

1991

# The development and application of high-speed digital switching trees for regular arithmetic arrays.

Lino. Del Pup  
*University of Windsor*

Follow this and additional works at: <http://scholar.uwindsor.ca/etd>

---

## Recommended Citation

Del Pup, Lino., "The development and application of high-speed digital switching trees for regular arithmetic arrays." (1991). *Electronic Theses and Dissertations*. Paper 3824.

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email ([scholarship@uwindsor.ca](mailto:scholarship@uwindsor.ca)) or by telephone at 519-253-3000ext. 3208.



National Library  
of Canada

Bibliothèque nationale  
du Canada

Canadian Theses Service    Service des thèses canadiennes

Ottawa, Canada  
K1A 0N4

## NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

## AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

**THE DEVELOPMENT AND APPLICATION OF HIGH-  
SPEED DIGITAL SWITCHING TREES FOR REGULAR  
ARITHMETIC ARRAYS**

by

Lino Del Pup

A Thesis

Submitted to the Faculty of Graduate Studies through the  
Department of Electrical Engineering in Partial Fulfillment  
of the Requirements for the Degree of  
Master of Applied Science at the  
University of Windsor

Windsor, Ontario

August, 1991



National Library  
of Canada

Bibliothèque nationale  
du Canada

Canadian Theses Service    Service des thèses canadiennes

Ottawa, Canada  
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-69891-8

Canada

Lino Del Pup 1991

© All Rights Reserved

## ABSTRACT

This work represents the author's contributions to a joint effort into the development of switching tree architectures. The work concentrates on concepts of CMOS design and lay-out stressing the optimization of parasitic and interconnect capacitance to increase circuit performance. Practical design techniques are discussed with regards to constructing pad frames and power distribution networks. These concepts are then applied to a set of three cells which trace the development of the switching tree cell. The clocking schemes, ROM architectures and general structure of each cell are discussed in detail. Finally, a set of standard cells is constructed using the switching tree structure, in  $3\mu$  CMOS technology. SPICE simulations indicate that cells performing 3-bit binary addition in one clock cycle are capable of functioning at clock speeds of up to 50 Mhz. Test results confirm the functionality of the cells up to the maximum speed of the Input/Output (I/O) pads, which is approximately 40 Mhz. Due to current I/O pad speed limitations, a set of ECL Compatible I/O pads is developed for use as part of the standard cell library. Simulation results indicate that the pads are capable of operating at speeds above 100 Mhz. Test results confirm the pads operation up to 65 Mhz, which is the maximum speed available from the test equipment. Based on the test results, it is estimated that the maximum speed of the  $3\mu$  ECL Compatible I/O pads is approximately 80 Mhz.

## ACKNOWLEDGEMENTS

*I would like to express my sincere thanks and appreciation to Dr. G. A. Jullien and Dr. W. C. Miller for their tremendous support and guidance throughout the progress of this thesis. Much thanks must also be given to my good friend and working partner Roger Grondin. In addition, I would also like to thank Mr. J. Novosad and Mr. A. Johns for many favours rendered. Thanks must also go to Jane Bailey for her support and patience during the writing process. Last but far from least, I would like to thank my parents, Mr. & Mrs. Achille and Edda Del Pup, for without them none of this would be possible.*

# TABLE OF CONTENTS

ABSTRACT	iv
ACKNOWLEDGEMENTS	v
LIST OF FIGURES	x
LIST OF TABLES	xviii
CHAPTER 1	
Introduction	1
1.1 Introduction	1
1.2 Thesis Organization	8
CHAPTER 2	
The Art of Design	10
2.1 Introduction	10
2.2 Dealing with Circuit Capacitance	11
2.2.1 Circuit Modelling for Proper Design	13
2.2.2 Parameterizing a Circuit Block	15
2.2.3 Summary of Modelling	25
2.3 Different Transistor Configurations	26
2.3.1 The Basic Transistor Layout	26
2.3.2 Parallel Devices with Merged Drains	27
2.3.3 Doughnut Devices	30
2.4 Practical Layout Considerations	31
2.4.1 Generating Pad Frames	31



2.4.2	Power Distribution	33
2.4.3	Routing	36
2.5	Summary	37
CHAPTER 3		
The Development of Switching Trees		39
3.1	Introduction	39
3.2	The ROM5 Cell	40
3.2.1	The Clocking Strategy	41
3.2.2	The Star Configured ROM	42
3.2.3	Test Results of The IC3WROMB Chip	48
3.2.4	Summary of ROM5 Design	49
3.3	The JROM Cell	50
3.3.1	The Binary ROM Cell	50
3.3.2	The New Clocking Strategy, JClock	54
3.3.3	Test Results of IC3WRRJT, WRXLT, WRJET	57
3.3.4	Summary of The JROM Design	60
3.4	The Switching Tree Cell	60
3.4.1	True Single-Phase Clocking (TSPC)	61
3.4.2	Svensson's True Single Phase Clocking	62
3.4.3	Negative Transition TSPC-1 D-Latch	66
3.4.4	Input / Output Buffering	67
3.4.5	The Complete Switching Tree Cell	69
3.4.6	Determination of Maximum ROM Height	71
3.4.7	Summary of The Switching Tree Cell Design	74
3.5	Heuristic ROM Cell Layout Techniques	74
3.6	Summary of The Development Work	82

## CHAPTER 4

Standard Cells	83
4.1 Introduction	83
4.2 Standard Sub-Cells	84
4.3 The 3-Bit Adder Cell	86
4.3.1 3-Bit Adder Simulation Results	88
4.4 The 3-Bit Multiplier Cell	91
4.4.1 3-Bit Multiplier Simulation Results	92
4.4.2 Test Results on The 3 Bit Multiplier Cell	94
4.5 The 4-Bit Adder Cell	95
4.5.1 4-Bit Adder Simulation Results	95
4.5.2 Test Results on The 4-Bit Adder Cell	98
4.6 The 6 Bit Multiplier Chip	98
4.7 The Development of 3 $\mu$ ECL Compatible Pads	99
4.7.1 Defining ECL and ECL Compatible Technology	101
4.7.2 Designing The ECL-Driver Pad	102
4.7.3 Designing The ECL-Receiver Pad	105
4.7.4 ECL Simulation Results	106
4.7.5 The Test Chip IC3WRECL	107
4.7.6 Test Results of IC3WRECL	108
4.8 Summary of Standard Cells	111

## CHAPTER 5

Summary and Future Directives	112
5.1 Summary and Conclusions	112
5.2 Future Directives	115

REFERENCES		117
APPENDIX A	A Summary on Dynamic Domino CMOS Logic	119
APPENDIX B	Complete Schematic Diagram of ROM5 Cell	123
APPENDIX C	Complete Schematic Diagram of JROM Cell	129
APPENDIX D	Schematic Diagram of 4-Bit Comparator Cell	133
APPENDIX E	Schematic Diagram of 5 to 3 Compressor Cell	136
APPENDIX F	Summary of Development of 1.2 $\mu$ CMOS I/O Pads	142
APPENDIX G	Schematic Diagram of 3-Bit Adder Cell	153
APPENDIX H	SPICE-Deck for 3-Bit Adder Cell	158
APPENDIX I	Schematic Diagram of 3-Bit Multiplier Cell	179
APPENDIX J	Schematic Diagram of 4-Bit Adder Cell	188
APPENDIX K	Basic Transmission Line Theory	196
APPENDIX L	ECL Receiver and Driver Spice-Decks	201
Vita Auctoris		206

## LIST OF FIGURES

1.1	A) Surface and, B) Cross-Sectional View of a Basic NMOS Transistor	1
1.2	Standard NMOS Circuit Design Technique	2
1.3	Localized Complimentary Well Construction of CMOS Process	3
1.4	Standard CMOS Logic Construction	3
1.5	A) Basic Clocking Scheme for Evaluating Dynamic Logic, and B) Pseudo Two-Phase Clocking Scheme	5
1.6	A) Nora Logic Block Illustrating Floating Circuit Nodes, and B) Domino Logic Block showing Single-Phase Clocking and Use of Signal Buffering	6
1.7	The Full Binary-Tree Structure with Tree Height, $n$ , Equal to Three	7
2.1	Illustrating Components of Interconnect Capacitance	12
2.2	Proper Input Circuit Model	13
2.3	Proper Output Circuit Model	14
2.4	Schematic Diagram of a 2 Input Nand Gate	15
2.5	Layout of 2 Input Nand Gate	16
2.6	Mask-Extracted SPICE-Deck of 2 Input Nand Gate	17
2.7	Complete Schematic for Simulating the Nand Gate	16
2.8	Additional Devices Added to the Original SPICE-Deck	18
2.9	Input/Output Waveforms from Nand Gate Simulation	19
2.10	Resulting Waveforms from Set of 9 SPICE Runs	19
2.11	Resulting Graph of Gate-Delay Vs. Output Loading	20
2.12	Current Waveforms from SPICE Simulations.	21
2.13	Schematic of Domino 2 Input OR Gate	21
2.14	Schematic of Input Delay Circuitry	22

2.15	Three Micron CMOS Layout of Domino 2 Input OR Gate	22
2.16	Illustration of Setup and Hold Time Definitions	23
2.17	Simulation Results with Setup-Time of 1.0ns	24
2.18	Simulation Results with Setup-Time of 2.5ns	25
2.19	Simulation Results with Setup-Time of 3.0ns	25
2.20	CMOS Layer Representations in Grey-Scale Patterns	26
2.21	A) Surface and B) Cross-Sectional View of a Basic NMOS Transistor	27
2.22	Layout of A) Unbiased, and B) Biased Parallel Devices	28
2.23	Inverter Layout Utilizing Parallel Devices	29
2.24	Illustration of Linear Connection of Inverter Cells	29
2.25	Doughnut Devices: A) Unbiased B) Biased Device	30
2.26	Proper Power Distribution Network within a Circuit	34
2.27	Illustration of Current Flow in Via Connection	35
2.28	Illustration of Proper Large Current Via Connection	36
2.29	Illustration of Interlaced S-Pattern Routing	36
2.30	Illustration Showing: A) Area Efficient Design Vs. B) Inefficient Design for Metal 2 Routing	37
2.31	Metal 2 Routing Layer Over 3-Bit Adder Cell	38
3.1	Clocking Strategy for Cell Operation	41
3.2	Determination of Maximum Decoder Evaluation Time, $\sigma$	42
3.3	Standard 4 by 8 Dynamic ROM Structure	43
3.4	Reorientation of ROM structure	44
3.5	Star Configuration ROM Structure	45
3.6	Illustrating The Grouping of ROM Devices into the Star Configuration	46
3.7	Illustration Showing the determination of Parasitic Drain Capacitance	47
3.8	The Complete 5-Bit STAR Configured ROM Cell Structure	49

3.9	Schematic Representation of a Full 3rd Order Binary Decision Tree	52
3.10	Layout of a Full 5th Order Binary ROM Tree	53
3.11	SPICE Simulation of The 5th Order Binary ROM Tree	53
3.12	Complete Schematic of JROM Evaluation Block	54
3.13	A) Timing Diagram and B) Clock Buffer for JROM Cell	56
3.14	Schematic Core of The Evaluation Block	56
3.15	Schematic of XLAS Cell	57
3.16	Illustration of Iso Edge Delay Value	59
3.17	Diagram Illustrating Alternative P and N-Type Latch Block Implementation	62
3.18	Schematics of P and N-type TSPC-1 Latches	63
3.19	Schematics of P and N-type TSPC-2 Latches	64
3.20	Illustration Showing How Clock Delay Causes Evaluation Disruption	64
3.21	Reverse Clock Distribution Method	65
3.22	Negative D-Latch Schematic	66
3.23	The Scaled Negative Transition D-Type Latch	66
3.24	50 Mhz Mask-Extracted D-Latch SPICE Simulation	67
3.25	Current Demand for D-Latch at 50 Mhz	68
3.26	Block Diagram of a Single Switching Tree Cell	69
3.27	A complete Block Diagram of a Switching Tree Cell	70
3.28	Switching Tree Pipelined Dynamic Logic System	70
3.29	Simulation Circuit Used From Determining Maximum Rom Height	72
3.30	Simulation Results From ROM Modelling Decks	73
3.31	Comparison of Different ROM Evaluation Approaches	73
3.32	Schematic of 'S1' Cell in 3-Bit Multiplier	75
3.33	A) First Transposition Choice, B) Resulting Interconnect Overlap	76
3.34	A) Second Choice for Transposition, B) Resulting Spatial Schematic	77

3.35	Illustration of Layout Pattern and How it is Replicated	78
3.36	A) Illustration of Starting layout Point, and B) The First Device Layout	79
3.37	A) Schematic, and B) Layout of The First Two Devices	79
3.38	A) Schematic, and B) Layout of The First Three Devices	80
3.39	Illustration of Alignment of Second Diffusion Column	80
3.40	Complete Layout of The 'S1' ROM Cell	81
4.1	Schematic Representation of a) <b>Inv1</b> Cell, b) <b>Inv3</b> Cell	85
4.2	A) Schematic Diagram and B) Layout of InBuff Cell	85
4.3	Schematic Representation of TSCP D-Latch Cell	86
4.4	Physical Layout of TSCP D-Latch Cell	87
4.5	Functional Description of 3-Bit Adder	87
4.6	Illustration of S2 Circuit Signals in 50 Mhz Simulation	88
4.7	50 Mhz SPICE Simulation of 3-Bit Adder Cell	90
4.8	Supply Rail Current Demand for 3-Bit Adder Cell @ 50 Mhz	90
4.9	Mask Layout of 3-Bit Adder Cell Showing Sub-Cell Placement	91
4.10	Mask Layout of 3-Bit Adder Cell in Northern Telecom 3 $\mu$ Process	92
4.11	3-Bit Multiplier Functionality	91
4.12	SPICE Simulation of S3 Bit of Multiplier Cell @ 25 Mhz	93
4.13	Supply Rail Current Demand for 3-Bit Multiplier Cell @ 25 Mhz	93
4.14	Mask Layout of 3-Bit Multiplier Cell in 3 $\mu$ CMOS Technology	94
4.15	4-Bit Adder Functionality	95
4.16	SPICE Simulation of 4-Bit Adder Cell @ 40 Mhz	96
4.17	SPICE Simulation of 4-Bit Adder Cell @ 30 Mhz	96
4.18	Supply Rail Current Demand for 4-Bit Adder Cell @ 40 Mhz	97
4.19	Mask Layout of 4-Bit Adder Cell in 3 $\mu$ CMOS Technology	97
4.20	Diagram Illustrating Implementation of Partial Products Algorithm	98

4.21	Complete Schematic Representation of 6-Bit Multiplier Cell	99
4.22	Physical Mask Layout of 6-Bit Multiplier Cell in 3 $\mu$ CMOS Technology	100
4.23	ECL 10K Series Specifications	101
4.24	Specifications for ECL Compatible I/O Pads	102
4.25	True ECL Compatible Output Driver Schematic	103
4.26	Final Schematic of ECL-Driver Pad	103
4.27	Mask Layout of ECL-Driver Pad	104
4.28	Mask Layout of ECL-Receiver Pad	105
4.29	Schematic Diagram of Two Stage ECL Receiver Pad	106
4.30	ECL Pads Simulation at 100 Mhz	107
4.31	The Layout of IC3WRECL	108
4.32	Illustration of Test Setup	109
4.33	Scope plots taken from ECL Compatible Board Test	110
4.34	ECL-Driver at 65 Mhz with Sinusoidal Excitation	111
A.1	Basic Definition of a Dynamic Domino Logic Gate	120
A.2	Illustration of Domino Logic Chain	121
A.3	A Circuit which Cannot Be Implemented by Domino Logic Gates	122
B.1	Schematic of Clock Cell, ClkBuff5	124
B.2	Schematic of 2 to 4 Decoder Sub-Cell	124
B.3	Schematic Representation of 3 to 8 Decoder Sub-Cell	125
B.4	Schematic of The Static Decoder Network	126
B.5	Schematic Representation of Input Buffer Cell	126
B.6	Schematic Representation of STAR Configured ROM Cell	127
B.7	Schematic Representation of Complete ROM5 Cell	128
C.1	Schematic Representation of The JClockBuff Cell	130
C.2	Schematic Representation of The JEvalBlk Cell	130



C.3	Schematic Representation of The ROM Cell	131
C.4	Schematic Representation of The JInBuff Cell	131
C.5	Schematic Representation of The JoutBuff Cell	132
C.6	Complete Schematic of The JROM Cell	132
D.1	Schematic Diagram of X > Y ROM Cell	134
D.2	Schematic Diagram of Y > X ROM Cell	135
D.3	Schematic Diagram of X Equals Y ROM Cell	135
E.1	Functionality of 5 to 3 Compressor Cell	137
E.2	Schematic of ClockBuff53 Cell	137
E.3	Schematic of OutBuff53 Cell	137
E.4	Schematic of InBuff53 Cell	138
E.5	Schematic Diagram of S0-Bit Cell	139
E.6	Schematic Diagram of S1-Bit Cell	140
E.7	Schematic Diagram of S2-Bit Cell	141
E.8	Complete Schematic Diagram of 5 to 3 Compressor Cell	141
F.1	Simplified Schematic realization of 1.2 $\mu$ CMOS Output Pad	143
F.2	Schematic Realization of 1.2 $\mu$ CMOS Input Pad	144
F.3	The I/O Signals of the 1.2 $\mu$ Cmos Pads Simulations at 100 Mhz	146
F.4	Current Demand of the 1.2 $\mu$ Cmos Output Pad simulated at 100 Mhz	146
F.5	Layout of the 1.2 $\mu$ CMOS I/O Pad Test Structure ICAWRIOT	147
F.6	Scope Plot of New DUT Board Supply Rail During Testing of WRIOT	148
F.7	Scope Plot of Output Pad Wave-form at 1 Mhz	149
F.8	Scope Plot of Output Pad Wave-form at 5 Mhz	149
F.9	Scope Plot of Output Pad Wave-form at 15 Mhz	150
F.10	Layout of ICAWRIO2	151

G.1	Schematic of ClkBuff3 Cell	154
G.2	Schematic Representation of Rom Cell S0	154
G.3	Schematic Representation of Rom Cell S1	155
G.4	Schematic Representation of Rom Cell S2	155
G.5	Schematic Representation of Rom Cell C3	156
G.6	Schematic Representation of OutBuff3 Cell	156
G.7	Schematic Representation of InBuff3 Cell	157
G.8	Complete Schematic Representation of 3-Bit Adder Cell	157
I.1	Schematic of CLkBuffM3 Cell	180
I.2	Schematic Representation of Rom Cell S0	180
I.3	Schematic Representation of Rom Cell S1	181
I.4	Schematic Representation of Rom Cell S2	182
I.5	Schematic Representation of Rom Cell S3	183
I.6	Schematic Representation of Rom Cell S4	184
I.7	Schematic Representation of Rom Cell S5	185
I.8	Schematic Representation of OutBuffM3 Cell	186
I.9	Schematic Representation of InBuffM3 Cell	186
I.10	Complete Schematic Representation of 3-Bit Multiplier Cell	187
J.1	Schematic of ClkBuff4 Cell	189
J.2	Schematic Representation of Rom Cell S0	189
J.3	Schematic Representation of Rom Cell S1	190
J.4	Schematic Representation of Rom Cell S2	191
J.5	Schematic Representation of Rom Cell S3	192
J.6	Schematic Representation of Rom Cell C4	193
J.7	Schematic Representation of OutBuff4 Cell	193
J.8	Schematic Representation of InBuff4 Cell	194
J.9	Complete Schematic Representation of 4-Bit Adder Cell	195

K.1	Distributed RLC Model for Modelling any Conductive Path	197
K.2	Lumped RLC Model for Simple Interconnect Systems	198

## LIST OF TABLES

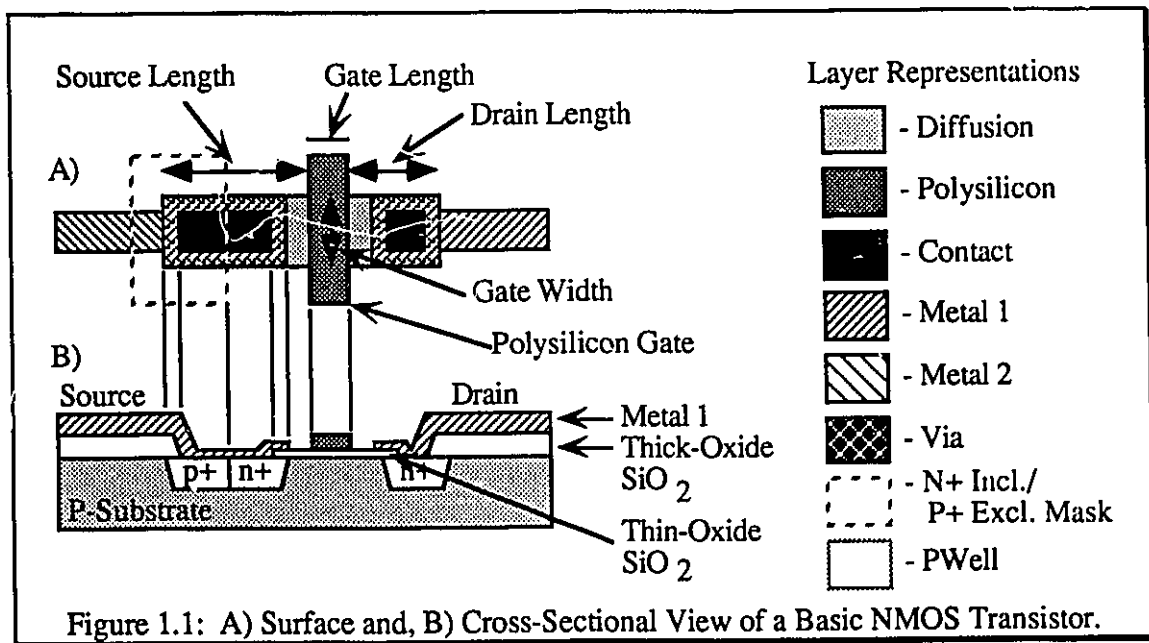
2.1	Interconnect Capacitance Values of Bell Northern 3 $\mu$ CMOS Process	12
3.1	Summary of Test Results on IC3WROMB	48
3.2	Possible Two Dimensional ROM Arrangement	51
3.3	Possible N Dimensional ROM Decomposition	51
3.4	Summary of Test Results on Test Chip IC3WRRJT	58
3.5	Summary of Test Results on Test Chip IC3WRXLT	58
3.6	Summary of Test Results on Test Chip IC3WRJET	58
F.1	Summary of Test Data from Testing of ICAWRIOT	148

# CHAPTER 1

## INTRODUCTION

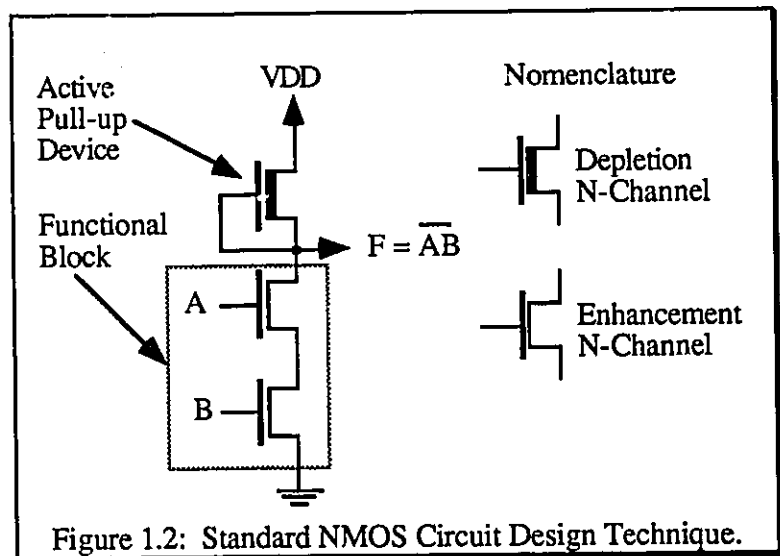
### 1.1 INTRODUCTION

Over the past decade or so, semiconductor research has made significant advancements in the development of smaller and faster technologies. The development of Metal Oxide Semiconductor (MOS) technology, particularly the n-channel or NMOS technology, has revolutionized the semiconductor industry and spawned a new era with the advent of the micro-computer chip. The main building block of NMOS is the n-channel field effect transistor (NFET) device. The basic structure of the NFET is illustrated in Figure 1.1.



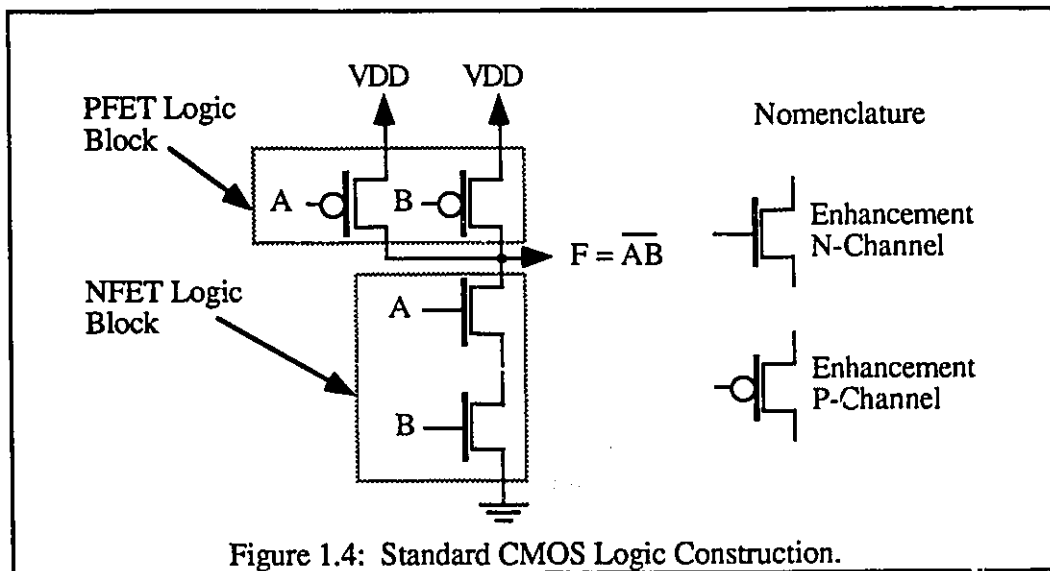
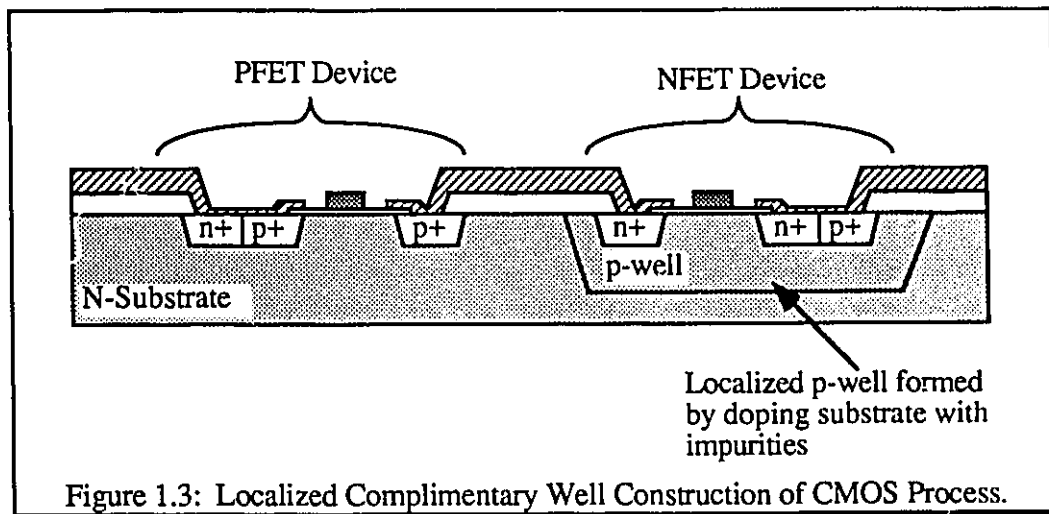
NFET devices can be classed into two types; enhancement(E)-mode and depletion(D)-mode. Both devices have a conducting channel which varies its conductance with applied gate voltage; the depletion-mode device has a conducting channel present with zero gate to source potential, the enhancement-mode requires a threshold voltage to be applied before a channel is present. In NMOS technology, the depletion-mode device is used as a load. The presence of a conducting channel at all times, guarantees that the output of the gate will reach the supply voltage for a logic '1' output [1]. Figure 1.2 shows the standard NMOS circuit design technique which utilizes the depletion-mode active pull-up device.

Using this NMOS technology, it is possible to pack thousands of NFET transistors onto a single chip and perform complicated logic functions. The complexity of the logic functions is limited due to the large static power



dissipation associated with large numbers of NMOS logic circuits; this limits the number of devices that can be adequately fabricated on a single chip using this technology. An alternate technology, CMOS (Complimentary Metal Oxide Semiconductor) uses, for example, p-type wells as 'surrogate' substrates diffused into an n-type substrate, allowing for the construction of both n-channel and p-channel devices on a single silicon substrate, as illustrated in Figure 1.3. Having both types of transistor allows the design of a logic family, commonly referred to as standard CMOS logic, which dissipates almost no static

power. Standard CMOS logic is constructed out of two logic blocks, one built from NFET devices and the other from PFET devices, as shown in Figure 1.4. The two blocks perform complementary logic switch functions, and hence, for any given input, only one block will provide a path between the output node and one of the power supply rails. This reduced power dissipation and also improvements in state-of-the-art denser technologies allowed devices to be packed closer together and increased the level of functional complexity that could be placed on a single chip. Standard CMOS logic construction, however, has several disadvantages.

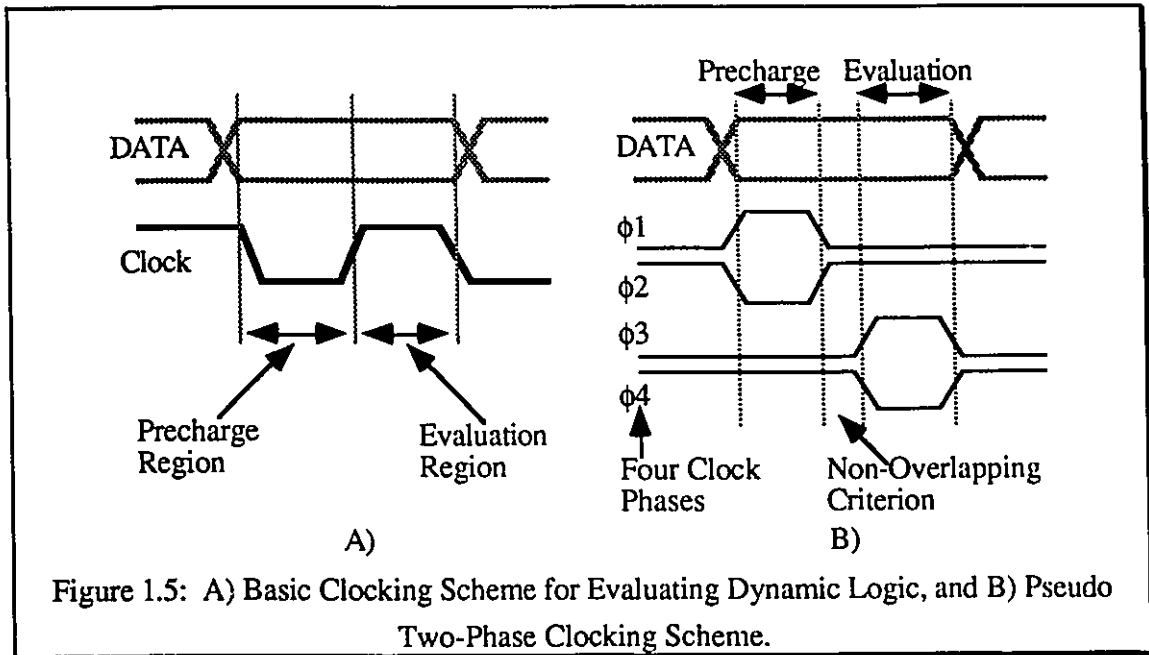


The mobility of holes, the majority carriers in the conducting channels of p-channel devices, is about 1/3 that of electrons, the carrier in n-channel devices. In order to provide channels with approximately the same conductance in both the n-channel and p-channel devices, the widths of the PFETs have to be increased. This increases the size of each logic gate and also increases the dynamic switching current due to the increased parasitic capacitance associated with the larger devices. It is possible to replace the PFET block with a single p-channel transistor, if special dynamic logic circuits are used. These logic circuits use temporary memory (stored charge on parasitic node capacitances) to replace the pull-up PFET logic block, thus reducing the size; the speed of dynamic logic circuits is also usually higher than that for static logic circuits because of the reduced load capacitance.

Dynamic logic families require the use of clocking schemes to first precharge the logic block and then allow it to evaluate, as shown in Fig. 1.5(a). A popular clocking technique is the pseudo two-phase clocking scheme which requires the routing of four non-overlapping clock signals, as illustrated in Fig. 1.5(b). This technique suffers from clock skew problems brought about by process deviations across the surface of large chips. Clock skew is associated with the delay experienced by clock signals as they are distributed along metal conductors across large chips. Since clocks are meant to synchronize logic circuits, synchronizing delays between one part of the circuit and another can often cause soft failures; this is particularly critical if several clocks are required for the synchronizing process (often a requirement in dynamic logic circuits). As chip areas increase, it is necessary to decrease the maximum clock rate so that the skew delays can be kept within timing tolerances. Dynamic circuits that use single phase clocks are clearly to be preferred. DOMINO [2][3][4] logic and NORA [5][6] logic are two dynamic logic families that operate with reduced numbers of clock lines. Recent results have produced, in dynamic logic, the equivalent of the edge triggered latch in static logic [7]. Thus dynamic circuits can



be synchronized with single phase clocks, reducing the clock skew problem to manageable proportions.



