE and contained numerous errors and bugs. Even though Berkeley has continued to improve upon the program, SPICE3 has never gained the popular acceptance of previous releases.

Throughout the 1980's several commercial releases of SPICE were developed. Among the most popular are Software Spectrum's MICRO-CAP, IntuSoft's IS_SPICE, MicroSim's PSPICE, and Meta-Soft's HSPICE. Although others have more recently developed based upon SPICE3, all of these that have been mentioned are based on SPICE2. The particular version of SPICE used to simulate the interconnections and the driving transistors was released in 1990. HSPICE v.h92b. by Meta-Soft incorporates all of the features of the original SPICE2 program as well as additional features such as lossy transmission lines, more advanced transistor models, an polynomial-dependent voltage sources, among many others. Two specific features of HSPICE over more generic SPICE versions that were used throughout the project are the support for more advanced models which were used to develop more accurate models of scaled devices and the lossy transmission line model which was used originally as a base

6.4.2 Transistor Modelling Using HSPICE

Similar to SPICE3, HSPICE is written in the C language which makes it more modular and portable than the original SPICE2 which is written in Fortran. The version used in this project, HSPICE v.h92b., by Meta-Soft incorporates all of the features of the original SPICE2 program and other circuit analysis programs, as well as enhancements and features developed in house by Meta-Soft. This flavour of SPICE adds the following features[49]: improvement on simulation convergence - a well known problem with SPICE2, accurate modelling of integrated circuits by allowing the use of a wide variety of commercial foundry models, automatic parameter generation of device models from a given set of measured data, lossless and lossy transmission line models for integrated circuit, package and PCB (Printed Circuit Board) technologies, a graphical viewing program for results analysis, and statistical and sensitivity investigation using Monte Carlo analysis. The two specific features of HSPICE which were of particular interest are the support for more advanced models which were used to develop more accurate

models of scaled devices and the lossy transmission line model which was used originally as a reference for initial simulations of interconnects.

In 1993-4, 0.50um technology was considered to be the next generation technology for most of the major semiconductor companies and in 1995-6 this technology is beginning to be used in the bulk manufacture of devices. In order to anticipate the problems that could be expected from submicron circuit designs in the future, simulation parameters were required for even smaller devices, such as 0.35um and 0.25um. For these deep submicron transistors, models were unavailable and it was necessary to extract new HSPICE parameters to describe these devices. The basic method used to extract these parameters was to use basic transistor scaling theory to adjust the transistor models for the 0.75um and 0.50um devices.

In general, each of the models presented below was originally developed for a 0.75μm or 1.0μm process. Through the use of scaling theory these models were adapted for smaller processes. These results are unrealistic only in that a 5V voltage source applied as V_{DD} is assumed for all scaling levels whereas in modern deep submicron devices 3.3V and 2.7V are more commonly used. This constant-voltage scaling is used primarily to simplify analysis of the results.

In addition, since higher source voltages result increase the coupled crosstalk voltage, this assumption provides a worst-case situation. The results were then interpreted using an extrapolation of the switching characteristics of these original devices to determine the expected transition times for scaled devices. Graph 6.2 below shows a comparison of the switching characteristic curve of the fourth device in a chain[§] of 0.50μm inverters which are falling from a high state to a low state.

[§] A chain of inverters is used in favour of a single inverter to provide a more realistic slope and transistion time than the output of an inverter driven dirrectly from an ideal voltage with a pulse waveform would supply.

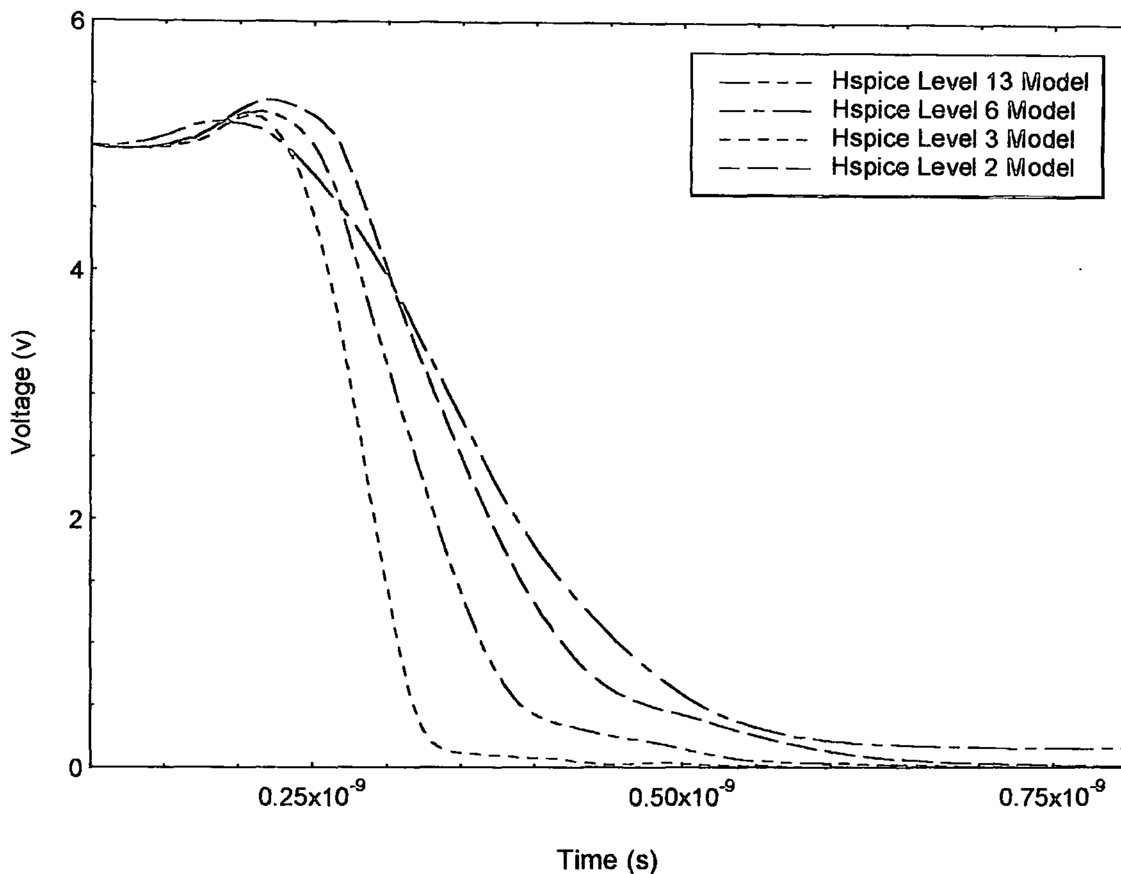

Fig.6.2. Comparison of HSPICE process models for the falling transition of a 0.50μm inverter .

The models were evaluated for accuracy using figure 6.2 and the output of the HSPICE file which listed device resistance, and device capacitance parasitics. As can be seen, there are significant changes in transition time, overshoot, and voltage levels between these models. It can be seen that in the level 6 model, the low state output is roughly 200mV above V_{DD}. Adjustments were made to the model to try to improve this output voltage to closer to 0V, but were unsuccessful. The output voltage for a high state, however, was within 2mV of V_{DD}. The slowest of the models were the level 6 and level 2 models and these two were disregarded as too optimistic in terms of worst-case crosstalk noise since crosstalk noise is dependent on the speed of the switching device.

The level 13 model was the most accurate based on 50% and 10%-90% transition times which were extrapolated from literature, but the level 13 model is also the most complex model and an voltage spiking irregularity was found in simulations of the 0.35μm model. Finally, the level 3 model parameters was available for the 0.75μm, 1.0μm and 1.5μm processes, while the level 13 model was only available for the 0.75μm process. To create models for the earlier processes would require a significant effort and for all of these reasons the level 3 model was implemented in the project simulations.

6.5 Comparison of Scaled Transistor Models

Using the level 3 transistor parameters, models were created for 0.35μm, 0.5μm transistors and existing commercial models were used for the older 0.75μm, 1.0μm, 1.5μm processes. Comparisons made between these inverters for each of these processes are described in this section. In each of the graphs on the following page, an inverter chain is used. A line resistance of 50Ω and a line capacitance of 1pF are used between each of these inverters to represent the typical loading conditions of large lines. Each of the inverters is minimum length for that process and is a constant width. A width of 3μm for n-channel devices and 7.5μm for p-channel devices is used for all of the scaling levels.

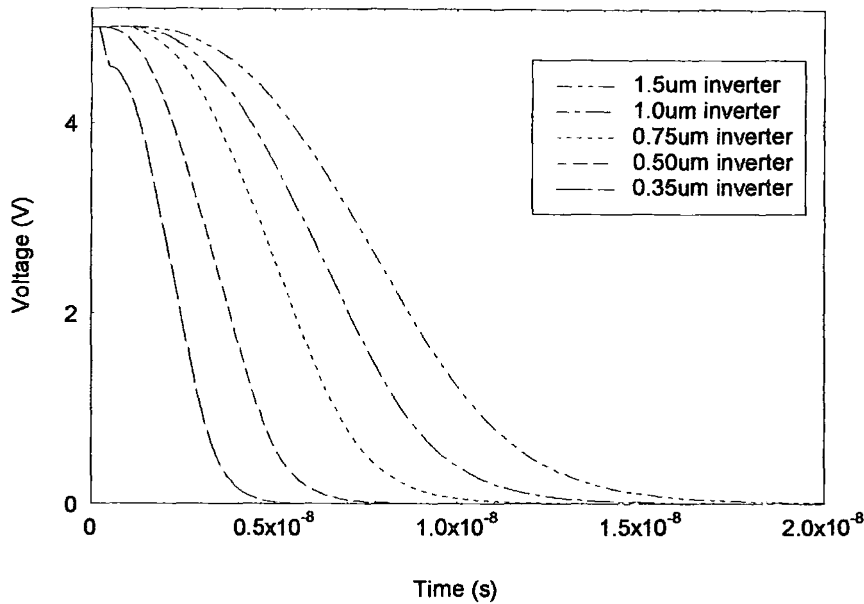

Fig.6.3. Comparison of the transition times of scaled transistors.

Figure 6.3 above shows a graph relating the transition times from high to low for the fourth transistor of the interconnect model for each of the scaled processes. The slight error that can be observed at the beginning of the transition for the 0.35μm inverter is caused by an inaccuracy in the model. This graph indicates the relative improvements in switching speeds offered by improvements in the process technology. It can be seen that this improvement is roughly linear through each succeeding generation.

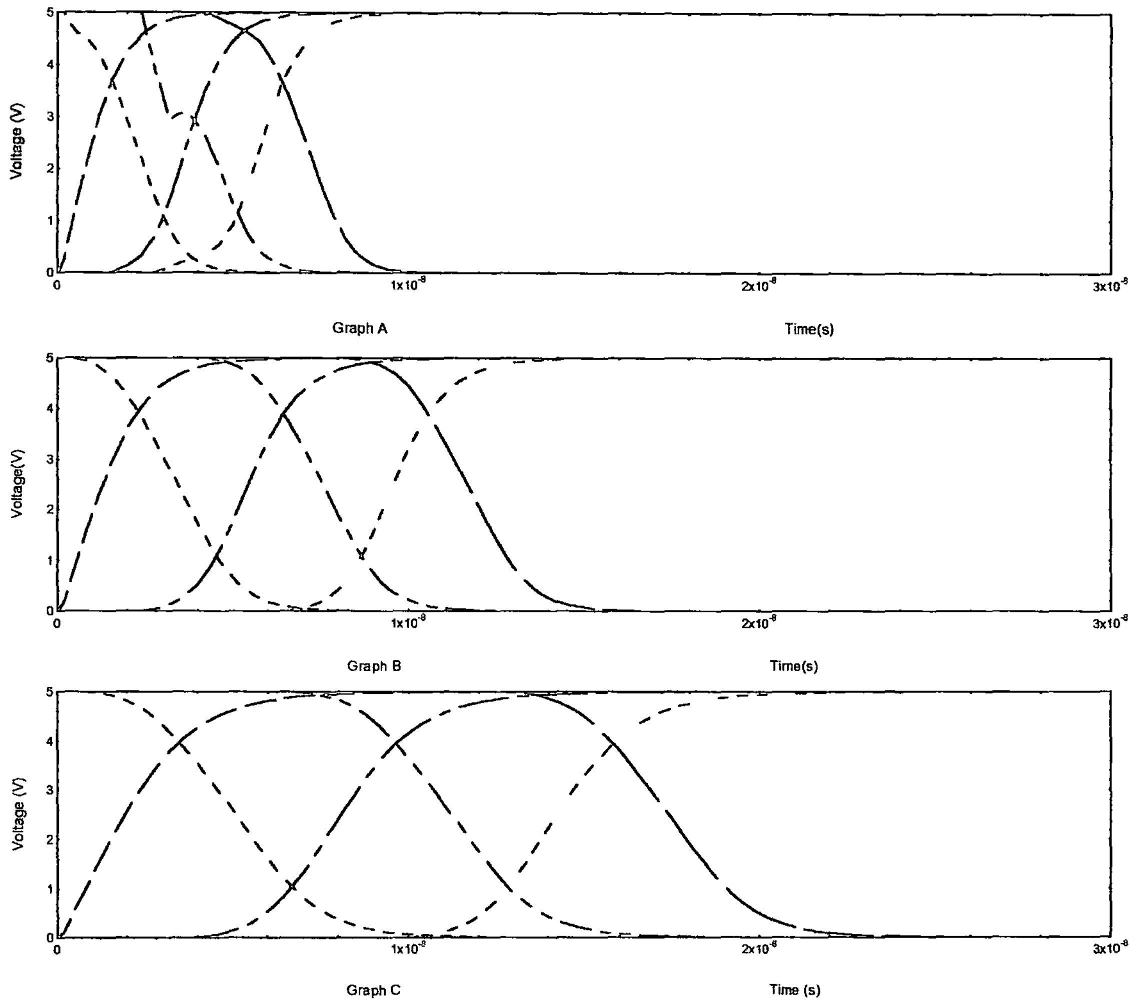

Fig.6.4. Comparison of the switching speeds for a chain of scaled inverters. Graph A. shows a 0.35μm inverter chain, while graphs B. and C. shows the switching of 0.50μm and 0.75μm devices respectively.

The graph shown in figure 6.4 shows the transition speed as the signal change propagates through each device in turn in a six device inverter chain. This graph shows essentially the same information as that shown in 6.3, but in a different form. In this case, the relative delay of a signal moving though an entire chain is shown, rather than the delay through a single device. Again the linearity of the speed improvement is can be seen. The flaw in the 0.35μm model is more evident in this graph. The cause of this error was

never discovered, but is only apparent under light loading conditions and has a relatively insignificant effect on the crosstalk results discussed in chapter 7.

7. Crosstalk Delay and Noise

7.1 Introduction

Crosstalk noise can be defined as the unwanted coupling of signals from one conductor onto another conductor. It is caused by the inherent capacitance and inductance that exists between neighbouring conductors. In the past at 3μm and 1μm technology levels, the problem of crosstalk noise in silicon circuits could effectively be ignored due to the fact that both inductive and capacitive coupling between signal wires was minimal, lines were typically short and device switching speeds were relatively slow . Inter-track inductance was negligible due to the highly resistive properties of silicon interconnects and the comparably low switching speeds of CMOS circuits. Coupling capacitance was also insignificant since the relatively large linewidths of the tracks and the large separations between neighbouring interconnects caused the self capacitance of tracks to dominate over their coupling capacitance.

As integration levels have increased over the years, the problem of crosstalk noise has gradually become more of an issue. Since the industry has followed quasi-ideal scaling rules in order to reduce problems with electromigration and high resistance, the coupling between adjoining tracks has increased. In the following section the problems that crosstalk noise presents to design and reliability engineers will be presented. It will be argued that, in this authors opinion, the most serious problem with continued use of quasi-ideal scaling rules at submicron geometries is crosstalk.

This chapter first looks at the actual problem that crosstalk noise presents to engineers. Following this, a simplistic example of the coupling of a signal from an active line to an inactive line is shown and the basic mechanism is explained. In the third section, the simulation results from scaled models and from scaled lines are presented and the noise waveforms from these models are evaluated. The problems caused by delay problems

due to capacitive coupling, a problem that may be termed "crosstalk delay", are examined in the fourth section. The results of a wide variety of simulations are then presented and analysed before concluding with a series of simulations that attempt to illustrate the impact of various strategies aimed at reducing the impact of crosstalk noise.

7.2 Introduction to Crosstalk Noise

Figure 7.1 shows a simplified schematic of a 2-bit signal bus being driven by two inverters which will be used to explain the effect of crosstalk. In this representation, the effects of inductance and resistance are ignored, the devices being driven at the end of the lines are omitted and the lines are modelled as a lumped capacitance. As will be shown in the graph in figure 7.3, the signal line is falling from 5V to 0V, while the coupled line is held constant at V_{high} = 5V. When the signal line rises, the voltage across the capacitors changes correspondingly. In the initial case at time, t, equal to zero, the coupled line and the signal line are both at 5V and the mutual capacitance between the two line, Cm, is not charged at all. The two self capacitances between the tracks and the ground plane are charged to a value, Q_s.

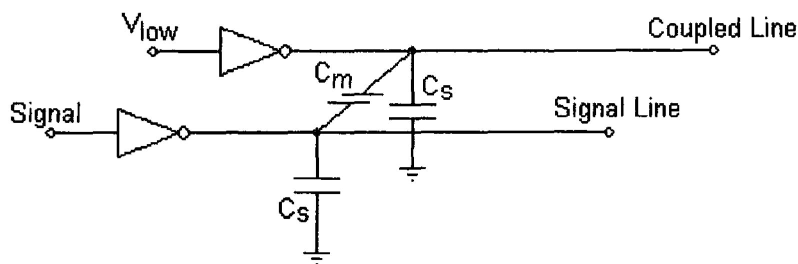

Fig. 7.1 Simple example of crosstalk coupling on a 2-bit signal bus.

When the input signal changes, the voltage of the signal line changes from 5V to 0V in a finite time interval. The mutual and self capacitances act as a capacitive divider and so, during this interval, the current though the inverter is given by:

$$i = \frac{C_m}{C_s}\frac{dV}{dt} \qquad (7.1)$$

Figure 7.2 can be used to simply figure 7.1 by considering the inverter driving the inactive line as a series resistance and by representing the switching of the inverter on the active line as an pulsed source.

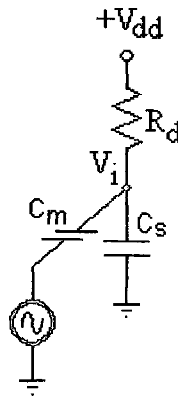

Fig. 7.2: A simplified representation of the circuit in fig. 7.1 while the signal line inverter is switching.