NORA logic utilizes a true two-phase clocking scheme; however, the logic suffers from noise margin degradation due to the lack of signal buffering. It is also very susceptible to capacitively coupled noise injection due to floating nodes found in the circuit path, as is illustrated in Fig. 1.6(a). The logic can be used to form a natural pipeline, since stages alternate precharge and evaluate, and so has been seriously considered for digital signal processing circuits, where throughput is more important than computational time.

Domino logic, as illustrated in Fig 1.6(b), utilizes a single-phase clock; every dynamic evaluation node, which is buffered with an inverter to prevent race conditions, maintains good noise margins. Domino logic cannot perform logic inversion, therefore complementary function Domino circuits are often required, which place large penalties on the overall area requirements. Domino logic forms a dynamic combinational chain that requires the placement of a static latch, at the end of the chain, to capture the output result.

Since inversion can easily be performed along with the static latch function, we can sometimes design the Domino chain to not require inversion; rather we perform it at the beginning, or wait until the end. This does not always work, however.

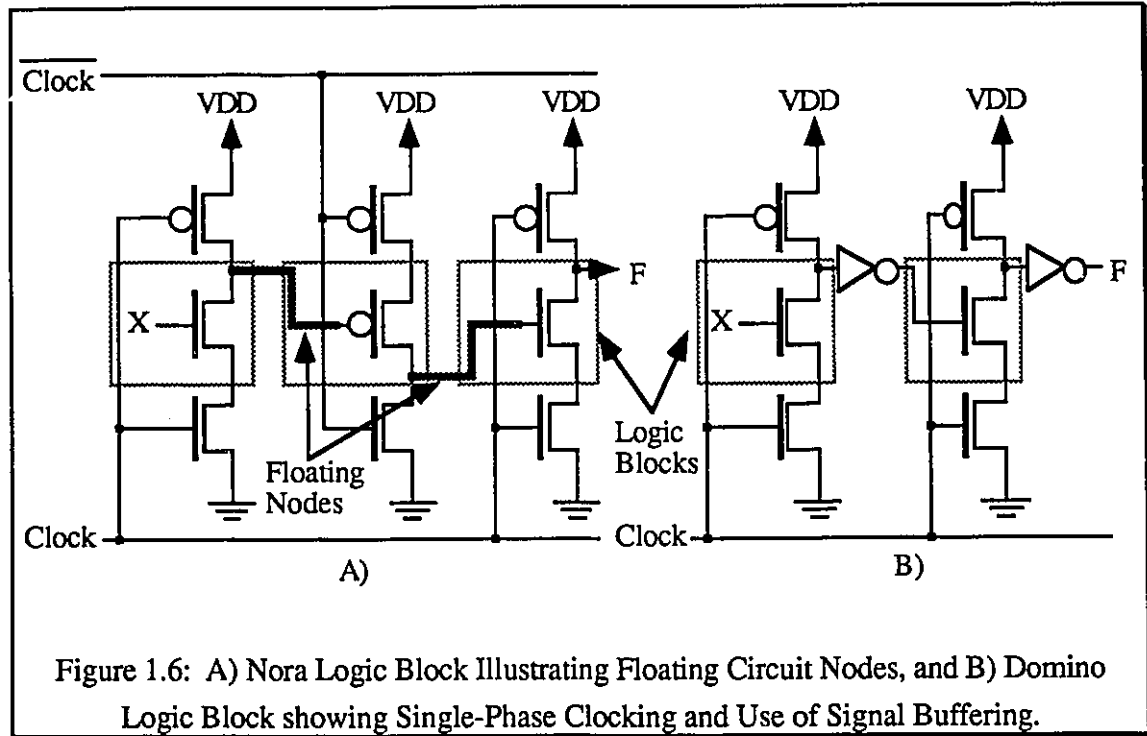


Figure 1.6: A) Nora Logic Block Illustrating Floating Circuit Nodes, and B) Domino Logic Block showing Single-Phase Clocking and Use of Signal Buffering.

One thing that all dynamic logic families have in common is that they implement a logic function through the use of a single logic block per stage, usually in the form of NFET devices which are arranged to implement a truth-table and can be thought of as a ROM (Read Only Memory) block, since the functionality is fixed at fabrication. This concept has led to much research and development in the area of ROM and programmable ROM structures. All ROM structures possess an area of devices which defines the programming of the ROM by the placement and absence of devices. The general shape of the ROM is dictated by the input signal decoding method which places conformal restrictions on the ROM's construction. Some ROM structures utilize a decoder network which produces two fields of output corresponding to cross-point information. This type of decoder network requires the ROM to be in the form of a two dimensional structure and

programming is done by placement of devices at the intersection of row and column signals. Though two dimensional decoding is the most obvious and popular method of decoding ROM structures it is possible to create multi-dimensional ROM structures. As the dimensionality of the ROM structure increases, the complexity of the decoder network decreases. Thus if the ROM dimensionality is taken to the extreme, meaning the ROM is addressed as an 'n' dimensional structure, where 'n' is defined as the number of input bits addressable by the ROM. The resulting ROM structure is known as a Binary-Tree or Switching-Tree illustrated in Fig. 1.7, and the decoder network resolves down to 'n' inverters.

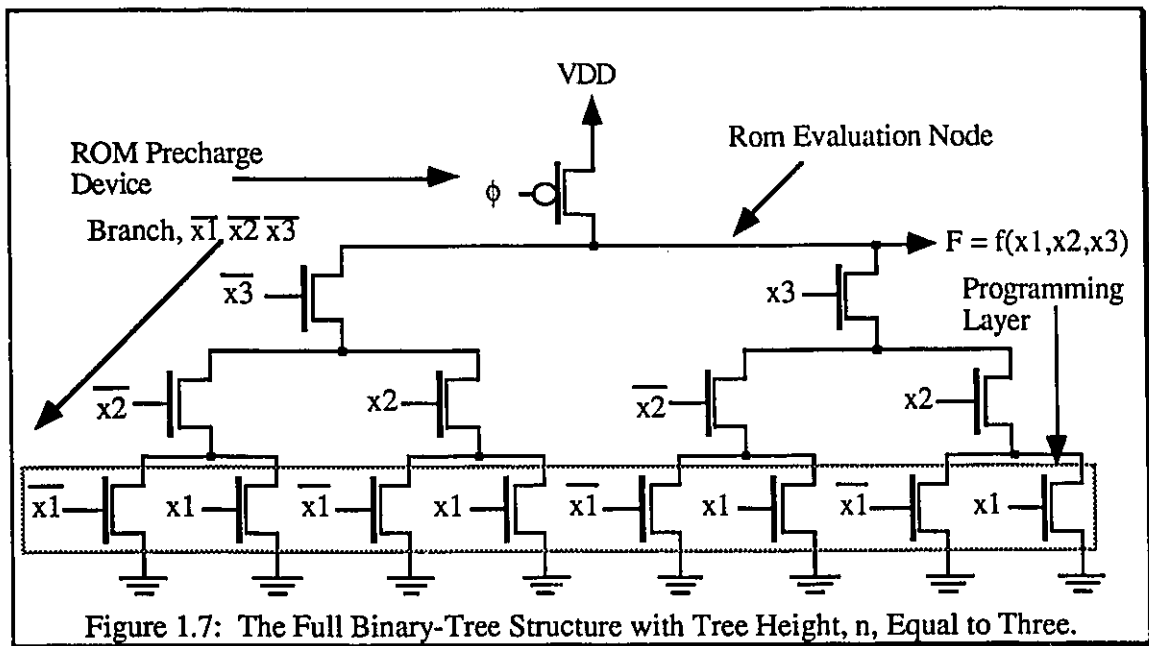


Figure 1.7: The Full Binary-Tree Structure with Tree Height, n, Equal to Three.

The complexity of the logic function which can be implemented using Switching-Trees is directly related to the height of the Switching-Tree, 'n'. In looking at the full Switching-Tree, Fig. 1.7, programming is done by leaving or removing the last device in each branch of the tree, which effectively creates or removes a path to ground for that particular branch. Once certain devices are removed this makes other devices in the tree no longer necessary or redundant. Thus the ROM requires to be minimized which removes

the redundant devices and increases the packing density of the ROM. This procedure though upsets the capacitive balance of the ROM's internal nodes, which can lead to charge sharing and charge redistribution problems. These issues need to be resolved in order to achieve a suitable and reliable logic block. Thus Switching-Trees are dynamic logic blocks which implement arithmetic functions which can be expressed in truth-table, or minterm fashion. Switching-Trees attempt to increase the functional density of logic circuit techniques as well as increasing operating speeds of standard CMOS technologies.

## **1.2 THESIS ORGANIZATION**

This thesis deals with the physical development of digital switching tree structures, which were implemented using dynamic logic blocks utilizing true single phase clocking techniques as well as true single-phase latches developed by Yaun & Svensson [7]. The body of this work represents the authors contribution in the team effort look into the automated design, of on the fly development, of standard switching tree cells for the implementation of random truth table, or minterm description of digital logic. This work consists of generating a library consisting of dynamic logic blocks designed using heuristic layout techniques for constructing the ROM portion of the switching tree's cells. The cells were generated by software known as 'WOODCHUCK' developed by Roger Grondin[8]. As part of the fixed cells in the library, I/O(Input/Output) pad drivers were developed to handle the increased speeds of the switching-tree structures.

**Chapter II** deals with practical design and layout considerations attained through years of layout experience. The chapter covers methods of improving pre-layout SPICE simulations, different device layout techniques, as well as summary of helpful hints.

**Chapter III** will document the development of digital switching trees, tracing through the three iterations of ROM cells. A close examination of dynamic logic as well as clocking techniques for fully pipelined bit serial architectures will be covered.

Determination of ROM-tree limitations with respect to tree heights and evaluation speed will be discussed. The evaluation of ROM-tree efficiency will also be examined using a speed, area, and power product for comparison purposes, and a heuristic rule set for performing personalized layouts of ROM-tree structures will be given.

**Chapter IV** presents a set of cells developed in the Bell Northern 3 $\mu$  CMOS technology which represents the switching tree standard cell library. The standard cell designs were developed as proof of concept designs. Included in the library is a set of ECL compatible I/O pads developed to handle the increased speed of the switching tree cells.

**Chapter V** offers a summary and conclusions of the work done as well as outlining future work direction.

# CHAPTER 2

---

## THE ART OF DESIGN

### 2.1 INTRODUCTION

As the title suggests, designing and laying out integrated circuits is as much an art as it is a science. How the designer chooses to assemble a circuit can dramatically influence the circuit's performance. Often the success or failure of a novel architecture may very well rest on the ability of the designer to fully utilize the technology which is being used for implementation. Thus this chapter will be devoted to discussing practical design and layout considerations attained through experience. It is the author's intention to impart some practical knowledge of VLSI design to new designers.

This chapter will be divided into two main sections. The first section will discuss dealing with circuit capacitance and capacitive loading effects, which are major issues in the design of high speed circuitry. The discussion will examine the origins of parasitic device capacitance and interconnect capacitance. Methods for estimating, and improving initial SPICE circuit simulations will be discussed through the use of examples. The second section of this chapter will deal with developing practical design experience. Very often the novice designer may develop a design which may meet all the design specifications but, due to poor judgement or lack of experience, the designer may construct the design in such a manner as to severely limit the design's capabilities.

## 2.2 DEALING WITH CIRCUIT CAPACITANCE

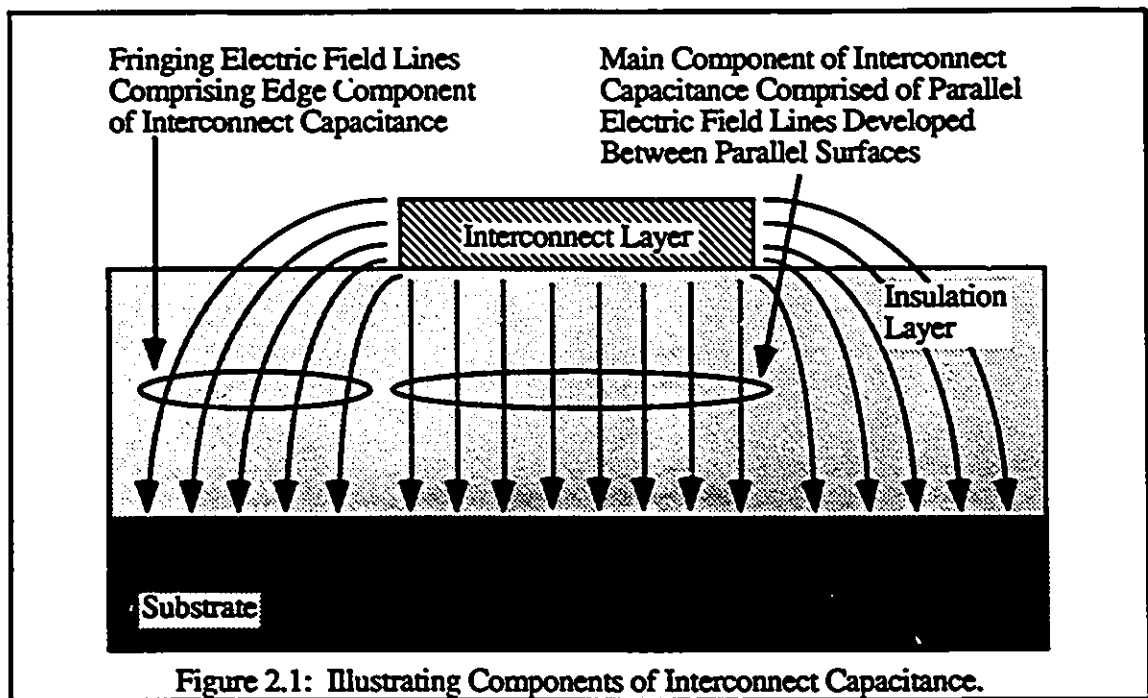
There are two main sources of circuit capacitance in CMOS circuits. The first source is known as parasitic device capacitance which is, most often, unwanted capacitance associated with the physical construction of a MOS device. Parasitic device capacitance exists between the three terminals of the MOSFET device and the bulk substrate connection. The local substrate or well of a MOSFET device lies directly under the device and comprises one combined plate of three parallel plate capacitors. The other plates are the drain, source, and gate region of the MOSFET device. For an in-depth discussion of the device physics with respect to capacitance refer to reference [9].

The second form of circuit capacitance exists between the interconnect layers of the CMOS technology and the substrate. As mentioned previously, all circuit components in a CMOS technology are designed on a silicon substrate, with layers of insulation between each layer. Thus all layers in the CMOS process will exhibit capacitance to the substrate; the magnitude is dependent on the position of that layer with respect to the substrate. This is why polysilicon, which is closer to the substrate than the metal interconnect layers, exhibits the highest capacitance per square. This value is even greater for the gate area where the polysilicon is separated from the substrate by only thin-oxide. The greater the thickness of total oxide between the layer and the substrate the lower the capacitance. Thus metal two, which is the top layer, has less per square capacitance than metal one, and metal one has less per square capacitance than polysilicon.

The capacitance associated with the interconnect layers is comprised of two components. The first component is the value of capacitance which exists between the two surfaces that comprise a parallel plate capacitor, neglecting any fringing effects of the electric field. This value of capacitance is defined in terms of area, and is derived from the microstrip-line capacitor model[10]. The second component which makes up the

interconnect capacitance is the edge component, which is due to the fringing effect of the electric field at the edges of the conductor. The edge component arises from the fact that the conductor has a relatively large thickness, which means that a vertical surface exists on the conductor which will produce arched electric field lines to the flat substrate surface below. The edge component is defined in terms of units of edge length. In comparing the relative lengths of the electric field lines, illustrated in Fig. 2.1, one can see that the edge component value will be smaller in size. The values of capacitance for the interconnect layers in the Northern Telecom 3 $\mu$  CMOS process are listed in Table 2.1, along with values for a 100 $\mu$  run of minimum width interconnect [11]. These values can be used to estimate interconnect capacitance of circuitry.

Layer to Field	Capacitance (in ff/ $\mu^2$ )	Edge Component (in ff/ $\mu$ )	Total Capacitance (in ff/100 $\mu$ )
Polysilicon	6.0e-2	5.0e-2	28.0
Metal 1	2.7e-2	4.0e-2	16.1
Metal 2	1.4e-2	2.0e-2	8.2





## 2.2.1 CIRCUIT MODELLING FOR PROPER DESIGN

In order to create a preliminary SPICE simulation of new circuitry which will accurately reflect the performance of the design once it has been laid out, great care must be taken to provide the simulation package with the appropriate circuit models. All SPICE device models require a model-card which lists important device parameters. Often the model is over specified in that some of the parameters can be calculated from a subset of the others. A choice is available to substitute the calculation with a single parameter value. Many of the parameters have default values; since errors are not generated if parameters are not provided (defaults substituted) it is important to examine the model parameters carefully if the defaults are not acceptable. In addition to the device models, device sizes are required. Since accurate values of perimeter and area of the source and drain nodes are required for each device, these values are most often generated from mask extractions; for preliminary designs good estimates must be obtained. Reasonable estimates of interconnect capacitance should also be added to preliminary simulation descriptions (accurate values are usually obtained from a good extraction utility for final simulations). Designers must therefore have an idea for the method in which each device will be laid out prior to constructing the SPICE-deck, and section 2.3 will discuss different transistor arrangements and how to determine their relative geometric measurements.

The first step to producing an accurate simulation description is to carefully parameterize the circuitry driving the circuit under consideration. It is essential to provide input signals with limited driving capability which accurately reflects the environment in

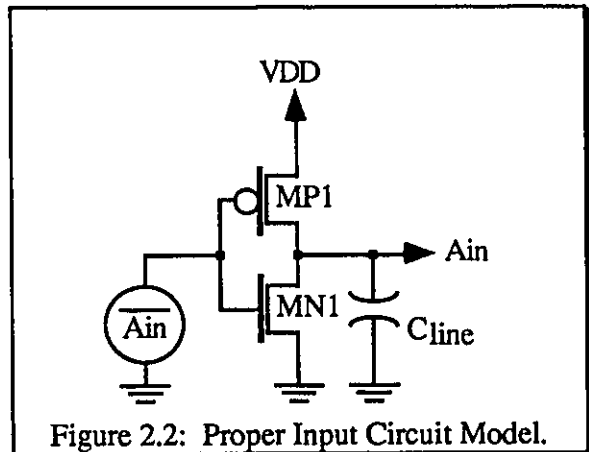
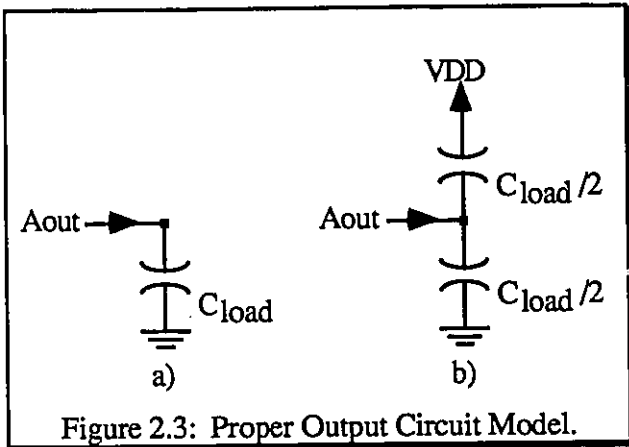


Figure 2.2: Proper Input Circuit Model.

which the circuit will have to function. Proper placement of all input signals in time, with relationship to control signals arriving at the cell boundary, is important for guaranteeing realistic time dependent behavior. This can be done very simply by using an inverter, a capacitor, and an ideal source to accurately model the input circuit behavior, as illustrated in Fig. 2.2. The sizes of MN1 and MP1 can be taken from the last stage of the preceding input circuitry, if it is known, or a best guess will do. The input loading capacitance is critical as it dictates the general shape and relative signal delay of the input signal. The time placement of each input signal can be adjusted in the ideal input source signal description.

Accurate estimates of the loading effect of the driven circuit, at the output of the circuit under simulation, have to be provided for the initial simulation. The load takes the form of lumped capacitance representing the interconnect and gate capacitance of the following stage, as



in Fig. 2.3. The value of the output capacitance can be varied through a number of SPICE simulations and the resulting data can be compiled to determine the performance of the circuit with respect to variations in output loading. An example of this will be carried out in the following section. Note that if SPICE has trouble converging in the initial DC analysis, or develops TIME-STEP-TOO-SMALL errors, then splitting the single load capacitor, as indicated in Fig. 2.3b, often corrects the problem. If this does not help, increasing the size of the capacitors by a factor of 2 or more can help. This can also be performed on internal circuit nodes which are highly isolated and have only a single capacitance value between that node and a power rail. In these cases, during high-impedance conditions, the SPICE

software may not converge in the initial DC analysis due to mathematical limitations associated with the finite precision representation of data. By replacing the capacitor by an equivalent voltage divider structure and slightly increasing its value, can sometimes eliminate the convergence problem without totally upsetting the circuits behavior.

## 2.2.2 PARAMETERING A CIRCUIT BLOCK

To simulate large circuit designs comprised of thousands of devices, which would create too many variables for SPICE type packages, will require the use of a Digital Simulation package. Digital simulation packages require well parameterized models of the small building block or sub-circuits, in order to accurately perform an accurate timing analysis of the complete circuit. In this section we will take two standard cells as examples and fully parameterize them, listing all necessary information required for digital circuit simulation.

The first circuit to be considered will be a simple NAND gate. The circuit diagram for the standard cell is shown in Figure 2.4 and the layout of the cell is shown in Figure 2.5. Looking at the circuit diagram it is obvious that the only path to ground for the output node is through two series NMOS devices, while two paths to VDD exist through two single PMOS devices. Since the current

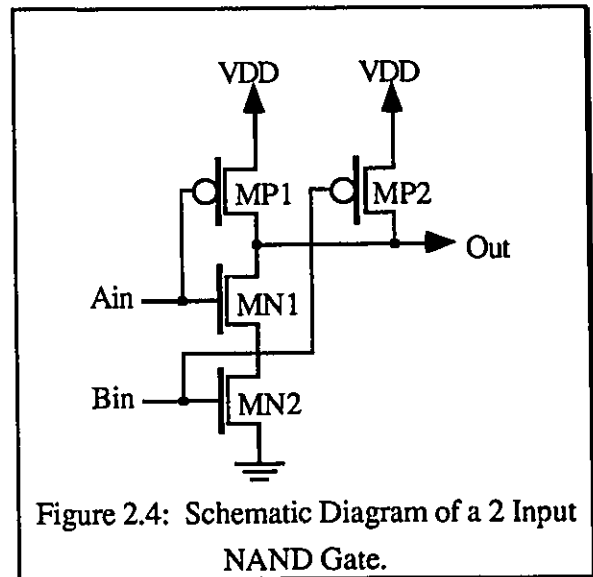


Figure 2.4: Schematic Diagram of a 2 Input NAND Gate.

mobility is approximately three times greater for the NFETs than the PFETs, the normal scaling of device widths will be three to one in order to achieve equal pull-up and pull-down times (switching times). But since the NMOS devices are in series, it is necessary to



have the same delay, it will suffice to have the cell switch from a high to low output and then back. Ain and Bin signals will be defined to perform this input switch.

```

*****
* n-channel enhancement mosfets extracted from*
*mask layout follow.*
*****

mn1 out ain 3 0 ntran w= 0.54e-05 l= 0.3e-05
+ad= 0.3888e-10 as= 0.81e-11 ps= 0.84e-05 pd= 0.252e-04
+nrd= 0.13333e+01 nrs= 0.27778

mn2 0 bin 3 0 ntran w= 0.54e-05 l= 0.3e-05
+ad= 0.3564e-10 as= 0.81e-11 ps= 0.84e-05 pd= 0.24e-04
+nrd= 0.12222e+01 nrs= 0.27778

*****
* p-channel enhancement mosfets extracted from*
*mask layout follow.*
*****

mp3 vdd ain out vdd ptran w= 0.162e-04 l= 0.3e-05
+ad= 0.10692e-09 as= 0.729e-10 ps= 0.252e-04 pd= 0.456e-04
+nrd= 0.40741 nrs= 0.27778

mp4 vdd bin out vdd ptran w= 0.162e-04 l= 0.3e-05
+ad= 0.10692e-09 as= 0.729e-10 ps= 0.252e-04 pd= 0.456e-04
+nrd= 0.40741 nrs= 0.27778

*****
* parasitic node capacitors follow*
*****

cn1 ain vdd 0.83956e-14
cn2 0 vdd 0.12475e-13
cn3 out vdd 0.18296e-14
cn4 bin vdd 0.70756e-14

cp1 ain 0 0.1644e-14
cp2 0 out 0.22552e-14
cp3 0 bin 0.1644e-14

*****
* parasitic coupling capacitors follow*
*****

cc1 ain out 0.162e-14

```

Figure 2.6: Mask-Extracted SPICE-Deck of 2 Input NAND Gate.

A set of nine SPICE runs will now be performed varying the load capacitance,  $C_{load}$ , from 50 to 500ff in 50ff increments. A set of input and output waveforms are illustrated in Fig. 2.9 showing the cell's functionality. A close-up view of the entire set of outputs for the nine runs is shown in Fig. 2.10 illustrating the variation in the gate-delay as a function of load capacitance. The gate delay for the cell is defined here by measuring the time between the 50% edge points of the input and output signals. Taking the acquired data and using a polynomial curve fitting program, as illustrated in Fig. 2.11, it is possible to produce an equation for the gate delay of the circuit with respect to load capacitance. This equation can then be used in a gate level simulator to perform switch level simulations of an entire chip composed of parameterized sub-cells. The resulting gate delay is given by eqn. (2.0).

$$\text{Gate-Delay} = 0.94727 + 9.3196e-4 * C_{load} + 2.3683e-5 * C_{load}^2 - 2.6154e-8 * C_{load}^3 \quad (2.0)$$

```

*****
* Input Circuitry *
*****
vain  ainb  0  pulse(0.05 4.95 1ns 3ns 3ns 7ns 20ns)
vbin  binb  0  pulse(0.05 4.95 1ns 3ns 3ns 17ns 40ns)

mn1001  ain  ainb  0  0  ntran l=3u w= 5.4u pd=19.8u ad=38.88p
mp1002  ain  ainb  vdd vdd ptran l=3u w=16.2u pd=30.6u ad=116.6p

mn1003  bin  binb  0  0  ntran l=3u w= 5.4u pd=19.8u ad=38.88p
mp1004  bin  binb  vdd vdd ptran l=3u w=16.2u pd=30.6u ad=116.6p

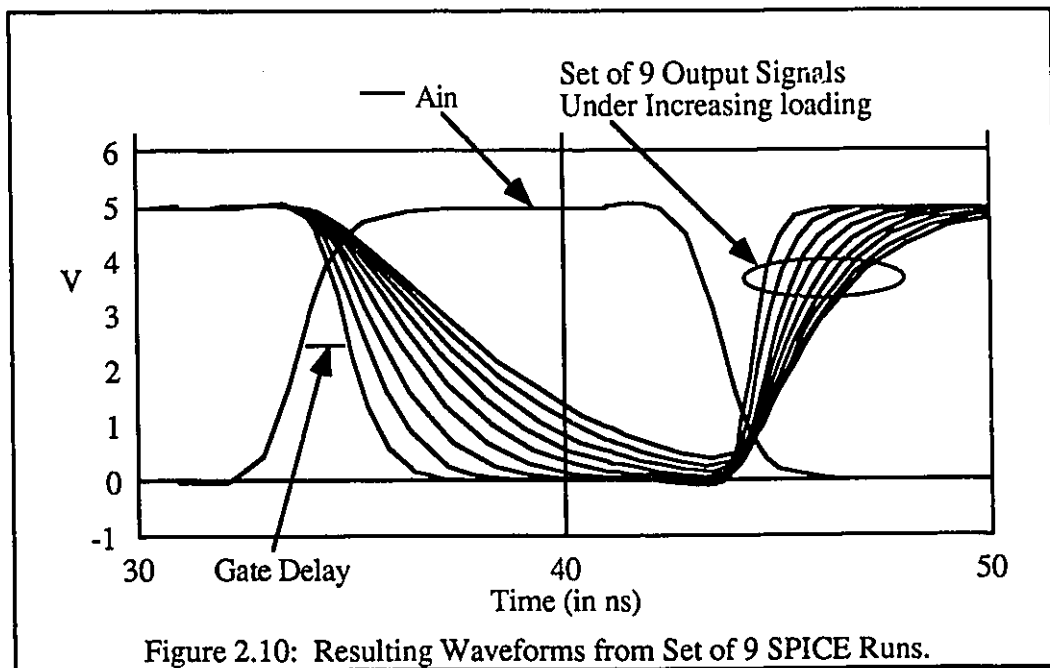
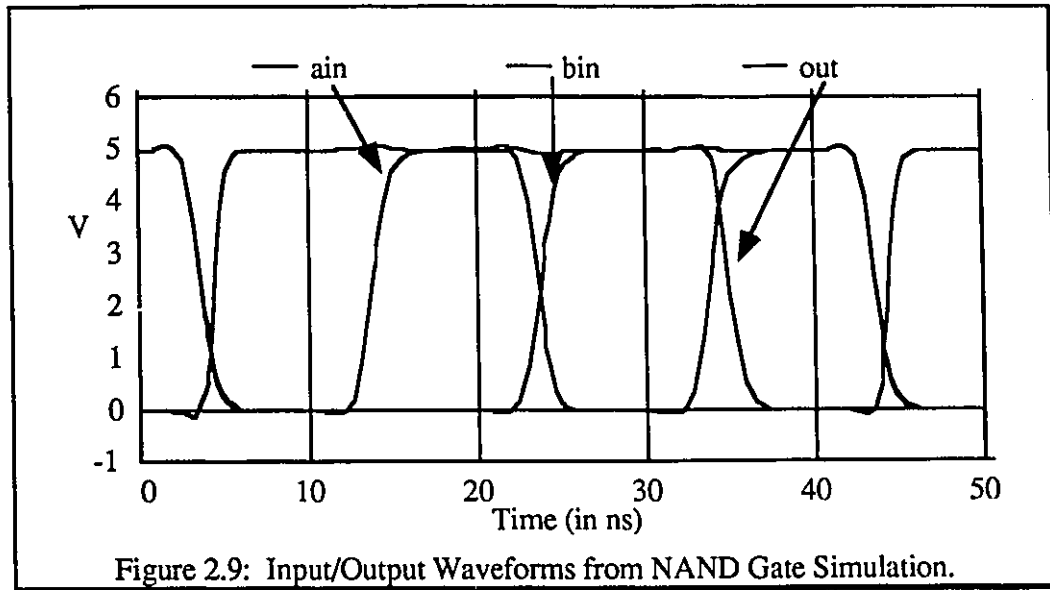
cain  ain  0  100ff
cbin  bin  0  100ff

*****
* Output Load *
*****
cload  out  0  100ff

*****

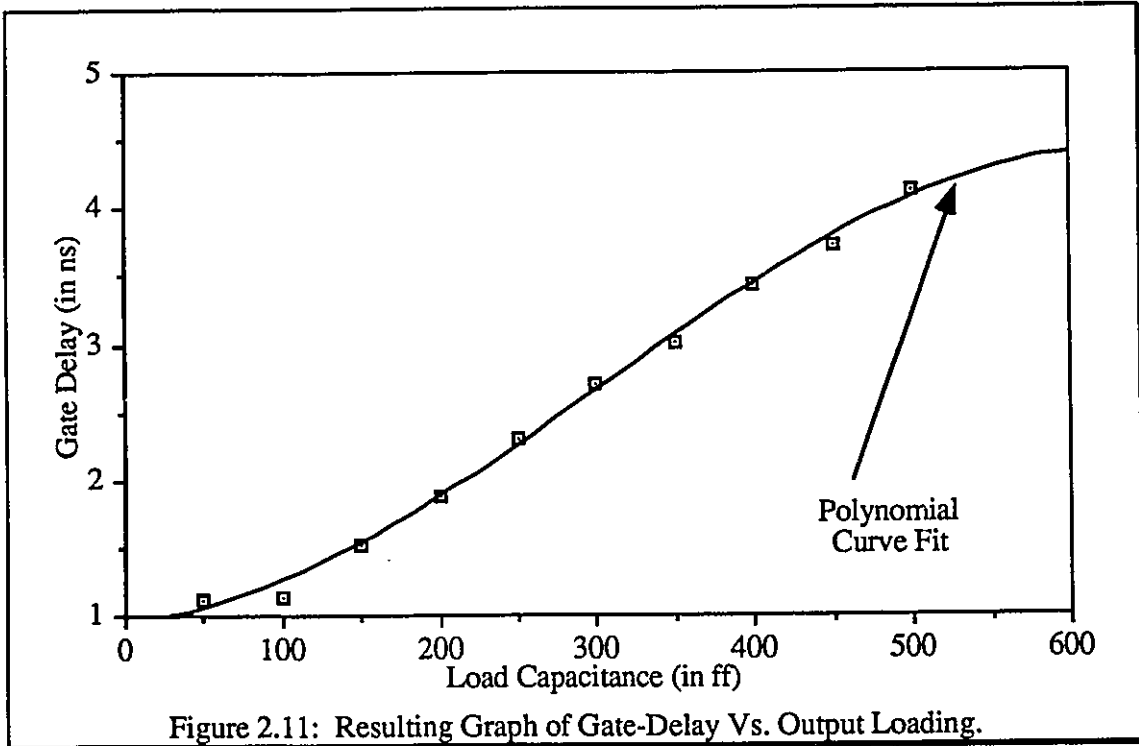
```

Figure 2.8: Additional Devices Added to the Original SPICE-Deck.



The power consumption of complex circuitry can also be determined through similar SPICE simulations, where both speed and load conditions need to be varied. For this static CMOS logic circuit the power consumption will be linearly dependant on the output load capacitance times the frequency to which that load capacitance is charged and discharged [12]. Thus running a large set of SPICE simulations would be a waste of time.

Instead if the switching current of the circuit is examined for two known load capacitances, then an equation can be determined to predict the power consumption of the circuit under



small variations in load conditions. In order to do this properly, the circuit under analysis must have its supply source isolated from the rest of the circuitry. The current waveforms in Figure 2.12 represent the supply current demanded by the NAND gate when driving a 100ff and 200ff load. By determining the energy drawn from the supply during switching, which is the area under the current 'spike' multiplied by the voltage of the power rail, then an equation can be determined for the power consumption as a function of load capacitance. The two values for power dissipation were determined to be 5.5 and 8.5mW/cycle. The equation for the power dissipation for the static CMOS NAND gate will take the form:

$$P_{diss} = (p_0 + p_1 * C_{load}/100) * freq_{out} \text{ mW}$$

The determined power dissipation equation is given in eqn. (2.1).

$$P_{diss} = (2.5e^{-12} + 3.0e^{-12} * C_{load}/100ff) * freq_{out} \text{ mW} \quad (2.1)$$



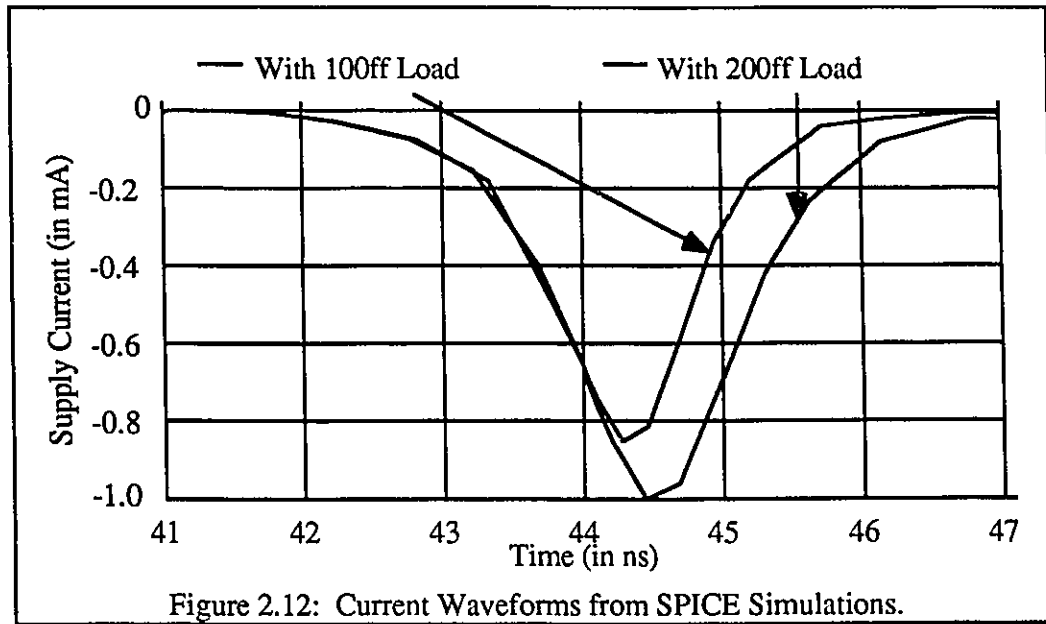


Figure 2.12: Current Waveforms from SPICE Simulations.

The next issue that will be examined is that of determining proper signal placement for cells that require clock synchronization. The cell that will be examined will be composed of a Domino

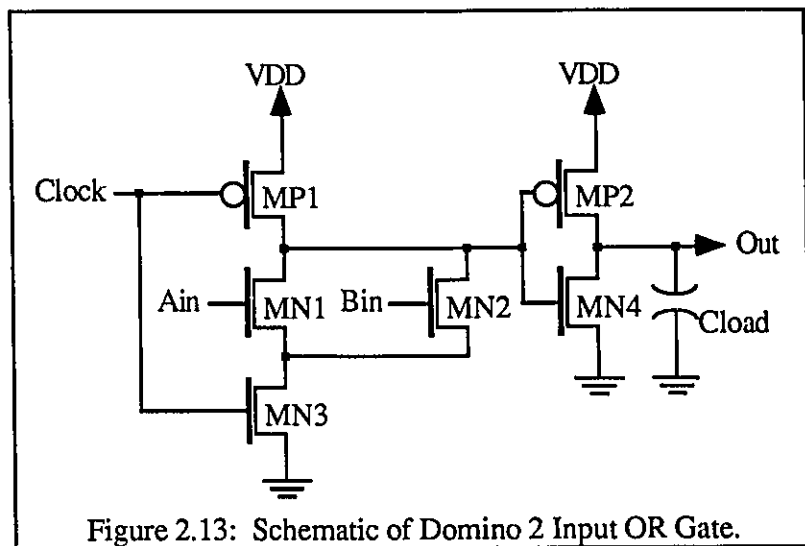


Figure 2.13: Schematic of Domino 2 Input OR Gate.

style 2 input OR gate, illustrated in Figure 2.13, and two inverters. See Appendix A for a summary of the domino logic family. For the purpose of this discussion, the cell in question will represent a complex logic block that has some static logic connected to the inputs, possibly comprising a decoder network with outputs having different critical path delays. To accomplish this the inverters will be placed in series with the 'Ain' input as illustrated in Figure 2.14. The other portion of the cell represents a functional block, namely the 2 input OR gate, which has clocking signals associated with it.

This portion of the cell may represent a complicated function designed in a non-race logic family like Domino Logic, or it may be a RAM or ROM based lookup table. The following discussion will explain the need

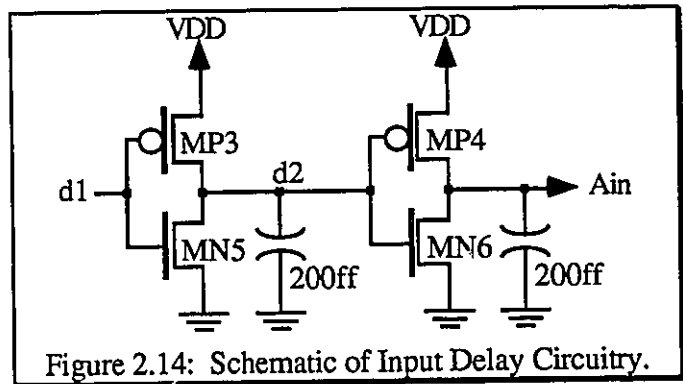


Figure 2.14: Schematic of Input Delay Circuitry.

for determining proper setup and hold times as well as how to determine maximum clock speed. The layout of the Domino 2 input OR gate is shown in Figure 2.15.

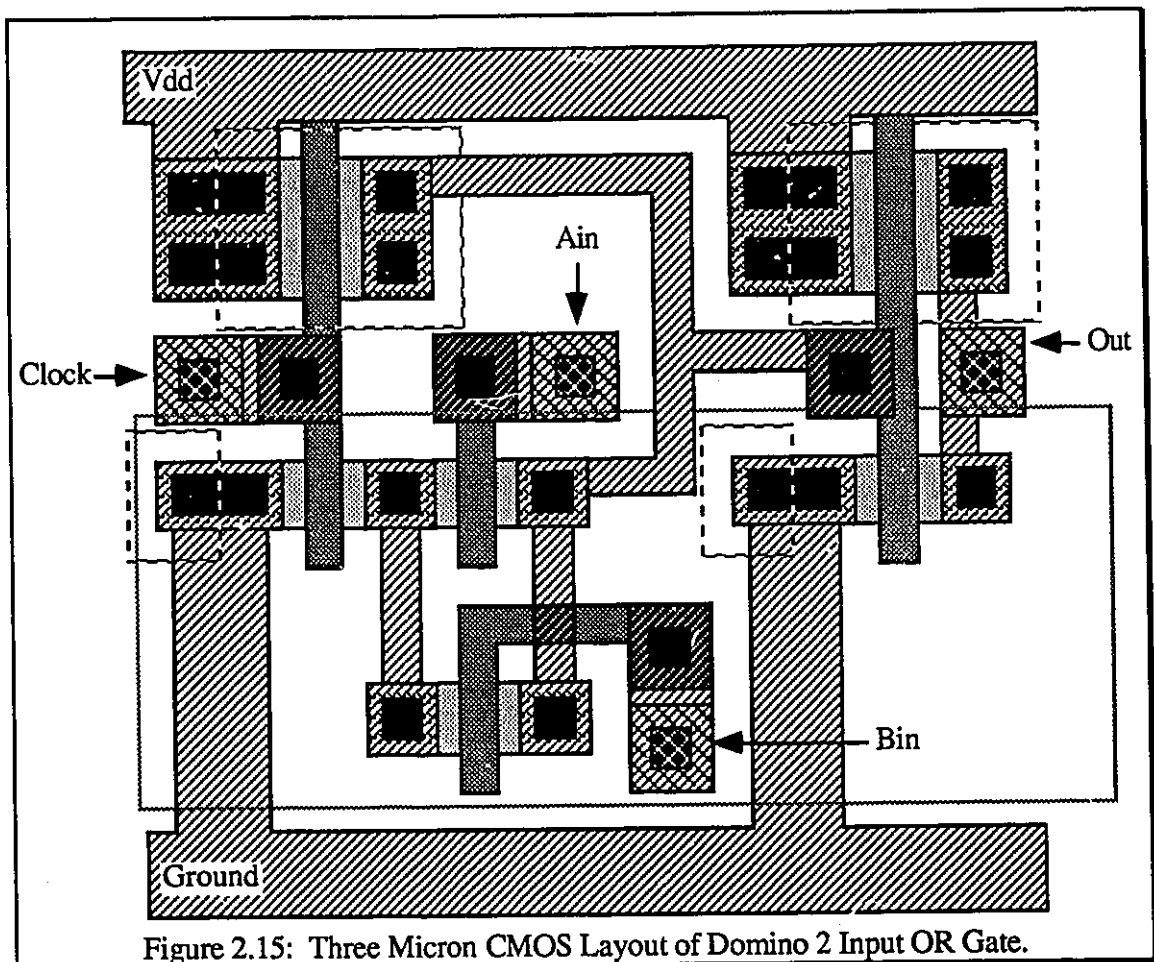
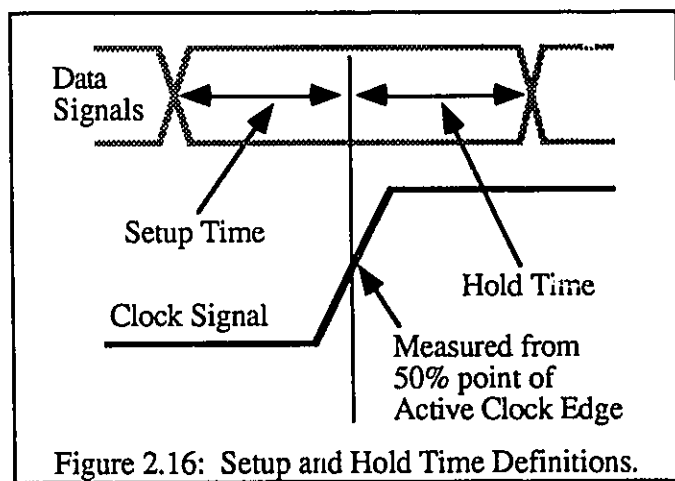


Figure 2.15: Three Micron CMOS Layout of Domino 2 Input OR Gate.

The term setup-time implies the time that the input signals must be stable before clocking of the cell can take place in order to guarantee correct results. The term hold-time implies the time that the input signal must be maintained after the clocking edge to insure correct functionality. These concepts are illustrated in Figure 2.16. Both of these terms are defined as minimum values since larger setup and hold times will always ensure functionality provided the data edges do not fall outside the window of the clock cycle. The setup and hold times for a circuit can be found by performing iterative SPICE runs and decreasing one of the values while holding the other constant. In order to do this, a complete SPICE-deck must be created which contains all input and output modelling circuitry. After experimenting with the signal edge placements with respect to the clock edge, to ensure functional simulations results, the iterative process can begin.

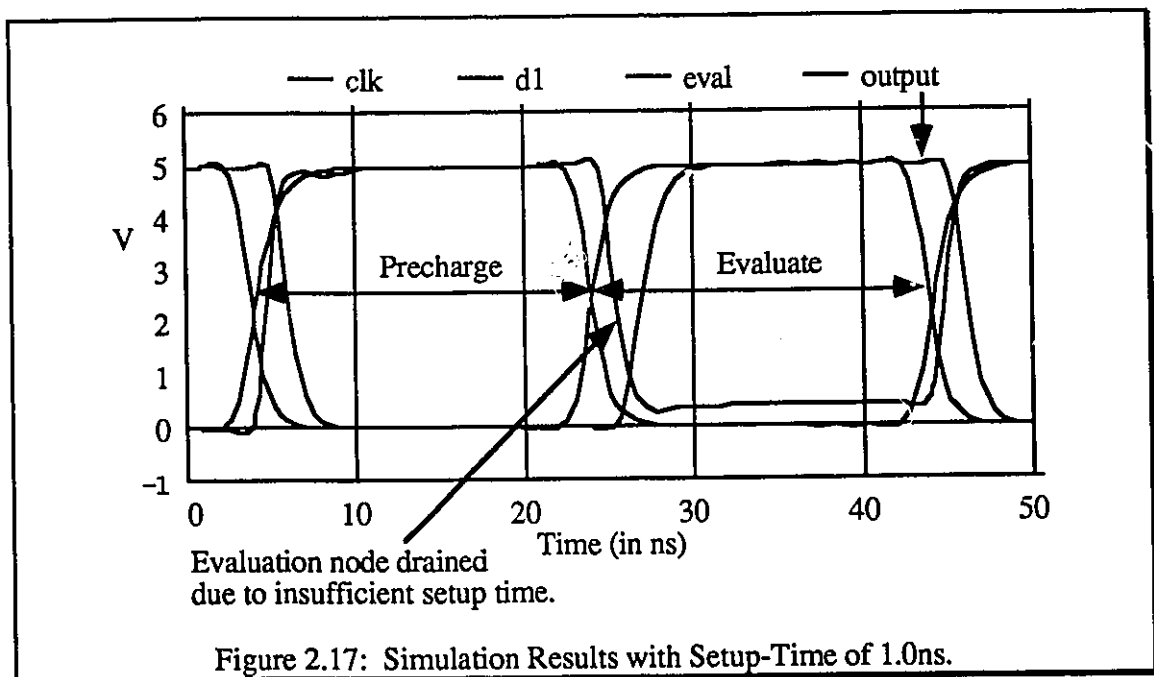
Since the cell being examined has only a single Domino logic block, there is no need to determine input signal hold time, as that is fixed to the width of the evaluation clock pulse as defined by the Domino logic family. So the only values

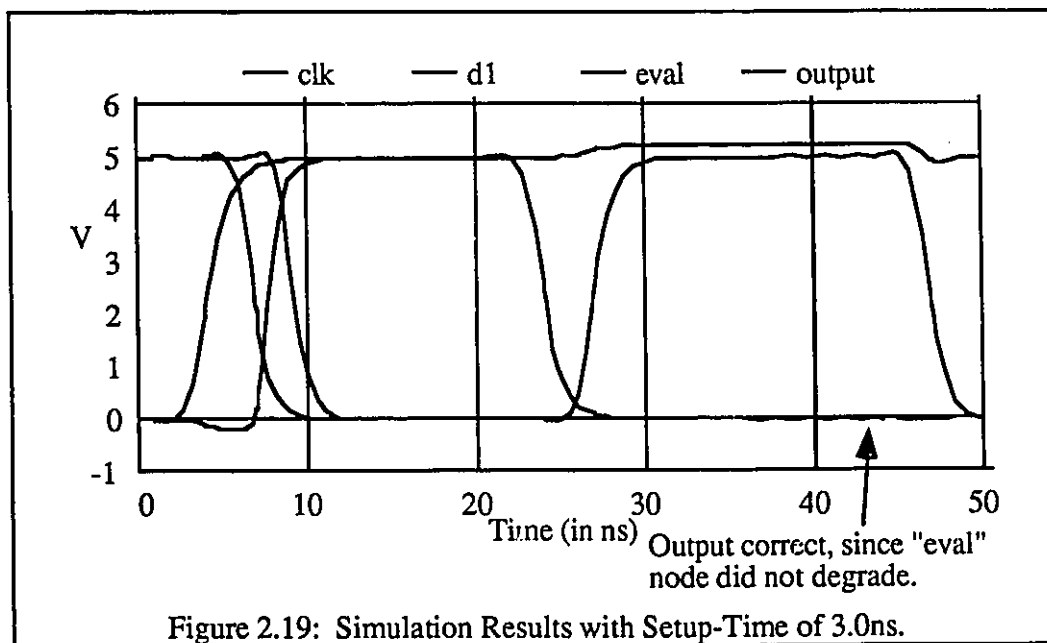
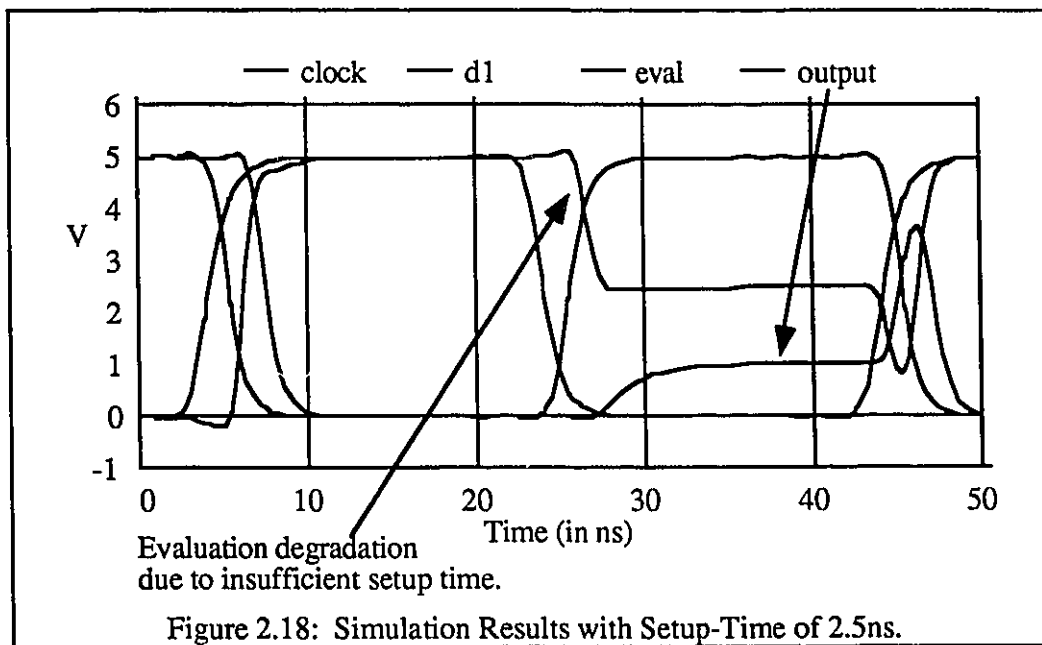


which the designer is concerned with are the input signal setup times, and the maximum evaluation time of the cell, which will determine the maximum clocking speed of the cell. This can be performed by a set of iterative SPICE runs to determine the minimum setup time. Each SPICE run will increase the delay of the clock signal hence increasing the setup time of the input signals. It is important to point out that if the SPICE software fails to run properly and sites an inability to attain DC convergence or a time-step-to-small error, this has no implication on the maximum performance of the circuit under test. It is merely an

inability of the software to produce results. Thus unless a SPICE run completes and indicates a functional failure, then no presumptions should be concluded about the circuit performance.

A complete SPICE-deck for the domino circuit can be found in Appendix II. The circuit is loaded down with a 200 ff load capacitance. Several initial SPICE simulations were performed to setup the initial input signals for proper functionality. The clock frequency for the simulations was chosen at a mid-range operating frequency of the technology; 25 Mhz. Moderate input signal edge times of 3 nanoseconds were chosen to ensure a realistic simulation of these signals at the gate inputs. The results of the iterative simulations, to determine the minimum value of setup-time, are illustrated in Figures 2.17 through 2.19. From these simulations it is determined that the minimum setup time required is 3ns. It can also be determined that, under the specified output load conditions, the maximum evaluation time for the cell is 5.3ns. Thus, assuming that the clock signal has a 50% duty cycle, then the minimum clock period is 11ns indicating a maximum clocking speed of approximately 90 Mhz.





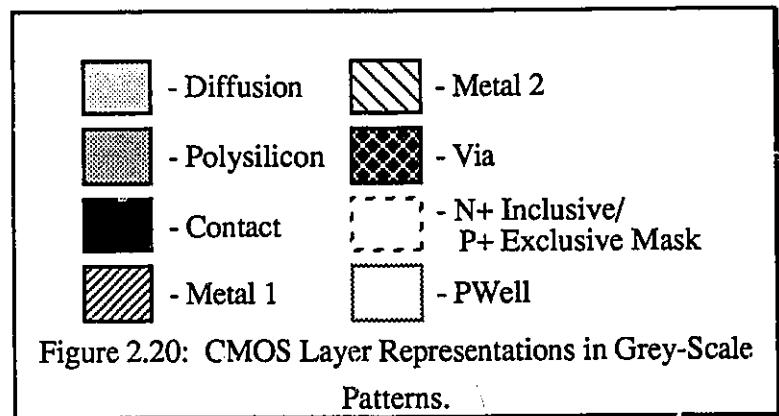
### 2.2.3 SUMMARY OF MODELLING

The last few sections have indicated the need for accurate assessment and modelling of I/O specifications. This can be done through the use of simple circuitry and the application of some common sense. Furthermore, the modelling process requires the need

to predetermine how devices are to be designed, which has a large bearing on the inter-circuit parasitic capacitance associated with the devices. Lastly, the simulations are necessary for determining gate-delay and cell evaluation times in order to accurately parameterize sub-cells for the use of digital simulation packages in order to accurately simulate large networks.

## 2.3 DIFFERENT TRANSISTOR CONFIGURATIONS

The following section will examine device lay-out configurations. To simplify the figures, some layers may be purposely left out. Figure 2.20 illustrates the standard



CMOS layer representations using simple gray-scale patterns.

### 2.3.1 THE BASIC TRANSISTOR LAYOUT

A simplified view of an NMOS transistor is illustrated in Fig. 2.21. The device illustrated has its well biased through the use of the split-contact on the source node. The split-contact in effect biases the local substrate to the ground potential. If the split-contact is replaced with a normal contact, then the two terminals would become interchangeable and their designation of source or drain terminal would be dependent on the respective terminal voltages. Note that each device well on a silicon substrate requires biasing, so that if a device is constructed without the split-contact on the source node, than a well plug will have to be placed somewhere in the local well.

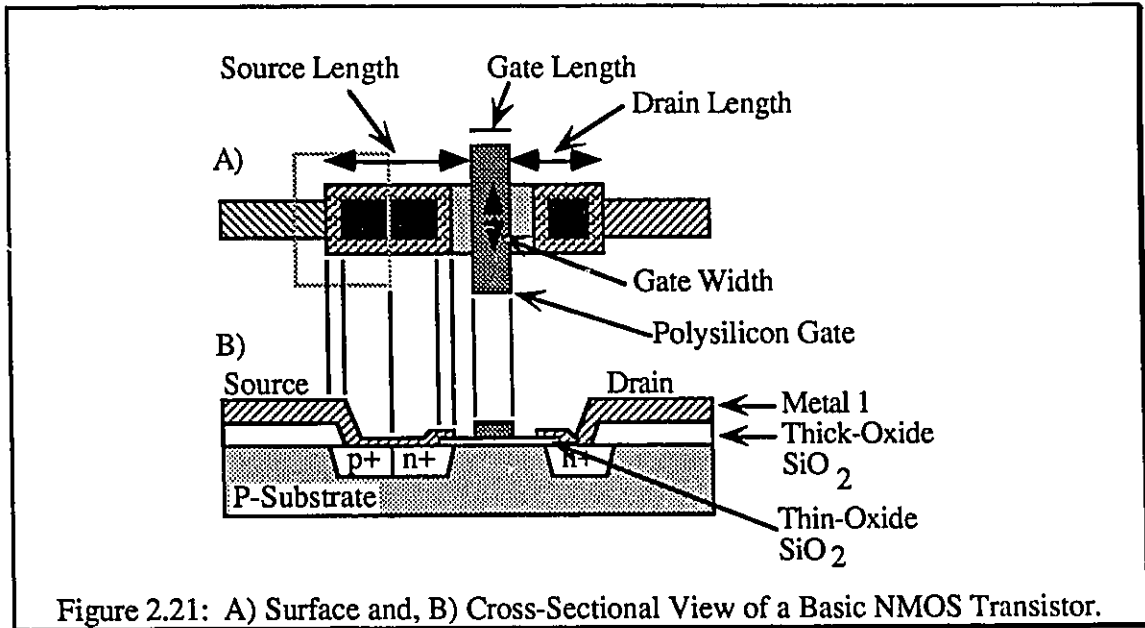


Figure 2.21: A) Surface and, B) Cross-Sectional View of a Basic NMOS Transistor.

This Figure also illustrates how to properly determine the length and width of the gate, drain, and source regions of the device.

In determining the area and perimeter of device nodes, the following equations are used:

$$\text{Area}(\text{drain, gate, and source}) = \text{Length} \times \text{Width} \quad (2.2)$$

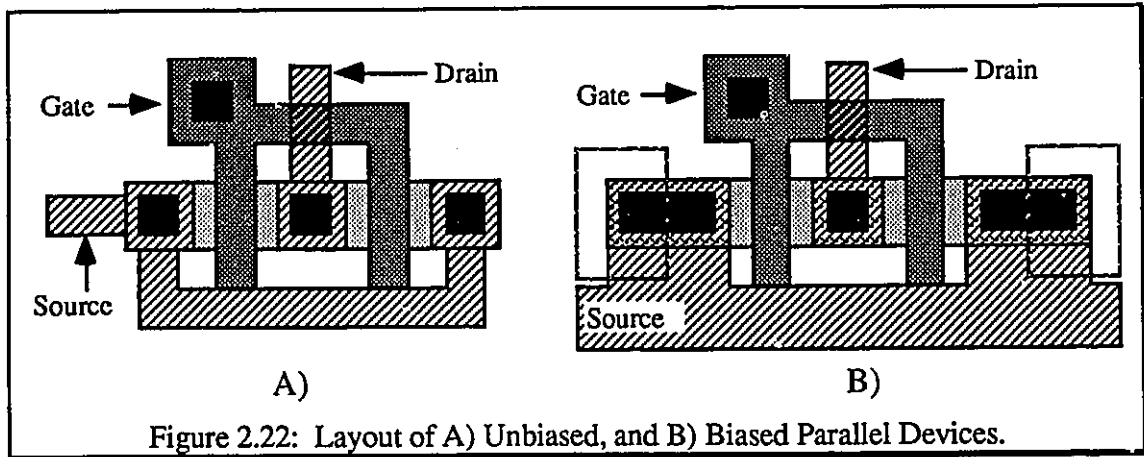
$$\text{Perimeter}(\text{drain or source}) = 2 \times \text{Length} + \text{Width} \quad (2.3)$$

Note that the perimeter of the drain and source regions consists of only three sides of the rectangular region. This is because the side adjacent to the gate region does not contribute any parasitic capacitance to the circuit because no real edge is formed with the substrate.

### 2.3.2 PARALLEL DEVICES WITH MERGED DRAINS

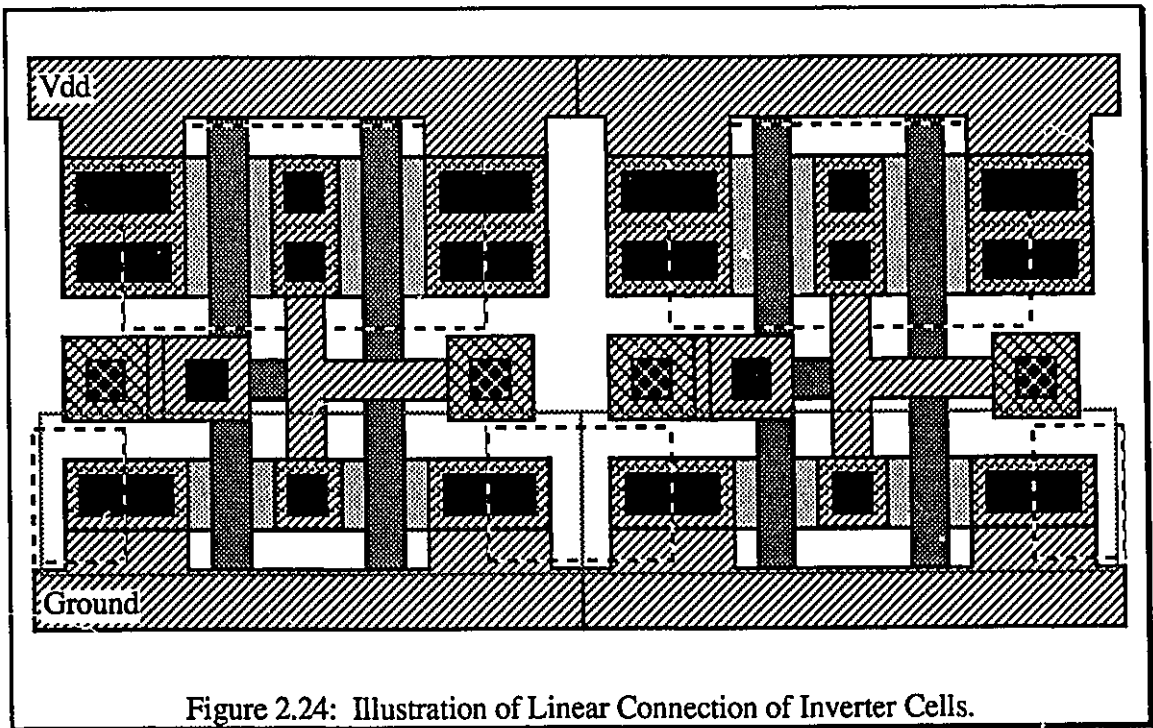
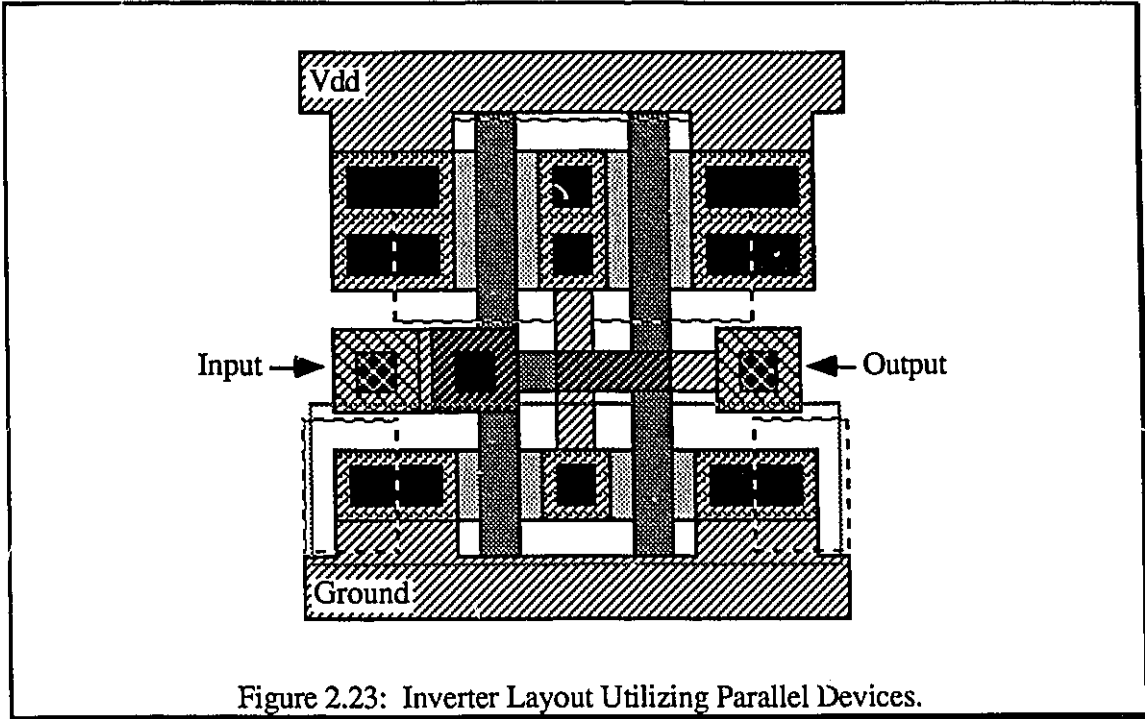
The devices illustrated in Fig. 2.22 are comprised of two similar devices merged at the drain node, so as to share only one drain contact and a single polygon of diffusion. The two sources are tied together with metal1. The resulting device will have the same effective gate width equal to the sum of the two devices gate widths,  $2W$ . The main component of the drain capacitance has been reduced by a factor of 1.6, while the edge

component has been reduced by a factor of  $(24 + \text{gatelength})/15$ , where gatelength is in design scale microns.