The current, i, through the series resistance of the device and the track (summed together as R_d in figure 7.2) causes a voltage drop at node V_i. Thus, the crosstalk noise seen at node V_i due to this voltage drop can be described by using equation (7.1) and Ohm's law to give:

$$V_{cr} = \frac{C_m}{(C_s + C_l)}\left(\frac{R_t}{R_d + R_t}\right)\frac{dV}{dt} \, (7.2)$$

where R_d and R_t are the device and track resistance respectively. The graph shown below in figure 7.3 shows the voltages seen on the two tracks. The line denoted as the active line in the graph is switching from high to low while the coupled line remains

fixed high. The crosstalk voltage is the dip down to approximately 3.9V seen in the inactive line during the transition of the active line.

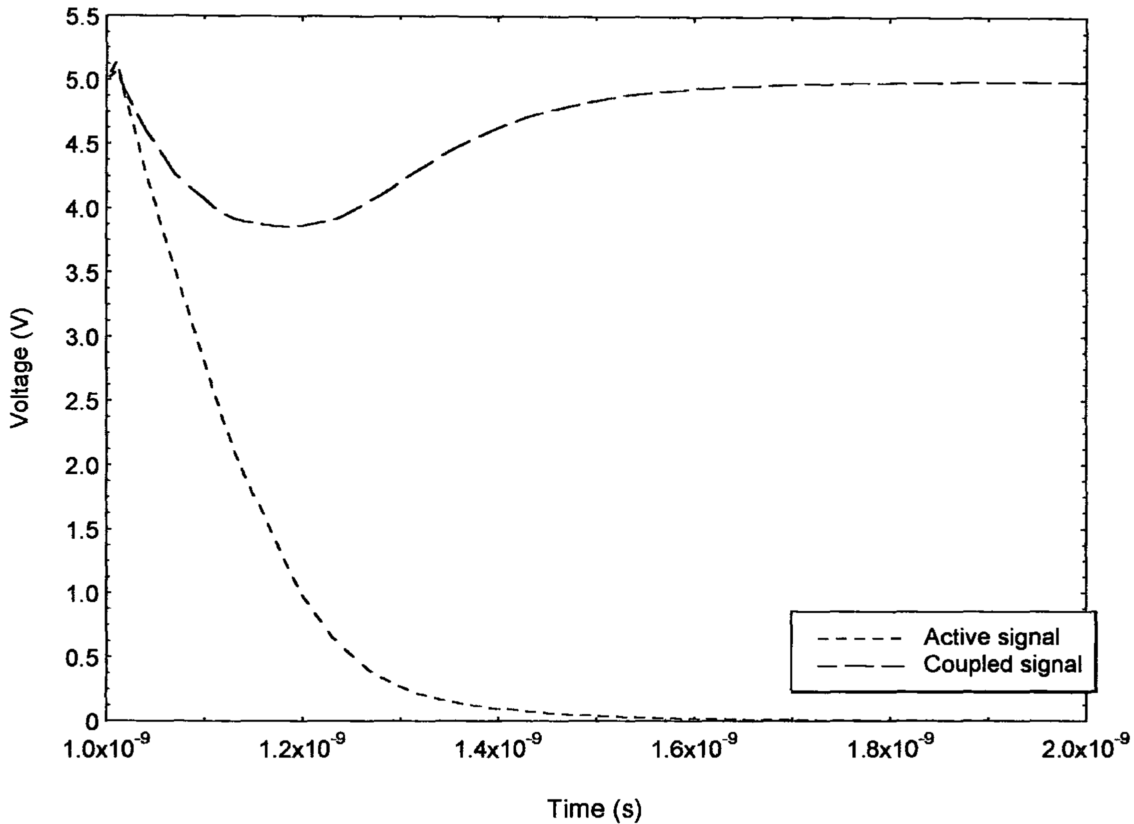

Fig. 7.3: Graph illustrating signal coupling between an active and an inactive line.

As stated at the beginning, this example ignores the line resistance and inductance, the gate capacitance and parasitic effects on the devices being driven, the effects of other neighbouring lines and is a lumped model of the distributed capacitance on the line. Although simplistic, the example shown in figure 7.3 illustrates the basic principles behind crosstalk noise and signal coupling.

Two measurements are mentioned in the following sections: peak crosstalk voltage and the $V_{50\%}$ crosstalk duration. The peak crosstalk voltage measures the maximum peak of the crosstalk signal. It is measured using the absolute value of the signal referenced to

the signal voltage when there is no coupling present. In other words, when the coupled line is held at 5V and the crosstalk signal causes a maximum drop of 500mV, this is referred to a 500mV peak crosstalk signal rather than a -500mV signal. The V50% pulse duration measures the width of the crosstalk pulse at half of the peak crosstalk voltage.

7.2.1 Analysis of Crosstalk Noise on Parallel Lines in Scaled Bus Structures

To simplify the explanation of signal coupling on a bus the example of a bus structure made up of 9 parallel bit lines each switching simultaneously will be used. In this case, all of the data on the bus is latched onto and off of the bus on a clock edge. On such a bus the worst case coupled crosstalk signal would occur when all 8 other bit lines are switching simultaneously in one direction while the central line remains at a constant voltage.

For example when a set of all low bits switch high while the central line remains low (the layout of these tracks will be shown in fig.7.7(a.). In this case, the centre line will charge up the mutual capacitance between it and all of the other neighbouring lines. The current required to create this charge must come from the driver of the line which will have a finite device resistance and through the distributed resistance of the line itself. While this mutual capacitance is charging through the in series resistance of the driver and the line the voltage on the line will increase to offset the charge imbalance created across this mutual capacitance and the signal on the centre line will seem to "follow" the signals on the other lines. This spurious signal is coupled from the other lines.

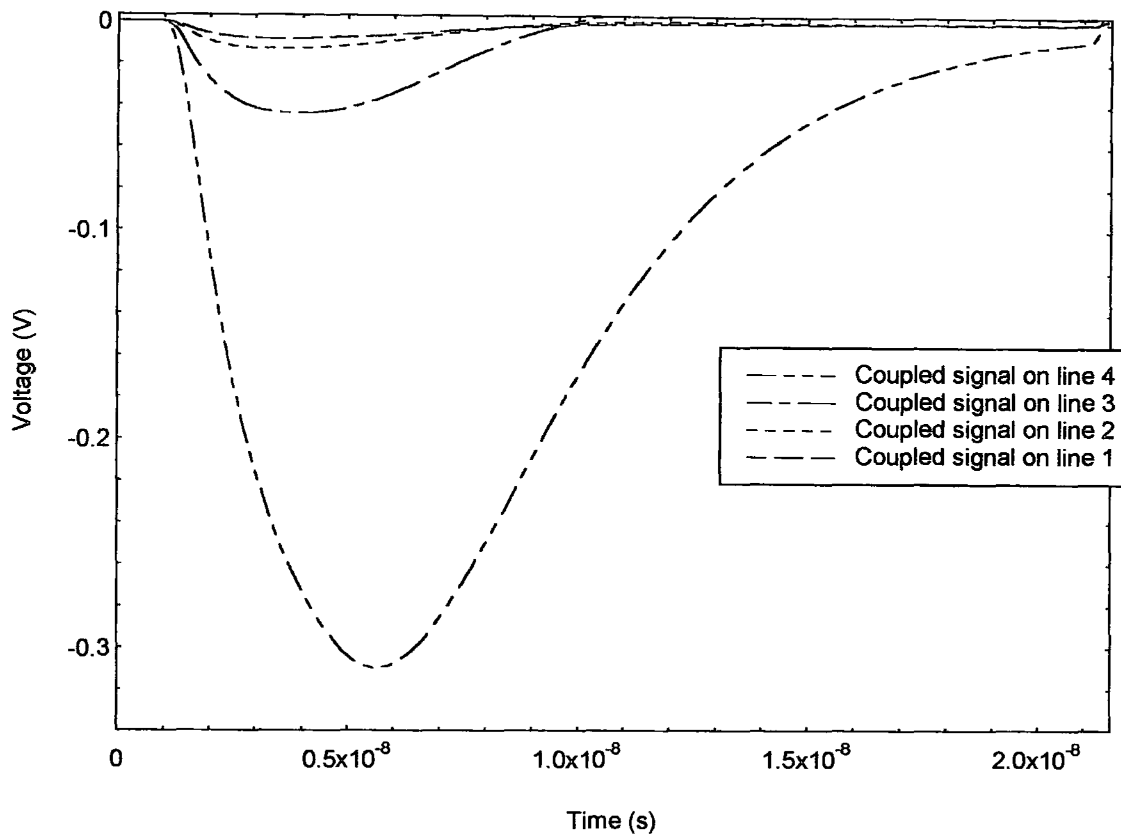

Fig. 7.4. Coupled crosstalk signal on each signal line in an 9 bit bus.

In the case shown above in figure 7.4, one single bit line is switching while the other nine neighbouring lines are held constant at 5V. The lines are driven (and are driving) 0.35μm inverters and are 0.1mm in length and are modelled using a 3-stage RLC π model. The pitch of the line is 1.0μm and the thickness and height of the line are both 0.4μm. The lines are assumed to be made of aluminium and the simulated temperature is 27°C. For reference in the graph and discussion below, the lines will be numbered across from one to nine, with the centre line labelled five.

Considering the graph shown in 7.4, it can be seen that the amplitude of the of the signal coupled is greatly dependent on the number of conductors between the active line and the coupled line. In a typical system with more than 9 conductors, the single coupled to the farthest conductors is less than 1% of the value of the signal coupled onto the nearest

conductors. At the same time, the amount of time required to determine the interconnect parameters, the length of the simulation file and the amount of time required to simulate the system are all dependent on the square of the number of conductors. Using simulations with up to 18 conductors, it was concluded that a reasonable compromise between simulation accuracy and complexity was to limit the system to less than nine conductors.

In figure 7.4, only half of the bit lines are shown since the other four are the same due to symmetry. In the figure it can be seen that the worse case coupling occurs between the centre line and the two closest neighbouring lines, labelled lines 3 and 4. The coupled signals on lines 1 and 2 are comparatively much smaller. This can also be seen in the capacitive matrix for the system, again divided in half for simplicity, shown below in equation (3.15).

$$\begin{bmatrix} 1.65 & 0.46 & 0.037 & 0.016 & 0.012 \\ 0.46 & 1.81 & 0.44 & 0.032 & 0.16 \\ 0.037 & 0.44 & 1.82 & 0.44 & 0.037 \\ 0.016 & 0.032 & 0.44 & 1.81 & 0.46 \\ 0.012 & 0.016 & 0.037 & 0.46 & 1.65 \end{bmatrix} \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} \\ C_{51} & C_{52} & C_{53} & C_{45} & C_{55} \end{bmatrix} \quad (7.3)$$

The coupled signals on the farther lines are effectively "shielded" by the much closer lines, in this case, C_{12}, is much greater than C_{13}, C_{14} and C_{15}. The signal coupled between lines 3 and 5 is approximately 25% smaller than that coupled between lines 4 and 5, but if the fourth conductor were not there, then the coupling would be much larger. This effect can be seen if we removed the shielding conductor.

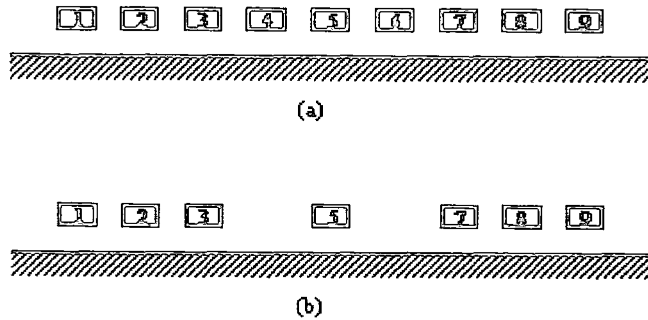

Fig. 7.5. Illustration of the nine bit bus. (a) the complete nine bit bus, and (b), with two conductors removed.

In figure 7.5, the fourth conductor in this system has been removed while the spacing between the conductors has remained constant. Thus, the shielding effect that the closest conductor has over the more distant conductors is removed, and the mutual capacitance between conductors 1 and 3 (which could now be renumbered as 3 and 4) increases when compared to the previous case.

$$\begin{bmatrix} 1.49 & 0.46 & 0.041 & 0.016 \\ 0.46 & 1.66 & 0.45 & 0.024 \\ 0.041 & 0.45 & 1.81 & 0.123 \\ & & & \\ 0.016 & 0.024 & 0.123 & 1.65 \end{bmatrix} \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} \\ C_{51} & C_{52} & C_{53} & C_{45} & C_{55} \end{bmatrix} \quad (7.4)$$

In this capacitance matrix, the blank row and column represent the removed conductor. Comparing C_{35} in the matrix shown in equation (7.3) and C_{35} in (7.4), there is an increase of more than three times as much mutual capacitance between the shielded line and the unshielded one. The other lines also show a much higher mutual capacitance to the centre conductor as well. Thus the introduction of a constant voltage conductor between two switching signal lines has a shielding effect on the coupled crosstalk noise

This effect of shielding bit lines will be discussed in the section on reducing the impact of crosstalk noise in section 7.4.

7.2.2 Analysis of Crosstalk Noise at Sampling Points along a Distributed Line

Due to the distributed self and mutual capacitances and line resistance in the coupled line and in neighbouring interconnect lines, the amplitude of the coupled crosstalk voltage is different when sampled at various points along a line. A series of simulations was made in order to examine the difference in crosstalk noise at different points along a line. Comparisons were then made between the coupling seen at various points along a distributed interconnect at different levels of interconnect scaling. For simplicity, only two of the scaled processes are discussed in this section, but simulations using the other technologies result in similar trends to those discussed.

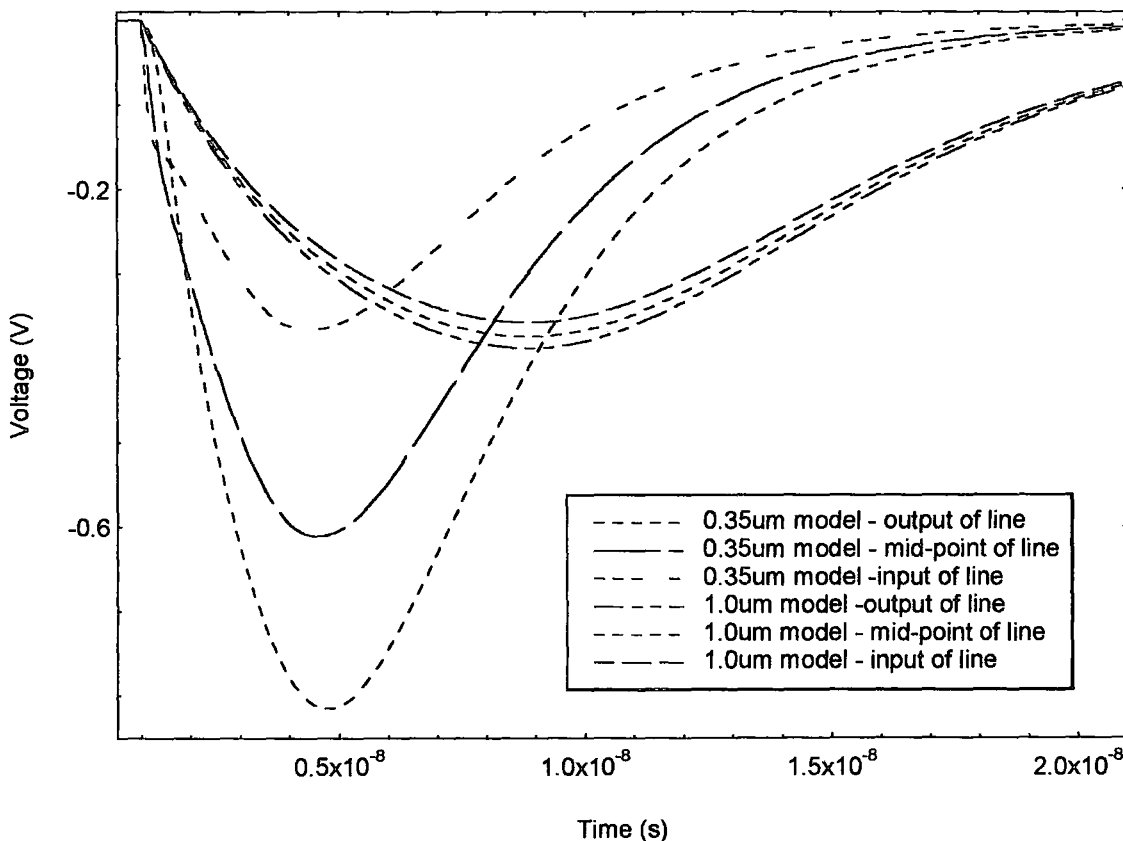

Fig.7.6. Comparison of the coupled voltage at various point along a line at different scaling levels.

In the graph shown above in figure 7.6, the input to the central line of a nine bit bus driving by inverters is held constant at 5V, while all of the neighbouring lines switch from 5V high to 0V low signals. The lines are driven and are driving inverters that are 3μm wide and are the minimum length for that process. The lines are all 1cm in length, with the geometries consistent with those described in the table in figure 5.7 for each process. All of the interconnections are modelled using a 3-stage RLC π model at 27°C. The amplitude of the coupled signal is shown at three points along the line: the output of the first inverter, the centre point on the line and the input to the second inverter.

As shown in figure 7.6, there is a large increase in the crosstalk signal between the two scaled processes. While the coupled signal seen on the 1μm track peaks at 390mV, the crosstalk signal seen on the 0.35μm tracks has a maximum amplitude of 788mV. The graph also illustrates the differences between the two processes in terms of the coupled signals seen at each of the points along the tracks. In the 1μm process, the percentage increase between the signals coupled at each of the three points is approximately 4%. This is significantly smaller than the percentage increase of the crosstalk voltage at the three points on the 0.35μm scaled technology which between 35% and 66%.

Figure 7.6 also shows that the duration of the crosstalk pulse is shorter with reduced device size. At the point at which the signal has reached 50% of the maximum voltage, the signal duration is 13.38ns and 6.69ns for the 1μm and 0.35μm processes respectively. Also worth noting is that the overall shape of the coupled signal is different between the two processes. It can be concluded from this graph and from others which compare all of the scaled processes, that not only does the coupled crosstalk signal increase with increased scaling, but signal propagation through the track is different. As scaling levels are reduced, this change in signal propagation is caused by a combination of the increased resistance of the track, the switching delay of the transistor being

reduced, and the wavelength of the interconnect signal approaching the physical geometries of the track causing transmission line propagation.