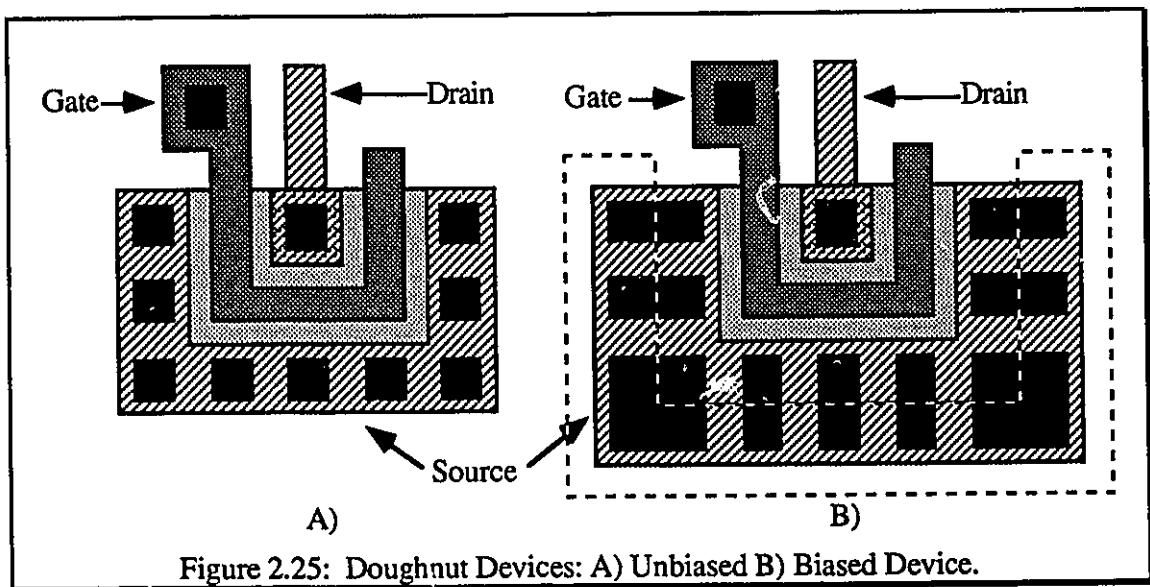
Thus a minimum gatelength of 9 design scale microns results in a parasitic drain capacitance to gate length ratio of approximately 1.87:1, normalized to a standard device. This ratio increases to a value of 3.12:1 when the device width is increased to 50 design scale microns. This merged design substantially reduces the loading of the output of the device by reducing the overall capacitance found at the drain node. The driving capability of the device is also enhanced since the parallel device has an overall lower 'on' resistance compared to a standard device with the same total gate length. The parallel device also provides an efficient method of creating large devices. This is illustrated in Figure 2.23 with the parallel design of a large inverter. Note the convenient routing of the power and ground rails in metal1. This allows for multiple cells to be connected in a linear array, as shown in Figure 2.24. Also note the placement of input and output connections with vias connecting to metal2. This allows the use of metal2 for the global routing of input and output signals. It is important to note that the vias in the cell are also aligned which allows for greater use of cell space for routing metal2 interconnect, since if the vias were randomly placed in the cell they would tend to block a large portion of the cell area against routing of metal2 signals.





### 2.3.3 DOUGHNUT DEVICES

The doughnut device is a logical extension of the parallel merged device. It offers an even higher parasitic drain capacitance to gate length ratio of  $0.393 + 0.0781 \cdot X$ , where  $X$  equals the length of the main straight section of gate measured from the diffusion edge to the inside corner of the polysilicon, and  $X$  must be greater than or equal to 12 design scale microns in order to be constructed. This gives a minimum dimension doughnut device a ratio of 1.33:1, when normalized with a standard device, which is poorer than a parallel merged device with an equivalent gate length, but a device of  $X$  equal to 50 design scale microns has a ratio of 4.75:1, which improves upon the parallel merged value of approximately 3.2:1. The doughnut device thus can offer a more efficient device which has less effective drain capacitance per unit of gate length when the required gate length is quite large. Figure 2.25 illustrates a) an unbiased, and b) a well-biased doughnut device. These devices, though offering better drive capability, tend to create less efficient layouts and hence should only be used where they are absolutely required.



## 2.4 PRACTICAL LAYOUT CONSIDERATIONS

The following is a collection of design techniques attained through several years of experience in chip designing and testing by the author.

### 2.4.1 GENERATING PAD FRAMES

The following points should be noted when determining the numbers of pads (pin count) required for a given test design:

- 1/ **Bring all output signals to individual pins.** Do not multiplex output pads unless it is absolutely necessary; if the output signals connected to that pad fail, there will be no way of absolutely determining what caused the failure.
- 2/ **Use individual clock signals** for each test structure on the chip, which are brought in on individual pins. This will allow for individual structures to be clocked while all others remain stopped. This will reduce the amount of noise during testing and will allow for the measurement of dynamic and static currents of the individual structures.
- 3/ If the test structures consist of a set of pipelined cells, where the clock signals are buffered through the pipeline, than the **clock signals at the end of the pipeline should be buffered and placed on output pads** so that the integrity and shape of the final clock signals can be 'scoped to determine if failure of a structure could be contributed to poor clock signals due to lack of sufficient buffering or degradation of signals due to noise or signal jitter.
- 4/ **Separate power and ground rails** should be used to separate internal circuitry rails from the noisy pad driver rails, since the large switching spikes generated by output pad drivers can disrupt the functionality of sensitive internal circuitry. This concept can be taken one step further, if the chip has internal analog and digital circuitry along with I/O pad

driver circuitry, then three separate rails should be used to ensure the greatest decoupling of the different circuit components. These separate rails will obviously be connected to separate pins to allow for the decoupling of those rails to be performed off-chip through the use of ceramic filter capacitors.

**5/ Use enough power and ground pins.** Do not restrict the amount of power that can be drawn by the circuitry by not using enough power and ground pins. There should be a power and ground pin between every three to five output pad driver pins depending on the amount of current that the output pads require. Unless you are absolutely sure, do not assume that one power pin can supply all the power to your circuitry and the output pads.

**6/ When laying out the pad frame do not create ground loops** with the metal conductors. Ground loops are any circular or annular pieces of metal which are connected to the same node. Thus the main I/O pad power and ground rails should be cut so that they do not form a complete circle. These metal loops constitute a single loop transformer which can produce loop currents in the conductor due to magnetic interference and can degrade the performance of the circuitry.

**7/ If pin limitations require the multiplexing of pins, then input pins can be multiplexed to different test structures** provided that the input signals only drive device gates. Do not connect input pins from different circuitry which are connected to drains or sources of devices because they can cause clipping of input signals. This can only be determined if the complete setup can be simulated using an analog simulator.

**8/ Consider tapping out critical circuit signals** through buffering circuitry, which will place a minimal amount of load on the circuit under test, to drive an output pin. This extra observability is often useful when determining circuit failure mechanisms. If it is felt that tapping out the signal may disrupt the circuitry, then duplicate the structure on the chip and keep the integrity of one while tapping out the second.

After these considerations in determining the number of pins that will be required, the next step is to look at the circuitry in order to place the pads close to the circuitry that they will be connected too. When wiring signals from the I/O pads, the use of metal2 for routing the long signal connections to the circuitry is a better choice over the metal1 routing layer, since the metal2 layer has a lower capacitance per square value than that of metal1.

Following these tips will ensure that the pad frame will provide the best performance and observability of the circuitry under test.

## **2.4.2 POWER DISTRIBUTION**

Power distribution in circuitry is an important issue. The most important thing to remember is that the power flowing into your circuitry is the life blood of your design. If you severely restrict the flow of power to your circuitry, then it will most likely fail or will be unable to perform to its peak ability. The following points should be observed:

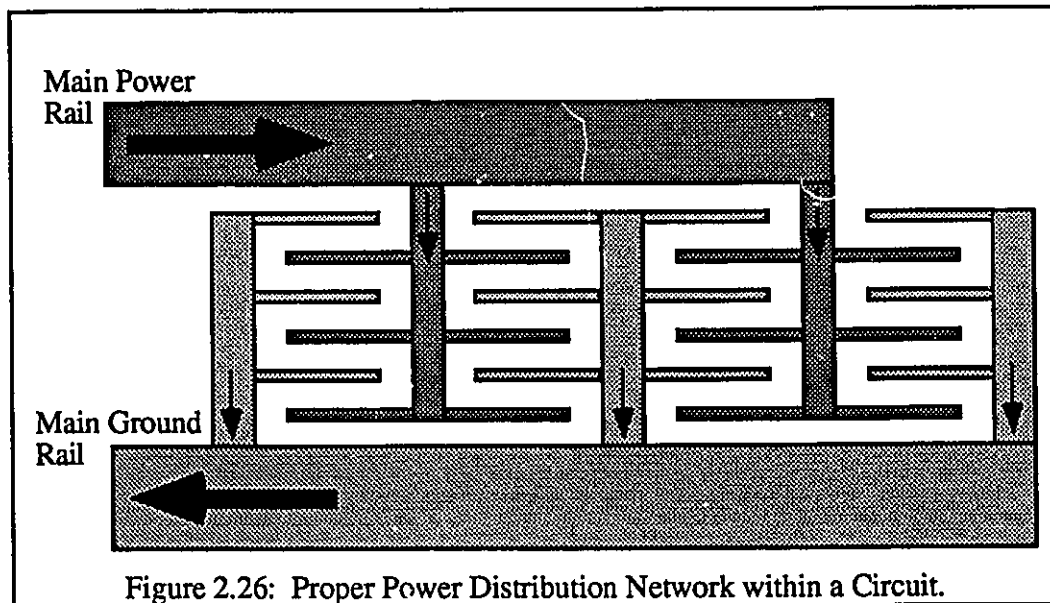
**1/ Metal 1 is more suitable for power rail distribution** since it offers a higher per square capacitance over the metal 2 routing layer. The high capacitance is advantageous on power rails as the increased energy stored in this extra capacitance will help support the power rails when current demand is high and help filter out high frequency noise components that exist on the power rail.

**2/ Make the power rails large enough.** Excessively thin or narrow power rails lead to dips in supply voltage at critical switching periods when the demand for current exceeds the power rail's ability to provide it. Secondly, an excessively thin or narrow ground rail will lead to a floating ground reference as excessive charge is dumped onto the ground rail at a level which is above the ability of the rail to instantly carry it away. Hence both rails must be of an adequate size. The normal procedure for sizing of the rails is to determine the maximum peak current demand which may be imposed by the circuitry at that point.

Then multiply that value by a safety margin value, say 1.2, and then divide that result by the current density per square of conductor value, which can usually be found in the design rule guide book for the technology. This will give the effective width of the power rail conductor. The ground rail conductor is then oversized slightly, say by 5% to 10% to ensure that a solid ground reference voltage is maintained. If the exact size of the power rail can not be determined, then make it large, use any empty space on the chip to increase the size of the power rails.

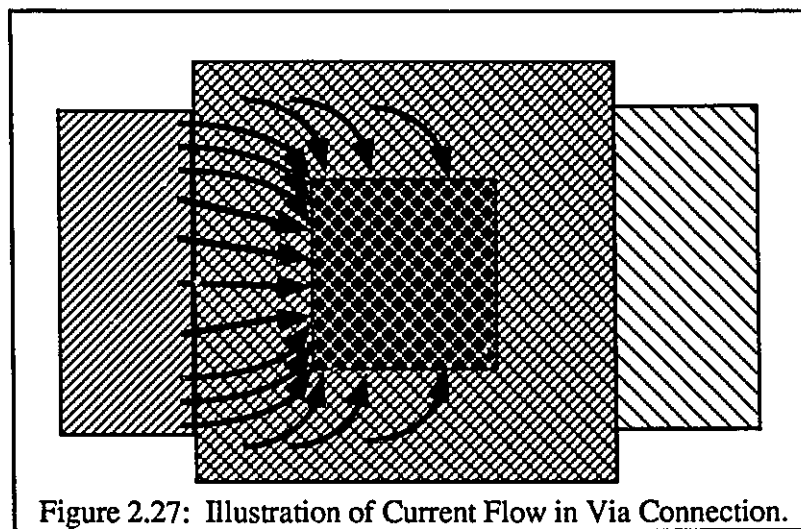
3/ For circuits which switch large amounts of current remember to **make the ground rail larger than the power rail** because CMOS circuitry is more susceptible to floating ground rails than to a drooping power rail.

4/ **Do not use polysilicon or large numbers of via jumpers** between metal1 and metal2 in the power rail construction. Both polysilicon and vias produce high resistance connections which will create larger voltage drops on the power rails during peak current demands.



5/ Treat the conductors' width along the power rail like a water plumbing system. Thus the main power rail is the widest and gets progressively smaller as it reaches the ends of the feed lines; this is illustrated in Figure 2.26. Also note in Figure 2.26 that no ground loops are formed in the power distribution network. The rails branch-out and terminate in the device circuitry, secondly the ground rails are slightly larger than the power rails.

6/ When vias must be placed in the power rail, realize the current carrying limitations of the vias. Vias have a higher per square resistivity than that of the metal interconnect, which means during peak current demands a substantial voltage drop may occur at these points. Secondly, the vias have a lower current carrying capability due to the physical construction of a via connection. A close-up view of a Via connection is illustrated in Figure 2.27.



The small arrows in the diagram indicate current movement, and their relative spacing indicates the relative current density in the Via connection. From this diagram you will note that the majority of the current tends to travel through the leading edge of the Via structure leading to higher current densities at the edges of the structure which may not be possible if the value of the per square current density for the conductor is exceeded. The resulting effect is a current bottle-neck at the via location.

The problem can obviously be alleviated by using more vias with an overall leading edge length larger than the conductors' width, as illustrated in Figure 2.28.

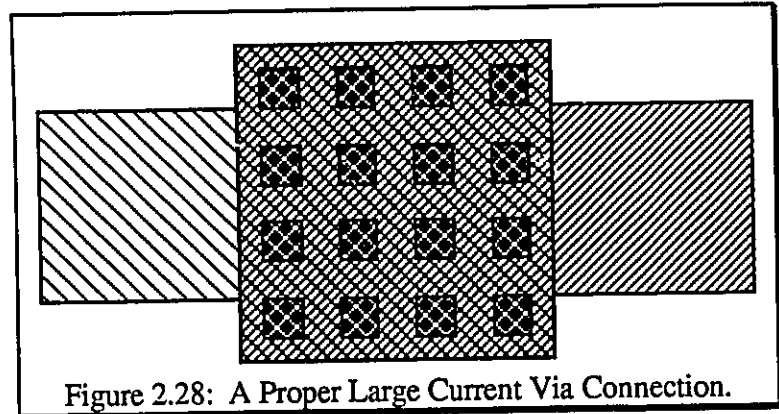


Figure 2.28: A Proper Large Current Via Connection.

### 2.4.3 ROUTING

When designing large cells, routing is very important. The problem of routing congestion is often due to poor pre-layout considerations. During the placement of sub-cell circuits into a larger cell, care must be taken to determine the paths of all interconnect and to leave adequate space for them. One

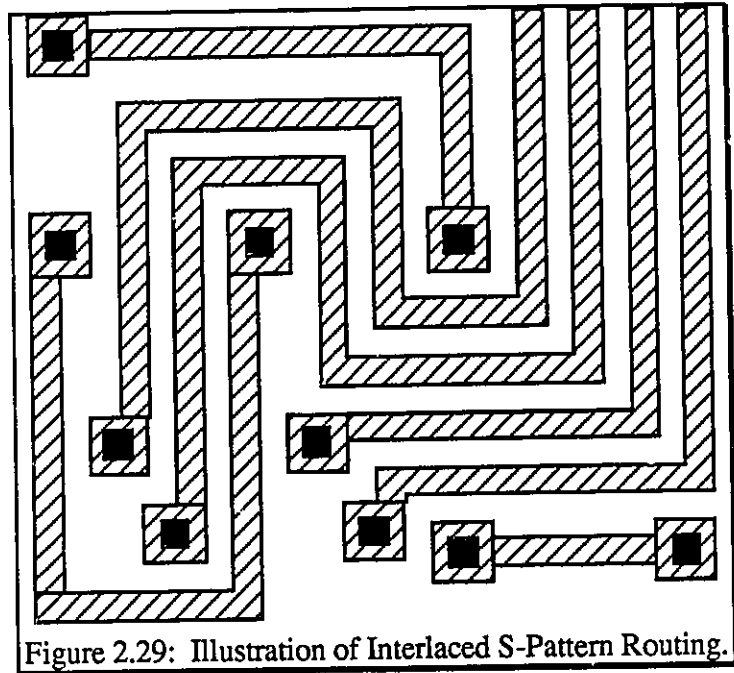


Figure 2.29: Illustration of Interlaced S-Pattern Routing.

helpful method for getting metal2 routing into and out of large cells is to use continuous interlaced 'S' patterns as illustrated in Figure 2.29. In order to do this the via locations within the cell must be placed for maximum efficiency. When vias in the cell are aligned they allow for greater use of cell space for routing metal2 interconnect. If the vias are randomly placed in the cell they tend to block a large portion of the cell area against routing of metal2 signals. This concept is more clearly seen in Figure 2.30. Fig. 2.30a) illustrates



a cell which has a large area for routing signals over it in metal2, while Fig. 2.30b) shows a poorly designed cell, which has almost no area free for routing of metal 2 signals.

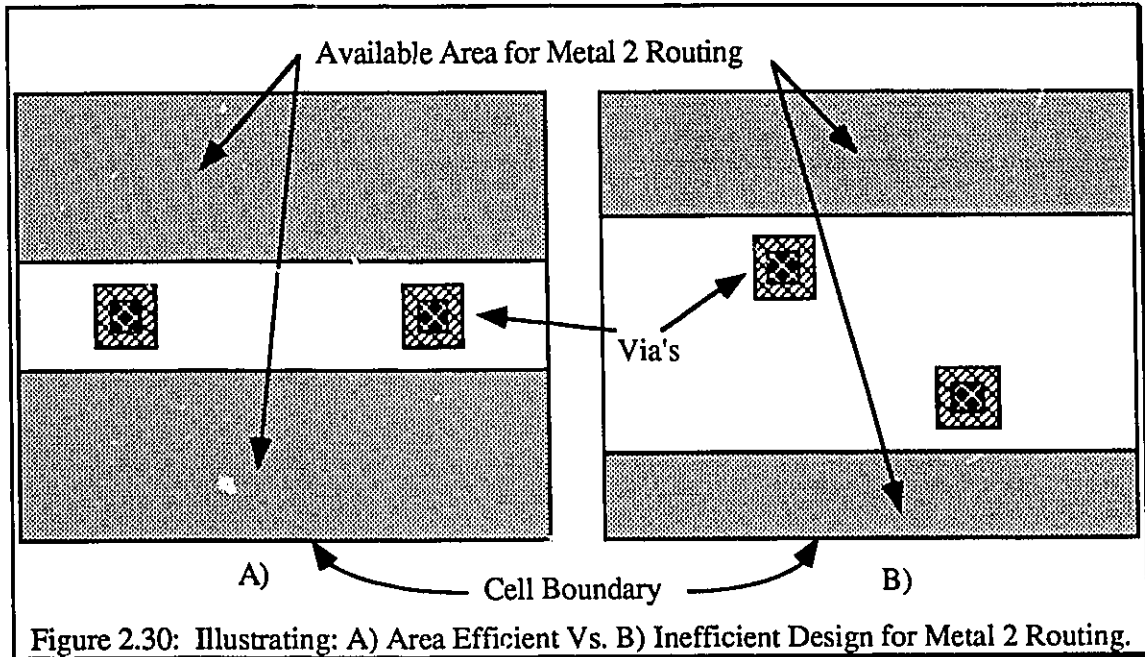


Figure 2.30: Illustrating: A) Area Efficient Vs. B) Inefficient Design for Metal 2 Routing.

When making short metal2 jumpers in a sub-cell, it is important to limit these jumpers to run in one direction, either horizontally or vertically. Having small metal2 jumpers running in both directions tends to severely reduce the available routing space in the metal2 layer. Also, keeping these jumpers as straight as possible will improve the metal2 routing efficiency.

Through the application of these simple rules it is possible to produce very efficient designs which utilize a majority of the cell space for routing. This is illustrated in Figure 2.31, which shows the metal 2 routing over a 3-bit adder cell.

## 2.5 SUMMARY

Developing good design practice is essential to becoming a good VLSI designer. Utilizing the design and simulation considerations given in this chapter should aid in this development.

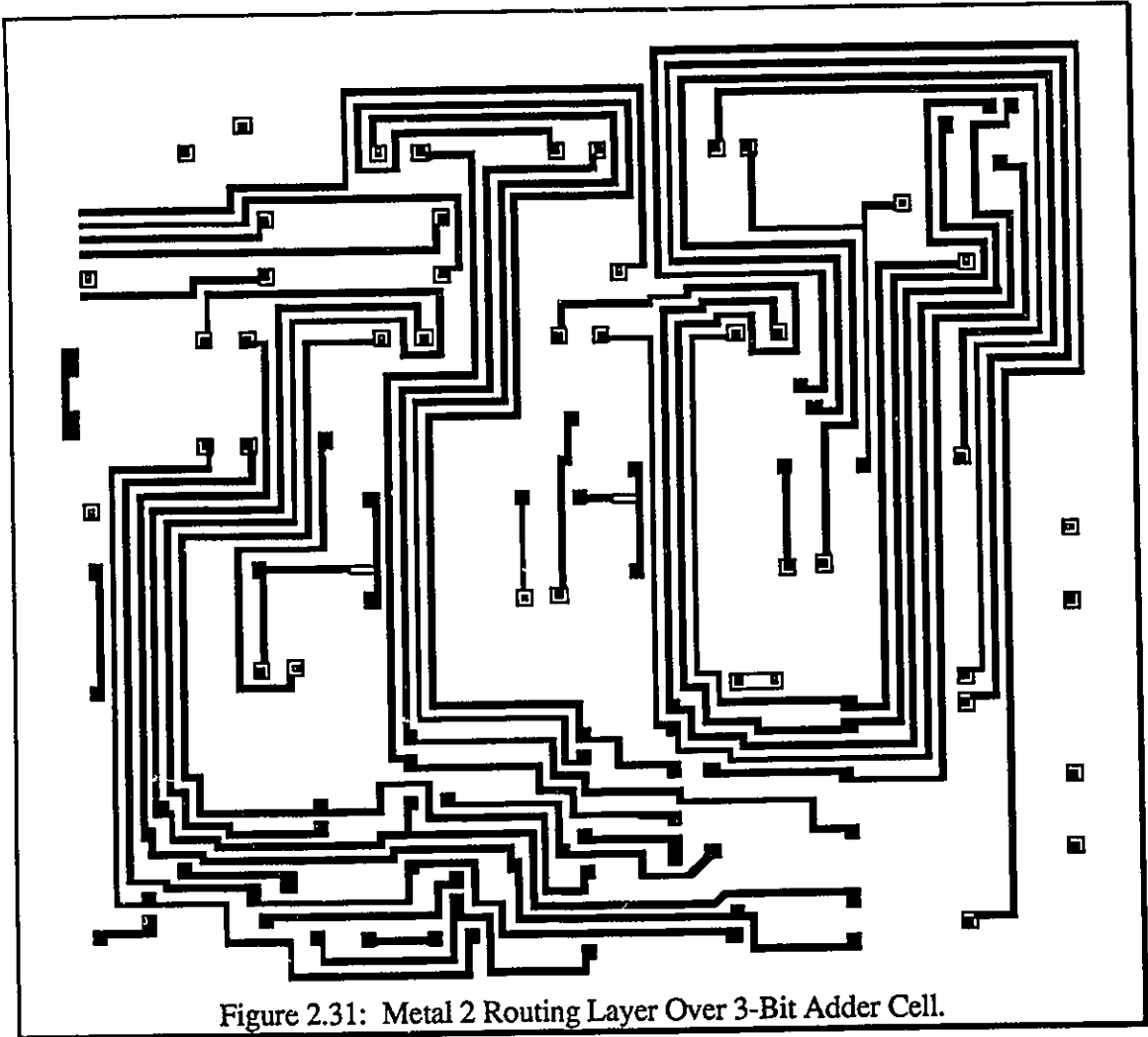


Figure 2.31: Metal 2 Routing Layer Over 3-Bit Adder Cell.

The key to developing high speed architectures is the minimization of capacitance. Hence smaller is always faster. The information in this chapter has come from practical experience obtained through trial and error approaches, and thus the author stresses experimentation during the design cycle as a method of solving design problems. It is most important to remember that common sense will be your guide in most design issues. Thus controlled experimentation proves to be much more worth-while than blindly grasping for solutions.

**CHAPTER**  
**3**  

---

**THE DEVELOPMENT**  
**OF SWITCHING TREES**

### 3.1 INTRODUCTION

The development of the switching tree cell is rooted in the work of Peter Bird, who developed a standard ROM cell used in RNS(Residue Number System) DSP(Digital Signal Processing) processing [13]. The ROM look-up cell consisted of a 32 position ROM, arranged in a 4 by 8 rectangular array, addressed by row and column decoders. The ROM look-up cell was designed using dynamic logic, utilizing pseudo two-phase clocking techniques. The ROM cell constituted a look-up table with a 5 bit address field which produced a 5 bit output word. The cell was capable of running at approximately 20 Mhz, utilizing a 3 micron CMOS technology. The switching tree cell is the end result of a two year research program to develop the most area and speed efficient ROM cell structure. Such a cell can implement any boolean function on the address variables and, in particular, those used in regular arithmetic arrays.

Several ROM architectures were developed and explored, during the course of the research presented here, to determine the most area efficient method of implementing high speed dynamic ROM structures. The first cell that was developed, which will be called the ROM5 cell, was a derivative of Peter Bird's RNS ROM cell. It utilizes pseudo two-phase clocking with a static CMOS logic implementation of standard row and column decoder networks. These networks address a new ROM structure, referred to as a STAR configured ROM. In this initial work we optimize the performance of existing technology

rather than attempting to develop a totally new design. The cell also represents a reference point against which we can judge the viability of future designs.

The second cell that was developed will be referred to as the JROM cell. This cell consists of a new dynamic single-phase clocking logic scheme developed by Jullien [14]. The logic is driven by a ROM cell which is built from a full binary tree of NMOS transistors. This cell leads to the final version of a minimized binary tree, referred to as a *Switching Tree* cell, the cell which utilizes the true single-phase clocking technique developed by Yuan and Svensson [7].

### **3.2 THE ROM5 CELL**

The ROM5 cell was designed to be an improvement to an existing design. The original design utilizes a dynamic logic configuration synchronized by a two-phase clocking technique. As discussed in the introduction, two-phase clocks are particularly susceptible to clock skew problems that eventually lead to latch transparency and system failure. The effects of clock skew on large designs, consisting of long chains of cells, require the system to operate at slower clock frequencies than that of the individual cells. This degradation in system performance, over the performance of the individual components, is due to process variations over the two dimensional surface area of the design. The clock line signals skew since one clock line might experience a higher per unit capacitance and the clock driving devices may not be perfectly matched. Thus the goal of this design was to attempt to develop a cell which optimized the performance problem in order to increase the system level performance of the existing technology.

The design goal was to create a cell where the maximum precharge and evaluation time met the specifications of a higher clock rate frequency than was capable of being driven by the VLSI Research Group library of I/O pads. The clocks are defined with minimal non-overlapping safety margins which allows the cell to run at slower speeds,

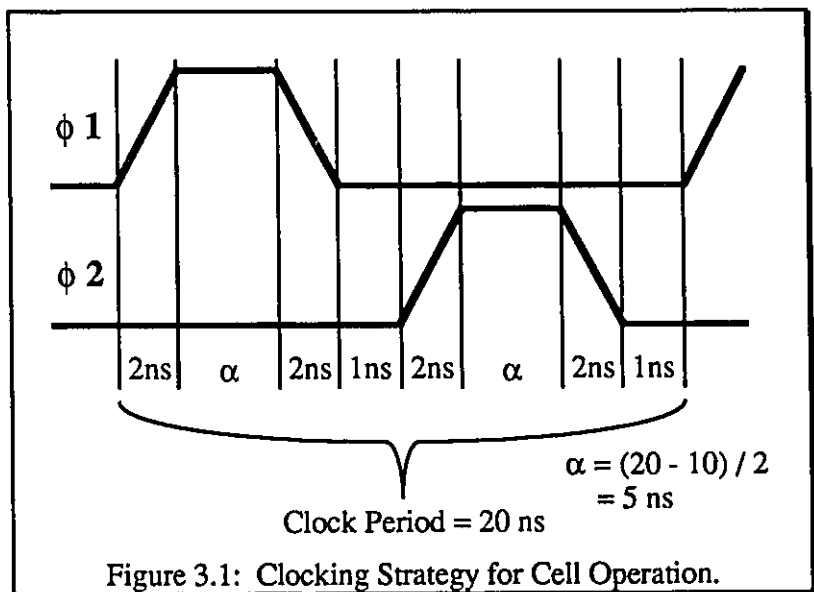
within a large system, with increased safety margins. The complete schematic diagram for the ROM5 cell can be found in Appendix B.

### 3.2.1 THE CLOCKING STRATEGY

The clocking specifications for the cell were defined as follows:

- 1/ The upper limit of the clocking speed to individual cells will be 50 Mhz (which implies a clock period of 20 nanoseconds (ns)).
- 2/ All signals entering and internal to the cell structure must maintain minimum signal rise and fall times of approximately 2 ns.
- 3/ The minimum safety margin for the non-overlapping of the clock phases will be set to 1 ns, which implies that the clock pulses will have a duration of  $\alpha$  ns. This implies that the ROM's precharge and evaluation cycles fall within the active portion of the clock pulse, which, assuming the circuitry is active from the 50% points of the clock edges, gives a maximum time of 7ns to perform the precharge and the evaluation function.

Thus the clocking signals are defined as illustrated in Figure 3.1, where  $\alpha$  represents the stable portion of the clock pulse. Only the Positive going clock phases are illustrated.

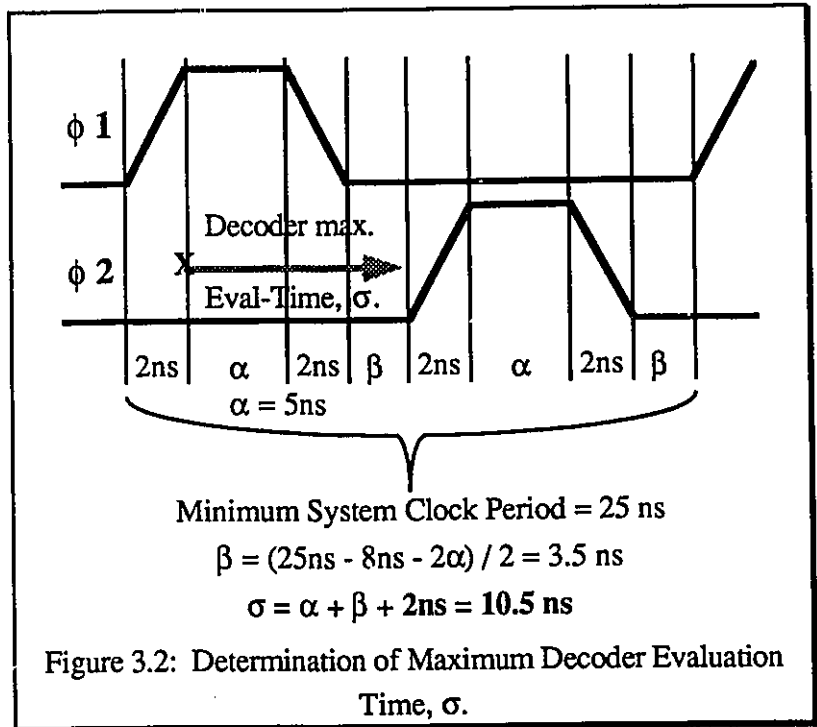


The next step is to

determine the acceptable critical path time delay for the static CMOS decoder network. Since the decoder network is a static implementation then the critical path delay for the

decoder need not be as stringent as the precharge and evaluation time of the ROM structure, since its evaluation time will be fixed by defining of the maximum system operating speed.

The maximum system level speed that is expected is 40 Mhz, which implies a 25ns clock period. Since data will be presented to the decoder at the rising edge of the first clock phase, the decoder must present valid data to the ROM structure just prior to



the ROM evaluation which begins at the rising edge of the second clock phase. Thus the maximum evaluation delay of the decoder is found to be 10.5ns, as illustrated in Figure 3.2. These specifications are used during initial SPICE simulations to determine if the circuitry has met the required specifications.

### 3.2.2 THE STAR CONFIGURED ROM

During the initial SPICE simulations it was found that the normal dynamic 32 location ROM structure implementation, illustrated in Figure 3.3, could not meet the required evaluation time of 7ns, as specified in section 3.2.1.

Initial mask-extracted SPICE simulations indicate that the ROM structure can precharge in 5ns but the evaluation time is limited to 9ns. The slow evaluation time of the ROM

structure is clearly due to the slow charge drain, from the parasitic capacitance associated with each ROM evaluation node, through minimum size NMOS devices.