7.2.3 Evaluation of the Contribution of Inductance to the Crosstalk Signal

Although the effect of the contribution of inductance at submicron technologies on the signal propagation and on signal coupling is discussed in general terms in section 2.2, the development of more accurate device models allowed a more specific and accurate assessment of the influence of inductance on the crosstalk simulation results. Since the inclusion of inductance into the simulation model increases the complexity of the model and thus results in longer extraction and simulation times, a comparison of identical models, one with inductance and one without, would allow a determination of the whether or not to include the inductance parameter in the interconnect model. If the effect of the inductance is sufficiently small, then this parameter can ignored for the general model at that scaling level.

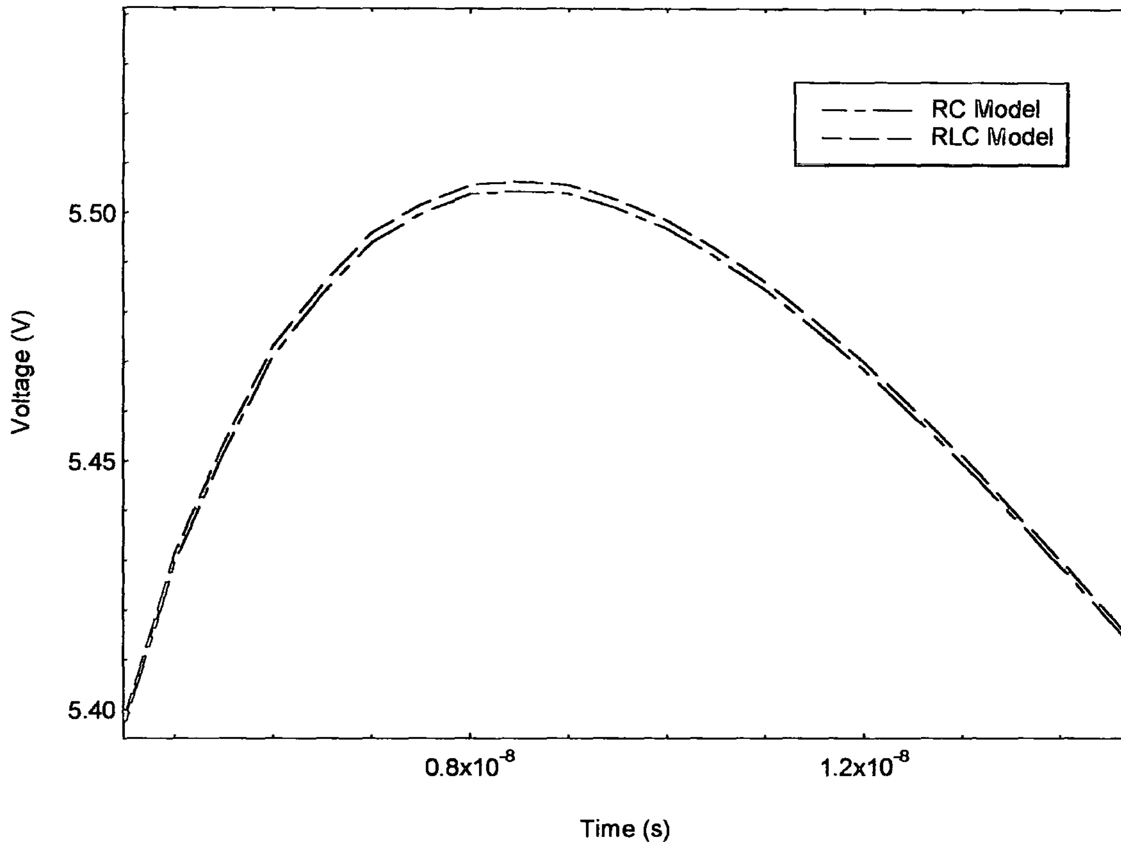

Fig. 7.7. Comparison crosstalk signal noise for RC and RLC line models on a 0.35μm process.

A comparison between the crosstalk signal of an RC interconnect model and an RLC model for the same interconnect is shown above in figure 7.7. In order to reduce resistance and increase the line inductance, tracks that were long(1cm) and wide(4μm) with relatively small separations(2μm) were used. Minimum 0.35μm transistors with relatively wide(8μm) channels widths were also used. The dielectric thickness and interconnect thickness were typical (0.4μm) for the estimated 0.35μm process. The simulation temperature was 27°C, V_{dd} is 5V and the number of parallel interconnections is three.

Figure 7.7 shows a small section of the peak of a coupled crosstalk signal. The input into the inverter driving the centre interconnect is held constant low, while the inputs into the inverters driving the two neighbouring signal lines are driven from high to low. The

difference shown between the two crosstalk signals on the centre line is noticeable but minimal. The largest difference between the results of the two models occurs at the peak of the crosstalk signal. At this point there is a 40mV difference between the RC and RLC models. This difference results in the RLC model having a nearly 8% increase in peak signal voltage over the RC model.

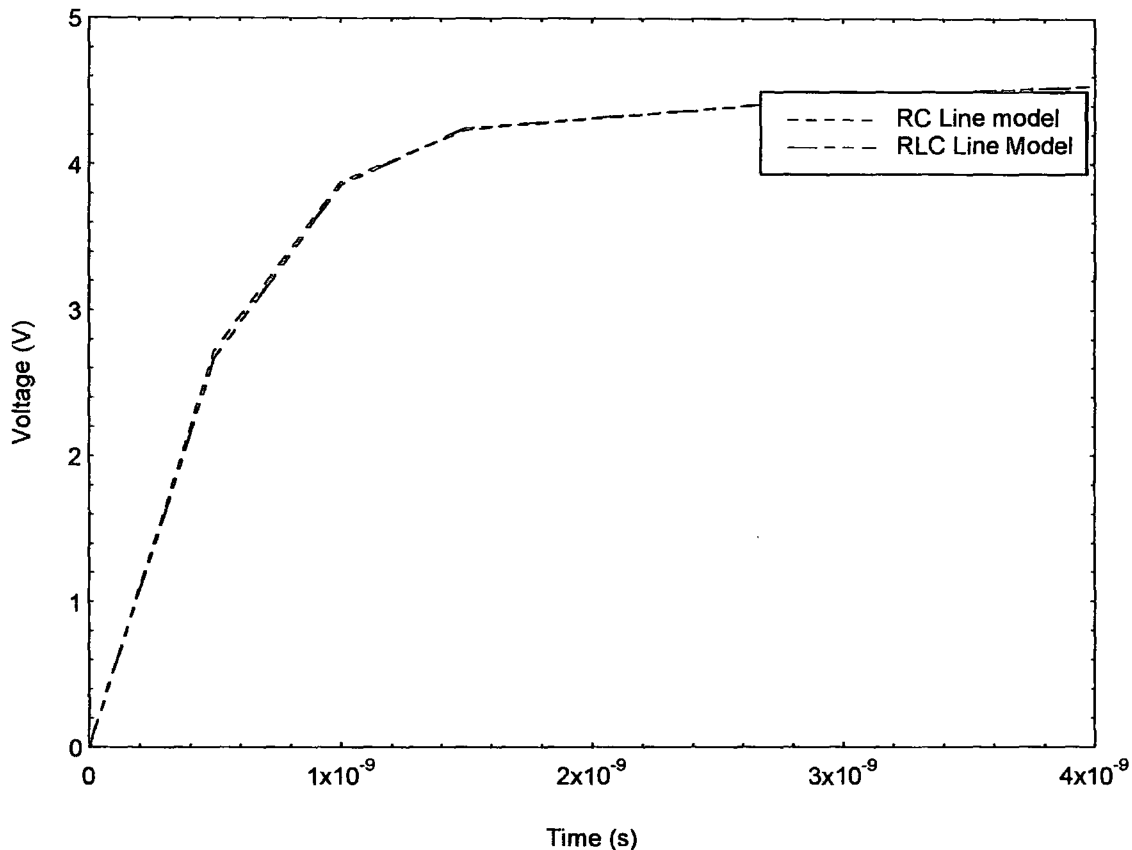

Fig.7.8. Comparison of the output slope of an interconnect of RC and RLC line models

The graph shown in figure 7.8 was produced using the same system described above for figure 7.7. In this case, the graph shows the difference between the slopes of the RC and RLC at the input to the line output inverter. It illustrates that the difference between the slopes for RC and RLC models is effectively negligible. Using a conductor and device system that was designed to maximise transmission line effects, there is only a 20mv at

the 50% V_{dd} point in the transition. This difference results in a difference between the rise times of the slopes of approximately 5ps.

The results from the graphs shown in figures 7.7 and 7.8 indicate that, while inductance contributes have an effect on the slope delay and the crosstalk voltage, this effect is relatively minor in typical interconnects at 0.35μm. For systems which involve the routing of timing critical global interconnects, such as clock networks, in deep submicron technologies, the influence of inductance on the system timing will be appreciable, but for shorter routing at 0.50μm and 0.35μm processes transmission line effects have only a minor impact.

7.2.4 Effect of Temperature on Crosstalk Noise

As illustrated in equation (7.2), crosstalk noise is a function of the device and track resistance as well as the switching speed of the driving device. Since all three of these parameters are temperature dependant, the crosstalk signal must also be dependent on the temperature of the system. Several simulations were performed to examine the influence of temperature on crosstalk.

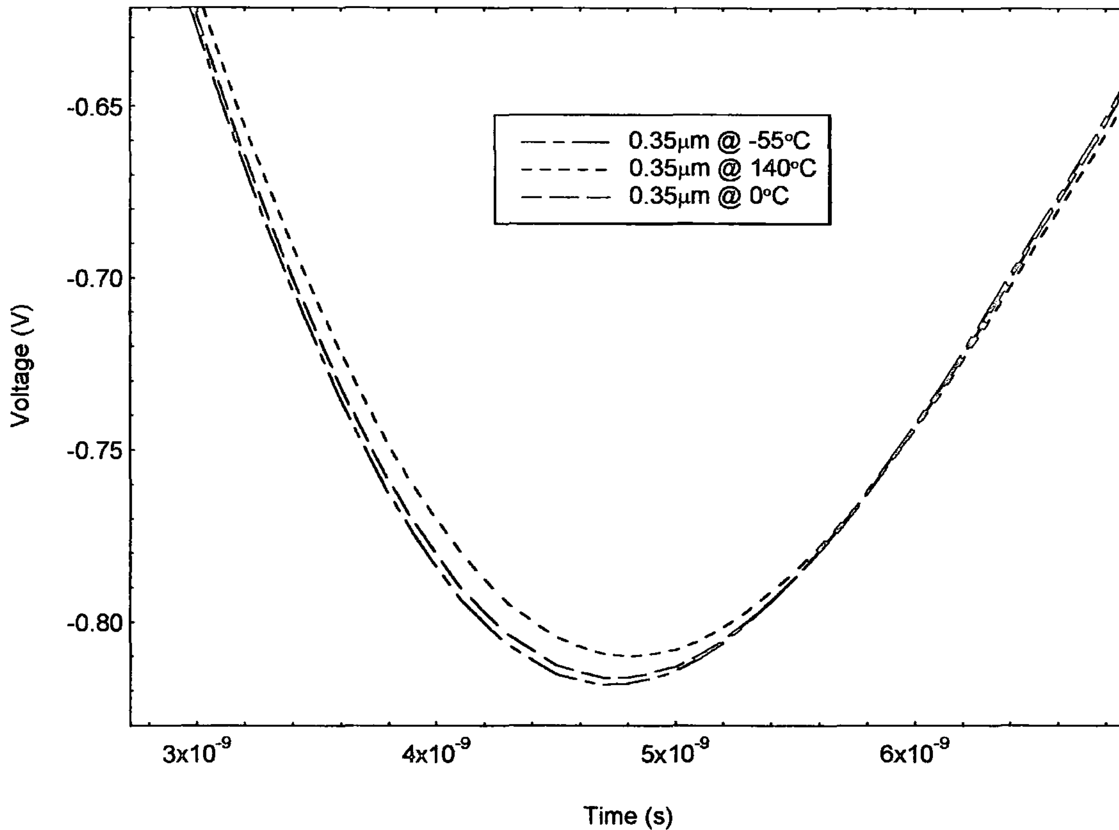

Fig.7.9 Comparison of coupled crosstalk voltage against simulation temperature

The system simulated in the graph above in figure 7.9 is a 0.35μm system of 8 conductors driven by, and driving, minimum channel length inverters. The bus is 1cm long and the interconnect dimensions are as described in table 5.7. The signal shown is the centre signal which is held at 5V by the inverter, while all of the other tracks are switching from high to low (5V -> 0V). In this case, all three of the systems are identical except for the change in simulation temperature.

Considering equation (7.2), there are two effects that are expected when the simulation temperature is increased. At higher temperatures the duration of the crosstalk pulse would be expected to be longer since the device would switch more slowly. The peak crosstalk voltage is expected to remain relatively constant since expected increase due to

the rise in line resistance is offset by the slower switching speed and increased device resistance. Both of these expectations can be seen in the graph shown in figure 7.9.

The low temperature signal is the fastest and has the largest peak coupled voltage. As the system heats up, the device switches more slowly and the device resistance increases, resulting in a reduced crosstalk signal with a longer duration. Measuring the voltage differences between the signals seen at different temperatures indicates that this signal variation is very minor. The difference in coupled signals between the simulations at $-55°C$ and 0°C results in a peak voltage difference of 2mV and a V50% delay of 23ps while the nearly 200°C temperature difference between the low and high temperature simulations resulted in a difference between the peak voltages of 6mV and a delay shift of 50ps.

The interconnect model used does not directly support accurate temperature simulations, since the change in dielectric properties and in second order resistance effects are not considered. Although these results are not precise in terms of the magnitude of the voltage and delay shifts, they do illustrate the general effect of temperature on crosstalk noise in a system. Although the change in signal is small, the temperature of the other simulations described in the paper is cited for reference.

7.2.5 Analysis of Crosstalk Noise in Scaled Bus Structures

As was discussed in the section on the quasi-ideal scaling of interconnects (section 5.3), the use of quasi-ideal scaling rules leads to an increased mutual to self capacitance ratio (C_m/C_s) which in turn leads to increased signal coupling. It is expected that the crosstalk noise in submicron circuits will be much higher than that seen in processes which use larger feature sizes. Using the interconnect layout dimensions described in table 5.7 a series of simulations were performed to analyse the effect that scaling had on the worst-

case crosstalk signal coupled onto the centre conductor in an 8-bit signal bus for different values of interconnect length.

The following graphs (figures 7.10-7.12) simulate the worst-case crosstalk coupled onto the centre track in an 8-bit bus. The bus is driven by inverters that are minimum channel length and fixed channel width (3μm for n-channel devices and 7.5μm for p-channel devices) and with the exception of the central interconnect, the inputs to all of these inverters all switch from a high signal to a low signal in 100ps. Vdd is 5V, the simulation temperature is 27°C, 3-stage π RLC models are used to model the lines, and the fourth track in an 8-bit bus is shown each of the set of graphs below.

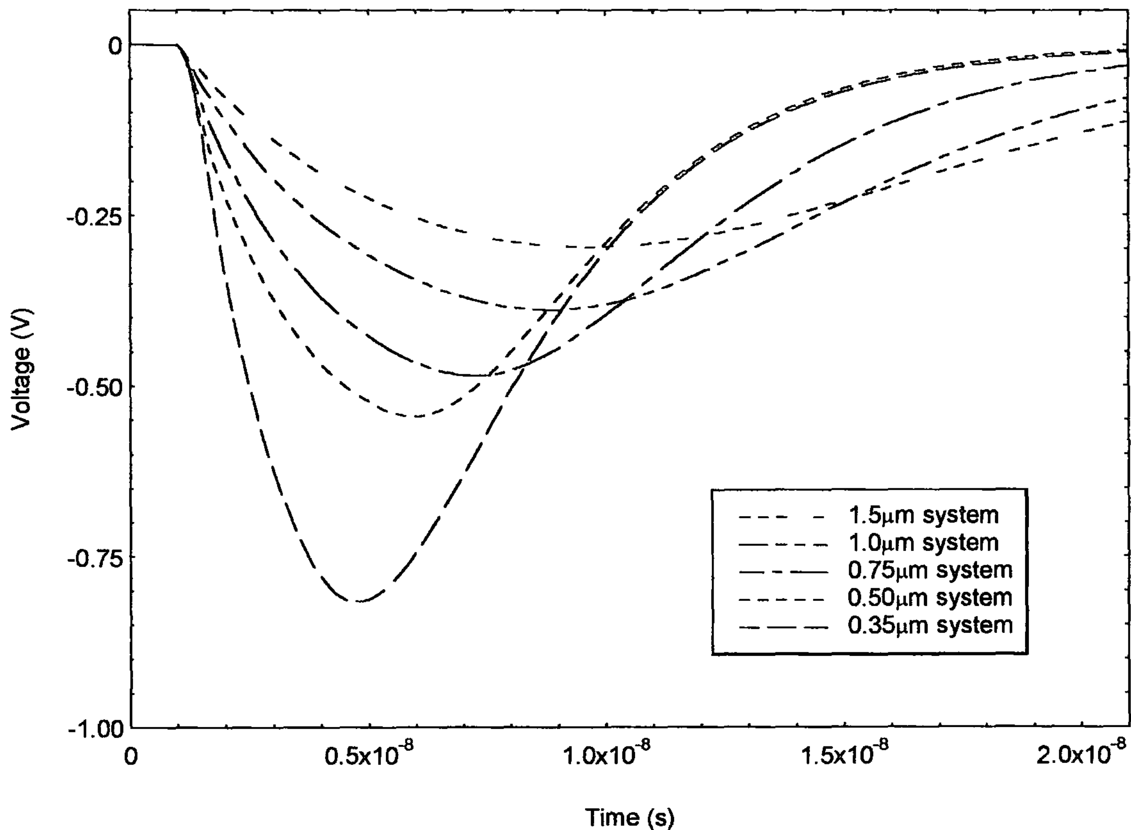

Fig.7.10. Comparison of crosstalk noise against scaling for a 1cm long 8-bit bus.

Figure 7.10 shows the maximum crosstalk signal using a long bus with a length of 1cm. For 0.35μm, the a peak crosstalk voltage is 816mV with a $V_{50\%}$ time delay of 6.7ns, while for the same system at 0.50μm there is a peak crosstalk voltage of 543mV with a $V_{50\%}$ time delay of 8.03ns.

It can be seen that the duration and the amplitude of the crosstalk pulse are both dependent on the scaling level. There is a large increase in the peak crosstalk voltage at all five scaling levels as the minimum feature size is reduced. This increase in coupled voltage at higher scaling levels is due to the increased resistance of the tracks, the increased mutual capacitance between the tracks and the faster switching speeds of the devices. Since the switching is improved at smaller dimensions, the width of the pulses get shorter at the scaling level increases.

Min. Feature Size (μm)	Peak Crosstalk (mV)	$V_{50\%}$ Pulse Duration (ns)
1.5	298	15.843
1.0	390	13.098
0.75	485	10.08
0.50	543	7.99
0.35	816	6.68

Fig. 7.11. Table summarising key features in the graph in figure 7.12.