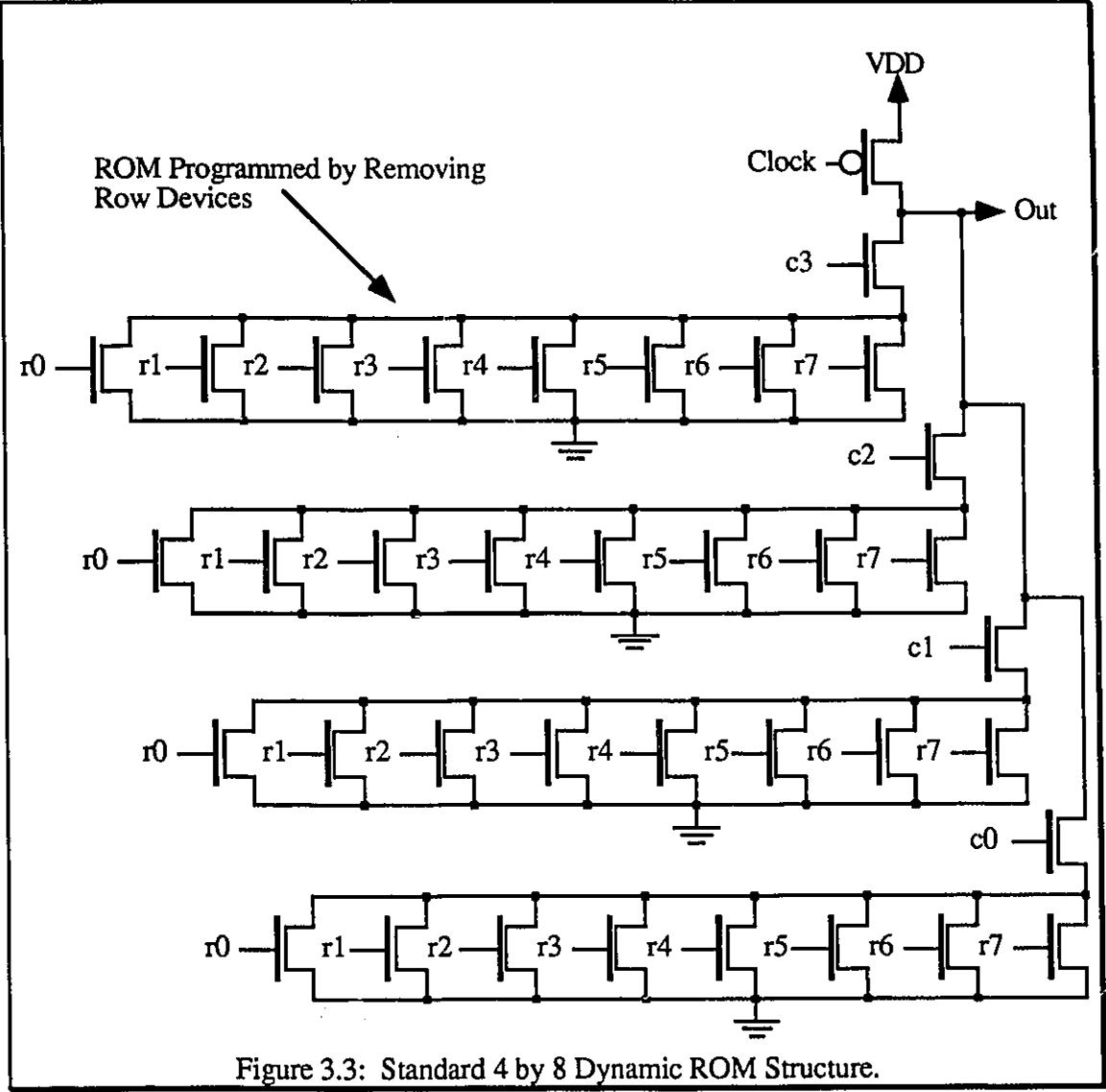


Figure 3.3: Standard 4 by 8 Dynamic ROM Structure.

The size of the column devices in the ROM were increased but had no significant effect on the discharge slope. Increasing the width of the row devices, while simultaneously increasing the size of the PMOS precharge device, ensures that charge redistribution within the ROM structure does not degrade the evaluation high level below the acceptable noise margins. This scaling up procedure, however, did not result in suitable increases in

evaluation speeds since the size that the row devices were required to be increased to, was limited by the acceptable size that the ROM structure could occupy. Thus a different method of implementing the ROM structure is required which reduces the effective parasitic capacitance of the ROM structure while maintaining an acceptable size.

Taking a close look at the ROM structure, it is evident that the bottom devices in the NMOS series chain should be made as large as possible in order to decrease the evaluation time, as dictated by device scaling theory [15]. Since the row devices are on the bottom of the series chain this will also make the ROM structure too large. A novel solution is to flip the ROM structure, as indicated in Figure 3.4, placing the four column devices on the bottom of the chain.

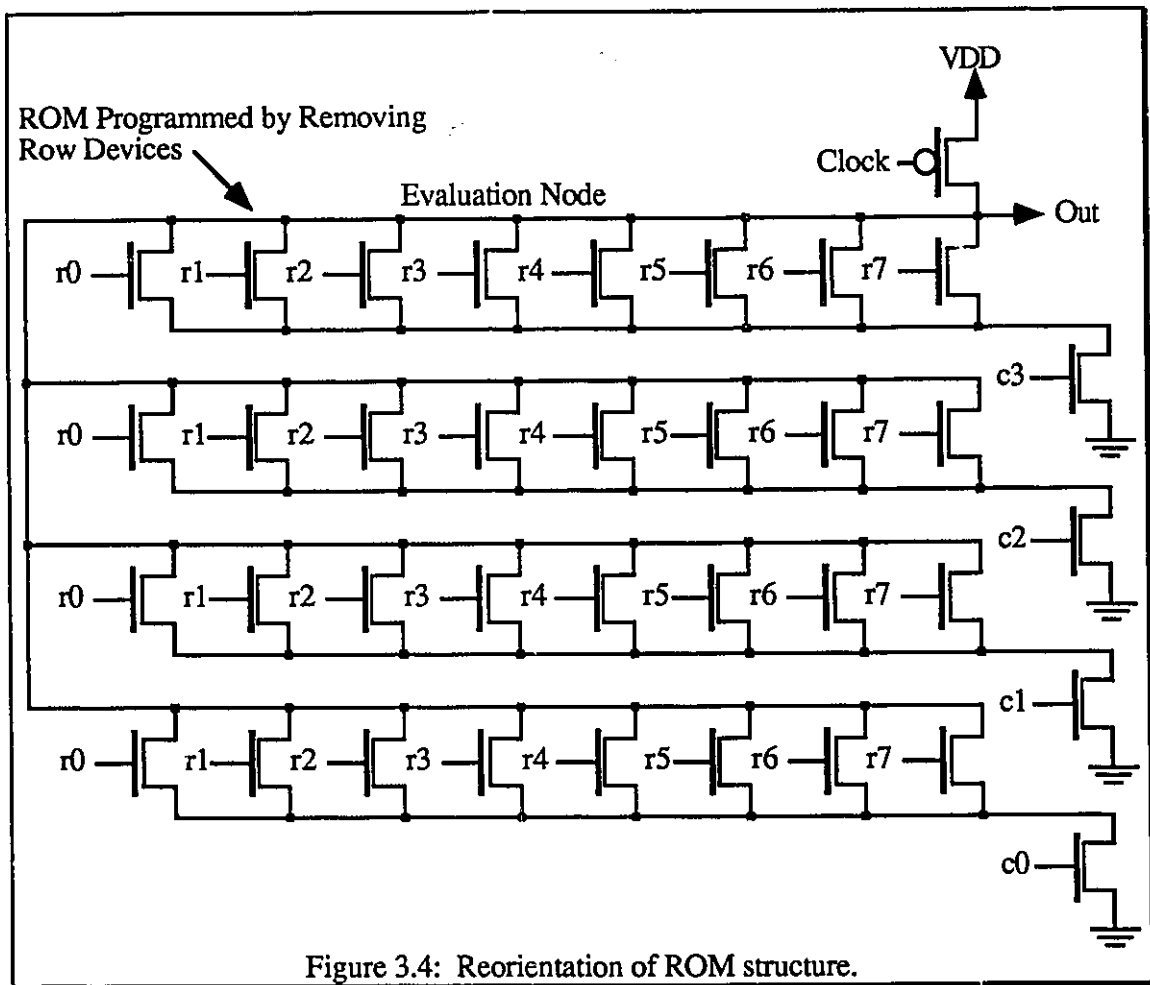
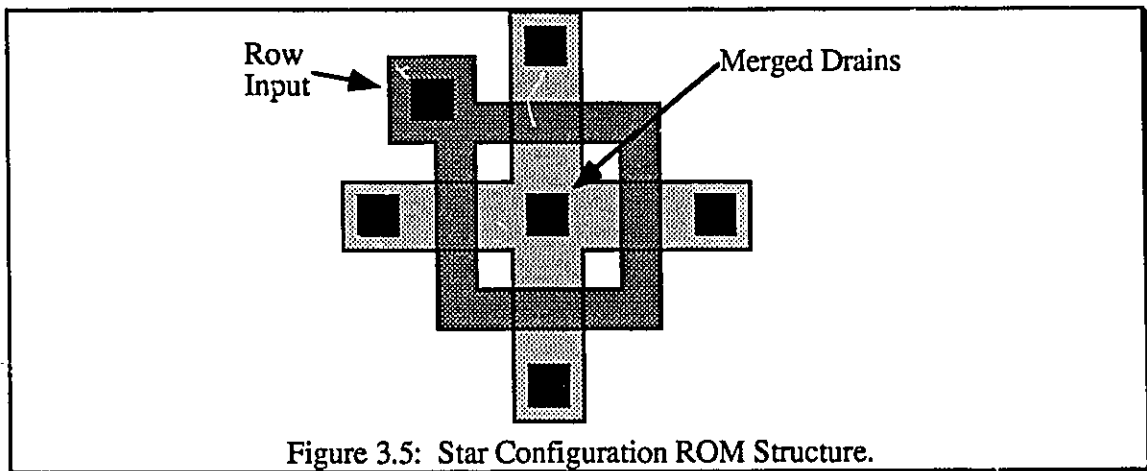


Figure 3.4: Reorientation of ROM structure.



Hence these four devices can be made quite large without paying a large penalty in area. Unfortunately, though, this reordering of devices creates a relatively large capacitance associated with the evaluation node, as all the drains of the row devices are connected to the evaluation node. SPICE simulations confirm that this particular structure, even with minimum sized row devices and large column devices, still evaluates slower than the normal design. Hence it is necessary to decrease the effective drain capacitance of the row devices in order to speed up the evaluation time.

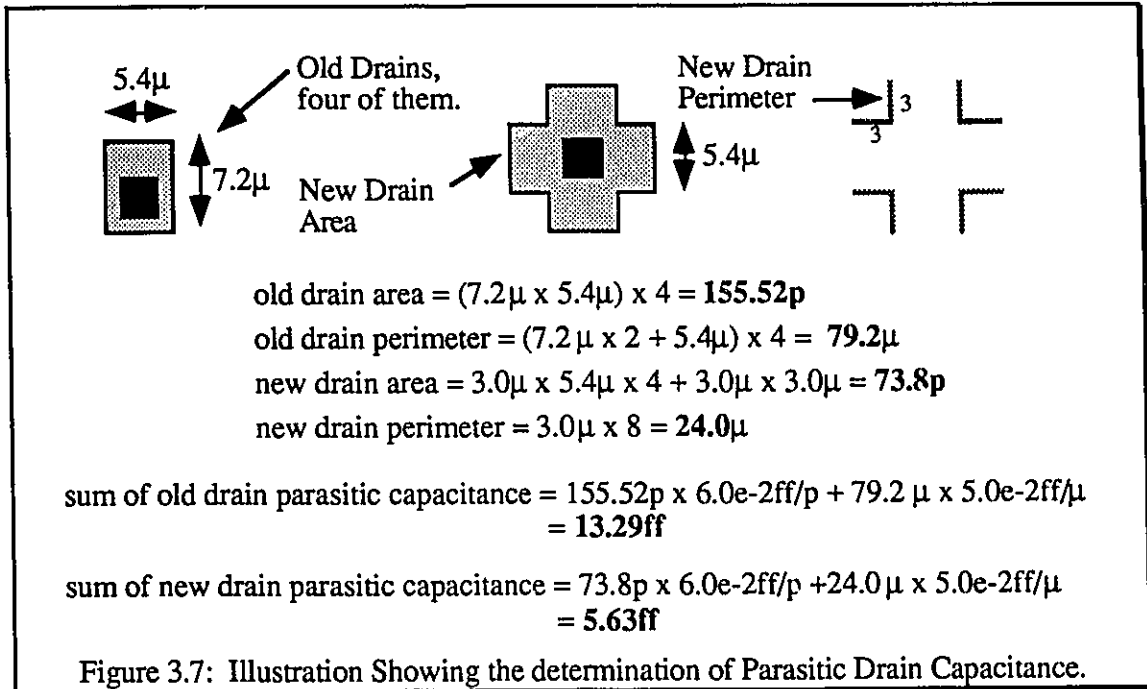
After much experimentation, a new ROM configuration has been developed which reduces the evaluation time to acceptable limits by merging the drains of the four row devices in the ROM structure which share the same gate signal. The new ROM configuration, Figure 3.5, is referred to as the STAR configuration after the appearance of the transistor diffusion blocks. The grouping of the ROM transistor into eight Star type diffusion blocks is better illustrated in Figure 3.6.



The effective parasitic drain capacitance associated with the drains of the row devices is reduced by a factor of 2.4, down to 42 % of the original value. These calculations are illustrated in Figure 3.7. This reduction in the parasitic drain capacitance on the evaluation



dynamic ROM structure which measures  $11.77 \times 10^3 \mu^2$ . This corresponds to a 56% increase in area in order to achieve the desired evaluation speed.



The STAR configured ROM cell has several advantages over the older styles of dynamic logic implementation, the most important of which is the significant increase in performance. The STAR ROM provides twice the performance for an acceptable 56% increase in area. The cells also utilize a full guard-ring of local well biasing in order to reduce the effects of substrate currents on the ROMs performance. The guard-ring will essentially collect any injected substrate current which might affect the ROM performance by unbiasing ROM devices during an evaluation cycle; this can lead to incorrect results. The row devices have been kept small which minimizes the input signal capacitance, thus insuring peak performance. The large column device approximates the same total input capacitance associated with the eight row devices, which means unbalanced ROM cell driving circuitry is not required.

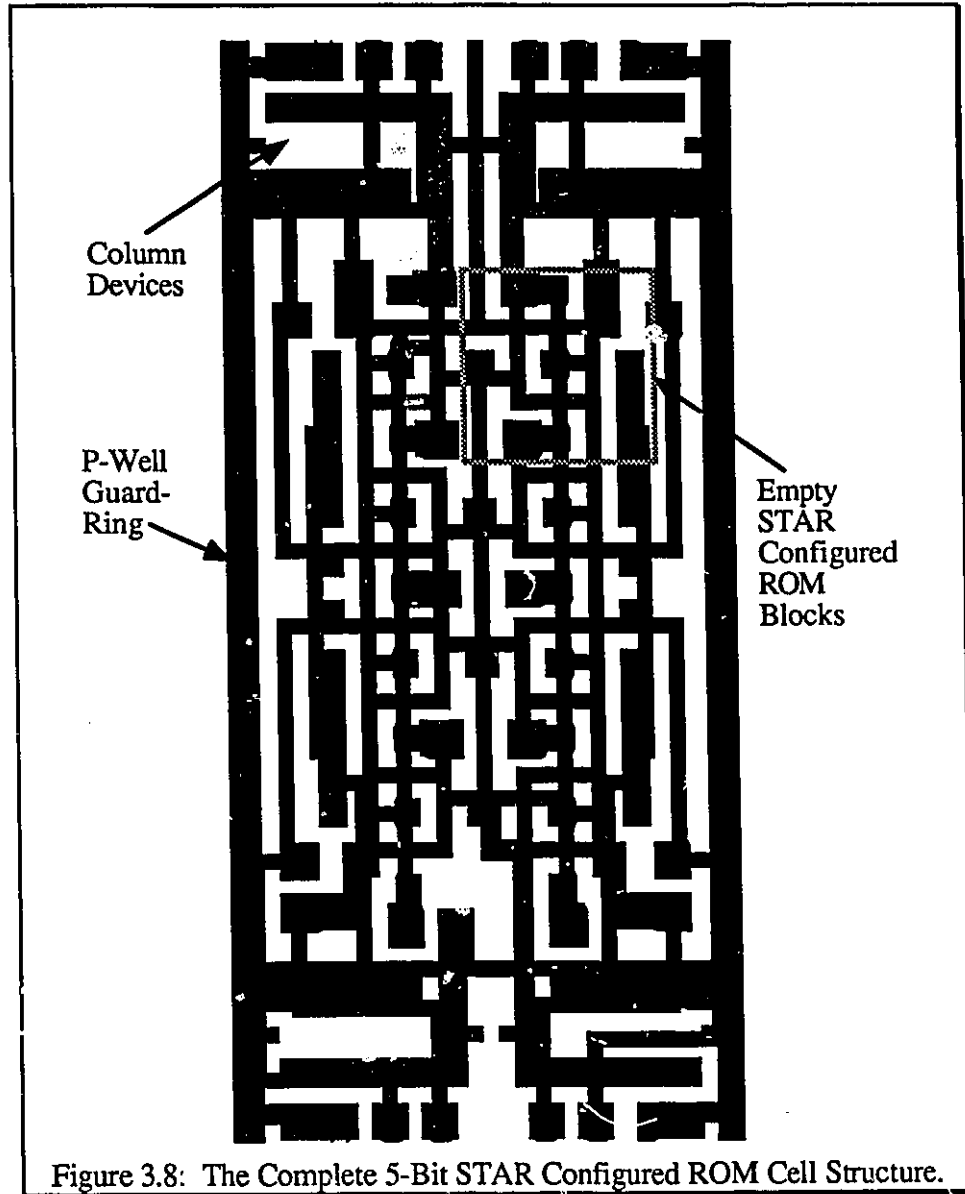
### 3.2.3 TEST RESULTS OF THE IC3WROMB CHIP

The ROM5 test chip, IC3WROMB, was tested using the Asix-2 tester. Functionality was determined through the use of an exhaustive test pattern. The test batch consisted of five 40-pin dip chips. The summary of the test results is given in Table 3.1.

Dev #	Maximum Clock Speed (in Mhz)	I <sub>dd</sub> @ Max. Speed (in mA)	I <sub>ddstatic</sub> (in mA)	V <sub>oh</sub> (in volts)	V <sub>ol</sub> (in volts)	V <sub>ih</sub> (in volts)	V <sub>il</sub> (in volts)
1	36	37.2	6.40	4.92	0.14	2.76	0.47
2	36	37.4	6.50	4.89	0.14	2.78	0.46
3	33	33.4	6.25	4.83	0.14	2.76	0.44
4	33	33.8	6.20	4.84	0.14	2.75	0.50
5	36	37.0	6.45	4.90	0.15	2.74	0.44

All five devices proved to be functional and had an average maximum speed of 35 Mhz, which is slightly less than the maximum speed of the existing I/O Pads (the design goal). The static current demand represents the current demand when all inputs are low and the clocking signals are still present. In this case four components of the static current flowed into the output pads which were connected to the clock lines to bring the clock signals off chip. Similarly, the current demand at the maximum speed contains components from the nine output pads switching dynamic current into the load capacitances as well as that to the ROM5 test cell.

The data outputs are very solid signals with a minimum voltage swing of 4.7 volts at maximum speed. Similarly the input noise margins were quite adequate as input high values required a minimum voltage of only 2.8 volts and input low values required a maximum voltage of 0.44 volts. These levels make this cell perfectly compatible with current TTL levels as well as being CMOS compatible.



### 3.2.4 SUMMARY OF THE ROM5 DESIGN

The ROM5 cell design proved to be a complete success. A test cell, IC3WROMB, was fabricated, tested and did prove to be capable of running solidly up to 36 Mhz. The cell design included a new ROM cell architecture that offered superior performance over previous designs developed by members of our VLSI group. From this study it is apparent that this approach, which utilizes well scaled static logic for the decoder network and a high

speed two dimensional ROM design, results in an excessively large cell with a high power dissipation level. It can then be concluded that this design style trades-off power and area for speed, and that an alternative architecture is necessary to improve upon the speed-power-area product.

### 3.3 THE JROM CELL

The JROM cell is the first attempt to develop a new cell design which utilizes single-phase clocking as well as a ROM structure which requires no additional decoder circuitry, allowing more time for the ROM to evaluate in the clock cycle. The following section traces the approach taken in deciding to implement 'n' dimensional ROM Trees or Switching Trees. A complete schematic diagram of the JROM cell can be found in Appendix C.

#### 3.3.1 THE BINARY ROM CELL

Normally, ROMs have always been designed to be addressed as two dimensional structures. The regular two dimensionally addressed dynamic ROM structure, with an address field of 'n' bits, is essentially composed of  $2^n$  NMOS row devices and  $2^{n/2}$  column devices with a single PMOS transistor which acts as the precharge device. Thus if  $n = 6$ , then the ROM would have 64 row device locations addressed in an 8 x 8 fashion. The ROM itself, though, could be arranged in several different arrangements, as illustrated by Table 3.2. From Table 3.2, it is obvious that the square ROM, i.e. the 8 x 8 ROM will be more area efficient since it requires less devices in the decoder network and also requires less address lines routed to the ROM. It can be shown that when decomposing a ROM to two dimensions using normal decoding techniques the ROM which is closest to being square will always achieve the most area efficient design.

Table 3.2: Possible Two Dimensional ROM Arrangement			
	Possible Arrangement	Number of Address Lines	Number of Decoder Devices *
1	64 x 1	64	384
2	32 x 2	34	192
3	16 x 4	20	108
4	8 x 8	16	96

\* Based on static decoder implementation using 2 to 4 decoders and AND gates to achieve the desired decoder functionality.

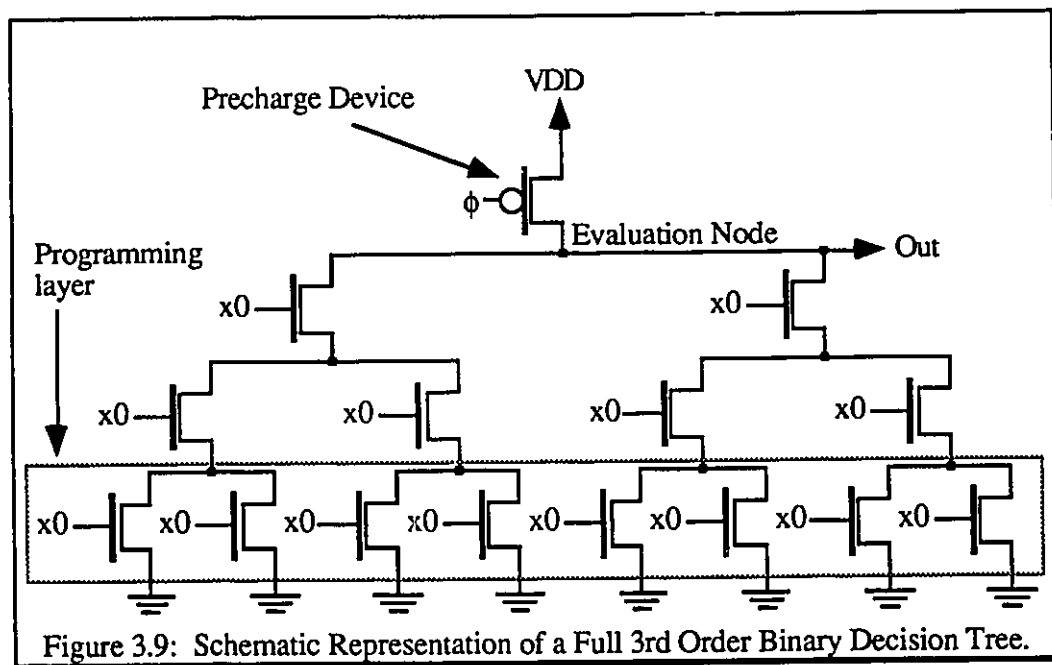
The next logical step is to then take a look at decomposing the ROM into all possible n space, while attempting to keep the ROM sections as square as possible. Possibly a four dimensional ROM structure may be more efficient than a two dimensional one? Thus table 3.3 illustrates the decomposition of an n = 6 ROM. From this table it appears that a 6th dimensional ROM structure is more efficient than any other method of decomposition. It might also be important to note that the 6th dimensional ROM essentially comprises a binary decision tree and the decoder network is nothing more than 6 inverters.

Table 3.3: Possible N Dimensional ROM Decomposition					
	Possible Arrangement	Number of Address Lines	Number of Decoder Devices *	Number of ROM Devices **	Total Device Count
1	8 x 8	16	96	64	160
2	8 x 4 x 2	14	86	64	150
3	8 x 2 x 2 x 2	14	54	112	166
4	4 x 4 x 4	12	84	64	148
5	4 x 4 x 2 x 2	12	52	96	148
6	4 x 2 x 2 x 2 x 2	12	20	120	140
7	2 x 2 x 2 x 2 x 2 x 2	12	12	126	138

\* Based on static decoder implementation as in Table 4.2.  
 \*\* Based on decomposition of ROM Structure.

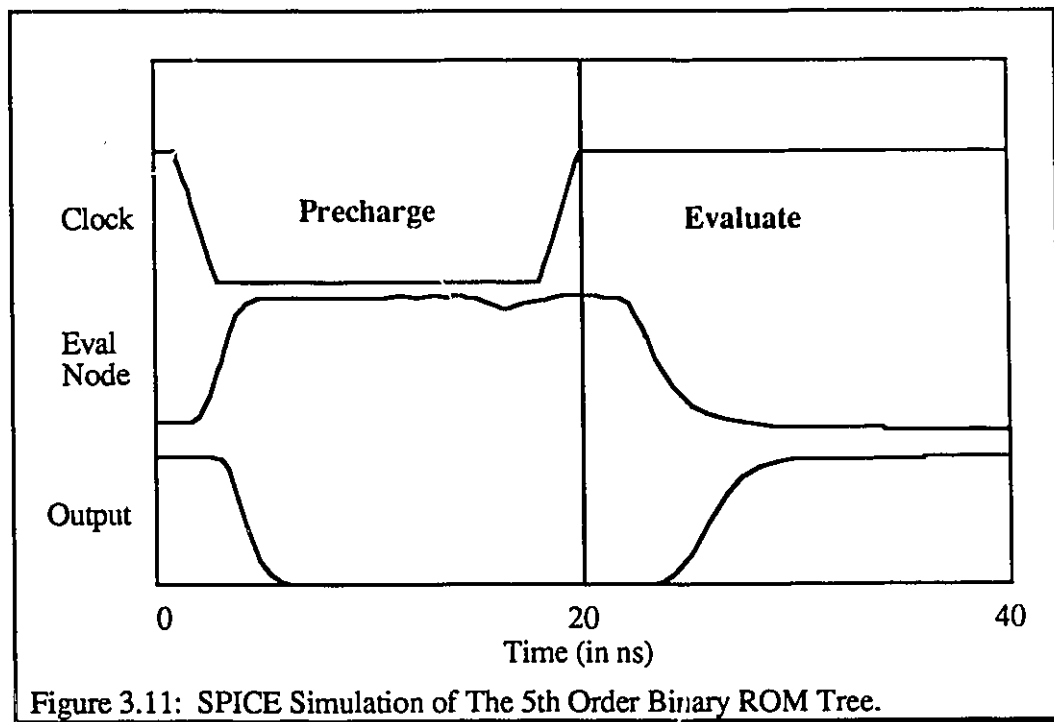
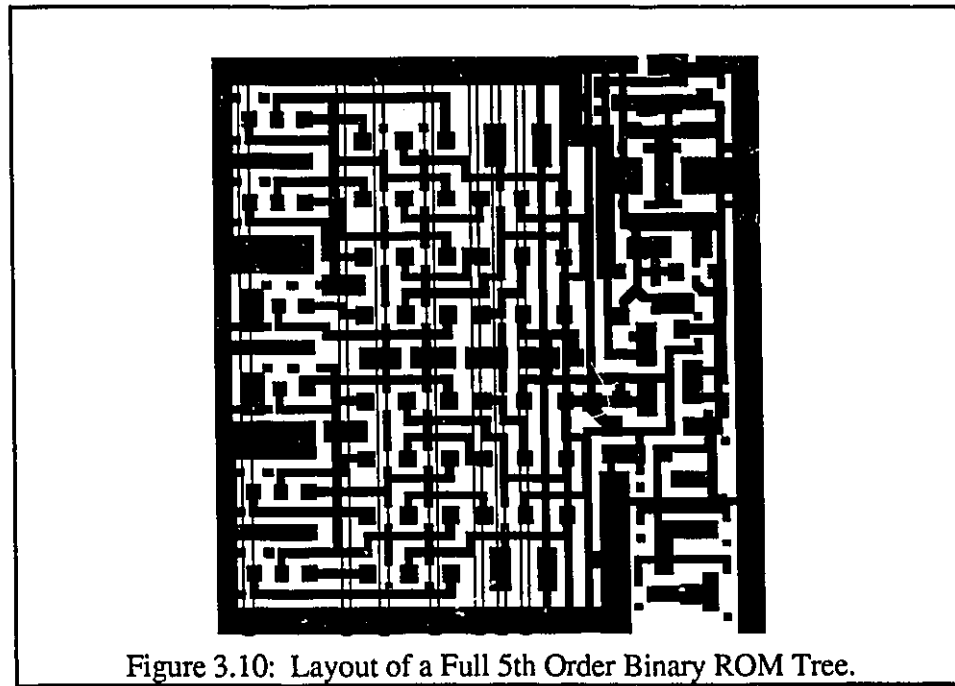
Thus from an analysis based on device count it was decided that the ROM structure could most effectively be designed in the form of a binary decision tree as illustrated in Figure 3.9, where the last level of ROM devices essentially comprises the programming

section. After the initial layout of a fifth order ROM tree it was found that the regularity of the ROM Tree's structure made the layout of the ROM very area efficient. A full fifth order binary ROM tree cell is illustrated in Figure 3.10. The cell consists of 62 NMOS devices and utilizes a complete diffusion guard ring around the perimeter of the cell. The cell only occupies 40,000 square microns in size, compared to 80,000 square microns/output bit in the ROM5 cell. Initial SPICE simulations indicate that a fifth order tree is functional and can operate at speeds up to 50 Mhz, as shown in Figure 3.11.



These speeds can probably be further increased if device scaling is used to optimize the performance of the ROM cell. The simulations also point out that charge sharing does exist within the ROM structure but does not appear to degrade the performance of trees with heights of five or less.





### 3.3.2 THE NEW CLOCKING STRATEGY, JCLOCK

The clocking strategy developed for the JROM cell was initially developed by Jullien [14] and utilizes a clocked charge pump circuit. The evaluation block, which not only performs the ROM evaluation but also has extra devices to make the cell useful for RNS applications, is illustrated in Figure 3.12. The signals  $X_{in}$  and  $X_{inbar}$  along with the Select signal are used to force the output signal to match the respective input signal associated with the ROM, for RNS purposes. The main clocking signals for the cell are the Phi and Iso or isolation clock signals. As indicated, the ROM evaluation node drives the circuitry. The two clock signals are generated locally within each ROM cell from the single global clock signal. The relationship of the clock signals is illustrated in Figure 3.13a) with the clock buffer circuit illustrated in Figure 3.13b).

The principle behind the operation of this clocking scheme is that, in a chain of ROM cells, each clock signal in the chain has a small positive delay in relationship to its parent signal. This relationship can be used to ensure that the proper presentation and evaluation of data is maintained from one cell to the next.

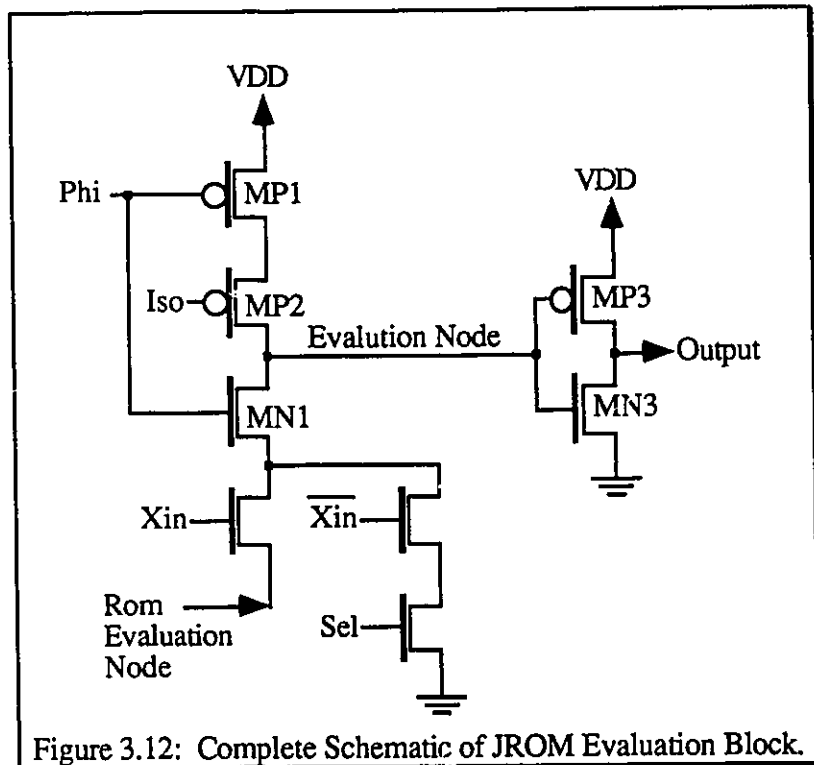


Figure 3.12: Complete Schematic of JROM Evaluation Block.

Though this idea does work in principal, further restrictions on the clock signals have turned out to make this approach unattractive.

The basic circuit functionality of the evaluation block will now be described and all references to devices and nodes will be made with respect to Figure 3.14, which represents the core of the evaluation block less the extra device used for RNS applications.