The table above is a summary of the peak coupled voltage and crosstalk pulse duration for the graph in figure 7.10. Comparing the ratio of the change in feature size to the ratio of the crosstalk voltage at that scaling level, the crosstalk voltage shown increases at a faster rate than the scaling level. In addition, the increase in crosstalk isn't linear with scaling level. This will be shown in a graph later in this section. The reduction in duration of the crosstalk pulse with scaling is consistent with the results obtained in chapter 6 when considering the switching speeds for loaded inverters.

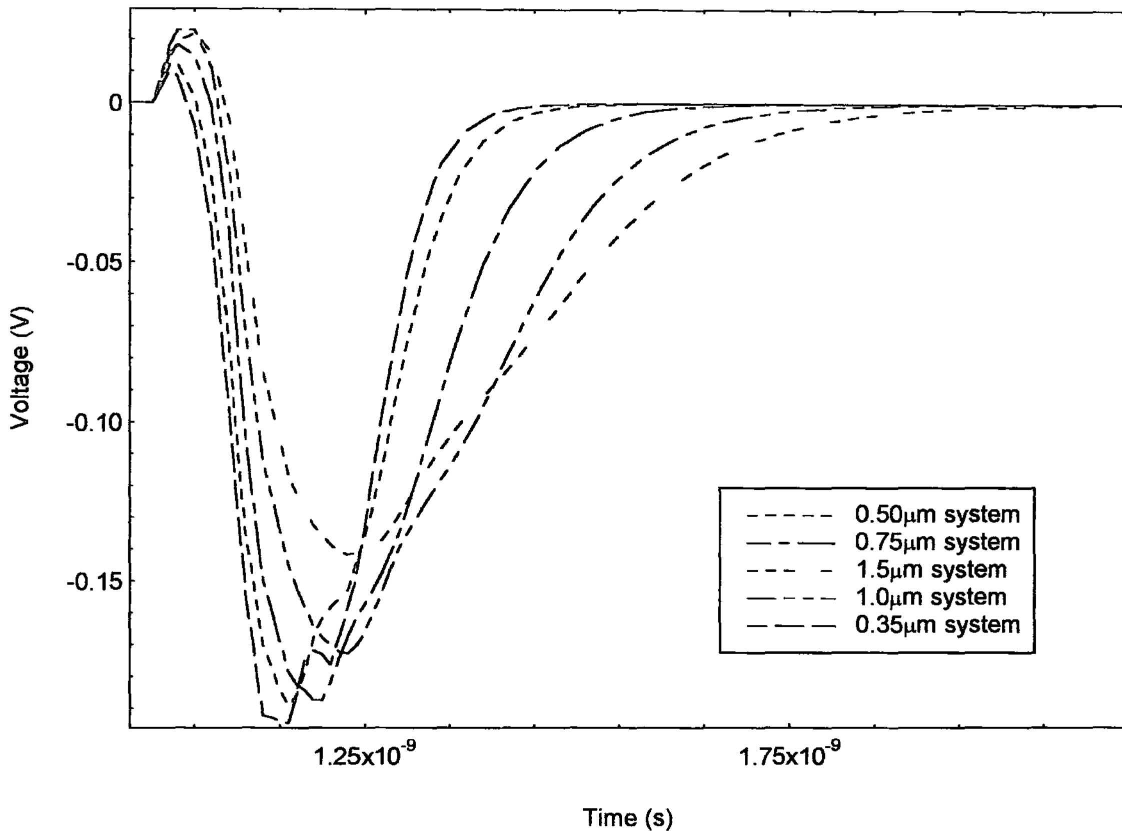

Fig. 7.12. Comparison of crosstalk noise against scaling for a 100μm long 8-bit bus.

The maximum crosstalk signal using a long bus with a length of 100μm is shown above in figure 7.12. This results in a peak crosstalk voltage of 195mV with a $V_{50\%}$ time delay of 180ps for the 0.35μm system, while for the 0.50μm system there is a peak crosstalk voltage of 189mV with a $V_{50\%}$ time delay of 190ps. A flaw, or "glitch", in both the 0.35μm and 0.50μm models can be seen just after the peak on the crosstalk voltage. This is caused by the method used to scale these two models from the original 0.75μm model and is observed in conditions in which there is only a small capacitive or resistive load on the transistor. This flaw can also be seen in the transistor scaling graphs in section 6.7.

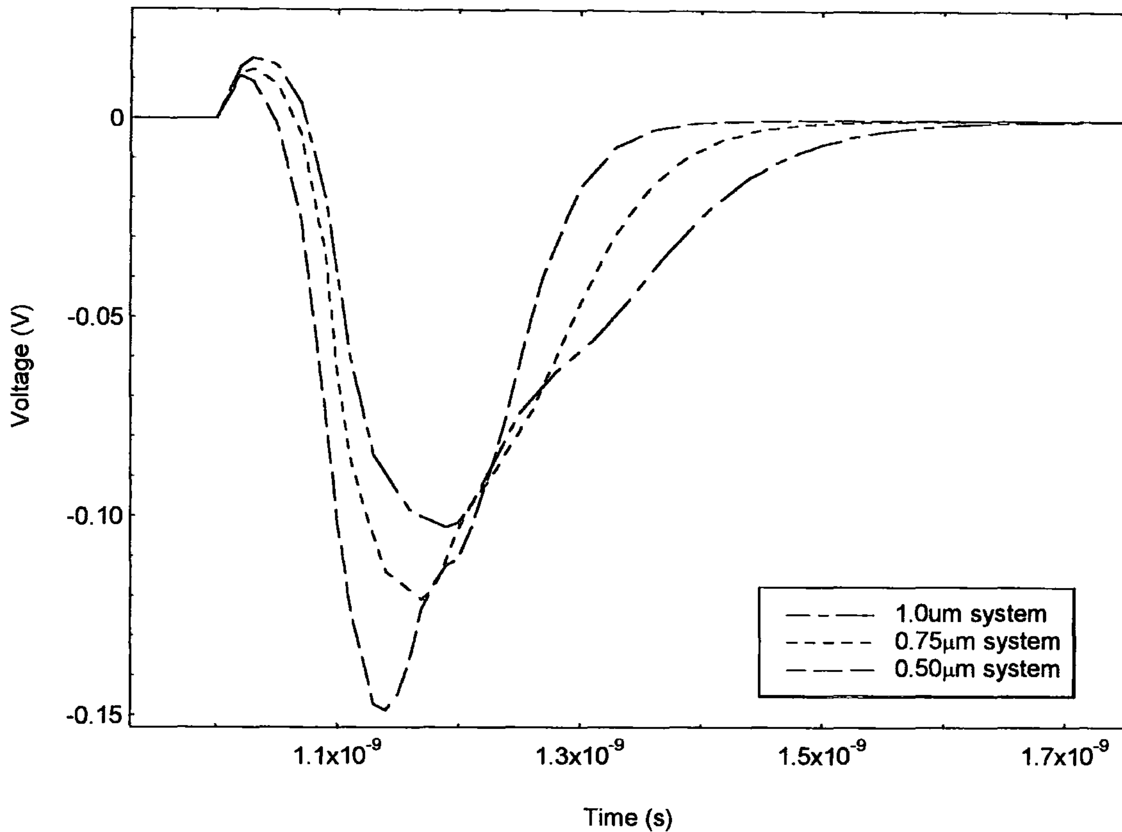

Fig. 7.13. Comparison of crosstalk noise against scaling for a 10μm long 8-bit bus.

A simulation showing the worst-case crosstalk signal on a short length bus is shown above in figure 7.13. The 0.35μm model is not used in this simulation because the load on the transistor is small and the flaw in the transistor model becomes extremely large resulting in erroneous values for both the duration of the coupled signal and magnitude of the signal. The peak crosstalk voltage on the 0.50μm track is 149mV with a $V_{50\%}$ time delay of 15ps and on 0.75μm system the peak crosstalk voltage is 120mV with a $V_{50\%}$ time delay of 19ps.

Considering the three graphs presented in figures 7.11-7.13, two primary conclusions can be made concerning the dependency of crosstalk on scaling and on track length. As track length is increased, the coupled crosstalk voltage increases as well. Considering equation 7.2 again, this is due to the ratios of Cm/(Cs+Cl) and Rt/(Rt+Rd) increasing towards

unity as the track becomes longer. When these two ratios approach unity further increases in track length will no longer result in increased crosstalk. So, unlike interconnect delay, it is difficult to define a "critical track length" - the point at which crosstalk noise becomes unacceptable, since, for certain devices or interconnect dimensions, the coupled crosstalk noise will never be a problem. It can also be seen in the three graphs, that the pulse duration is dependent on track length as well. This can be explained by the fact that longer tracks result in larger loads which take longer make the transition.

7.3 Introduction to Crosstalk Delay

From the perspective of a circuit designer, crosstalk causes several difficulties. The most obvious problem is that signal coupling from one line to another can lead to errors, or "glitches" in the operation of the circuit. In the situation of the bus of a microprocessor, the coupling may only occur when certain bits patterns are present on the line which can lead to difficulties detecting the errors. In addition, these errors may only occur in certain conditions of temperature, operating voltage, or processing conditions leading to reduced yield or erroneous products within the market. This problem is alleviated, but not eliminated by the fact that electrical and design rule checks for the avoidance of crosstalk noise are recently being considered within the semiconductor industry.

A second problem caused by the coupling of signals within a circuit can be termed as "crosstalk delay". Again, this is most easily explained by using the example of a bus on a microprocessor. The bit lines of nearly all microprocessors are synchronously latched** and within such a bus it would be a common occurrence for all of the bit lines to be switching voltage levels at the same time. In the most severe case, all of the lines surrounding a central line could be switching from a high voltage state to a low state at the same time that this central line is switching from low to high. The situation in a bus

** In other words, all of the bit lines switch simulataneously by triggering all of them to the switching of a clocking signal.

driven by inverters is shown below in figure 7.14. This condition is differentiated from the situation of crosstalk noise by the fact that, in this case, the signal line affected is switching rather than remaining at a constant value.

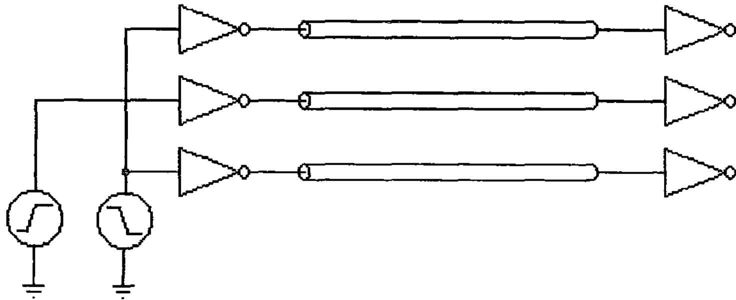

Fig. 7.14 Illustration of crosstalk delay.

In this example, though it isn't shown within the illustration for purposes of simplicity, the lines are coupled by the mutual capacitances and inductances between them. Whenever the line in the centre is falling from a high state to low state at the same time as the neighbouring lines are falling, the mutual capacitance and inductance between them will remain uncharged and the signals will switch with a minimum switching delay. On the other hand, whenever the centre line is rising from low to high while the other two lines are falling from high to low, the field in the capacitance and inductance between them will switch polarity and these parasitic components will discharge first and then charge up again. In this case the line will see a maximum switching delay.

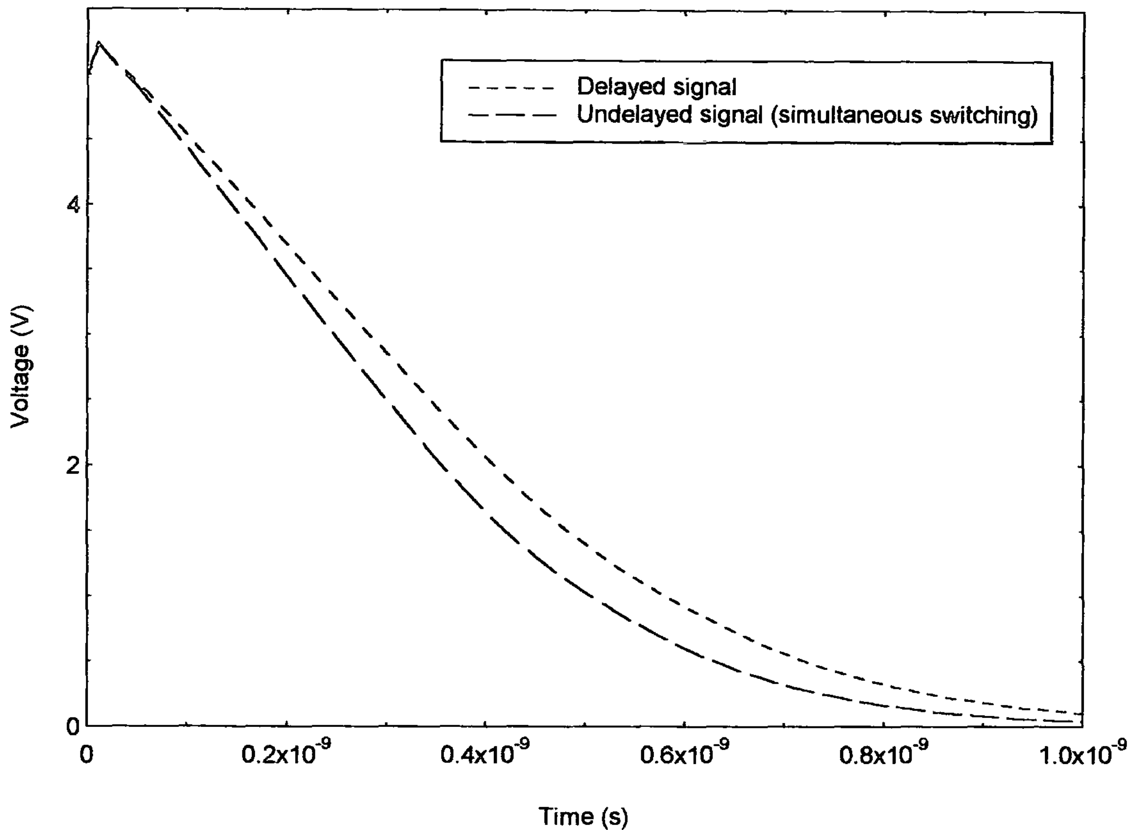

Fig. 7.15. Example of crosstalk delay showing difference in the slopes between the delayed and undelayed signals.

The example shown above in figure 7.15 illustrates the difference between delayed and undelayed slopes caused by the switching of signals on neighbouring lines and the mutual inductance and capacitance. The undelayed signal occurs when all of the lines switch in parallel while the delayed signal shown occurs when all of the other neighbouring conductors switch in one direction, while the signal conductor shown switches in the opposite direction. These two signals shown at the worst-case situations for crosstalk delay. Other different signal combinations will result in delay values between these two extremes. In this example, the slope (10%-80%) of the delayed signal is approximately 19% longer than that of the undelayed signal. The delayed signal is approximately 40ps slower than the undelayed signal when measured at the $\frac{1}{2} V_{dd}$ point.

UNIV DUNELM

This difference in delay becomes a serious problem in synchronous design. At the one extreme, all of the signals will be synchronised and the signals may not need to charge the parasitic components between them. In this case they may "arrive" at the receiving latch before they are designed to, the erroneous signals are latched and a minimum delay problem occurs because the latch hold time is violated. On the other hand, the signals will see a maximum inductance and capacitance and will arrive later than they are supposed to, miss the clock pulse and an erroneous past signal will be latched as the current value. This is referred to a setup time violation. For a circuit to work as designed the margin between these two states must always be considered.

In the past this delay was not a serious problem. The relatively large self capacitance of lines overwhelmed the much smaller mutual capacitance and the marginal change in crosstalk delay was minimal. As the ratio of self capacitive to mutual capacitance is reduced through the use of SOI (Silicon On Insulator) technologies or through the use of quasi-ideal scaling rules which reduce the mutual capacitance at a slower scaling rate than the self capacitance, this will no longer be true. In these cases, rather than designing for a fixed RC delay, a more complex task of designing for a best and worst case RC delay through a line will have to be considered.

7.3.1 Analysis of Crosstalk Delay in Scaled Bus Structures

Since crosstalk delay and crosstalk noise are manifestations of the same effect, the discussion in 7.2.5 about scaling and crosstalk noise applies to crosstalk delay. Increased crosstalk delay is expected at submicron device dimensions due to the increased mutual to self capacitance ratio (C_m/C_s). Thus, it is expected that the minimum and maximum delays through an interconnect will get larger as feature sizes are reduced through the use of scaling.

Using the same system models used to produce figures 7.10, 7.12 and 7.13, simulations were performed to evaluate the change in worst-case crosstalk delay produced at different tracks lengths and different scaling levels. The following graphs (figures 7.16-7.17) simulate the worst-case crosstalk delay coupled on the centre track in an 8-bit bus. In figure 7.16, the bus is driven by inverters that are all switching from a low signal to a high signal except for the central interconnect which is switching from high to low to produce the delayed signal. To generate the undelayed signal, all signals are switching simultaneously from high to low. For figure 7.17, the signals listed are reversed. The signal on the centre track (the fourth track in the bus) is shown in both sets of graphs.

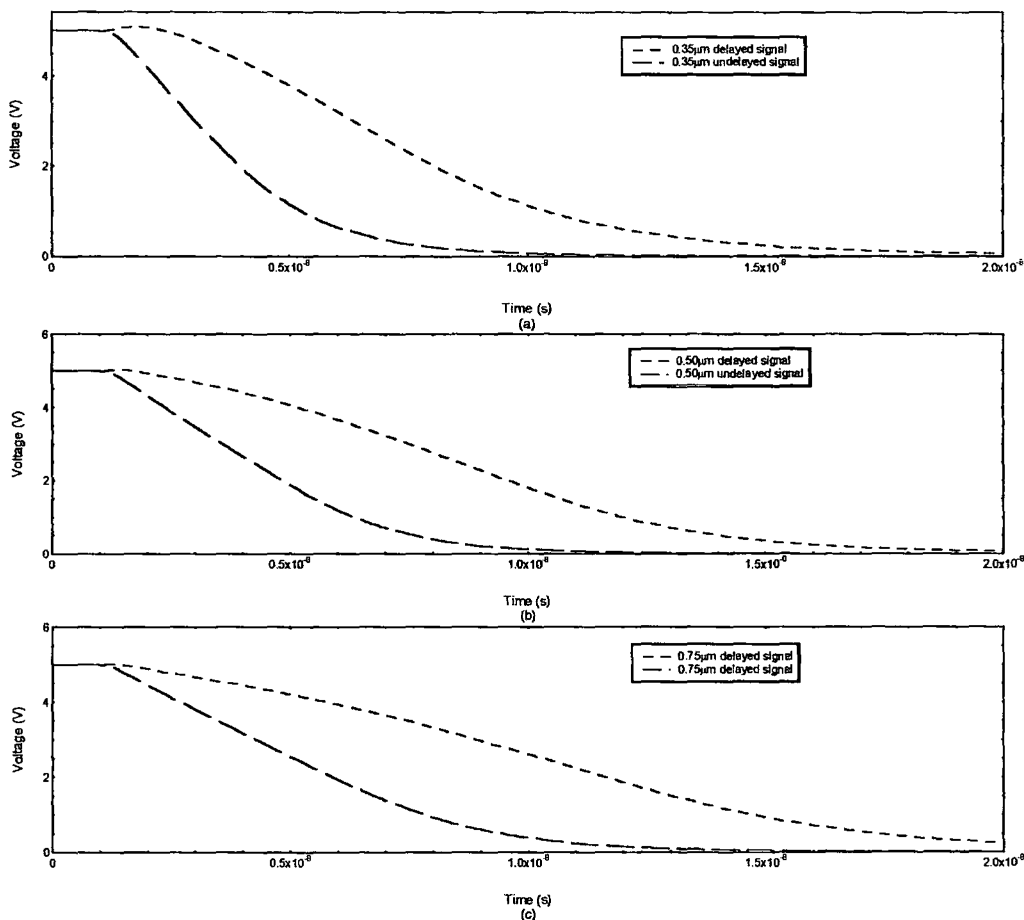

Fig.7.16. Comparison of crosstalk delay against scaling for a 1cm long 8-bit bus for three processes: (a.) 0.35μm, (b.) 0.50μm and (c.) 0.75μm.