At the start of the precharge phase, location 'i' of the clock cycle, device MP1 becomes active charging up the precharge node 'Pchrg'. A few nanoseconds later, at position 'ii', devices MP2 and MN1 both turn off causing the evaluation node to be in a high impedance condition, where the past evaluation state is maintained through capacitive charge storage. As data arrives at the input to the ROM structure during the Precharge phase, the ROM will either establish or not establish a connection to ground from the evaluation node. Then at the beginning of the evaluation phase, position 'iii' of the clock cycle, device MP2 begins to become active, dumping the stored charge onto the evaluation node and creating a path through MP1 and MP2 to the Vdd rail from the evaluation node.

This lasts for a short period of time until the delayed edge of Phi reaches the circuitry at position iv of the clock cycle. At this point the ROMs isolation period is over and the evaluation phase begins. The ROM can now influence the state of the evaluation node. If there is no connection to ground through the ROM structure then the evaluation signal level will drop in accordance with the degree of charge sharing produced by the connection of the ROM to the evaluation block. On the other hand, if the ROM structure does make a connection from the evaluation node to the ground rail, then the charge will begin to be drained from the evaluation node. The

evaluation node will then be discharged after a finite period of time until position 'v' of the clock cycle is reached. This point marks the end of the clock cycle as the Iso signal

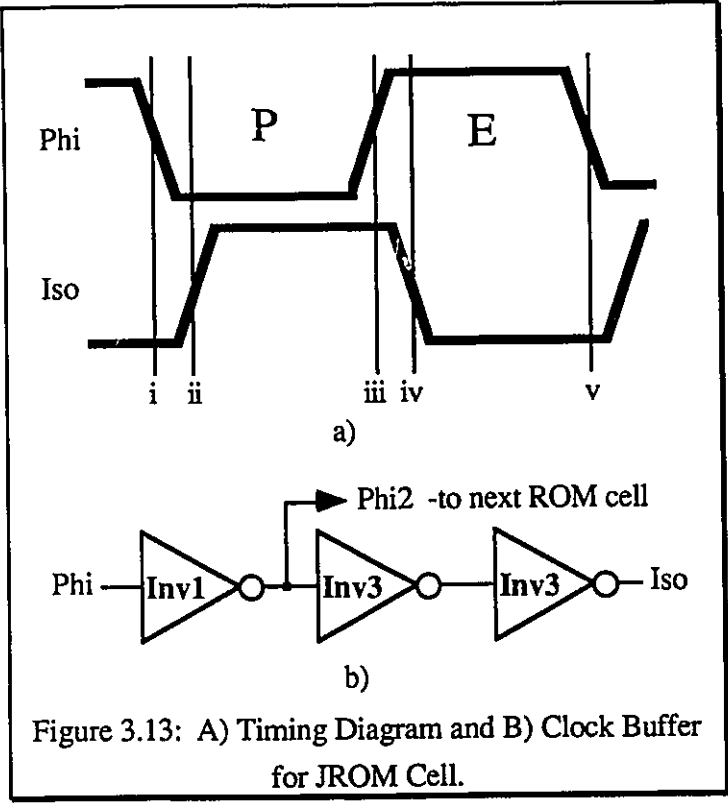


Figure 3.13: A) Timing Diagram and B) Clock Buffer for JROM Cell.

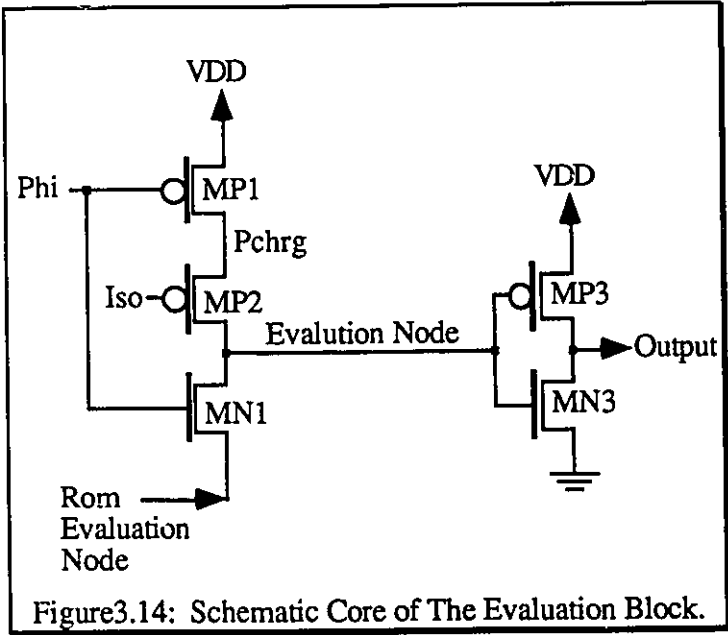
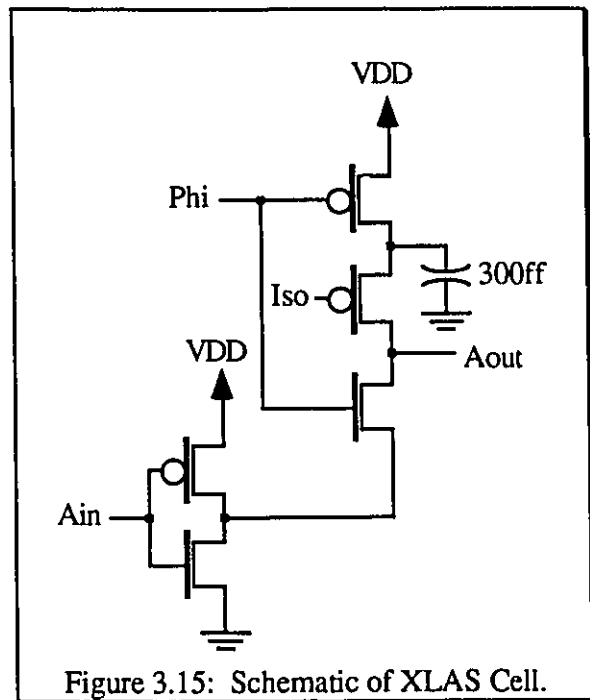


Figure 3.14: Schematic Core of The Evaluation Block.

switches which begins to isolate the evaluation node followed by the change in Phi's signal level which completes the isolation of the evaluation node and marks the start of another precharge phase.

A latch element was also designed which functioned in the same manner. The name of the latch is XLAS which is an abbreviation for X-Latch-Structure. The schematic diagram of the cell is illustrated in Figure 3.15. The latch is designed to be as small and as fast as possible with the ability to tightly pack the latches together. Thus the output inverter is removed from the evaluation block and additional capacitance is added on the precharge node to compensate for the lack of drive capability of the cell.

An extra PMOS device is added on the input, creating an inverter which drives the evaluation portion of the cell. It is found that this arrangement yields a faster evaluation time, with better voltage levels rather than just using a single NFET to drive the latch evaluation block. SPICE simulations of the XLAS cell indicate that it is capable of running in excess of 100 Mhz and has a limited loading capability of approximately 250 ff with an output voltage swing of 4.2 volts.



### 3.3.3 TEST RESULTS OF IC3WRRJT, WRXLT, WRJET

Three proof of concept chips were fabricated to verify the functionality of the new cell design. The first chip IC3WRRJT consists of individual JROM cells with the clock buffer cell removed so that both clock phases can be externally controlled. The second test

chip, IC3WRXLT, consists of different test structures utilizing the XLAS latch design. The final test chip, IC3WRJET consists of a series of five JROM cells to prove the pipelining ability of the new cell design. The summary of the chip test results is given in Tables 3.4 through 3.6.

Dev #	Maximum Clock Speed (in Mhz)	Idd @ Max. Speed (in mA)	Iddstatic (in $\mu$ A)
1	25	1.25	250
2	22	1.00	250
3	22	1.00	250
4	25	1.00	250
5	22	1.00	250

Dev #	Maximum Clock Speed (in Mhz)	Idd @ Max. Speed (in mA)	Iddstatic (in mA)	Max. Clock Speed for 10 Series Chain (in Mhz)
1	25	3.25	13.25	12.5
2	25	3.00	12.75	13.3
3	25	3.75	15.00	13.0
4	25	3.00	13.00	12.5
5	25	3.00	11.00	12.5

Dev #	Maximum Clock Speed (in Mhz)	Idd @ Max. Speed (in mA)	Iddstatic (in mA)
1	10	6.75	1.5
2	10	6.75	1.75
3	10	7.00	1.75
4	9.5	6.50	1.5
5	* Damaged in Shipping*	----	----

Though all devices did prove to be functional, the performance level was extremely low compared to SPICE simulation results. The test chip IC3WRRJT which contains an individual JROM cell minus the clock buffer proved to be invaluable in determining the reason for the poor performance. The test results show that under controlled clocking the JROM cell will run to the maximum speed of the tester (25 Mhz). When the cell is placed

in a series chain, the maximum clocking speed is severely degraded. The reason lies in the placement of clock edges.

During the testing of the IC3WRRJT cell the edge placement of the second clock signal Iso was varied with respect to the main clock signal, Phi, as illustrated in Figure 3.16. It was found that, for a delay of less than 2.5 nanoseconds between edges, the output low voltage period,

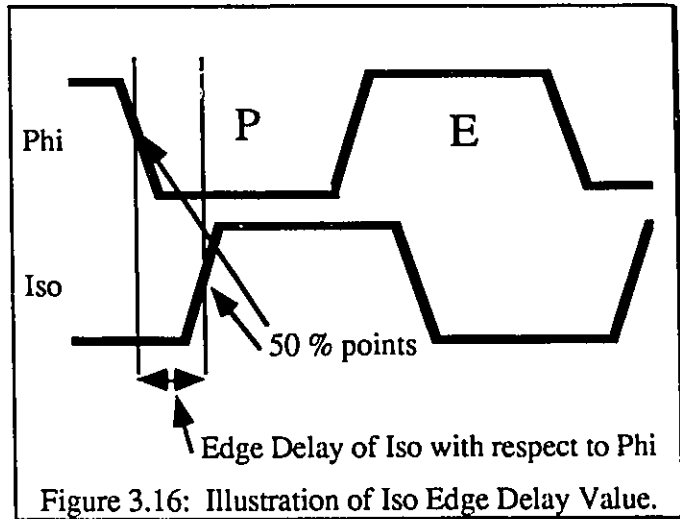


Figure 3.16: Illustration of Iso Edge Delay Value.

$V_{ol}(\Delta t)$ , decreases in width until the output fails when the edge delay drops to approximately 1.7 nanoseconds. Similarly, for edge delays of greater than 5.0 nanoseconds, the output voltage returns to zero after each evaluation phase. This indicates that the evaluation node is not fully isolating from the precharge node, which leads to the evaluation node being pulled up to  $V_{dd}$  and causing the output to go low during each precharge phase. This effectively erases the correct data value. Thus it is apparent that there exists a small window for the delay of the second local clock signal, Iso, which is 2.5 to 5.0 nanoseconds with respect to the Phi signal. It is felt that this critical time requirement may cause cells to fail due to a combination of process variations and changes in temperature or from noise injection on the Iso line. These scenarios would lead to random error patterns which could not be traced back to any particular area in the logic structure.

### **3.3.4 SUMMARY OF THE JROM DESIGN**

The development of the JROM cell proved that the concept of ROM implementation through the use of binary decision trees or Switching Trees is a viable concept. Unfortunately, the single-phase clocking method that was initially developed was not satisfactory for high speed implementation of these cells due to the strict requirement for the placement of the internal clock signals.

The switching tree ROM style is now further explored using a more satisfactory form of single-phase clocking scheme.

### **3.4 THE SWITCHING TREE CELL**

Two developments are now described that result in the final single phase pipelined switching tree architecture. The first is a minimization procedure that allows the programmed binary tree ROM to be minimized in a graph theoretic way. The second is adoption of the true single-phase clocking scheme developed by Yuan and Svensson [7].

In order to minimize the tree a rule set was developed by the VLSI Research group [ref]. In a companion thesis to this, the rule set is simplified and used in a heuristic software package to automatically generate minimized and merged switching tree architectures [ref]. The optimum solution is based not only on a graph theoretic rule set but also rules which govern the layout of the ROM cells themselves based on circuit theory and lay-out considerations. For a complete listing of the rule set and the method for decomposing the ROM cells through the use of a software heuristic algorithm see [8].

After the application of ROM decomposition rules it is found that most ROMs experienced a decrease in transistor count by as much as 90%, with an average decrease of about 30%. This decrease in ROM devices produces ROMs which have the same



functionality as the original but can both be built on a smaller silicon area and operate at faster speeds due to decreased parasitic capacitances. These advantages make the idea of implementing standard boolean logic through the use of decomposed ROM structures very appealing.

The adoption of the single-phase clocking technique [7], seems to offer a matched pipelining technique to the switching tree complex blocks. Several different styles of pipeline latches were investigated and preliminary simulations of the latches confirmed the results presented in the publication.

### **3.4.1 TRUE SINGLE-PHASE CLOCKING (TSPC)**

The standard method of clocking large pipelined systems has been through the use of multi-phase clocking schemes. Inherent in all these schemes is the need to pass two or more clock signals to all the cells in the pipelined array. This physical requirement introduces the problem of clock skew, which limits the size and speed of large circuits by causing latches to become transparent causing circuit failure. As the area of the chip increases and the line widths of the metal paths decrease, the net effect causes increased variations in signal propagation. Variations in the fabrication process across the chip causes clock distribution elements to have different gate delays. This effect is increased as minimum geometric feature size decreases. Thus it becomes more difficult to maintain relative circuit speeds for large designs on smaller technologies while utilizing multi-phase clocking schemes, unless great care is taken to minimize the problem.

This problem does have one solution; The use of true single phase clocking schemes. A true single phase clocking scheme (TSPC) is one which requires only one clock signal to be distributed across the chip which passes from cell to cell boundary. The TSPC design allows all latch and evaluate logic blocks to operate with only one clock signal, and eliminates the generation of local compliment signals or alternative phases.

Such a system is not as sensitive to process variations, except for clock delay problems, since there is no mechanism for latch elements to become transparent and fail. This type of a system is theoretically capable of higher clock frequencies at the large system level.

### 3.4.2 SVENSSON'S TRUE SINGLE PHASE CLOCKING

All TSPC techniques are essentially defined by the circuitry they employ. Most techniques require the use of alternating P and N-type latch or logic blocks to remove possible race conditions by only allowing alternative elements to evaluate while the other elements are in a latched or precharge mode. This concept is illustrated in Figure 3.17.

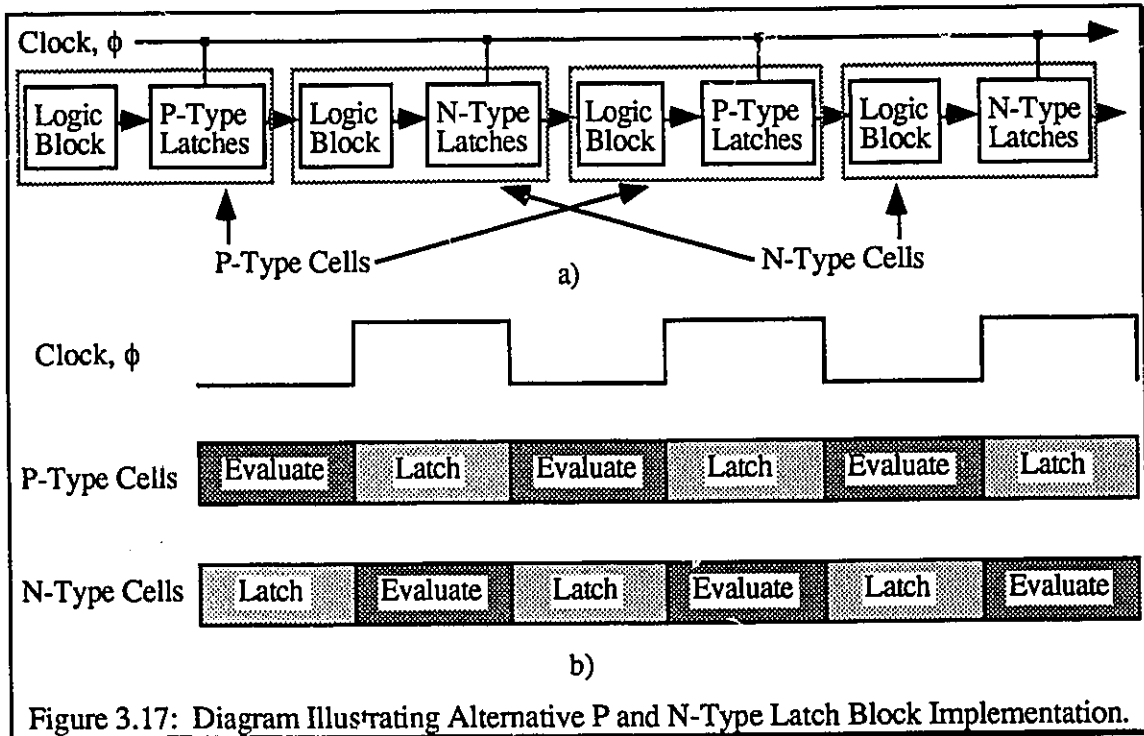


Figure 3.17: Diagram Illustrating Alternative P and N-Type Latch Block Implementation.

The pipeline illustrated in Fig. 3.17a) is composed of logic separated by alternating P and N-type latch elements. By definition the P-type latch element is composed mainly of PMOS devices which implies the latch data can only be changed during the negative portion of the clock signal, and is latched during the positive portion of the clock signal. Similarly, the N-type latch is constructed out of mainly NMOS devices which is latched during the

negative portion of the clock signal, and can be written to during the positive portion of the clock signal. Thus, as can be seen in Fig. 3.17b), all odd cells will evaluate during one-half of the clock cycle while the even cells are latched, then the even cells will evaluate during the second-half of the clock cycle while the odd cells are latched.

For these types of systems the only clock problems which can affect the performance of the system that employ them is clock slope sensitivity and clock delay. Clock slope sensitivity refers to the ability of the pipelining system to continue to properly control the flow of data through the pipeline under the conditions of slow rising and falling clock edges. This is because the clock transitions control the critical flow of data which can be disrupted if the clock edges become comparable to the gate delay of the logic blocks.

In the case of Svensson's TSPC type one circuit (TSPC-1) and type two circuits (TSPC-2), illustrated in Figures 3.18 and 3.19, they can function with slope edges of 20ns and 2ms respectively, which is, at a

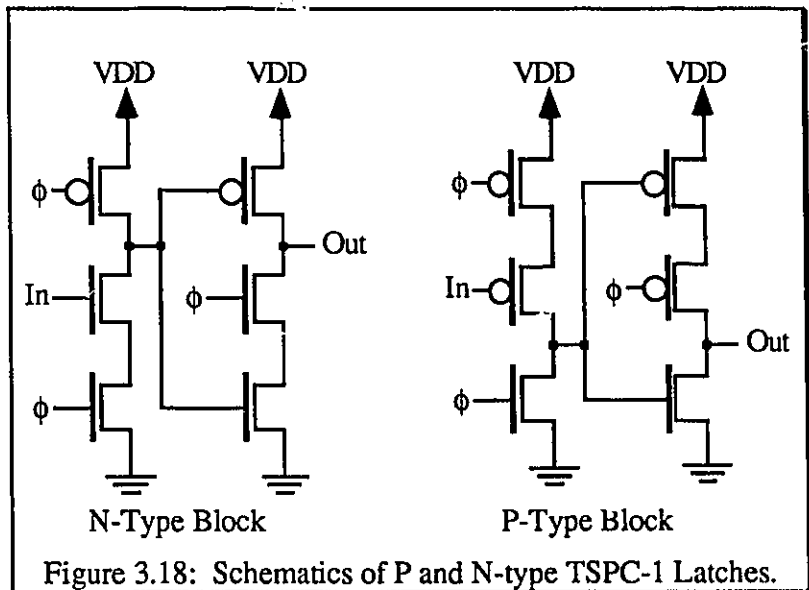


Figure 3.18: Schematics of P and N-type TSPC-1 Latches.

minimum, an order of magnitude greater than the gate delay of the Rom Tree logic cells.

The clock delay problem is very similar to the clock edge sensitivity problem. Clock delay problems occur when the clock delay between cells approaches, or is greater than, the gate delay of the logic blocks. Evaluation of the cells is disrupted by

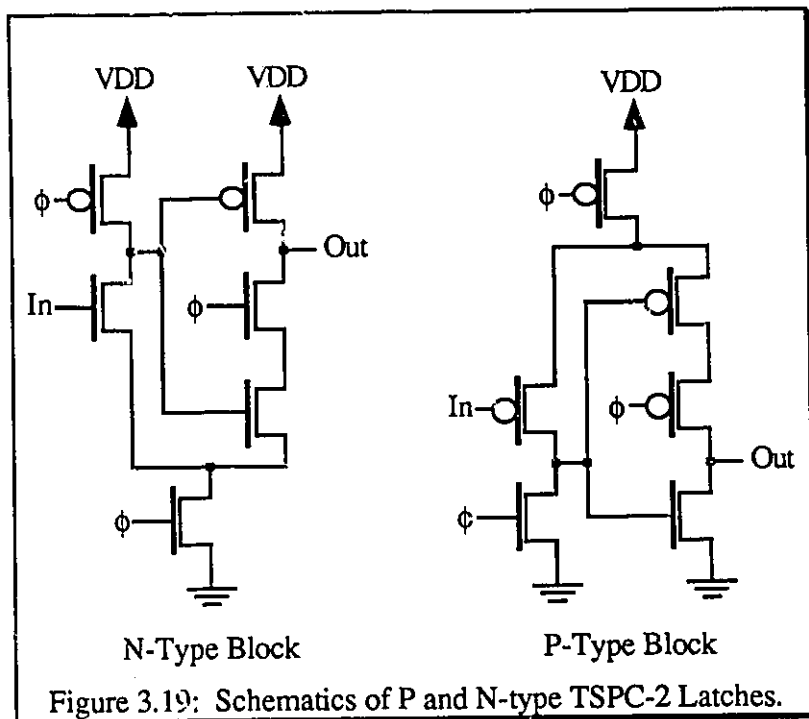


Figure 3.19: Schematics of P and N-type TSPC-2 Latches.

data changing at the end of the evaluation cycle, as shown in Fig. 3.20.

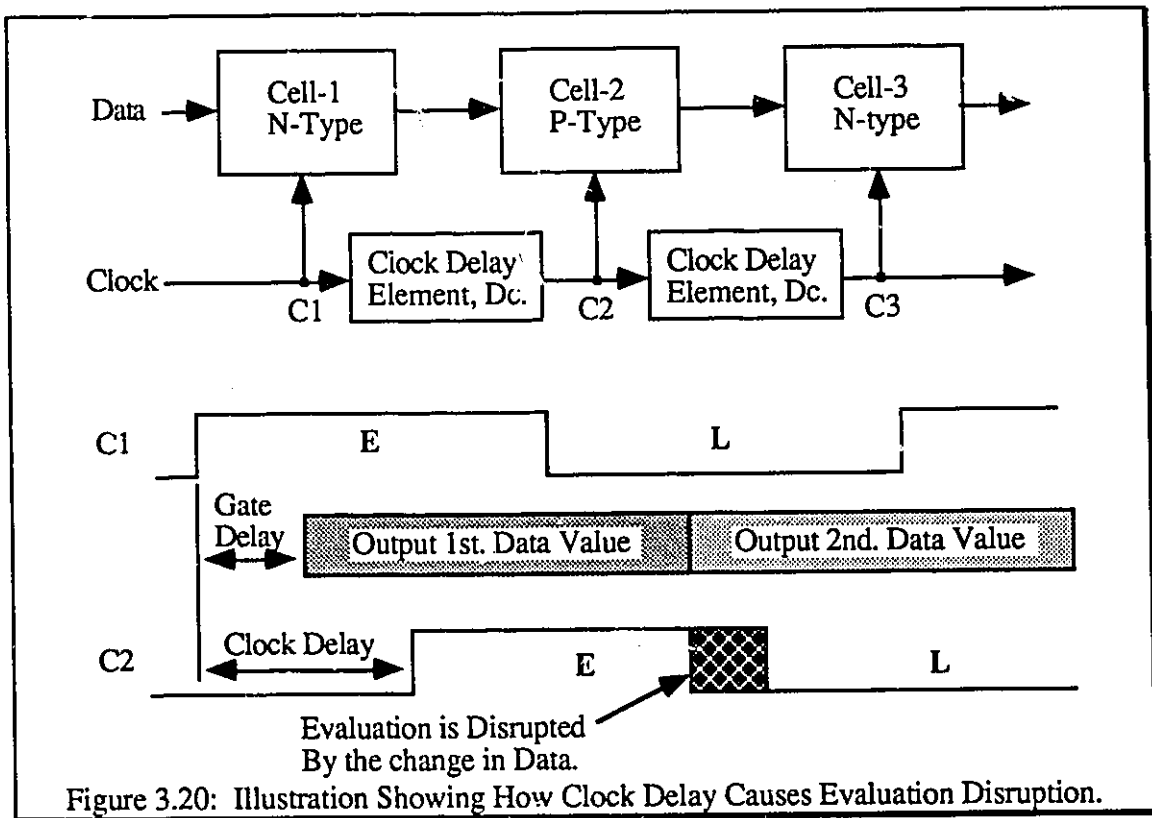


Figure 3.20: Illustration Showing How Clock Delay Causes Evaluation Disruption.

Normally this is not a problem unless the clock signal travels over a long path, possibly comprised of polysilicon sections which offer more resistance to signal transmission due to their higher per square capacitance to field oxide values. Several solutions can be found to this problem. One solution is to use a reverse clock distribution technique. Here the clock signal is distributed in the opposite direction to the flow of the data stream. In this method the evaluation of latter logic blocks will be guaranteed to fall within the stable data region of the previous logic block because the reverse distribution creates a safety margin; in essence negating the effects of clock delay, as illustrated in Figure 3.21. A second solution is to use latch components which are comprised of alternate P and N-type latch elements to create a latch which acts as a master/slave latch. Such a latch is the D-type Latch structure.

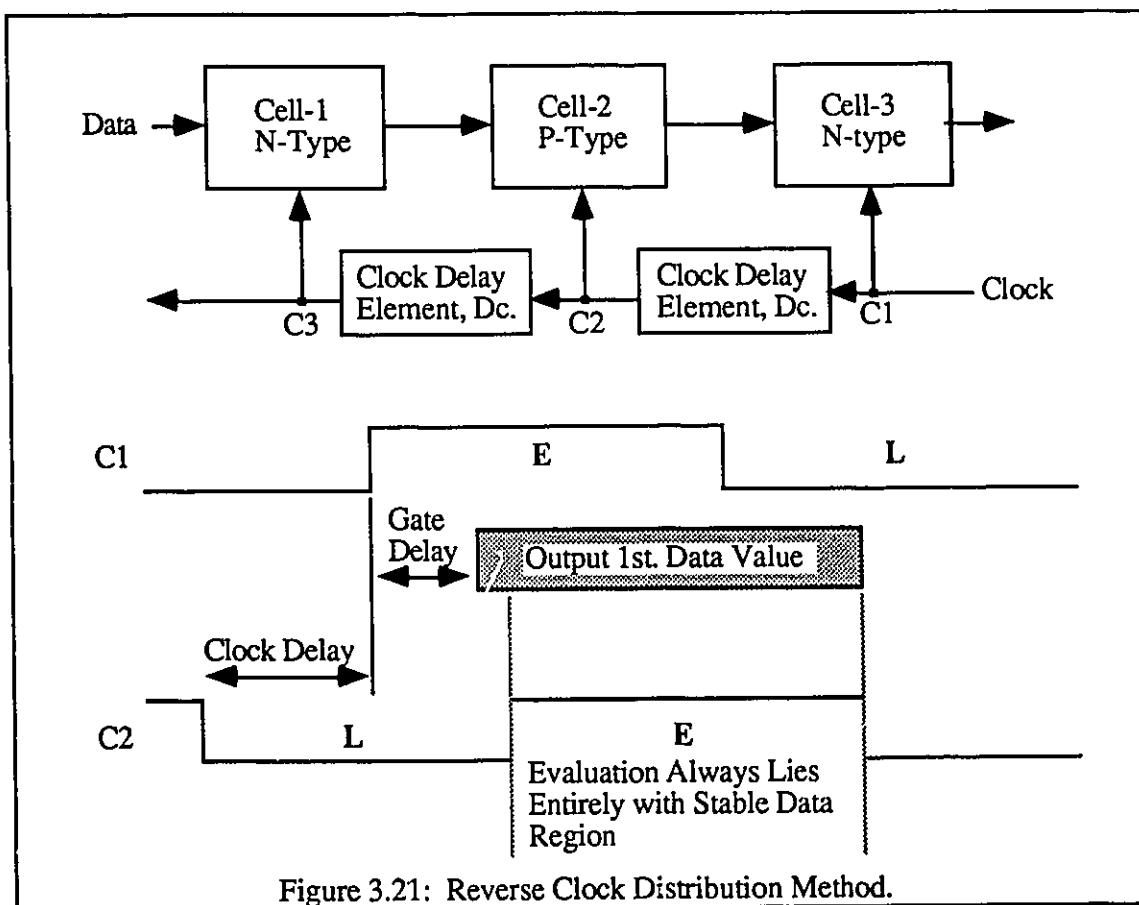


Figure 3.21: Reverse Clock Distribution Method.

### 3.4.3 NEGATIVE TRANSITION TSPC-1 D-LATCH

For the initial proof of concept standard cell design it was decided that the safest latch for implementation was the D-type Latch, which does not suffer from clock delay problems. It was felt that at testing time, if problems were to occur, the limitation of possible failure

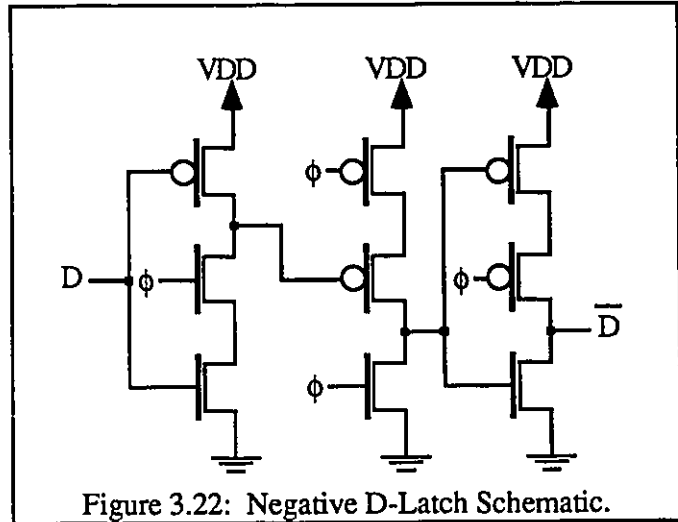


Figure 3.22: Negative D-Latch Schematic.

scenarios would be important. The schematic representation of the negative transition D-latch is illustrated in Figure 3.22. The choice of the negative transition latch over the positive transition latch was made based on the fact that the negative transition latch properly mates with the ROM implementation, requiring no clock signal inversion. The latch was simulated using a 'hand crafted' SPICE-deck; it was found to be very stable.

The devices are scaled to be able to drive a maximum output load capacitance of 250 ff with edge times no greater than 3 ns per edge. The resulting circuit is shown in Figure 3.23. A mask-

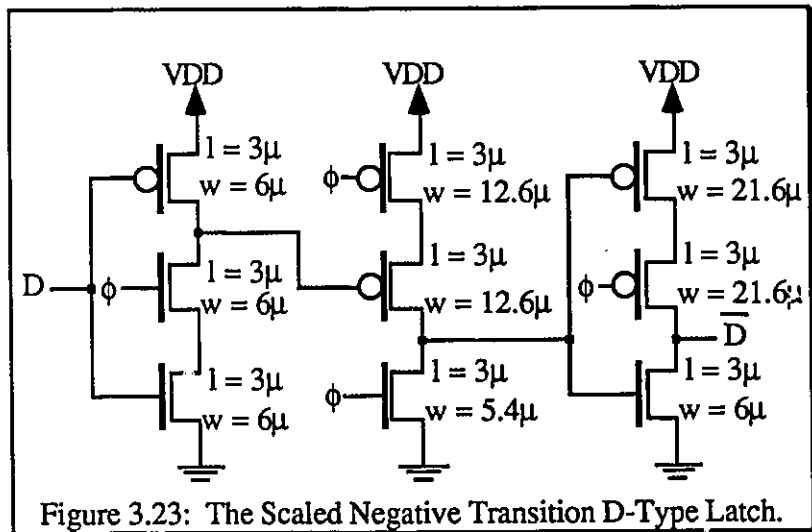


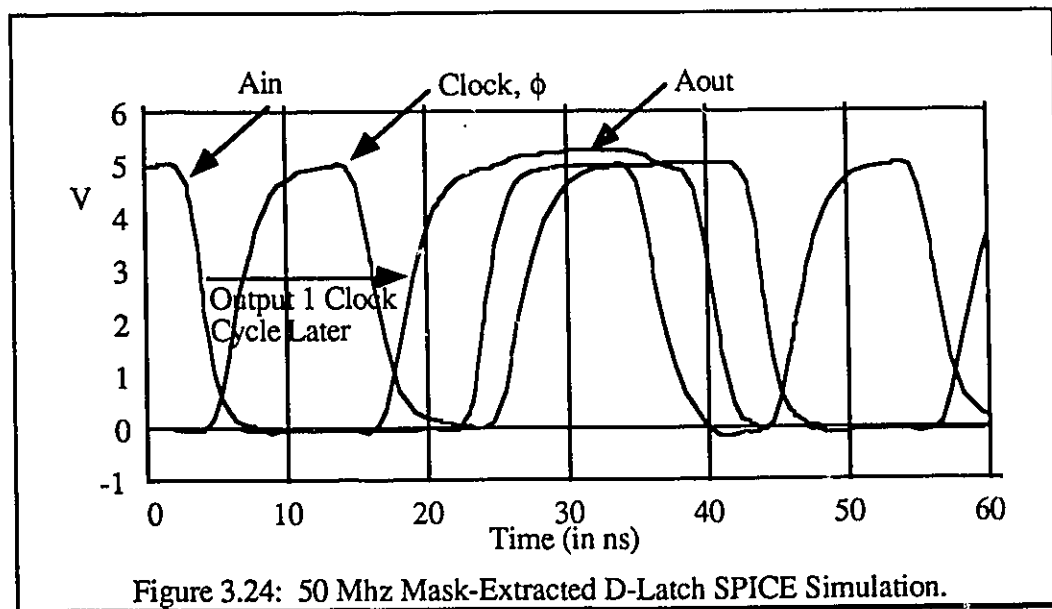
Figure 3.23: The Scaled Negative Transition D-Type Latch.