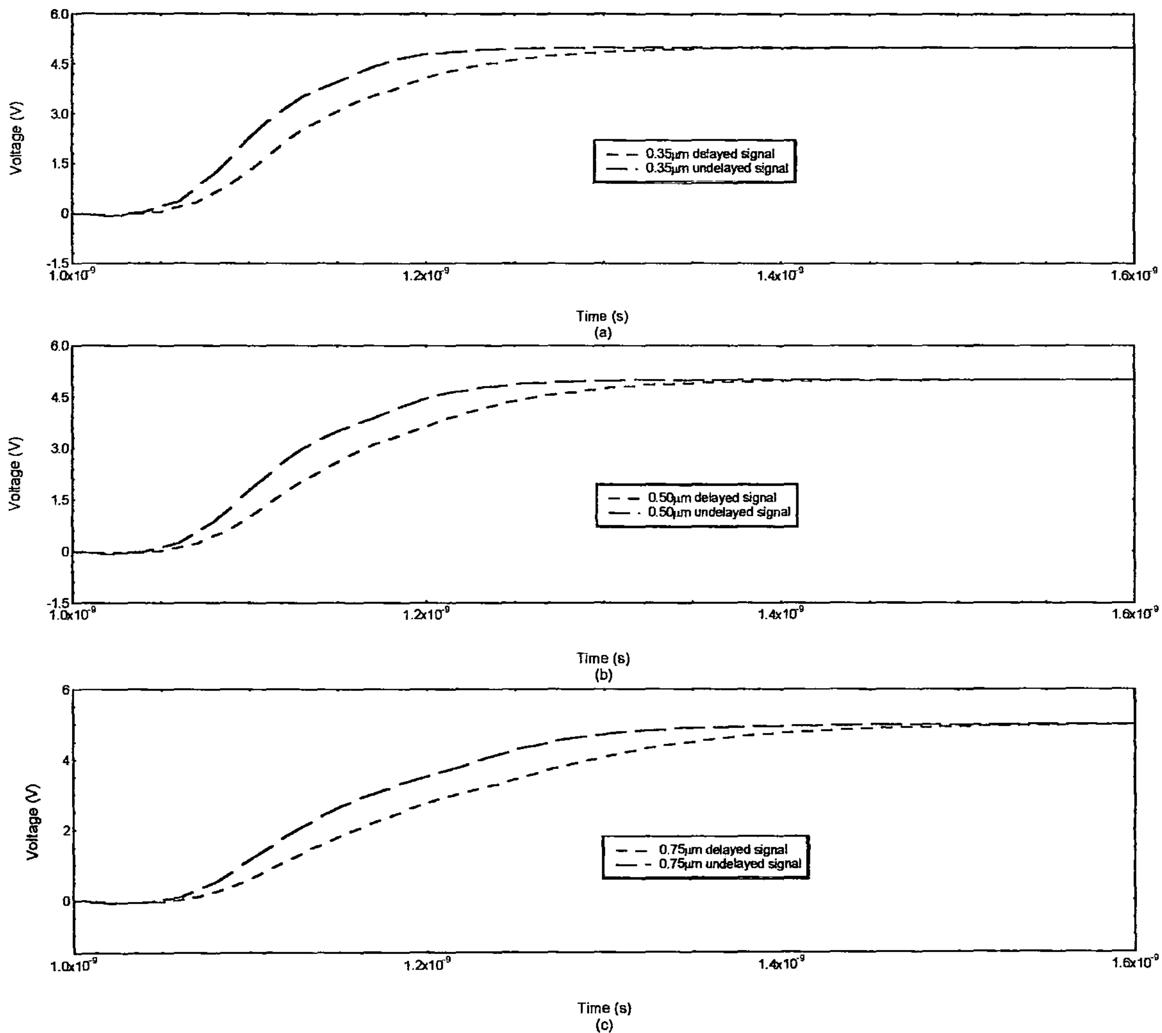

Fig.7.17. Comparison of crosstalk delay against scaling for a 100µm long 8-bit bus for three processes: (a.) 0.35µm, (b.) 0.50µm and (c.) 0.75µm.

In figures 7.16 and 7.17, the worst-case crosstalk delay seen on the centre line of a 1cm and a 100µm 8-bit bus is shown for each of three scaling levels. It can be seen that, the rise time improves as feature size is reduced and similarly the margin difference between the minimum and maximum improves. Due to the fact that the slope is much longer at larger device sizes, and the fact that the margin between the minimum and the maximum

crosstalk delay becomes slightly smaller as device size is reduced, the overall percentage amount that the margin of the slope becomes smaller as process size is reduced.

Process	100μm length (ns)	1cm length (ns)
0.35μm	0.20 (78%)	3.713 (21%)
0.50μm	0.29 (79%)	4.369 (26%)
0.75μm	0.40 (80%)	5.221 (29%)

Fig.7.18. Summary of 50% crosstalk delay measurements and percentage margin that crosstalk delay is to slope for different interconnect scaling and length values.

The table in figure 7.18 shows that crosstalk delay becomes slightly less of a problem as the process size is reduced. This result is not intuitive when you consider the crosstalk noise results until the slope of the line is taken into account. Since the margin remains nearly constant at each device size, but the slope becomes much worse, the overall effect is that crosstalk delay improves slightly with scaling.

7.4 Methods of Reducing Crosstalk

As has been shown in the previous graphs in this section, coupled crosstalk signals can reach a magnitude and duration sufficient to introduce false signals into the system resulting in logic errors and unpredictable performance. The problem of crosstalk delay creating large signal propagation delay margins and reducing valid times in synchronous circuits can also result in unpredictable performance. In order to reduce the risk associated with these two problems, it is important to consider methods that can be used to reduce crosstalk while keeping device densities high.

One method of reducing crosstalk is to reduce the substrate thickness in order to provide a shielding plane in close proximity to the interconnect lines. The effect of this reduction

can be seen by examining the self and mutual capacitances in figure 5.4. The mutual capacitance is reduced slightly and the self capacitance increases substantially. These changes, reduce the mutual capacitance to self capacitance (Cm/Cs) ratio which in turn reduces signal coupling. Minimising the substrate thickness can be done by physically reducing the substrate thickness, by using increasing the substrate doping density to increase the substrate conductivity or through the use of new technologies such as SOI (Silicon-On-Insulator) which allow extremely thin substrates to be used. All of these methods create their own additional problems and none are currently practical.

The effect of shielding which was mentioned briefly in section 7.2.1 can also be used to dramatically reduce crosstalk effects. This shielding can be achieved in two different ways: through the use of a form of stripline structure in which a reference plane is introduced above the signal line as well as below it, and by using neighbouring wires held at a fixed voltage. These two methods can be used alone or in combination to reduce the mutual capacitance of the system while increasing the self capacitance and possibly reducing the device density.

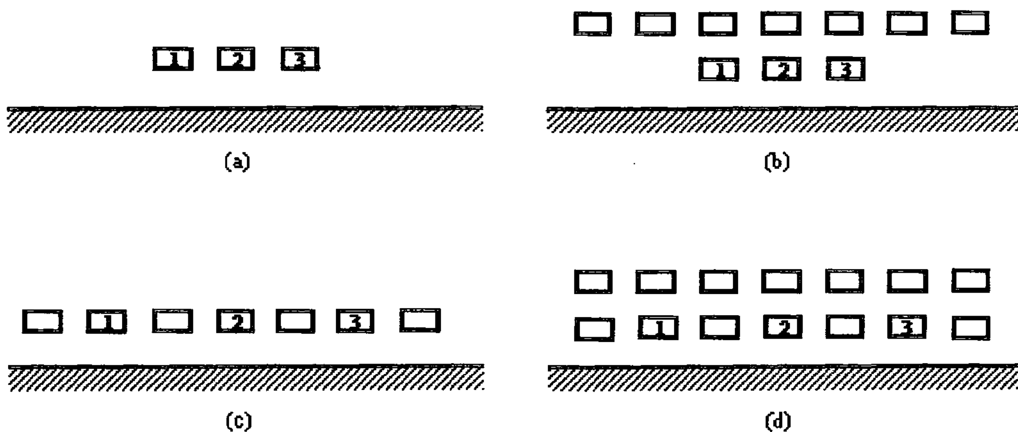

Fig.7.19. Different layout techniques that can be used to reduce crosstalk effects: (a) normal 3-bit bus, (b) stripline configuration, (c) shielded interconnects, and (d) shielded stripline (pseudo-coaxial) configuration.

The normal layout for a 3-bit bus is shown above in figure 7.19(a). with three signal lines above a cross-hatched reference plane. In figure 7.19(b), a reference plane of fixed voltage conductors (the unmarked lines) is added to the metallisation layer above the bus forming a stripline configuration. Shielding conductors are added in between the bit lines in figure 7.19(c) and in figure 7.19(d) both of these techniques are used to form a pseudo-coaxial configuration around each bit line.

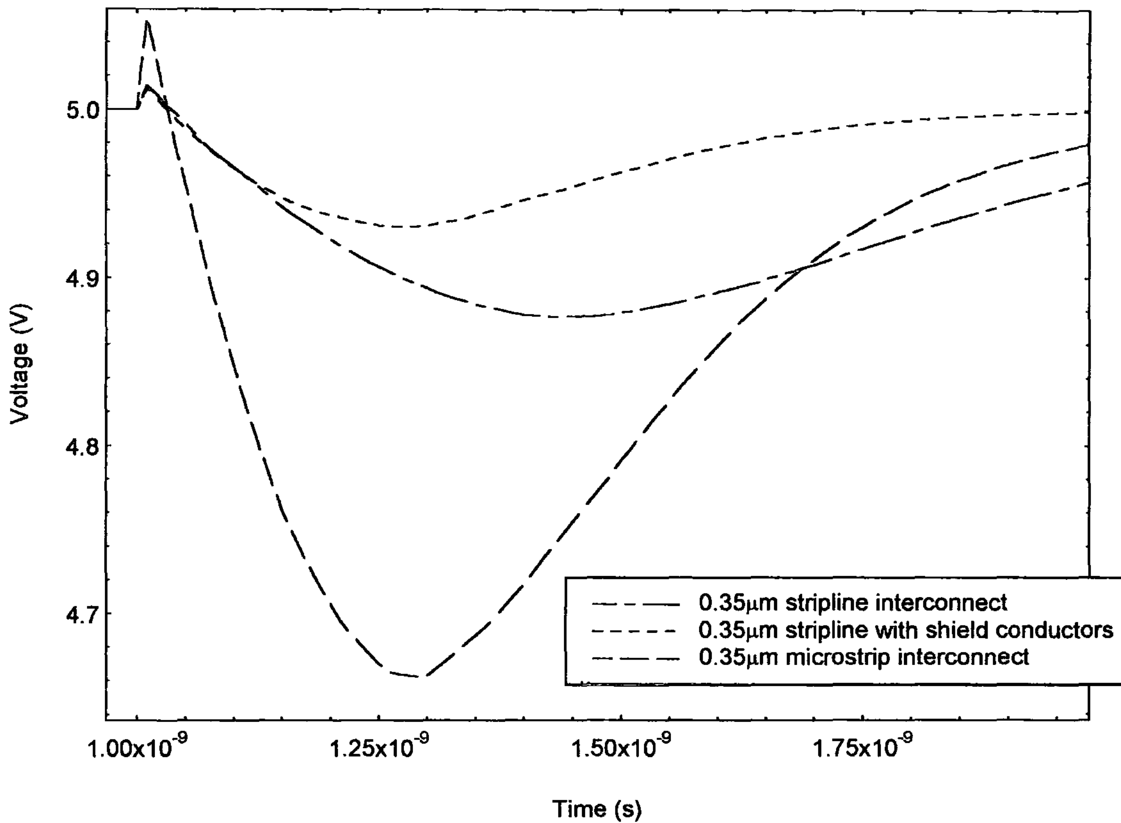

Fig.7.20. Comparison of layout techniques used to reduce crosstalk effects.

A comparison of the stripline and the pseudo-coaxial configurations is shown above in figure 7.20. A three bit bus driven by 0.35μm inverters with two of the signal lines switching from high to low while the centre line is held fixed at 5V is used in each of the examples. The standard microstrip configuration using interconnect dimensions from the table in figure 5.4 is shown for reference and this results in a peak voltage of 336 mV.

The stripline configuration using five closely spaced conductors fixed to 0V results in peak voltage of 123mV which is nearly a third of the microstrip configuration. The pseudo-coaxial configuration in which seven conductors held at 0V are used to surround the signal lines results in a peak voltage of 0.065mV which is less than a fifth of the microstrip system and nearly half the stripline configuration.

Although these methods result in substantial reductions in signal coupling they have two serious disadvantages. The most serious problem with this method is the loss in device density due to loss in routing space and routing flexibility. Another difficulty is that the increased self capacitance due to the closer spacing of the shielding conductors results in a longer RC delay and increased power dissipation. An added bonus of these techniques is that, since there are so many fixed voltage lines, there is more flexibility in power routing to devices.

Other methods that can be used to reduce the effects of signal coupling are to increase the separation between the tracks, or to reduce the interconnect thickness while offsetting the increased resistance by widening the metal pitch. Both of these dramatically reduce the system device density and can be disregarded for the same reasons as were mentioned for constant-R scaling in section 5.3.

One final solution is the introduction of signal buffers along long parallel lines. This reduces the problems associated with crosstalk effects by isolating the long track into two shorter tracks. The use of signal buffers on long bus lines also reduces the signal delay by reducing the load driven by the output device. The use of buffers increases power dissipation, however, and increases system complexity. For large parallel signal lines, it is the best solution to the problem of crosstalk.

8. Conclusions

Conclusions on each of the sections are presented below describing significant results obtained in each of the chapters and conclusions that can drawn from these results.

8.1 Electrical Modelling of Interconnections

In this chapter an accurate model for modelling distributed RC and RLC interconnect is described. A π model was determined to be more efficient in HSPICE than a corresponding T-model due to the calculation method used in HSPICE. Simulations were then used to determine the number of sections required to accurately model distributed RC and RLC interconnections. It was determined from these simulations that three stages provided sufficient accuracy while remaining computationally efficient. This conclusion was then compared with references in literature and was found to closely match other published results.

8.2 Capacitance Parameter Calculation

Several methods for calculating capacitance from interconnect dimensions are evaluated and explained. Lewis's method for determining capacitance is examined and is used as an example to illustrate the method chosen for this project: Matthaei's method. Algorithm for this method is derived from literature and this method is evaluated for accuracy. It is found that there are problems using this technique at deep submicron scaling levels, but it is determined that this is not an issue for this project. Finally, the results from Matthaei's method are compared with other results published in literature and found to agree with closed-form results published by Yuan and Trick. Another comparison with a numerical technique developed by Weeks and published by Chang find that Matthaei's method consistently underestimates the values calculated by Weeks and published by Chang within a small percentage difference. It is postulated that this may be due to some undocumented variable.

8.3 Inductance and Resistance Parameter Calculation

In this section, a method is described for calculation of inductance parameters based on the capacitance matrix determined using Matthaei's method. A method is also given for the determination of track resistance. Second-order high-frequency resistance effects are considered and then discounted as adding little accuracy to the model at the frequencies under consideration. This is verified using references from published papers.

8.4 Interconnection Scaling

The ideal and quasi-ideal methods for determining interconnect dimensions when an integrated circuit is scaled are described. Using quasi-ideal scaling parameters and sources in literature, estimated minimum design rules for metal 1 in a submicron process are derived. An evaluation of the variation in capacitance with changes interconnect dimensions is made and basic trends in the dependence of capacitance on changes in interconnect parameters are made.

8.5 Transistor Scaling and Modelling

The general theory for constant-field and constant voltage MOSFET transistor scaling are described. Using constant-voltage scaling rules and existing 1.0μm and 0.75μm HSPICE models, new 0.50μm and 0.35μm models are derived which are shown to have expected switching characteristics under loaded and unloaded conditions.

8.6 Crosstalk Delay and Noise

The concept of crosstalk noise and crosstalk delay are described using simple examples. A basic analytical model showing the dependency of crosstalk on certain circuit parameters is presented. The dependency of coupled crosstalk voltage and the duration of the pulse width are examined for variations in temperature, track length, inductance

and scaling level are analysed using HSPICE simulations of the model presented in this thesis. It was determined that temperature had negligible effect on the coupled crosstalk signal, but that the limitations of the model prevented a completely accurate estimation. It was found that inductance had larger effect than temperature, but using worst-case conditions it was found that it s absence would cause less than a 10% error.

Track length was found to greatly increase the coupled crosstalk signal, but analysis found that on extremely long lines the coupled signal will peak at a value determined by circuit parameters of the driving device, the receiving device and the line itself. Scaling was found to increase crosstalk significantly and was found to cause logic errors on extremely long (>1cm) 0.35μm parallel bus lines under worst case conditions. The crosstalk signals coupled at certain sampling points along a distributed line are examined at two different scaled processes. It was found that the coupled voltage along the distributed line varied by a wider margin in submicron interconnects than it did in larger structures.

Crosstalk delay is introduced as a concept and the problems that it can lead to in synchronous design are considered. Crosstalk delay is evaluated at different track lengths and at different scaling levels and it was found that the margin between the minimum and maximum delay decreases slightly with reduced device dimensions even when the faster slopes of submicron devices are considered. Although this reduces the problems associated with crosstalk delay, it is still considered to be a serious problem in synchronous design.