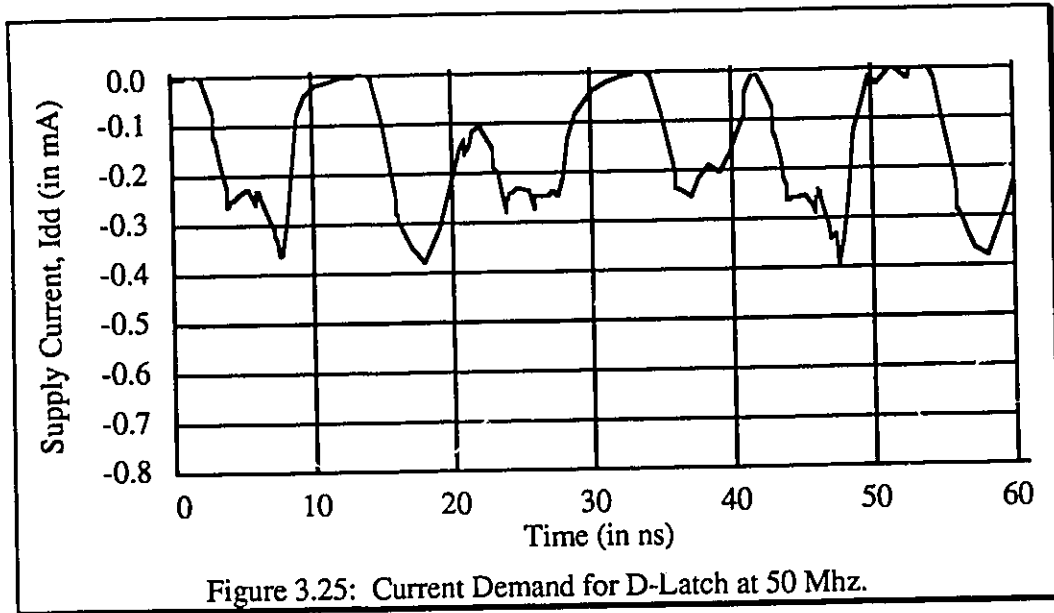
extracted SPICE deck was run to ensure that the I/O specifications were achieved. The

SPICE simulation results from the 50 Mhz run are shown in Figure 3.24, with the current waveform in Figure 3.25. The dimensions of the D-Latch are 220 x 120 design microns, and the latches are designed to be placed end to end to maximize cell packing. Also it is important to note that no output inverter is required after the D-latch since inversion of the evaluation is achieved by the D-latch itself, thus making the D-latch a more practical choice for implementation.

### 3.4.4 INPUT / OUTPUT BUFFERING

The input and output buffering of the switching tree cells is an important part of the cell design. Inadequate drive capability can cause cells to fail by generating extremely slow signal edges, which can occupy the entire clock cycle. These cells may also drive a large number of other cells depending upon the design of the circuitry, thus sufficient output buffering is essential.





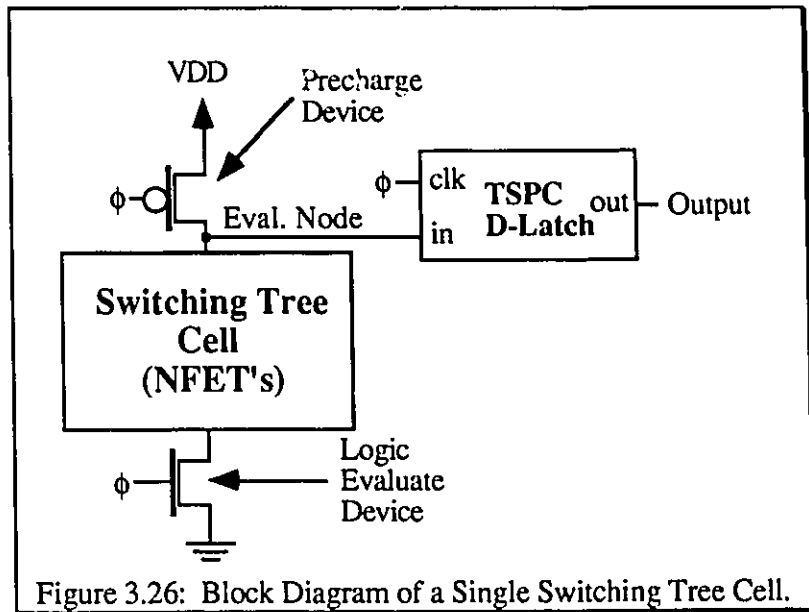
The output loading of each cell will be dependent on the particular circuit being designed; here it will be assumed that a worst case output load of 500ff will be encountered. Hence adequate buffering will be provided to drive this load with minimal signal degradation of signal edges to a maximum of 6 nanoseconds per edge.

For the preliminary proof of concept design, we place a limit of  $n = 6$  on input cell height. With this restriction it is possible to determine that the worst case input loading, due to the ROM construction is equal to one half of the total number of transistors across the bottom of the ROM times a reduction factor after decomposition, which is estimated to be approximately 0.8. Hence, the maximum number of transistors that must be driven by a given input buffer is approximately 26 devices. Thus input buffering is made adequate enough to drive the 26 gates of the ROM devices along with all the interconnect capacitance associated with those gates. A good estimate for interconnect capacitance in 3 micron CMOS technology is 10ff per 100 microns of minimum width line. The gate capacitance associated with a  $5.4\mu$  minimum length gate can be determined to be approximately 16ff. If it is assumed that the maximum interconnect length required for the ROM signals will be less than 500 microns, then a total load capacitance of  $26 \cdot 16\text{ff} + 5 \cdot 10\text{ff} = 466\text{ff}$  is found.

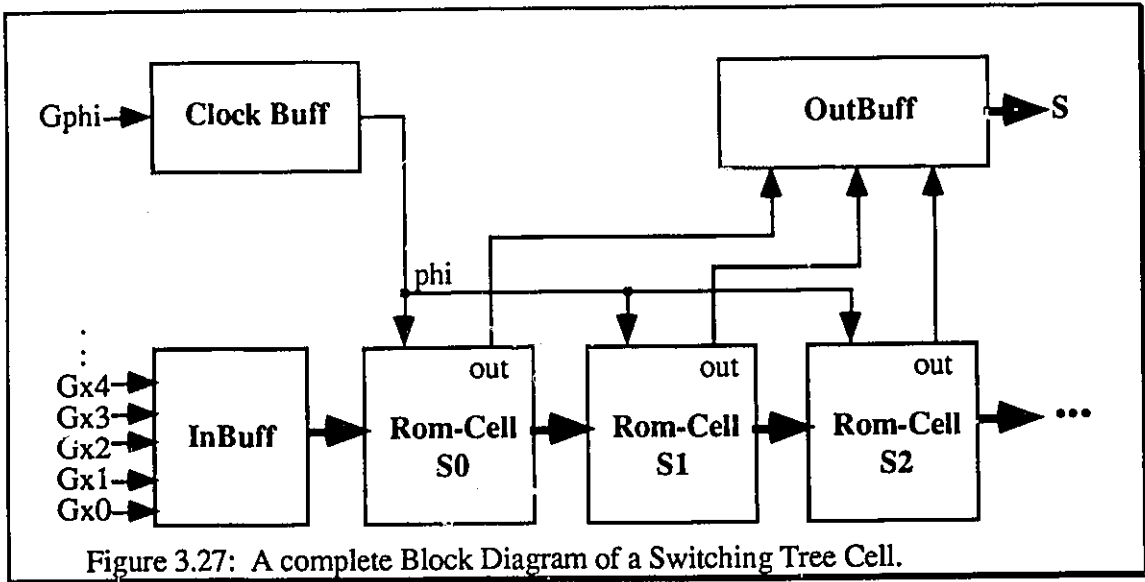


### 3.4.5 THE COMPLETE SWITCHING TREE CELL

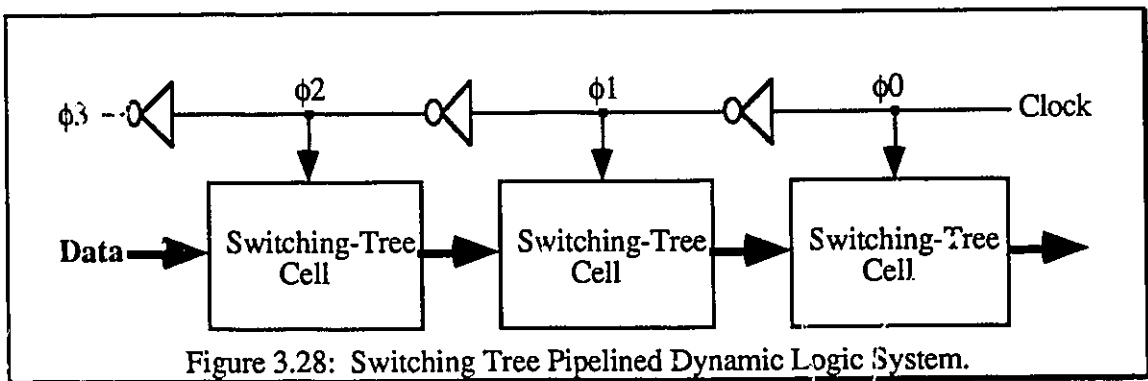
The block diagram of a single Switching-Tree ROM cell is illustrated in Figure 3.26. The cell implements standard boolean functions utilizing a Domino style logic implementation of the binary ROM tree whose output is latched



using a dynamic TSPC D-latch. All ROM inputs are shared amongst the ROM cells of the switching tree cell, requiring only local buffering and inversion of the input signals to drive the ROM cells. This design allows the use of just a single clocking signal to control the precharge and evaluation of the ROM in conjunction with the latching function. A complete switching tree cell is illustrated in Figure 3.27. All components of the switching tree cell essentially remain the same for fixed input and output field sizes, except for the contents of the ROM cells.



A general floor plan, or footprint, of a switching tree cell can be assembled. A finished cell is obtained by programming the ROM contents. This strategy allows for large series chains to be constructed where the clock signal requires a simple inversion between each cell to create alternating precharge and evaluate cells, as illustrated in Figure 3.28. It is suggested that reverse clocking techniques should be used to ensure against any clock delay problems. With this technique it should be possible to design logic systems which can function at the maximum speed of the slowest component in the pipeline.



### 3.4.6 DETERMINATION OF MAXIMUM ROM HEIGHT

It is necessary to determine the maximum ROM height,  $n$ , which can be constructed using the implementation style chosen, in order to ensure the functionality of the logic block. The maximum height of the ROMs is controlled by three factors: i) the expected clocking rate; ii) the size of the ROM devices; iii) the worst case acceptable degradation of noise margin due to charge redistribution in the ROMs during evaluation. We have set a lower bound for the maximum clock rate of any standard cell design to 25 Mhz, and, for reasons of simplification, all ROM devices are fixed in size to  $l = 3\mu$  and  $w = 5.4\mu$ , which allows for a standard gate matrix layout approach. The worst case noise margins that are acceptable are set equal to full CMOS levels less 1.5 volts on the inside of the high and low level. This implies that the worst case ROM evaluation low voltage is 1.5 volts and the worst case ROM evaluation high voltage is 3.5 volts. This allows a safety margin of approximately 0.5 volts per side, since the D-latch has been designed to record an input signal with an input swing of approximately 2.0 volts centered at 2.5 volts.

A sample set of SPICE simulations were generated in an attempt to determine the maximum tree height under the above conditions. The ROM circuit model, which simulates a full binary tree ROM structure by using an individual path, is illustrated in Figure 3.29, for  $n = 4$ . The simulation is driven to invoke the worst case charge sharing problem, which occurs when the ROM path pulls the evaluate node low on a previous evaluation by turning all ROM devices on. Device MN1, in Fig 4.29, is turned off during the next precharge phase, using the Dsw signal; this causes the entire ROM branch to remain fully discharged. In the next evaluation, all the ROM devices are turned on except MN4 which is the lowest ROM device in the tree. This will cause charge redistribution to occur as the added capacitances of the internal ROM nodes 'int1' through 'int3' are connected to the evaluation node, 'eval', which will lower the evaluation node voltage in proportion to the ratio of the capacitances.

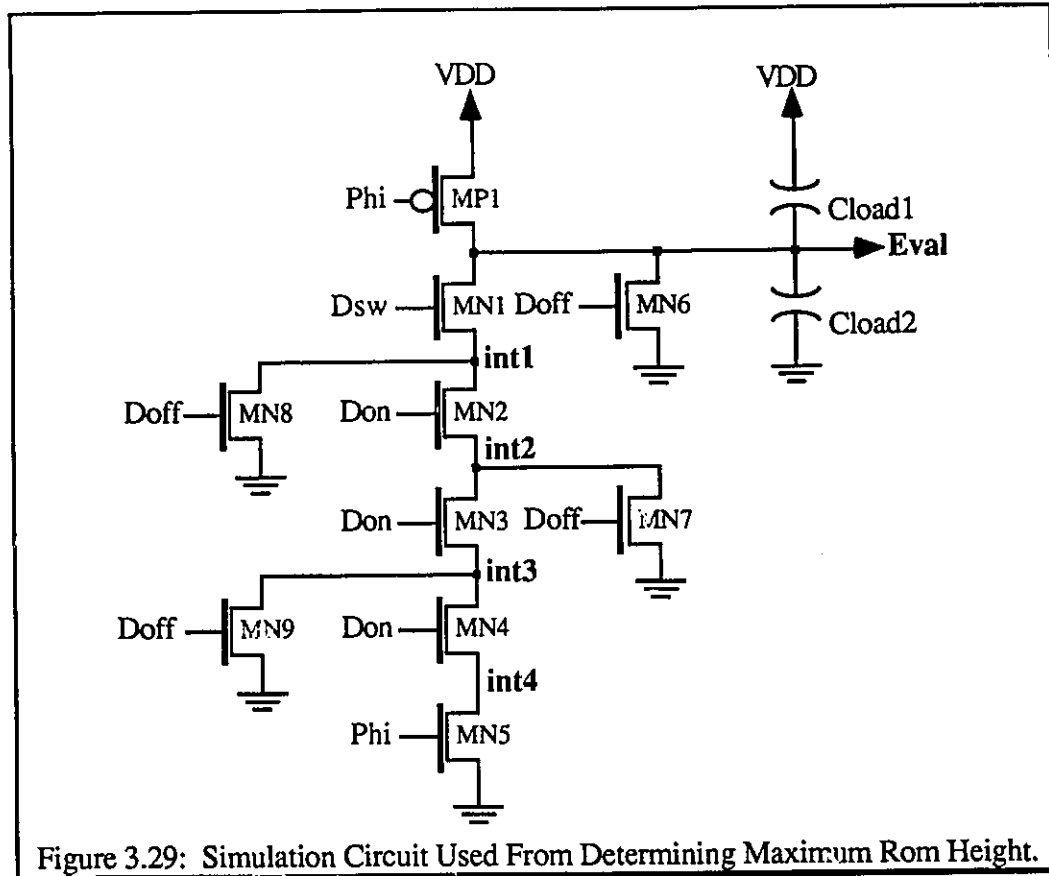
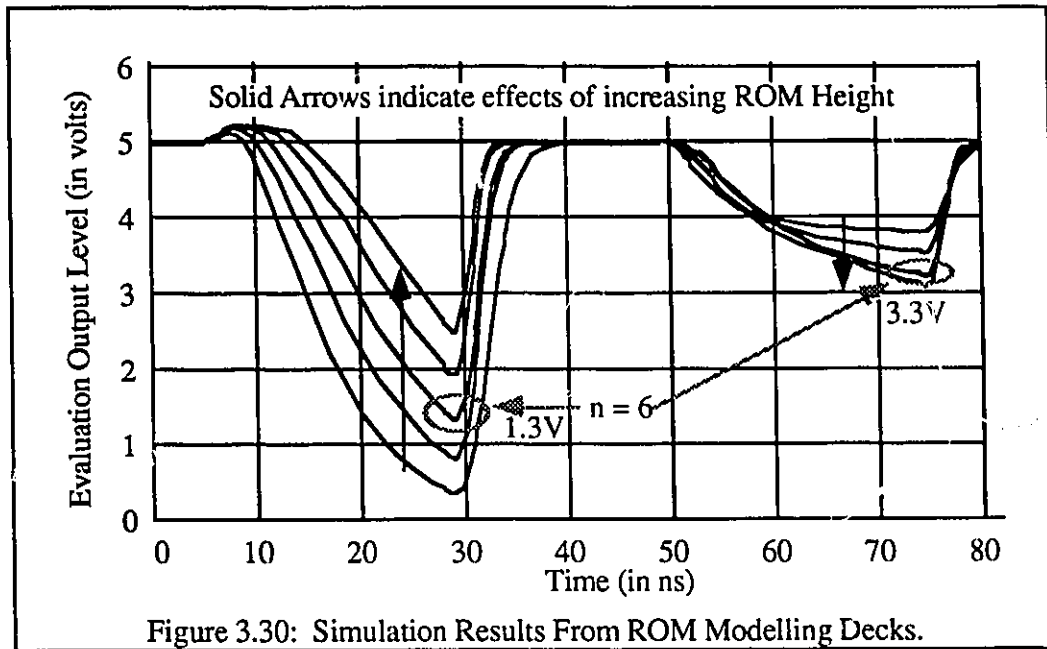


Figure 3.29: Simulation Circuit Used From Determining Maximum Rom Height.

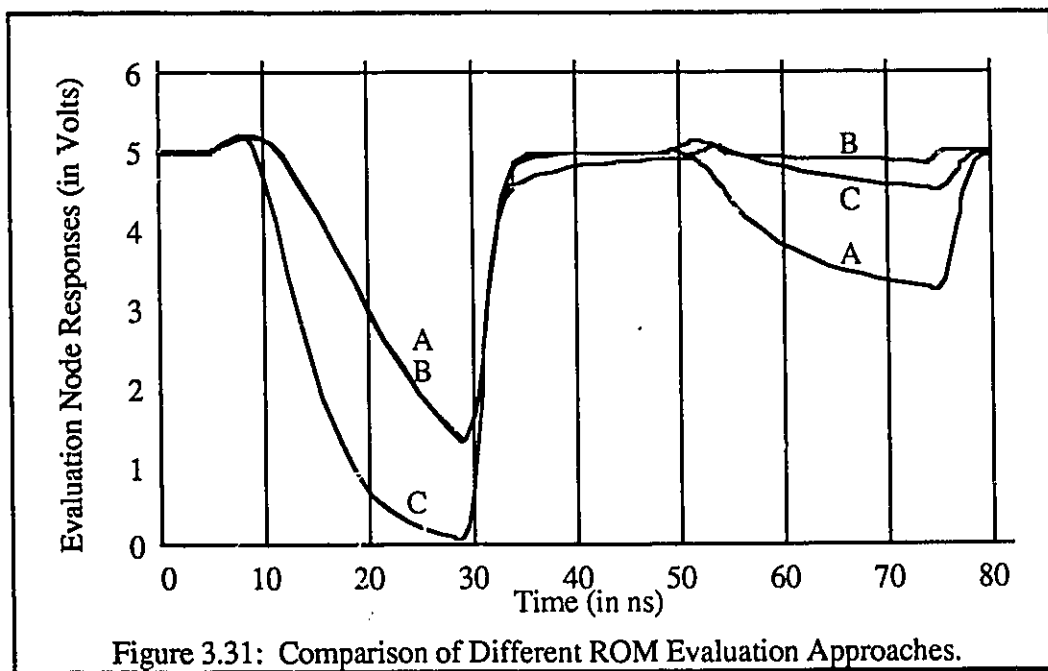
The initial simulations were 'tuned' based on many previous mask extracted simulations of height 4 trees. Extra capacitance was placed on the evaluation node as a lumped representation of the complex web of capacitances associated with the overlap of conductors and devices. A value of 150ff for both cload1 and cload2 produced evaluation signals close to the mask extracted simulations of previous tree structures. Figure 3.30 shows the evaluation node response from a set of simulations up to  $n = 8$ . As indicated on the figure, a maximum height of  $n = 6$  will still meet the specifications.

The charge sharing problem in series chains can be reduced by allowing all nodes to pre-charge, through the chain, prior to evaluation. In this way, minimal charge is bled from the evaluation node during a subsequent logic '1' evaluation. The SPICE-deck models were modified to simulate the effect of precharging the entire ROM prior to each evaluation.

The actual circuitry can easily be implemented by simply ORing all input signals with an inverted clock signal.



This will turn on all the ROM devices during the precharge cycle allowing for all the internal nodes to be charged.



The evaluation response for this type of precharged ROM is shown in Figure 3.31 as signal B, while signal A represents the evaluation of the normal circuit. As can be seen, the precharging causes increased gate delay for logic '0' evaluation while maintaining a higher logic '1' evaluation level. We can now improve the performance for the logic '0' evaluation by scaling the ROM devices [15]. The resulting evaluation node is shown as signal C in Figure 3.31. This final simulation shows that the combination of precharging the ROM cell along with the use of device size scaling produces enhanced performance results with heights of  $n = 6$  functioning at clock speeds approaching 100 Mhz. ROM cell heights as high as  $n = 10$  may be possible for clock speeds below 20 Mhz.

### **3.4.7 SUMMARY OF THE SWITCHING TREE CELL DESIGN**

The implementation of minimized binary ROM trees in Domino styled logic blocks in conjunction with true single-phase clocking techniques has produced a ROM cell capable of implementing complex logic functions in an area efficient design while performing at high throughput rates. The switching tree cells have the ability to create large pipelined logic chains without suffering excess performance degradation due to the effects of clock skew which is present in all multi-phase clocked systems. The study has shown that tree heights of up to six are possible without any optimization, while heights of up to 10 may be possible by employing device scaling and ROM precharging techniques.

## **3.5 HEURISTIC ROM CELL LAYOUT TECHNIQUES**

The ROM cells that were implemented in the standard cell designs were laid-out manually using simple heuristic techniques to minimize the size of the ROM area. The following section will discuss these heuristic procedures through the aid of a simple example. It is expected that these techniques may be transferred to a CAD tool for automatic tree synthesis, in a future research project.

The ROM cell chosen for this discussion is the 'S1' ROM cell from a 3-bit multiplier cell.

The layout procedure begins by taking the minimized switching tree cell schematic, shown in Figure 3.32, and transposing ROM branches, in order to maximize the number of adjacent devices with common gates, without creating any cross-over local interconnect problems. Looking at Figure 3.32, it can be seen that there are two devices which require the 'y1' signal and two devices which require the 'x0' signal and two

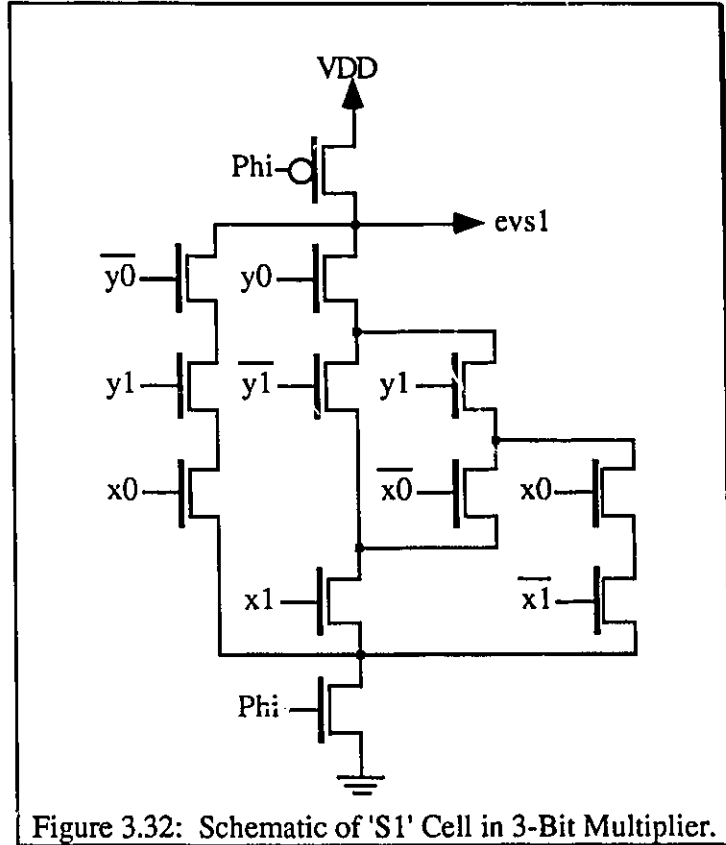
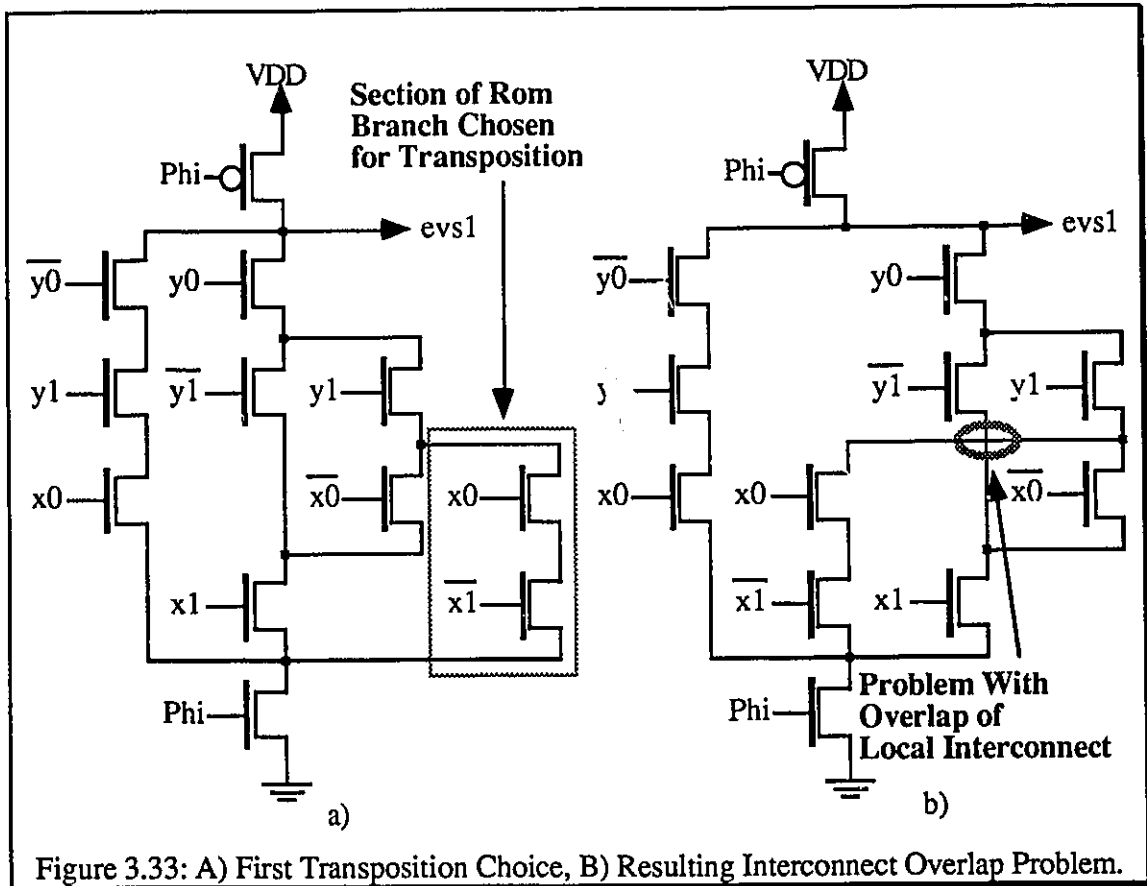


Figure 3.32: Schematic of 'S1' Cell in 3-Bit Multiplier.

devices which require the 'x0' signal. Thus we will rearrange ROM branches, or sections of ROM branches, to bring these common devices side by side without causing any cross-over of local interconnect. For example, if the section of the ROM branch containing the 'x0' and  $\overline{x1}$  inputs, shown in Figure 3.33A), were to be transposed, as shown in Figure 3.33B), the result will bring the 'x0' devices side by side, but will also create a cross-over of local interconnect. This transposition is clearly not satisfactory. If, however, the larger section of four devices, Figure 3.34A), are transposed, as shown in Figure 3.34B), then both sets of devices with common gate signals will now be adjacent, with no cross-over of local interconnect. This transposition is acceptable and will aid in compacting the layout of

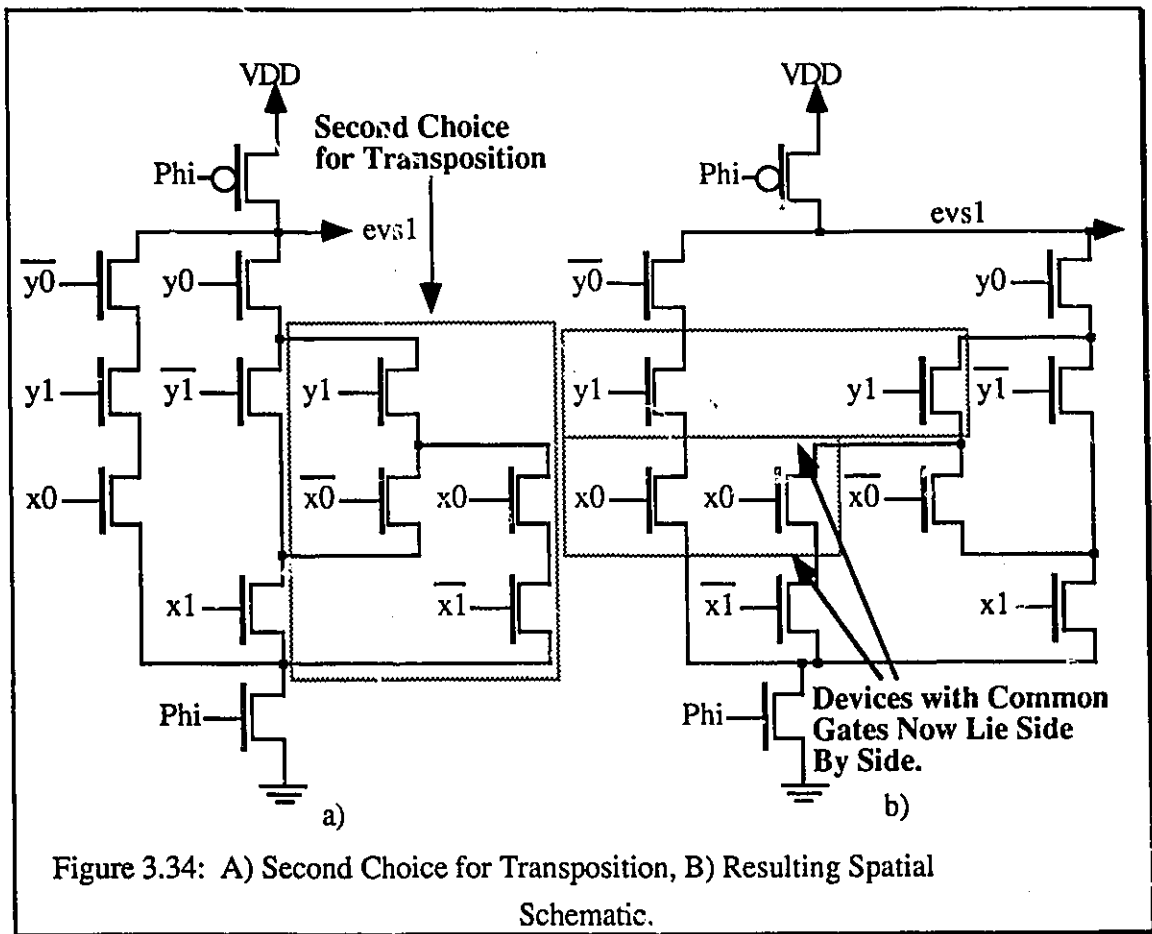
the ROM cell. If the ROM cell is more complicated, then this transposition process continues until the best rearrangement is obtained.



This pre-layout procedure attempts to reduce the number of via connections to input signal lines down to one via per input signal. Similarly, it also attempts to reduce the number of contacts used for routing the input signals down to one contact per input signal. There are three reasons for doing this. Firstly, the number of vias and contacts used to route input signals are a major cost in cell area. Minimizing these components effectively increases the compactness of the ROM cell design. Secondly, when routing the input signals, which will be travelling horizontally across the cell, they must mate with adjacent ROM cells that share the input signals. If the number of via connections to the metal2 routing layer is reduced to one via per input signal, then the job of routing the input signals



is made substantially easier since each input signal only connects to one via before passing through the cell. Obviously, this will not always be the case, but minimizing the number of vias simplifies the routing problem. Thirdly, when the devices are transposed, so that devices with common gates are adjacent to each other, then the length of polysilicon used to interconnect them is kept relatively short, which reduces both the capacitive loading on the input buffers and the series resistance, thus improving the performance of the entire cell.