Several techniques for reducing the effects of crosstalk are described and discussed. Two different shielding strategies are considered and evaluated for the reduced in crosstalk coupling estimated. The use of buffers on long lines is recommended as the best solution to signal coupling on global interconnections. Other strategies that should be considered are avoiding long parallel routing (i.e. routing the interconnects in such as way as to avoid crosstalk problems), mixing power and ground lines with signal lines wherever

possible, using large separations on long parallel lines and using an intermediate metal layer primarily for power routing (rather than using the highest level) in order benefit from the pseudo-stripline configuration.

Appendix A: Implementation of Matthaei's Method in PASCAL

This appendix contains the complete program implementing Matthaei's method on an MS-DOS system. It is written in PASCAL and is specifically optimised for Turbo Pascal v.4.0.

```
Program Thick_Conductor_Capacitance_Calculation_v20(Input,Output,File1,Printer);

(*************************************************************************)
(* Name:      CC22.Pas (Capacitance Calculation Program version 22) *)
(* Programmer: Patrick F. Mahoney                    *)
(*         Microelectronics Research                 *)
(*         Durham University - Science Site          *)
(*         Durham    DH1 3LG                         *)
(*         U.K.                                      *)
(*    email: p.f.mahoney@durham.ac.uk                *)
(*                                                   *)
(* Compiler: Turbo Pascal (>version 3.0)             *)
(*        (UCSD-compatible src available on request) *)
(*                                                   *)
(* Credits:                                          *)
(*                                                   *)
(*    The method used in this program was taken from a paper by: *)
(*    George L. Mattaei.                             *)
(*                                                   *)
(*                                                   *)
(*                                                   *)
(*                                                   *)
(*************************************************************************)

Uses CRT,Printer,DOS;              { Uses Screen Library Unit     }
                              { And DOS and Printer Units    }

Const Pi=3.141593;
    MaxN=8;                      { Maximum number of conductors }
    U0=1.25663E-6;                 { Permeability of FS         }
    E0=8.854E-12;                 { Permittivity of FS         }
    Eo=3.9;                      { Relative permittivity of SIO2 }
    Es=7.2;                      { Relative permittivity of SI  }
    Rh=0.67;                      { Horiz. match point constant  }
    Rv=0.8;                      { Vert. match point constant   }
    VConst=1;                     { Voltage at Zk induced by Qm  }
    Rho=2.8E-8;                    { Bulk resistivity of Al(Ohm*m) }

Type  Complex=Record                { Complex number used for Zk   }
          R:Real;
          I:Real;
       End;

    ComplexCrd=Record              { Complex number used for Bm   }
          R:Real;               { Boolean holds spacial orient- }
          I:Real;              { tation for use in PePo calcs }
          H:Boolean;              { H=True when Bm is horizontal }
       End;

    Conductors=Record              { Holds the information needed }
          W:Real;               { for multiple conductor       }
          H:Real;              { systems with dissimilar      }
          T:Real;              { conductors.  W,T,H,S are     }
          S:Real;              { Width, Thickness, Height     }
       End;                    { and Separation respectively  }
    Name   = String[32];              { The name of a file         }
```

```
    List   = Array[1..[4*MaxN]] of ComplexCrd; { List of B coordinates     }
    Coords = Array[1..[8*MaxN]] of Complex;   { List of Z coordinates     }
    Vector = Array[1..[8*MaxN]] of Real;     { Temp PePo for Inverstion }
    Matrix = Array[1..[8*MaxN]] of Vector;    { PePo Matrix              }
    CMatrix= Array[1..MaxN,1..MaxN] of Real; { Capacitance Matrix       }
    DMatrix= Array[1..MaxN] of Conductors;   { Dimension matrix         }
    RMatrix= Array[1..MaxN] of Real;         { Resistive matrix         }

Var  Blist   : List;                  { List of Bm coordinates      }
     Zlist   : Coords;                { List of Zk coordinates      }
     PePo    : Matrix;                { He and Ho matrix            }
     IPePo   : Matrix;                { He and Ho matrix for ind.   }
     CapMat  : CMatrix;               { Resulting capacitive matrix }
     IndMat  : CMatrix;               { Inductive matrix            }
     N       : Integer;               { Number of conductors in system}
     ResMat  : Vector;                { Resistive matrix            }
     Choice  : Char;                  { The menu choice letter      }
     MenuFlag : Boolean;              { A flag for the menu         }
     Answer  : Char;                  { Reply for various queries   }
     Ch      : Char;                  { Dummy value for Readkey()   }
     Dims    : DMatrix;               { List of conductors dimensions }
     I,J     : Integer;               { Various For loop variables  }
     Layout  : Char;                  { Type of layout used         }
     Status  : Integer;               { Flag of current program status}
     Err     : Boolean;               { Error Flag from function    }
     Leng    : Real;


(*****************************************************************)
Procedure Intro;
(* Describes the program and the calculation method used        *)
(*****************************************************************)

Begin
    ClrScr;                         { Turbo Pascal Clear Screen }
    Writeln('Capacitance Calculation Program v16');
    Writeln('————————————————');
    Writeln;
    Writeln;
    Writeln('This program enables the rapid calculation of the capacitance');
    Writeln('and inductance parameters for multiple layer, finite thickness,');
    Writeln('multiple conductor systems of rectangular conductors.  It uses');
    Writeln('an extremely efficient algorithm for calculations that was');
    Writeln('described in a paper by G.Matthaei et. al. in IEEE Trans. on');
    Writeln('CAD of ICs (Vol.11 No.4 April 1992 p.513).');
    Writeln;
    Write('Press any key to continue');
    Answer:=Readkey;
End;

(*****************************************************************)
Function Lowercase(Ch:Char):Char;
(* Changes the inputted letter to lowercase                     *)
(*****************************************************************)
Begin
    If ((Ord(Ch)>64) and (Ord(Ch)<91)) then LowerCase:=Chr(Ord(Ch)+32)
       Else LowerCase:=Ch;
End;


(*****************************************************************)
Procedure Menu(Var Status:Integer; Var Choice:Char);
(* Main menu                                  *)
(*****************************************************************)
```

```
Begin
   ClrScr;
   Writeln('Choose:');
   Writeln('  A. Input a conductor system.');
   Writeln('  B. Import a system from file.');
   If (Status=3) then
   Begin
      Writeln;
      Writeln('  C. Print results.');
      Writeln('  D. Export data to text file.');
      Writeln('  E. Export data to SPICE(tm) file.');
      Writeln('  F. Export data to Medici(tm) file.');
   End;
   Writeln('  Q. Exit program.');

   Choice:=' ';
   Repeat
      If (Keypressed) then Choice:=LowerCase(Readkey);
   Until (((Choice='a') or (Choice='b') or (Choice='q') or (Ord(Choice)=27)) or
        (((Choice='c') or (Choice='d') or (Choice='e') or (Choice='f')) and (status=3)));
   If (Ord(Choice)=27) then Choice:='q';

End;



(*****************************************************************)
Procedure SIUnits(Value:Real; Var File1:Text);
(* Converts an inputted value into a standard SI prefix number  *)
(*****************************************************************)

Var Point : Integer;
  Neg   : Boolean;

Begin
   Point:=0;                          (* Resets the Point counter *)

   If (Abs(Value)<1) and (Abs(Value)>1E-18) then
   Begin
      While (Abs(Value)<0.9999) do  (* Keep going until at final value *)
      Begin
         Point:=Point+1;      (* Move along the string as it divides *)
         Value:=Value*1000;         (* Divide down to the next unit *)
      End;
   End
     Else If (Abs(Value)<1E12) and (Abs(Value)>1) then
     Begin
        Point:=10;
        While (Abs(Value)>1000) do
        Begin
           Point:=Point+1;
           Value:=Value/1000;
        End;
     End;

   Write(File1,Value:9:2,' ');
   Case Point of                (* Choose proper prefix based on Point *)
                                 1: Write(File1,'m');
                                  2: Write(File1,'u');
                                  3: Write(File1,'n');
      4: Write(File1,'p');
      5: Write(File1,'f');
      6: Write(File1,'a');
     11: Write(File1,'k');
     12: Write(File1,'M');
     13: Write(File1,'G');
```

```
      14: Write(File1,'T');
    End;
End;

{*************************************************************************}
Function NumConductors(System:Char):Integer;
{* Get the total number of conductors in the inputted system       *}
{*************************************************************************}

Var Correct:Boolean;
   N:Integer;

Begin
    ClrScr;
    Writeln;
    Case System of
       '1':Begin
            Writeln('Planar System of Identical Conductors');
            Writeln('——————————————');
          End;
       '2':Begin
            Writeln('Planar System of Non-Uniform Conductors');
            Writeln('———————————————');
          End;
       '3':Begin
            Writeln('Non-Planar System of Non-Uniform Conductors');
            Writeln('————————————————');
          End;
       Else Begin
             Writeln('Something has gone horribly wrong.');
             Halt(1);
           End;
    End;

    Writeln;

    Repeat
      Correct:=True;
      Write('Number of conductors: ');
      Readln(N);

      If (N<1) then
      Begin
        Writeln('Input error.  Must be a positive number of conductors.');
        Correct:=False;
      End;
      If (N>MaxN) then
      Begin
         Write('Due to memory limitations, systems can only contain ');
         Writeln(MaxN,' conductors.');
         Writeln;
         Correct:=False;
      End;
    Until Correct;
    NumConductors:=N;
    Write('Conductor Length:    ');
    Readln(Leng);
End;


{*************************************************************************}
Procedure Identical(Var N:Integer; Var Dims:DMatrix);
{* Input for systems of identical conductors all in the same plane *}
{*************************************************************************}

Var W,T,H,S:Real;
```

```
Begin
   N:=NumConductors('1');

   Write('Conductor width:      ');
   Readln(W);
   Write('Conductor thickness:  ');
   Readln(T);
   Write('Conductor height:     ');
   Readln(H);
   Write('Conductor Separation: ');
   Readln(S);
   For I:=1 to N do
   Begin
      Dims[I].W:=W;
      Dims[I].H:=H;
      Dims[I].S:=S;
      Dims[I].T:=T;
   End;
End;

{*****************************************************************************}
Procedure SemiIdentical(Var N:Integer; Var Dims:DMatrix);
{* Input for systems of non-uniform conductors all in the same plane *}
{*****************************************************************************}

Var HT:Real;

Begin
   N:=NumConductors('2');
   Write('Height of conductors above ground plane: ');
   Readln(HT);
   Writeln;

   For I:=1 to N do
     With Dims[I] do
     Begin
        Write('Width of Conductor #',I,':      ');
        Readln(W);
        Write('Thickness of Condcutor #',I,':  ');
        Readln(T);
        Write('Separation of Conductor #',I,': ');
        Readln(S);
        H:=HT;
     End;
End;

{*****************************************************************************}
Procedure NotIdentical(Var N:Integer; Var Dims:Dmatrix);
{* Input for systems of non-uniform conductors in the different planes *}
{*****************************************************************************}

Begin
   N:=NumConductors('3');

   For I:=1 to N do
      With Dims[I] do
      Begin
        Write('Width of Conductor #',I,':      ');
        Readln(W);
        Write('Thickness of Conductor #',I,':  ');
        Readln(T);
        Write('Height of Conductor #',I,':     ');
        Readln(H);
        Write('Separation of Conductor #',I,': ');
        Readln(S);
      End;
End;
```

```
{*************************************************************************}
Procedure Convert(N:Integer; Var Dims:DMatrix; Var B:List; Var Z:Coords);
{* Converts the Width, Separation, Height and Thickness stats into   *}
{* complex plane coordinates.  Not well-documented for the simple    *}
{* reason that it is pretty straightforward.                 *}
{*************************************************************************}

Var I:Integer;                 { For-loop index            }
   Index:Real;                 { Index used to calculate centre }

Begin
   Index:=0;                         { Index for the middle of the }
   For I:=1 to N do                  { table                   }
      Index:=Dims[I].W+Dims[I].S+Index;
   Index:=-((Index-Dims[I].S)/2);

   For I:=1 to N do
   Begin
      B[I].I:=Dims[I].H;             { For base of conductors      }
      B[I].R:=(Index+(Dims[I].W/2));
      B[I].H:=True;
      Z[(2*I)-1].R:=(B[I].R-(Rh*Dims[I].W/2));
      Z[(2*I)-1].I:=B[I].I;
      Z[(2*I)].R:=(B[I].R+(Rh*Dims[I].W/2));
      Z[(2*I)].I:=B[I].I;

      B[(I+N)].I:=(Dims[I].T+Dims[I].H);  { for top of conductors       }
      B[(I+N)].R:=Blist[I].R;
      B[(I+N)].H:=True;
      Z[(2*(I+N))-1].R:=(B[(I+N)].R-(Rh*Dims[I].W/2));
      Z[(2*(I+N))-1].I:=B[(I+N)].I;
      Z[(2*(I+N))].R:=(B[(I+N)].R+(Rh*Dims[I].W/2));
      Z[(2*(I+N))].I:=B[(I+N)].I;

      B[(I+2*N)].I:=(Dims[I].H+(Dims[I].T/2));
      B[(I+2*N)].R:=Index;           { for left-side of conductors }
      B[(I+2*N)].H:=False;
      Z[(2*(I+2*N))-1].I:=(B[(I+2*N)].I-(Rv*Dims[I].T/2));
      Z[(2*(I+2*N))-1].R:=B[(I+2*N)].R;
      Z[(2*(I+2*N))].I:=(B[(I+2*N)].I+(Rv*Dims[I].T/2));
      Z[(2*(I+2*N))].R:=B[(I+2*N)].R;

      B[(I+3*N)].I:=(Dims[I].H+(Dims[I].T/2));
      B[(I+3*N)].R:=(Index+Dims[I].W);   { for right-side of conductors }
      B[(I+3*N)].H:=False;
      Z[(2*(I+3*N))-1].I:=(B[(I+3*N)].I-(Rv*Dims[I].T/2));
      Z[(2*(I+3*N))-1].R:=B[(I+3*N)].R;
      Z[(2*(I+3*N))].I:=(B[(I+3*N)].I+(Rv*Dims[I].T/2));
      Z[(2*(I+3*N))].R:=B[(I+3*N)].R;
      Index:=Index+(Dims[I].W+Dims[I].S);
   End;


End;

{*************************************************************************}
{* Complex calculation routines                              *}
{* These are all pretty straightforward and shouldn't need explaining *}
{*                                                  *}
{*************************************************************************}

Procedure ComplexAdd(Var Z1,Z2,Z3:Complex);
{********** Adds two complex numbers together **************}

Begin
   With Z3 do
```

```
    Begin
       R:=(Z1.R+Z2.R);
       I:=(Z1.I+Z2.I);
    End;
End;


Procedure ComplexPowers(Var Z1,Z2:Complex; Power:Integer);
(******** Takes the nth power of a complex number **********)
Var Temp:Complex;
   I:Integer;

Begin
    Z2:=Z1;
    If (Power>1) then
    Begin
     For I:=1 to (Power-1) do
     Begin
        Temp:=Z2;
        Z2.R:=((Temp.R*Z1.R)-(Temp.I*Z1.I));
        Z2.I:=((Temp.I*Z1.R)+(Temp.R*Z1.I));
     End;
    End;
End;

Procedure ComplexMult(Var Z1,Z2:Complex; Num:Real);
(******* Multiplies a complex number by Num ***************)
Begin
    Z2.R:=Z1.R*Num;
    Z2.I:=Z1.I*Num;
End;

Procedure ComplexSqrt(Var Z1,Z2:Complex);
(******** Takes the square root of a complex number *******)

Var T,R:Real;

Begin
    R:=Sqrt(Sqr(Z1.R)+Sqr(Z1.I));     { Converts to polar coordinates   }
    If (Z1.R=0) then
    Begin
       If (Z1.I<0) then T:=-(Pi/2)   { Takes care of problem of taking }
          Else T:=(Pi/2);           { ArcTan() of infinity           }
    End
      Else
      Begin
         T:=ArcTan(Z1.I/Z1.R);
         If (Z1.R<0) and (Z1.I<=0) then T:=T-Pi;  { Corrects for      }
         If (Z1.R<0) and (Z1.I>0) then T:=T+Pi;  { quadrant problems }
         { And sets the angle between +180 and -180                 }
         { As opposed to +360 and 0                          }
      End;
    R:=Sqrt(R);                    { These two take the square root  }
    T:=T/2;
    Z2.R:=R*Cos(T);                { Converts to complex coordinates }
    Z2.I:=R*Sin(T);
End;

(*****************************************************************************)
Procedure Arcsin(I:Complex; Var J:Complex);
(** Complex Arcsin procedure                         **)

(** Kreysig page 757.  Works out the Arcsin of a complex    **)
(** number without the limits imposed by the Taylors        **)
(** theory expansion.                                **)
(** Uses ln(z)=ln(x+iy)=((x^2+y^2)^.5)+i*arctan(y/x)        **)
(** And  ArcSin(z)=-i*ln(iz+sqrt(1-z^2))                **)
```

```
(** Due to floating point and quadrant errors, it's been      **)
(** bandaged a bit, but it does work for everything, I've     **)
(** put through it.                                           **)

Var Temp:Complex;
   Flag:Boolean;

Begin
   If ((I.I=0) and ((Abs(I.R))>1)) then Flag:=True
      Else Flag:=False;
   ComplexPowers(I,Temp,2);              { I = I^2          }
   Temp.R:=1-Temp.R;                     { I = 1 - I        }
   Temp.I:=-Temp.I;
   ComplexSqrt(Temp,Temp);               { I = I^.5         }
   Temp.R:=Temp.R-I.I;
   Temp.I:=Temp.I+I.R;                   { I = I + i*I      }
   I:=Temp;                              { switch variables }
   If (I.R=0) then
      If (I.I<0) then J.R:=-(Pi/2)  { Takes care of problem of taking }
         Else J.R:=(PI/2)          { ArcTan() of infinity           }

      Else J.R:=Arctan(I.I/I.R);          { J = i*Ln(I)      }
   If (I.R<0) and (I.I<0) then J.R:=J.R-Pi;    { Corrects for      }
   If (I.R<0) and (I.I>=0) then J.R:=J.R+Pi;   { quadrant problems }

   If ((I.R=0) and (I.I=0)) then J.I:=0
      Else J.I:=-Ln(Sqrt(Sqr(I.R)+Sqr(I.I)));   { J = i*Ln(I)      }
   If Flag then J.I:=-J.I;
   If ((J.I<1e-10) and (J.I>-1e-10)) then J.I:=0; { floating error fix }
End;


Procedure MatrixInverse(N:Integer; Var Poutput:Matrix);
(******** Calculates the inverse of a matrix ****************************)

Var Determ:Real;
   I,J,K,L,L1:Integer;
   Ipivot:Array[1..(8*MaxN)] of Integer;
   Pivot:Array[1..(8*MaxN)] of Real;
   Index2:Array[1..(8*MaxN), 1..2] of integer;
   Irow,Icol,Jrow,Jcol:Integer;
   Swap,Amax,T:Real;

Begin
  Writeln;
  Write('Inverting Matrix');

  Determ:=1.0;
  For I:=1 to N do
     Ipivot[I] := 0;

  For I:=1 to N do              { Find the pivot element        }
  Begin
   Write('.');
   Amax:=0;
   Icol:=-1;
   For J:=1 to N do
    If (Ipivot[J]<>1) then
    Begin
     For K:=1 to N do
     Begin
      If ((Ipivot[K]-1)>0) then Exit;
      If ((Ipivot[K]-1)<0) and ((Abs(Amax)-Abs(Poutput[J][K]))<0) then
      Begin
       Irow:=J;              { Store row and column of pivot   }
       Icol:=K;
```

```
        Amax:=Poutput[J][K];
       End;
      End;
     End;
    if (Icol<0) then
    Begin
     Writeln('Invalid Matrix');
     Exit;
    End;                              { If Icol             }
    Ipivot[Icol] := Ipivot[Icol]+1;

    If (Irow<>Icol) then              { Swap the rows to put the pivot  }
    Begin                             { element on a the diagonal       }
     Determ:=-Determ;
     For L:=1 to N do
     Begin
      Swap := Poutput[Irow][L];
      Poutput[Irow][L] := Poutput[Icol][L];
      Poutput[Icol][L] := Swap;
     End;                             { For L Loop          }
    End;                              { If Irow             }
    Index2[I][1] := Irow;
    Index2[I][2] := Icol;
    Pivot[I] := Poutput[Icol][Icol];
    Determ := Determ * Pivot[I];


    Poutput[Icol][Icol]:=1;           { Divide pivot row by pivot element }
    For L:=1 to N do
     If (Pivot[I]<>0) then
      Poutput[Icol][L]:=(Poutput[Icol][L]/Pivot[I])
     Else
     Begin
      Writeln('Invalid Matrix');
      Exit;
     End;                             { Else Pivot          }


    For L1:=1 to N do                 { Reduce the non-pivot rows       }
     If (l1<>Icol) then
     Begin
      T:=Poutput[L1][Icol];
      Poutput[L1][Icol]:=0;
      For L:=1 to N do
       Poutput[L1][L]:=Poutput[L1][L]-Poutput[Icol][L]*T;
     End;                             { If ICol             }
   End;                               { For I Loop          }


   For I:=1 to N do                   { Swap the columns around         }
   Begin
    L:=N+1-I;
    If (Index2[L,1]<>Index2[L,2]) then
    Begin
     Jrow:=Index2[L,1];
     Jcol:=Index2[L,2];
     For K:=1 to N do
     Begin
      Swap:=Poutput[K][Jrow];
      Poutput[K][Jrow]:=Poutput[K][Jcol];
      Poutput[K][Jcol]:=Swap;
     End;                             { For K Loop          }
    End;                              { If Index            }
   End;                               { For I Loop          }

  End;                                { Procedure MatrixInvert }
```

```
Procedure IndMatInv(N:Integer; Var Poutput:CMatrix);
{******** Calculates the inverse of a matrix ****************************}

Var Determ:Real;
   I,J,K,L,L1:Integer;
   Ipivot:Array[1..(8*MaxN)] of Integer;
   Pivot:Array[1..(8*MaxN)] of Real;
   Index2:Array[1..(8*MaxN), 1..2] of integer;
   Irow,Icol,Jrow,Jcol:Integer;
   Swap,Amax,T:Real;

Begin
 Writeln;
 Write('Inverting Matrix');

 Determ:=1.0;
 For I:=1 to N do
    Ipivot[I] := 0;

 For I:=1 to N do                  { Find the pivot element          }
 Begin
  Write('.');
  Amax:=0;
  Icol:=-1;
  For J:=1 to N do
   If (Ipivot[J]<>1) then
   Begin
    For K:=1 to N do
    Begin
     If ((Ipivot[K]-1)>0) then Exit;
     if ((Ipivot[K]-1)<0) and ((Abs(Amax)-Abs(Poutput[J][K]))<0) then
     Begin
      Irow:=J;                { Store row and column of pivot    }
      Icol:=K;
      Amax:=Poutput[J][K];
     End;
    End;
   End;
  if (Icol<0) then
  Begin
   Writeln('Invalid Matrix');
   Exit;
  End;                              { If Icol             }
  Ipivot[Icol] := Ipivot[Icol]+1;

  If (Irow<>Icol) then             { Swap the rows to put the pivot   }
  Begin                            { element on a the diagonal       }
   Determ:=-Determ;
   For L:=1 to N do
   Begin
    Swap := Poutput[Irow][L];
    Poutput[Irow][L] := Poutput[Icol][L];
    Poutput[Icol][L] := Swap;
   End;                             { For L Loop          }
  End;                              { If Irow             }
  Index2[I][1] := Irow;
  Index2[I][2] := Icol;
  Pivot[I] := Poutput[Icol][Icol];
  Determ := Determ * Pivot[I];


  Poutput[Icol][Icol]:=1;          { Divide pivot row by pivot element }
  For L:=1 to N do
   If (Pivot[I]<>0) then
    Poutput[Icol][L]:=(Poutput[Icol][L]/Pivot[I])
   Else
```

```
    Begin
      Writeln('Invalid Matrix');
      Exit;
    End;                              { Else Pivot        }


   For L1:=1 to N do               { Reduce the non-pivot rows        }
    If (l1<>Icol) then
    Begin
      T:=Poutput[L1][Icol];
      Poutput[L1][Icol]:=0;
      For L:=1 to N do
        Poutput[L1][L]:=Poutput[L1][L]-Poutput[Icol][L]*T;
    End;                              { If ICol          }
  End;                                { For I Loop        }


  For I:=1 to N do                 { Swap the columns around        }
  Begin
   L:=N+1-I;
   If (Index2[L,1]<>Index2[L,2]) then
   Begin
    Jrow:=Index2[L,1];
    Jcol:=Index2[L,2];
    For K:=1 to N do
    Begin
      Swap:=Poutput[K][Jrow];
      Poutput[K][Jrow]:=Poutput[K][Jcol];
      Poutput[K][Jcol]:=Swap;
    End;                              { For K Loop        }
   End;                               { If Index        }
  End;                                { For I Loop        }

End;                                  { Procedure MatrixInvert }

(**********************************************************************)
(* Even and Odd Distribution Functions - He() and Ho()              *)
(* These really are the core of the program and the method          *)
(* To really understand them, you need to read Matthaei's paper     *)
(* and maybe plug these formulas into a two and three dimensional   *)
(* graphing program (at least that's what I did)                    *)
(*                                                                   *)
(**********************************************************************)

Function He(Z:Complex; B:ComplexCrd; W:Real):Real;
(************ Works out the even potential function *****************)

Var Temp1,Temp2:Complex;

Begin
    Temp1.R:=((Z.R-B.R)/(W/2));
    Temp1.I:=((Z.I-B.I)/(W/2));
    Arcsin(Temp1,Temp2);
    He:=((-1/(2*PI))*Abs(Temp2.I));
End;                                       { Function He() }

Function Ho(Z:Complex; B:ComplexCrd; W:Real):Real;
(*********** Works out the odd potential function ******************)

Var Temp,Temp2:Complex;   { Temp variables are needed since functions }
                          { can't be used for complex routines        }
Begin
    Z.R:=Z.R-B.R;
    Z.I:=Z.I-B.I;
    ComplexPowers(Z,Temp,2);   { Temp = Z ^ 2              }
    Temp.R:=(Temp.R-Sqr(W/2));
    ComplexSqrt(Temp,Temp);    { Temp = (Temp - (W/2)^2)^0.5  }
```

```
   If (Z.R<0) then
     Ho:=Z.R+Temp.R
     Else Ho:=Z.R-Temp.R;
End;                                    { Function Ho() }


Procedure CalcMatrix(N:Integer; Dims:Dmatrix; Var Blist:List; Var Zlist:Coords;
              Var PePo,IPePo:Matrix);

(*** Brings it all together and works out the overall array ***)

Var D,Temp:Real;
   M,Mnew,K,A:Integer;
   BCompConj,BPrime:ComplexCrd;       { CompConj -> Complex conjugate     }
   ZCompConj,ZPrime:Complex;          { Prime -> Num multiplied by -i     }

Begin
  D:=-1;                         { Finds the reflection value of     }
                                 { Dielectric interface              }
  Writeln;
  Write('Calculating PePo Matrix');
  For M:=1 to (4*N) do
  Begin
     Write('.');
     Mnew:=(M Div 4);
     If ((M Mod 4)<>0) then Inc(Mnew);
     For K:=1 to (8*N) do
     Begin
      If (Blist[M].H) then    (* If B is horizontal *)
      Begin
       BCompConj.R:=Blist[M].R;     { Takes the complex conjugate of }
       BCompConj.I:=-Blist[M].I;    { Zk and Bm                      }

       PePo[K,M]:=He(Zlist[K],Blist[M],Dims[Mnew].W)+D*He(Zlist[K],BCompConj,Dims[Mnew].W);
       PePo[K,(M+4*N)]:=Ho(Zlist[K],Blist[M],Dims[Mnew].W)+D*Ho(Zlist[k],BCompConj,Dims[Mnew].W);
      IPePo[K,M]:=He(Zlist[K],Blist[M],Dims[Mnew].W);
      IPePo[K,(M+4*N)]:=Ho(Zlist[K],Blist[M],Dims[Mnew].W);
      End                            { If (Blist[M].H)     }
        Else
        Begin
         ZPrime.R:=Zlist[K].I;      { Multiplies Zk and Bm by -i for   }
         ZPrime.I:=-Zlist[K].R;     { symmetry reasons in for          }
         BPrime.R:=Blist[M].I;      { vertical calculations            }
         BPrime.I:=-Blist[M].R;

         ZCompConj.R:=-Zlist[K].I;  { Takes the complex conjugate      }
         ZCompconj.I:=-Zlist[K].R;  { and multiplies by -i             }

         PePo[K,M]:=He(Zprime,BPrime,Dims[Mnew].T)+D*He(ZCompConj,BPrime,Dims[Mnew].T);
         PePo[K,(M+4*N)]:=Ho(ZPrime,BPrime,Dims[Mnew].T)+D*Ho(ZCompConj,BPrime,Dims[Mnew].T);
        IPePo[K,M]:=He(Zprime,BPrime,Dims[Mnew].T);
        IPePo[K,(M+4*N)]:=Ho(ZPrime,BPrime,Dims[Mnew].T);

        End;                         { Else (Blist[M].H)   }
     End;                            { For K Loop          }
  End;                               { For M Loop          }
  Writeln;
End;                                 { Procedure CalcMatrix }

Procedure CapCalc(N:Integer; Var PePo,IPePo:Matrix; Var CapMat,IndMat:CMatrix);
(**** Calculates the capacitive matrix.  The G value is ignored ****)

Var Q,Qi:Vector;
   M,P,K,B,V:Integer;

Begin
  MatrixInverse((8*N),PePo);
```

```
 MatrixInverse((8*N),IPePo);

 For P:=1 to N do          { Cycles through each conductor           }
 Begin
  For M:=1 to (4*N) do     { Cycles through each side of each conductor }
  Begin
    Q[M]:=0;           { Resets the current Q value             }
    Qi[M]:=0;
    For K:=1 to (8*N) do  { Cycles through each of the two "match      }
    Begin              { points" on each side of each conductor     }

                    { These IF statements check to see if the    }
                    { current match point (K) is on the current  }
                    { conductor (P) and, if so, sets the         }
                    { conductor to unity, else to zero           }
      If (K=(2*P-1))        Or (K=(2*P))        Or
        (K=((2*N)+(2*P-1))) Or (K=((2*N)+(2*P))) Or
        (K=((4*N)+(2*P-1))) Or (K=((4*N)+(2*P))) Or
        (K=((6*N)+(2*P-1))) Or (K=((6*N)+(2*P))) Then V:=VConst
        Else V:=0;
                    { This addition statement performs the matrix }
                    { multiplication of the PePo matrix with the }
                    { V matrix.  PePo[K,M] is the Pe matrix value }
                    { while PePo[K,(M+4*N+1)] is the Po component }
      Q[M]:=Q[M]+V*(PePo[M,K]);
      Qi[M]:=Qi[M]+V*(PePo[M,K]);
    End;                             { For K Loop      }
  End;                               { For M Loop      }
  Writeln;
                    { Adds the charges of each of the four       }
  For B:=1 to N do        { conductor sides to get total charge        }
  Begin
    CapMat[B,P]:=E0*Leng*Eo*(Q[B]+Q[(B+N)]+Q[(B+2*N)]+Q[(B+3*N)]);
    IndMat[B,P]:=E0*(Qi[B]+Qi[(B+N)]+Qi[(B+2*N)]+Qi[(B+3*N)]);
  End;                               { For B Loop      }
 End;                                { For P Loop      }
End;                                 { Procedure CapCalc }

Procedure IndCalc(N:Integer; Var IndMat:CMatrix);

Begin
   IndMatInv(N,IndMat);
   For I:=1 to N do
      For J:=1 to N do
         IndMat[I,J]:=IndMat[I,J]*Leng*E0*U0;
End;

Procedure ResCalc(N:Integer; Var Dims:DMatrix; Var R:Vector);

Begin
   For I:=1 to N do
      With Dims[I] do
          R[I]:=(Leng*Rho)/(W*T);
End;


Procedure ShowResults(N:Integer; Var Capmat,IndMat:CMatrix; Var ResMat:Vector);
Begin
 If (N<4) then
 Begin
   ClrScr;
   Writeln;
   Writeln('Capactive Matrix');
   Writeln('—————');
   Writeln;
   For I:=1 to N do
   Begin
```

```
      For J:=1 to N do
      Begin
         SIUnits(CapMat[I,J],Output);
         Write('F     ');
      End;                    { For J Loop   }
      Writeln;
   End;                       { For I Loop   }
   Writeln;
   Writeln('Inductive Matrix');
   Writeln('————————');
   Writeln;
   For I:=1 to N do
   Begin
      For J:=1 to N do
      Begin
         SIUnits(IndMat[I,J],OutPut);
         Write('H     ');
      End;
      Writeln;
   End;
   Writeln;
   Writeln('Resistive List');
   Writeln('————————');
   For I:=1 to N do
   Begin
         SIUnits(ResMat[I],Output);
         Write('Ohms   ');
   End;
   Writeln;

   Writeln('Press any key to continue');
   Ch:=Readkey;
  End
  Else
  Begin
    ClrScr;
    Writeln;
    Writeln('Systems with more than 4 conductors are not able to fit on the screen');
    Writeln('Try printing out the matrix on a wide carriage printer or outputting');
    Writeln('to a text file and looking at the file using a text viewer at a higher');
    Writeln('resolution.');
    Writeln;
    Writeln('Press any key to continue.');
    Ch:=Readkey;
  End;

End;                          { Procedure ShowResults }


Procedure OutputResults(Var Dims:Dmatrix; N:Integer; Var CapMat:CMatrix;
                  Var File1:Text);
(* A straightforward procedure to print the output of the calculation to *)
(* a device specified by the input into File1.  Can be the printer, the  *)
(* screen or a DOS text file                                 *)


Var Day,Month,Year,DayofWeek:Word;     { Used to hold the date information }


Begin
   GetDate(Year,Month,Day,DayofWeek);
   Writeln(File1);
   Writeln(File1,'Capacitive Calculation Program Results: ');
   Writeln(File1);
   Writeln(File1,'Date: ',Day,'/',Month,'/',Year);
   Writeln(File1);
   Writeln(File1);
```

```
   Writeln(File1);
   Writeln(File1);
   Writeln(File1,'Conductor     Width     Height     Thick    Separation.');
   Writeln(File1,'----------------------------------------------------------');
   Writeln(File1);
   For I:=1 to N do
   Begin
      Write(File1,'  #',I,'     ');
      SIUnits(Dims[I].W,File1);
      Write(File1,'m');
      SIUnits(Dims[I].H,File1);
      Write(File1,'m');
      SIUnits(Dims[I].T,File1);
      Write(File1,'m');
      SIUnits(Dims[I].S,File1);
      Writeln(File1,'m');
   End;


   Writeln(File1);
   Writeln(File1);
   Writeln(File1,'Capacitive Matrix');
   Writeln(File1,'-----------------');
   Writeln(File1);

   For I:=1 to N do
   Begin
     For J:=1 to N do
     Begin
       SIUnits(CapMat[I,J],File1);
       Write(File1,'F    ');
     End;
     Writeln(File1);
   End;

   Writeln(File1);
   Writeln(File1,'Inductive Matrix');
   Writeln(File1,'-----------------');
   Writeln(File1);

   For I:=1 to N do
   Begin
     For J:=1 to N do
     Begin
       SIUnits(IndMat[I,J],File1);
       Write(File1,'H    ');
     End;
     Writeln(File1);
   End;

   Writeln(File1);
   Writeln(File1,'Resistive Matrix');
   Writeln(File1,'-----------------');
   Writeln(File1);

   For I:=1 to N do
   Begin
      SIUnits(ResMat[I],File1);
      Write(File1,'Ohms    ');
   End;


End;

Procedure WriteToSpice(N:Integer; Var C,L:CMatrix; Var R:Vector;
                  Var File1:Text);
```

```
Var I,J:Integer;
  K:Real;
  Day,Month,Year,DayofWeek:Word;    { Used to hold the date information }


Begin
   GetDate(Year,Month,Day,DayofWeek);
   Writeln(File1);
   Writeln(File1,'* Date: ',Day,'/',Month,'/',Year);
   Writeln(File1);
   Writeln(File1);

   { Options }
   Writeln(File1,'.OPTION POST');
   Writeln(File1,'.TRAN 0.5n 51.1n');
   Writeln(File1);


   { Voltage Sources }
   Writeln(File1,'VDD   3 0 DC 5V');
   For I:=1 to N do
      Writeln(File1,'VIN',I,' ',I,'1  0  PULSE(0V 5V 1n 0.1n  0.1n  50n 51.5n)');
   Writeln(File1);

   { Driving Transistors}
   For I:=1 to N do
   Begin
      Writeln(File1,'M1',I*2-1,'    3   ',I,'1    ',I,'2    3   PMOS3   L=u W=7.5u');
      Writeln(File1,'M1',I*2,'    3   ',I,'8    ',I+1,'0    3   PMOS3   L=u W=7.5u');
   End;

   Writeln(File1);

   For I:=1 to N do
   Begin
      Writeln(File1,'M2',I*2-1,'  ',I,'2   ',I,'1     0    0   NMOS3   L=u W=3u');
      Writeln(File1,'M2',I*2,'  ',I+1,'0   ',I,'8     0    0   NMOS3   L=u W=3u');
   End;

   { Diagonal first }
   Writeln(File1,'* Transmission lines with no coupling');
   For I:=1 to N do
   Begin
      Writeln(File1,'C',I,'a     ',I,'2     0     ',C[I,I]/6);
      Writeln(File1,'L',I,'b     ',I,'2     ',I,'3     ',L[I,I]/3);
      Writeln(File1,'R',I,'b     ',I,'3     ',I,'4     ',R[I]/3);
      Writeln(File1,'C',I,'b     ',I,'4     0     ',C[I,I]/3);
      Writeln(File1,'L',I,'c     ',I,'4     ',I,'5     ',L[I,I]/3);
      Writeln(File1,'R',I,'c     ',I,'5     ',I,'6     ',R[I]/3);
      Writeln(File1,'C',I,'c     ',I,'6     0     ',C[I,I]/3);
      Writeln(File1,'L',I,'d     ',I,'6     ',I,'7     ',L[I,I]/3);
      Writeln(File1,'R',I,'d     ',I,'7     ',I,'8     ',R[I]/3);
      Writeln(File1,'C',I,'d     ',I,'8     0     ',C[I,I]/6);
   End;
   { Then the coupled inductances }
   Writeln(File1,'* Mutual inductances');
   For I:=1 to N do
     For J:=(I+1) to N do
     Begin
        K:=(L[J,I]/(Sqrt(L[J,J]*L[I,I])));
        Writeln(File1,'K',J,I,'b   L',I,'b   L',J,'b     ',K/3);
        Writeln(File1,'K',J,I,'c   L',I,'c   L',J,'c     ',K/3);
        Writeln(File1,'K',J,I,'d   L',I,'d   L',J,'d     ',K/3);
     End;

   { Finally, the coupled capacitances }
   Writeln(File1,'* Mutual capacitances');
```

```
    For I:=1 to N do
       For J:=(I+1) to N do
       Begin
           Writeln(File1,'C',J,I,'a    ',I,'2    ',J,'2    ',C[J,I]/6);
           Writeln(File1,'C',J,I,'b    ',I,'4    ',J,'4    ',C[J,I]/3);
           Writeln(File1,'C',J,I,'c    ',I,'6    ',J,'6    ',C[J,I]/3);
           Writeln(File1,'C',J,I,'d    ',I,'8    ',J,'8    ',C[J,I]/6);
       End;
    Writeln(File1,'.END');
End;

Procedure OutputToFile(N:Integer; Select:Char; Var Dims:DMatrix;
                Var CM,LM:CMatrix; Var RM:Vector);

(* Checks to make certain that the filename is valid *)

Var L,Dot:Integer;                  { L is length, Dot is dot position }
   Valid:Boolean;                   { Is the filename valid          }
   File1:Text;
   Filename:Name;

Begin
    ClrScr;
    Writeln;
    Writeln;
    Case Select of
       'A':Begin
             Writeln('Output to ASCII Text File');
             Writeln('————————');
          End;
       'B':Begin
             Writeln('Output to SPICE Text File');
             Writeln('————————');
          End;
    End;

    Writeln;
    Writeln('Type "quit" to exit back to the main menu.');
    Writeln;

    Repeat
       Valid:=True;
       Write('File name (Output.');
       If (Select='A') then Write('Txt): ')
         Else Write('Sp): ');
       Readln(Filename);
       L:=Length(Filename);


       If (L>12) then
       Begin                    { This checks to make certain the }
          Writeln('Filename too long. Extra ignored.');
                                { filename isn't too long and if  }
          Delete(Filename,12,(L-12)); { it is, then it truncates it    }
          L:=12;            { If L statement }
       End;

       Dot:=0;
       I:=1;
       While ((I<L) and (Valid)) do
       Begin
           { This massive IF statement just checks to make certain
             that only letters and/or numbers are being used as the
             filename and that the filename only has one dot (ie. '.')
             in it                                  }
          If (((Dot<>0) and (Filename[I]='.')) or
             ((Ord(Filename[I])<48) and Not(Ord(Filename[I])=46)) or (Ord(Filename[I])>122) or
```

```
        ((Ord(Filename[I])>58) and (Ord(Filename[I])<65)) or
        ((Ord(Filename[I])>90) and (Ord(Filename[I])<97))) then
      Begin
        Writeln;
        Writeln('Invalid File name.  Non-readable character: ',Filename[I]);
        Writeln('Please type in another or type quit to abort');
        Writeln;
        Valid:=False;
      End;            { If Dot statement }

      If Filename[I]='.' then Dot:=I;
      Inc(I);
    End;              { For L Loop      }
  Until (Valid);

  If (L=0) then             { This uses the default file name if the }
  Begin                     { user doesn't input one                 }
    Filename:='Output';
    L:=6;
  End;                      { If L statement   }
                            { This adds on a tail '.txt' if another  }
                            { has not been already been specified    }
  If ((Dot=0) and (L<=8)) then
  Begin
    If (Select='A') then Filename:=Concat(Filename,'.txt')
     Else Filename:=Concat(Filename,'.sp');
  End
    Else If ((Dot=0) and (L>8)) then
      Begin
        Delete(Filename,8,(L-8));
        If (Select='A') then Filename:=Concat(Filename,'.txt')
          Else Filename:=Concat(Filename,'.sp');
      End;            { Else If statement }

                             { This makes all of the characters }
  For I:=1 to Length(Filename) do    { lowercase                   }
    Filename[I]:=Lowercase(Filename[I]);

  If ((Filename<>'quit.txt') and (Filename<>'quit.sp'))
    Then Writeln('Writing to file: ',Filename,'.')
      Else Select:='C';

  Writeln;
  Writeln;

  If (Select<>'C') then
  Begin
    Assign(File1,Filename);
    Rewrite(File1);
    If (Select='A') then OutputResults(Dims,N,CapMat,File1)
     else Begin
            Writeln(File1,Filename,'    Coupled ',N,' line SPICE file');
            WriteToSpice(N,CapMat,IndMat,ResMat,File1);
          End;
    Close(File1);
    Writeln('File successfully written.');
    Writeln('Press any key to continue.');
    Ch:=Readkey;
  End                 { If filename statement  }
    Else
    Begin
      Writeln('Aborted.  File not written.');
      Writeln('Press any key to continue.');
      Ch:=Readkey;
    End;              { Else statement       }
End;                  { Procedure OutputToFile }
```

```
Begin
 Intro;                 { Introduce the Program                    }

 Status:=1;   { Contains the status of the program through   }
          { the loop:  1. nothing to calculate, nothing calculated   }
          {          2. something to calculate                 }
          {          3. nothing to calculate, something calculated }
          {          4. end program                        }

 Repeat
   Menu(Status,Choice);  { Show menu and store input in Choice        }
   Case Choice of         { Do specified action based on Choice       }
     'a':Begin
          Writeln;
          Writeln('Does the system:');
          Writeln('  a. Contain identical conductors all in the same plane.');
          Writeln('  b. Contain non-uniform conductors all in the same plane.');
          Writeln('  c. Contain non-uniform conductors in different planes.');

          Repeat
              Layout:=Lowercase(Readkey);
          Until ((Layout='a') or (Layout='b') or (Layout='c') or (Ord(Layout)=27));

          If (Layout='a') then Identical(N,Dims)
            Else If (Layout='b') then SemiIdentical(N,Dims)
                Else If (Layout='c') then NotIdentical(N,Dims);
          If (Ord(Layout)<>27) then Status:=2;
        End;                          { Case Choice A }
     'b': Begin
          ClrScr;
          Writeln('Not implemented yet.');
          Writeln;
          Writeln('Press any key to return to menu.');
          Answer:=Readkey;
        End;
     'c': Begin                       { Case Choice B }
          ClrScr;
          Writeln;
          Writeln;
          Writeln('Print Results');
          Writeln('———————');
          Writeln;
          Writeln('  A. To screen');
          Writeln('  B. To printer');
          Writeln;
          Repeat
              Answer:=LowerCase(Readkey);
          Until ((Answer='a') or (Answer='b') or (Ord(Answer)=27));

          Case Answer of
              'a': ShowResults(N,CapMat,IndMat,ResMat);   { Writes to screen      }
              'b': OutputResults(Dims,N,CapMat,Lst);
                                 { Print results }
          End;                     { Case Answer   }
        End;                       { Case Choice C }

     'd': OutputToFile(N,'A',Dims,CapMat,IndMat,ResMat); { Write data to text file }

     'e': OutputToFile(N,'B',Dims,CapMat,IndMat,ResMat); { Write data to text file }


     'q': Begin
          Writeln;
          Writeln('Exit (y/n)?');
          Repeat
              Answer:=LowerCase(Readkey);
          Until ((Answer='y') or (Answer='n'));
```

```
        If (Answer='y') then status:=4;
      End;                        { Case Choice Q }

    Else Begin
        Writeln('This has not yet been implemented.');
        Writeln('Press any key to continue.');
        Ch:=Readkey;
      End;                        { Case Choice Else }
  End;                            { Case Choice  }

  If (Status=2) then              { Does the calculations }
  Begin                           { If they haven't been  }
    Convert(N,Dims,Blist,Zlist);
    CalcMatrix(N,Dims,Blist,Zlist,PePo,IPePo);
    CapCalc(N,Pepo,IPePo,CapMat,IndMat);
    IndCalc(N,IndMat);
    ResCalc(N,Dims,ResMat);
    Status:=3;                    { Nothing to calculate  }
  End;                            { If (EndFlag)  }
 Until Status=4;
End.                              { Main Program  }
```

References:

[1] A.Feller, H.R.Kaupp and J.J.Digiacomo, "Crosstalk and reflections in high-speed digital systems," *Proc. Fall Joint Computer Conf.*, Dec. 1965, pp. 511-525.
[2] I.Catt, "Crosstalk (noise) in digital systems," *IEEE Trans. Electron. Comput.*, vol. EC-16, pp. 743-763, Dec. 1967.
[3] J.A.DeFalco, "Reflection and crosstalk in logic circuit interconnections," *IEEE Spectrum*, pp.44-50, July 1970.
[4] J.C.Isaacs and N.A.Strahkov, "Crosstalk in uniformly coupled lossy transmission lines," *Bell System Tech. Journal*, vol. 52, no. 1, pp. 101-115, Jan. 1973.
[5] W.Su, S.M.Raid, A.Elshabini-Raid and T.Poulin, "Crosstalk analysis of multisection multiconductor lines," *IEEE Trans. on Instr. and Measure*, vol. 41, pp. 926-931, Dec. 1991
[6] L.T.Pillage and R.A.Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Comput.-Aided Design*, vol. 9, pp. 352-366, Apr. 1990.
[7] T.Tang and M.Nakhla, "Analysis of high-speed VLSI interconnections using the asymptotic waveform evalutation technique," *IEEE Trans. Comput.-Aided Design*, vol. 11, no.3, pp. 341-352, March 1992.
[8] V.K.Tripathi and J.B.Rettig, "A SPICE model for multiple coupled microstrips and other transmission lines," *IEEE Trans. Microwave Theory Tech.*, MTT-33, no. 12, pp. 1513-1517, Dec. 1985.
[9] V.K.Tripathi and R.J.Bucolo, "Analysis and modelling of multilevel parallel and crossing interconnection lines," *IEEE Trans. Electron Devices*, ED-34, no. 3, March 1987.
[10] D.A.Papaioannou, K.Z.Dimopoulos and J.N.Avarisiotis, "Simulation of crosstalk in high-speed multilayer off-chip interconnections," *Microelect. Journal.*, vol. 24, pp. 747-757, 1993.
[11] K.Chang, N.H.Chang, S.Oh and K.Lee, "Parameterized SPICE subcircuits for multilevel interconnect modelling and simulation," *IEEE Trans. Circuits and Systems*, vol. 39, no. 11, pp. 779-788, Nov. 1992.
[12] J.Kim and J.McDonald, "Transient and crosstalk analysis of slightly lossy interconnection lines for wafer scale integration and wafer scale hybrid packaging - weak coupling case," *IEEE Trans. Circuits and Systems*, vol. 35, November 1988.
[13] T.Sakurai, "Approximation of wiring delay in MOSFET LSI", *IEEE Jour. Solid-state Circuits*, vol. 18, pp. 418-426, August 1983.

[14] Y.V.Rajput, "Modeling distributed RC lines for transient analysis of complex networks," *Int. Jour. of Electronics*, vol. 36, p. 709, May 1974.
[15] G.D.Mey, "A Comment on 'The modelling of Resistive Interconnects for Integrated Circuits'", *IEEE J. Solid State Circuits*, SC-18(2), p. 200-203, April 1983
[16] H.B.Bakoglu, "Circuits, Interconnections and Packaging for VLSI," Addison-Wesley, 1990.
[17] A.K.Goel, *High-speed VLSI Interconnections*, Wiley, 1994, p.24
[18] H.B.Palmer, "The capacitance of a parallel-plate capacitor by the Schwartz-Christoffel transformation," *Trans. AIEE*, vol. 56, pp. 363, March 1937.
[19] R.H.Jansen, "High-speed computation of single and coupled microstrip parameters including dispersion, high-order modes, loss and finite strip thickness," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-26, pp. 75-82, Feb. 1978.
[20] E.T.Lewis, "An analysis of interconnect line capacitance and coupling for VLSI circuits," *J. Solid State Elec.*, Vol. 28, pp. 741-749, Aug. 1984.
[21] K.C.Gupta, R.Garg and I.J.Bahl, *Microstrip and Slotlines*, Artech, 1979.
[22] H.Yuan, Y.Lin and S.Chiang, "Properties of interconnection on silicon, sapphire and semi-insulating GaAs substrates," *IEEE Trans. Electron Devices*, vol. 29, April 1992.
[23] Y.Saad and M.H.Schultz, "GMRES: a generalised minimal residual algorithm for solving non-linear symmetric systems," *SIAM Jour. on Scientific and Statistical Computing*, vol. 7, pp. 856-869, July 1986.
[24] T.F.Hayes and J.J.Barrett, "Modeling of multiconductor systems for packaging and interconnecting high-speed digital IC's," *IEEE Trans. Computer Aided Design*, vol. 11, pp. 424-430, April 1992.
[25] K.Nabors and J.White, "Fastcap: A multi-pole accelerated 3-D capacitance extraction program," *IEEE Trans. on Computer Aided Design*, vol. 10, pp. 1447-1459, November 1991.
[26] K.Nabors, S.Kim and J.White, "Fast capacitance extraction of general three-dimensional structures," *IEEE Trans. Microwave Theory Tech*, vol. 40, pp.1496-1506, July 1992.
[27] W.T.Weeks, "Calculation of coefficients of capacitance of multiconductor transmission lines in the presence of a dielectric interface," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-18, pp. 35-43, Jan. 1970.
[28] M.Kanon, et.al, "Efficient techniques for the inductance extraction of complex 3-D geometries," *Proc. Int. Conf. on Computer Aided Design*, November 1992.
[29] P.A.Brennan, N.Raver and A.Ruehli, "Three dimensional inductance computations with partial element equivalent circuits," *IBM Jour. of Res. and Develop.*, vol. 23, pp. 661-668, November 1979.
[30] N.P.van der Meijs and A.J. van Genderen, "An efficient finite element method for submicron IC capacitance extraction," *Proc. DAC*, 1989
[31] N.P.van der Meijs and A.J. van Genderen, "3-Dimensional Finite Element Modeling of Integrated Circuit Capacitance," *Algorithms and Parallel VLSI Architectures*, Elsevier, 1991
[32] G.L.Matthaei, et.al, "Computation of Interconnect Distributed Capacitances and Inductances," *IEEE Trans. Computer Aided Design*, Vol. 11, pp. 512-523, April 1992.
[33] P.Silvester, *Modern Electromagnetic Fields*, Prentice-Hall, 1968, pp.101-103.
[34] C.P.Yuan and T.N.Trick, "A simple formula for the estimation of the capacitance of two-dimensional interconnects in VLSI circuits," *Electron Device Letters*, EDL-3, no. 12, pp. 391-393, Dec. 1982
[35] M.Michaelides, "Microstrip transmission lines: impedance matching, coupling and filtering," *Mullard Technical Comms.*, vol. 12, no. 116, pp. 167-185, Oct. 1972.
[36] T.Sakurai and K.Tamaru, "Simple formulas for two- and three-dimensional capacitances", *IEEE Trans. on Electron Devices*, vol. ED-30, no. 2, Feb. 1983.
[37] M.I.Elmasry, "Capacitance calculation in MOSFET VLSI," *IEEE Electron Device Letters*, EDL-3, no.1 pp.6-7, Jan. 1982.
[38] W.H.Chang, "Analytical IC metal-line capacitance formulas", *IEEE Trans. Microwave Theory Tech.*, MTT-34, pp. 608-611, Sept. 1979.
[39] C.Wei, R.F.Harrington, J.R.Mautz and T.K.Sarkar, "Multiconductor transmission lines in multilayered dielectric media," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-32, pp. 439-450, April 1994

[40] R.L.Wigington and N.S.Nahman, "Transient analysis of coaxial cables considering the skin effect," *Proc. IRE*, vol. 45, pp. 166-174, February 1957.
[41] C.Yen, Z.Fararinc and R.L.Wheeler, "Time domain skin effect model for transient analysis of lossy transmission lines," *Proc. IEEE*, vol. 70, pp. 750-757, July 1982.
[42] J.Kim and J.McDonald, "Transient and crosstalk analysis of slightly lossy interconnection lines for wafer scale integration and wafer scale hybrid packaging - weak coupling case," *IEEE Trans. Circuits and Systems*, vol. 35, November 1988.
[43] H.Hasegawa and Seki, "Analysis of interconnection delay on very high-speed LSI/VLSI chips using an MIS microstrtip line model," *IEEE Trans. Microwave Theory and Tech.*, vol. 32, December 1984
[44] R.H.Dennard et.al., "Scaling of MOSFET devices", *IEEE circuits and Devices*, SC-9, p.256, Jan. 1974.
[45] Eurochip Ltd, Zone Industrialle, 13106 Rousset Cedex, France.
[46] D.W.Dobberpuhl, et. al., "A 200MHz 64-b dual-issue CMOS microprocessor," *IEEE Jour. Solid-State Circuits*, vol. 27, no. 11, pp. 1555-1564, Nov. 1992.
[47] S.Bothra, M.Kellam and C.M.Osborn, "Analysis of the effects of scaling on interconnect delay in ULSI circuits," *IEEE Trans. Electron Devices*, vol. 40, no. 3, pp.591-597, Mar. 1993.
[48] Inmos Ltd, 1000 Aztec West, Almondsbury, Bristol, BS12 4SQ, Great Britain.
[49] Meta-Software Inc., *HSPICE User's Manual*, Meta-Software Inc., Campbell, CA July 1991.