A description of the device placement and routing heuristics used is now presented.

Firstly, all core ROM devices, that is the devices which form the branches of the ROM trees, act as simple pass transistors, meaning that the sources of the devices are not biased and hence these devices are susceptible to CMOS latchup [16]. For this reason all individual ROM trees are enclosed within a ring of p+ diffusion connected to the ground

rail with localized well plugs. This well plug ring serves to capture any substrate currents attempting to enter the ROM tree area, and thus removes the potential for device latchup. Secondly, all input signals to the ROM tree are routed in the metal 2 routing layer, which is reserved for this purpose. Hence all local interconnections between devices in the ROM tree will be made using the metal 1 routing layer. The layout commences with a column of devices in a strip of diffusion with gate signal connections to the immediate left of the diffusion. A second column of devices will begin in a strip of diffusion to the right of the first strip. The gate connections for the second device column will be placed to the right of the diffusion strip. The described layout pattern will be repeated with as many columns of devices as required to complete the ROM cell. This pattern is illustrated in Figure 3.35.

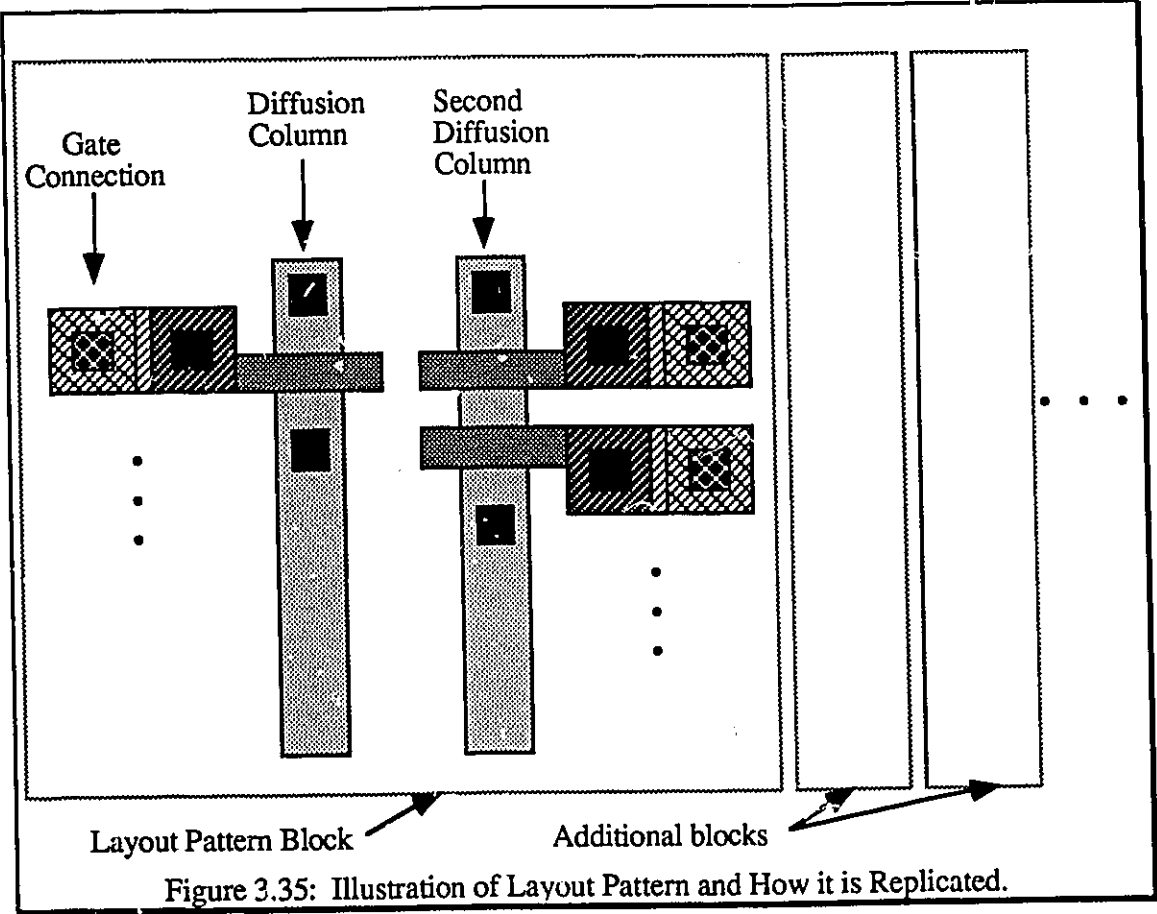


Figure 3.35: Illustration of Layout Pattern and How it is Replicated.

The layout of the ROM tree begins by taking the first device on the left most branch of the ROM tree, as illustrated in figure 3.36A), and placing that device along with the required contact and via to form the input signal routing, as illustrated in Figure 3.36B).

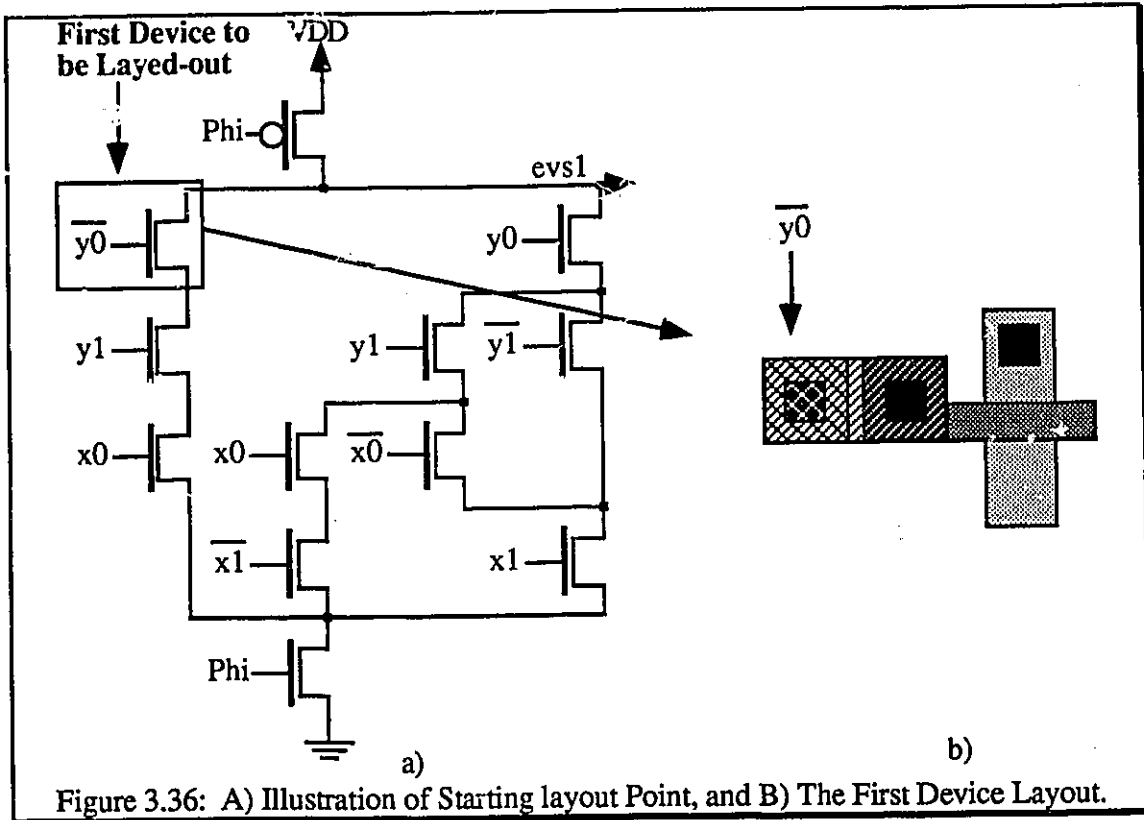


Figure 3.36: A) Illustration of Starting layout Point, and B) The First Device Layout.

Next, the second device in the left most branch is placed so that the diffusions run in a vertical column. If the drain/source connection of the two devices requires a third connection then the contact is placed between the two gates, otherwise, as is the case in this example, the two gates are placed at minimum spacing; this is illustrated in Figure 3.37. The third device in the left most branch will be layed out in

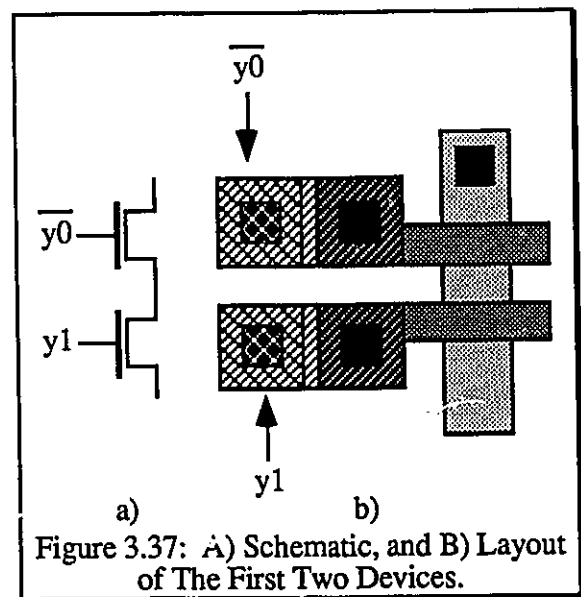


Figure 3.37: A) Schematic, and B) Layout of The First Two Devices.

the same fashion. In this case, since the three devices have no contacts in between the gates, the input signal connections will be shifted up in order to make room, as illustrated in Figure 3.38.

When the first branch is completed the second branch of the schematic placed in the second column of diffusion. The starting point for the second column is in alignment with the first column. For example, if a device in the second column of diffusion shares the same gate signal as a device in the first column, then those gates are aligned and the layout continues from that point. In this particular example the two devices with the 'x0' gate signal are aligned, as shown in Figure 3.39.

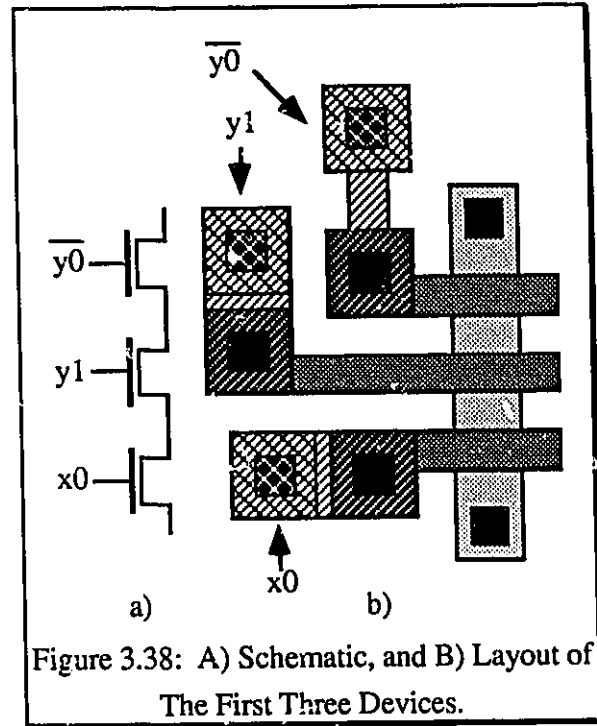


Figure 3.38: A) Schematic, and B) Layout of The First Three Devices.

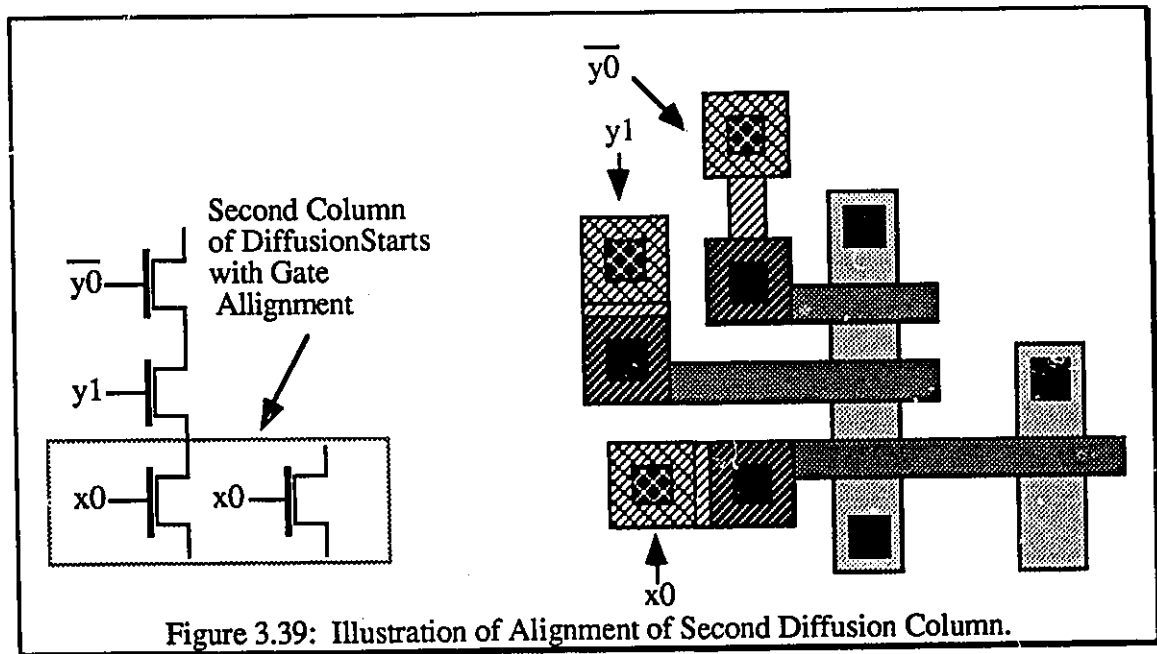


Figure 3.39: Illustration of Alignment of Second Diffusion Column.

We continue to lay-out the second column of diffusion in the same fashion as the first, except that all gates are placed to the right of the second column. The third column of the ROM cell is laid-out a similar fashion by repeating the layout pattern block; this process continues until the ROM lay-out is completed, as illustrated in Figure 3.40.

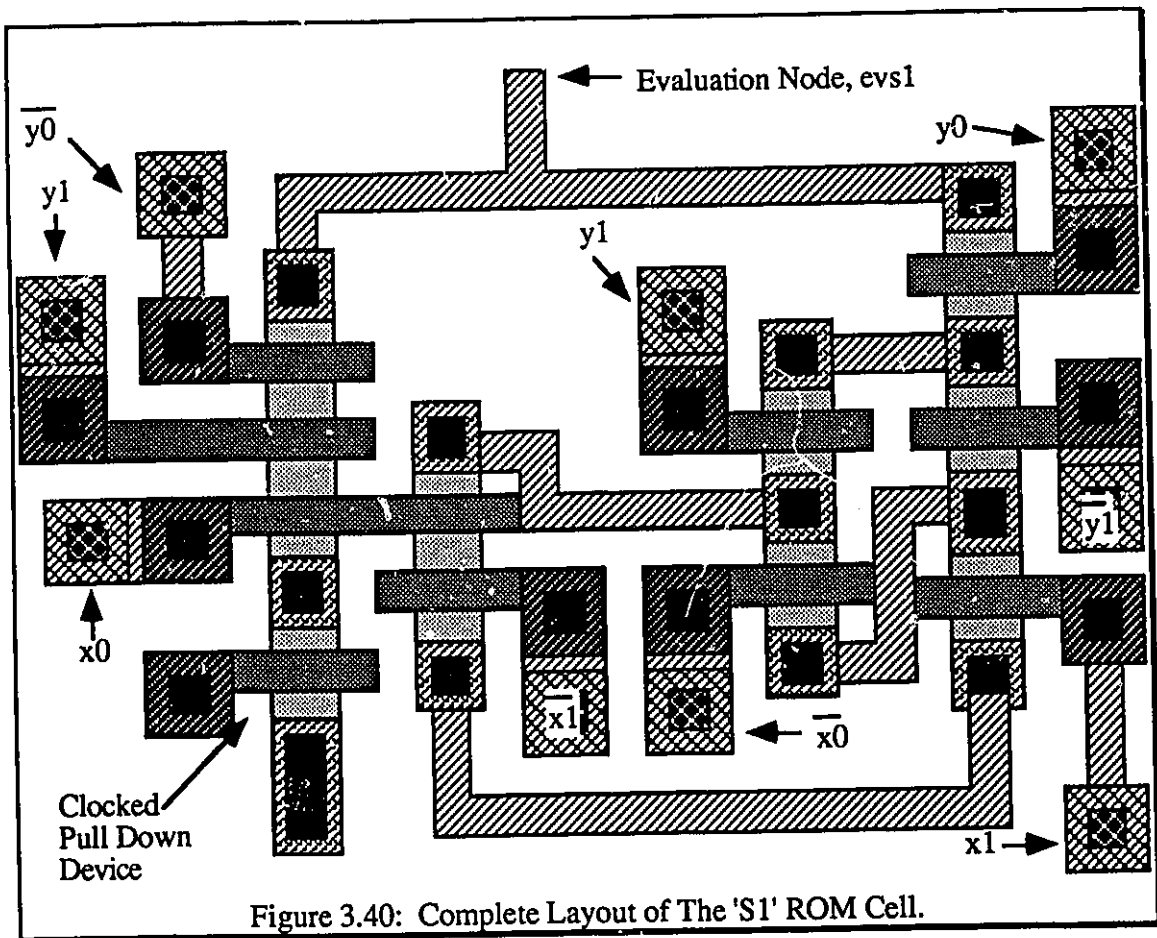


Figure 3.40: Complete Layout of The 'S1' ROM Cell.

We now surround the ROM cell with a guard ring of p+ diffusion which will be formed as an N+exclusive/P+inclusive mask and covered in metal 1 connected to the ground rail. Finally, the input signals are routed across the cell starting from the top; contacts and vias are adjusted to help complete the routing of the metal 2 layer of input signals. The signals must also be aligned so that other cells can be placed on either side. Though the author did not develop any standard method of doing this, most cells had fixed

neighbouring cells, so, in a synthesis package, any good routing scheme will be appropriate.

### **3.6 SUMMARY OF THE DEVELOPMENT WORK**

This chapter has covered the development of the switching tree cell through three versions of ROM based cells. The design and workings of each cell have been fully discussed. Several clocking techniques have also been described giving relative strengths and weaknesses. The final ROM cell design utilizes the true single-phase clocking technique of Yuan and Svensson which was found to be a very solid method of clocking linear pipelined architectures. The cell also utilizes minimize binary ROM trees, which appear to have an area advantage over other methods of implementing standard boolean logic. Also a set of heuristics has been provided for hand layout techniques of ROM cells. These heuristics may be useful in the development of a rule set for a ROM cell synthesizer package.

# CHAPTER 4

---

## STANDARD CELLS

### 4.1 INTRODUCTION

The following chapter discusses the development of several standard cell designs, constructed as proof of concept designs for the switching tree technique discussed in Chapter 3. The following cells are discussed:

- 1 A 3-bit adder
- 2 A 3-bit multiplier
- 3 A 4-bit adder

Other smaller cells have been generated such as a 4-bit comparator and a 5 to 3 compressor; the complete schematics for these designs can be found in Appendices D and E.

All the cells have been designed in the Northern Telecom 3 micron CMOS process. A test chip is discussed which was constructed using the three bit adder and multiplier cell along with the standard TSPC D-Latch cells. The design constitutes a fully pipelined 6-bit multiplier chip, which performs the multiplication through the standard sum of partial products algorithm. Finally, a set of ECL compatible I/O pads, that were developed as part of the high performance standard cell library, are presented.

One of the main problems associated with developing high-speed circuitry, is the difficulty of maintaining data rates throughout the entire system. Most high-speed systems require the need to maintain chip-level speeds off-chip, such as inter-chip communications in the case of multiple chip systems. This poses some difficulty, since the ever smaller

geometric feature size of new technologies reduces the average nodal capacitance of the circuitry, and in effect increases the available toggle-rate of the implemented circuitry. As signals are passed off-chip, to form board level communications, they encounter a large change of load capacitance. In order to maintain the same switching speeds at the output of the chip, as experienced by the internal circuitry, output drivers are required to provide large current gain.

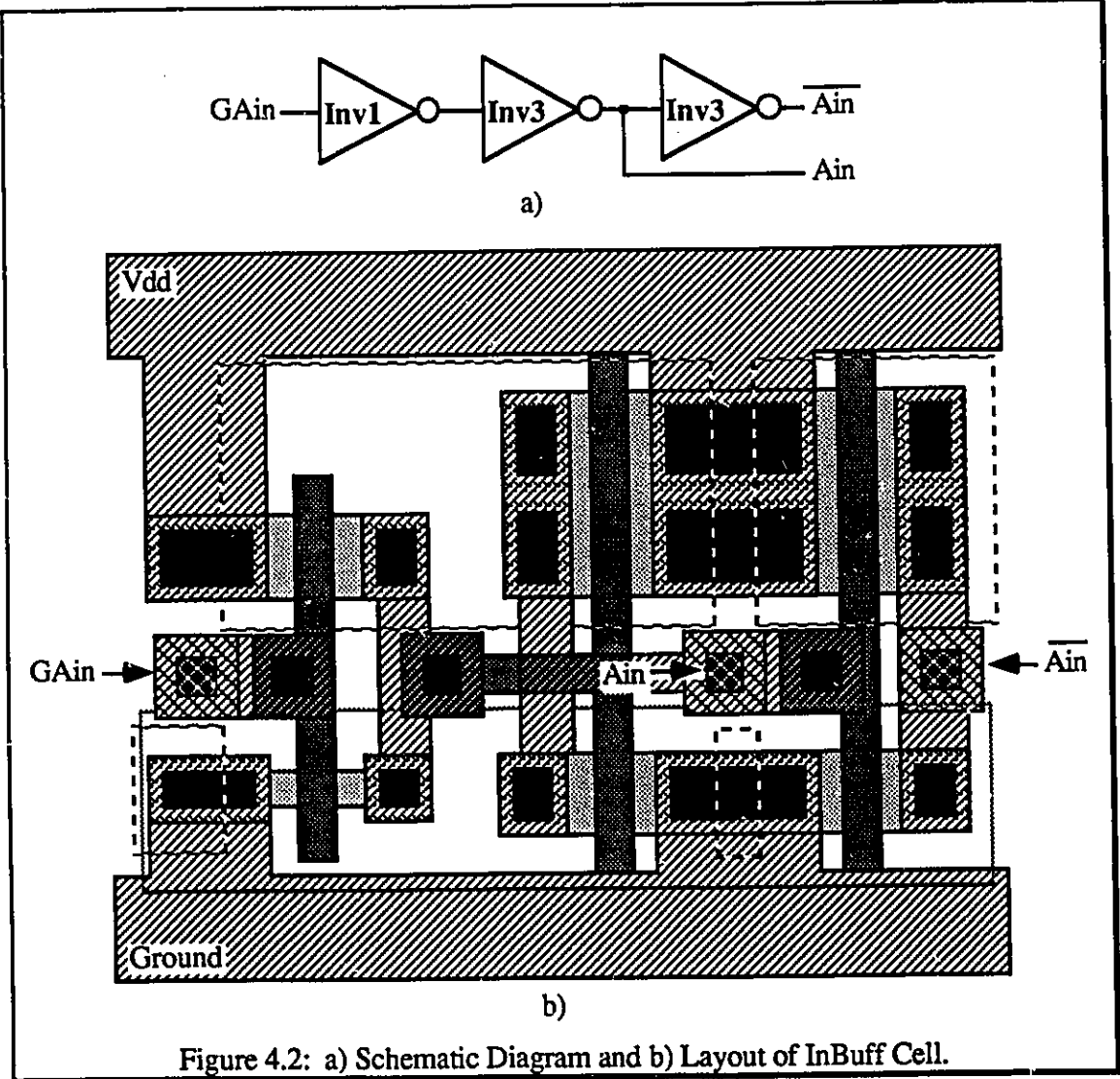
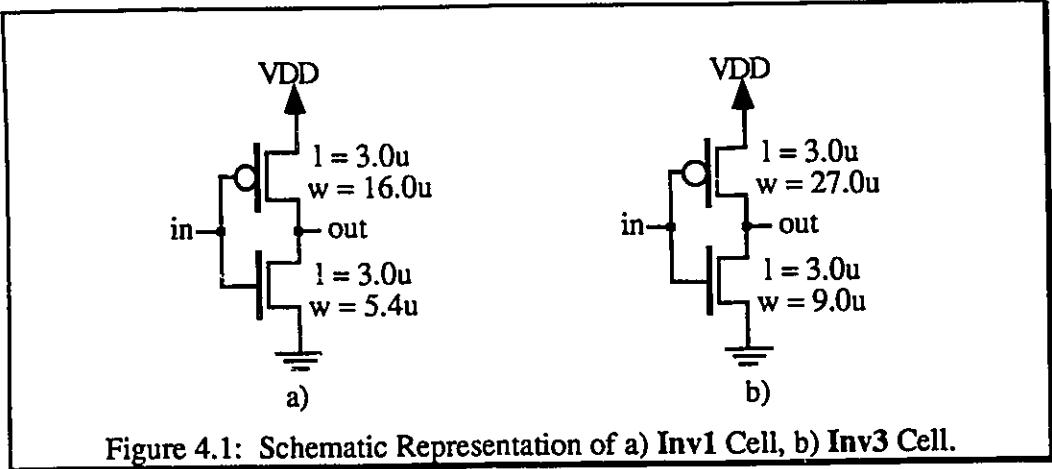
For maximum performance (switching speed) it is necessary to reduce the voltage swing of the logic levels at the chip output pins. Any reduction in voltage swing will result in a reduction in change of charge in output capacitance loads. Thus, for a given current drive, a change in logic level will be accomplished in a smaller time when the voltage swing is reduced. We discuss, in this chapter, the development of ECL (Emitter Coupled Logic) Compatible I/O pads which utilize the reduced signal swing levels of ECL technology while providing compatibility with a commercial discrete logic family. This is important for interface considerations.

As a companion design to the VLSI Research Group  $3\mu$  cell library I/O pad interface, we also present the development of a 5v swing robust output pad driver, and input protection circuit for the  $1.2\mu$  Northern Telecom CMOS technology. These are modelled after the John Carr designs in the  $3\mu$  CMOS technology [17]. Since the  $1.2\mu$  CMOS pad interface circuits are not part of the  $3\mu$  standard cell library, they are presented, separately, in Appendix F.

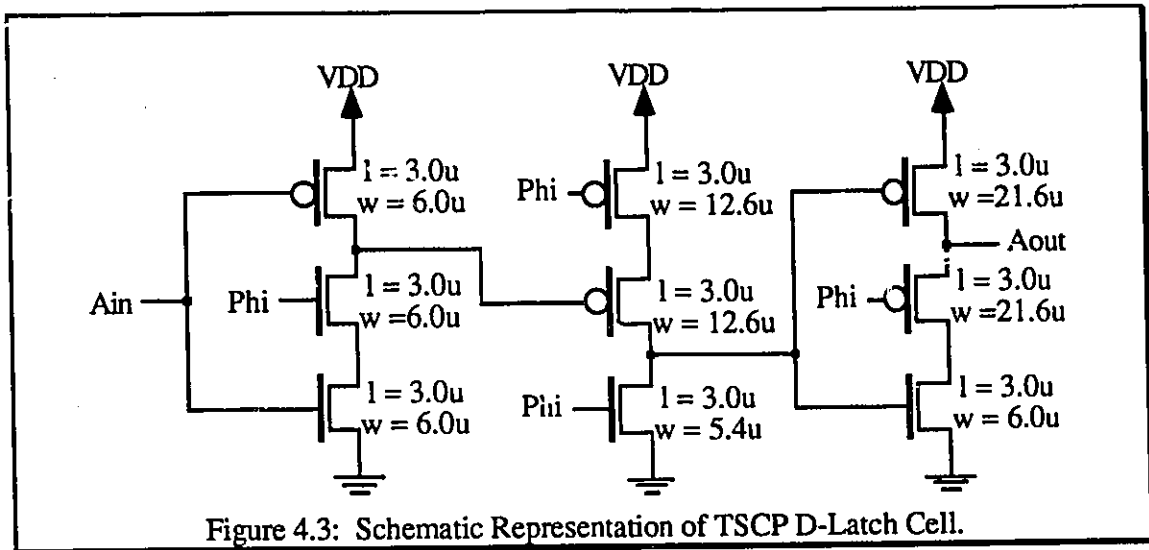
## **4.2 STANDARD SUB-CELLS**

A number of basic building-block sub-cells were designed which have been used extensively throughout the design of the standard cells. In Figure 4.1 the schematic diagrams of various inverter blocks are shown. The InBuff sub-cell comprises three inverter sub-cells as illustrated in Figure 4.2a).





The InBuff layout, illustrated in Figure 4.2b), shows how the devices are merged together in common wells using drain merging techniques to create a compact and efficient sub-cell design. The TSPC D-latch cells schematic is illustrated in Figure 4.3, while the layout is given in Figure 4.4. Note the close correspondence of device placement between the schematic diagram and the physical layout.



### 4.3 THE 3-BIT ADDER CELL

The three bit adder performs a standard binary addition. It requires two 3-bit input words plus a carry-in bit which allows the addition of larger bit fields by concatenating multiple adder cells. Thus a 6-bit addition can be performed with two cells in two clock cycles. The 3-Bit adder produces a 4-bit output field which consists of a 3-bit output word and a carry overflow-bit, Cout, as illustrated in Figure 4.5. The 3-bit addition is performed in one clock cycle. SPICE simulations indicate that the cell is capable of running at clock speeds up to 50 Mhz. The full schematic realization of the 3-bit adder can be found in Appendix G.

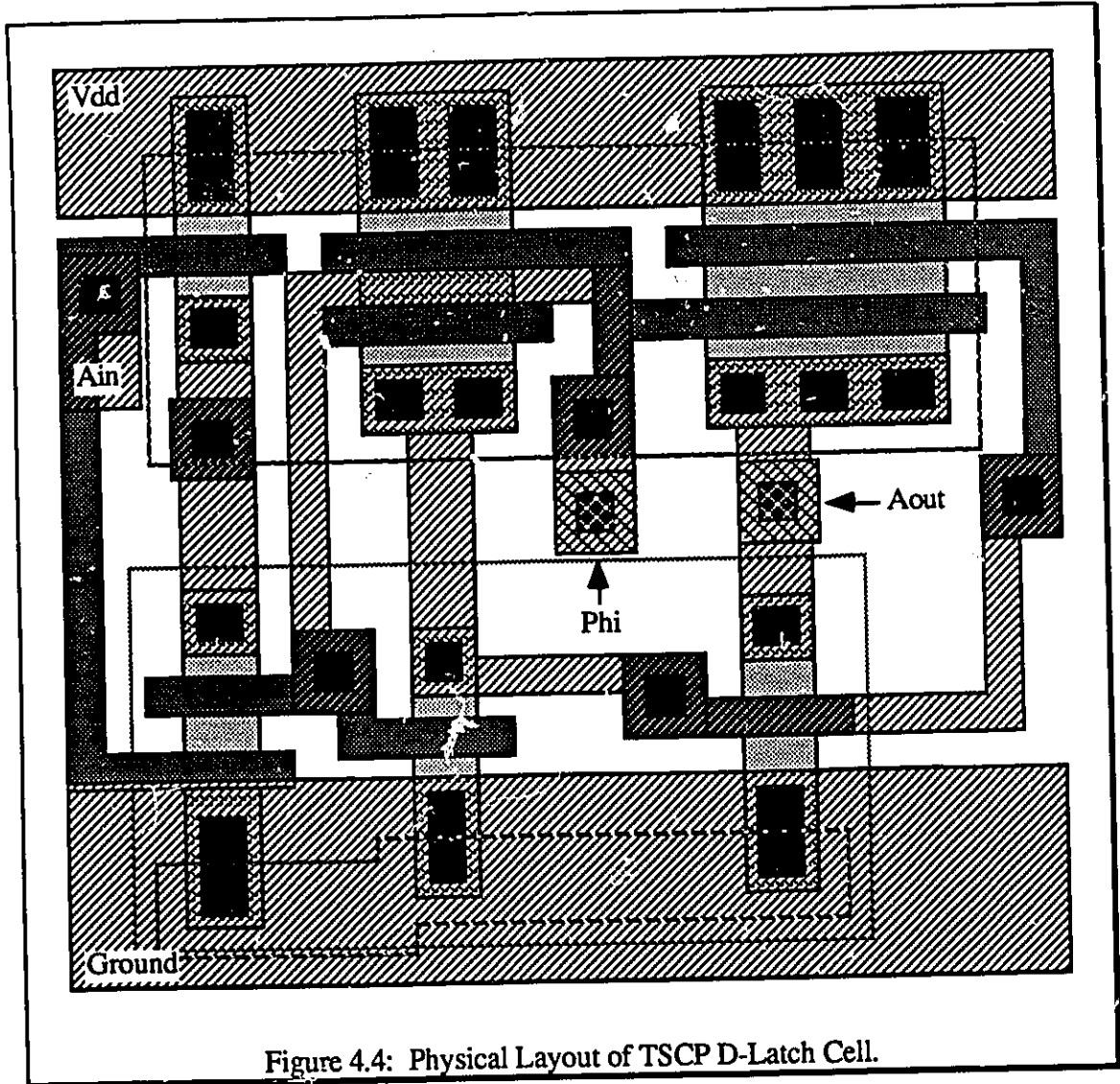


Figure 4.4: Physical Layout of TSCP D-Latch Cell.

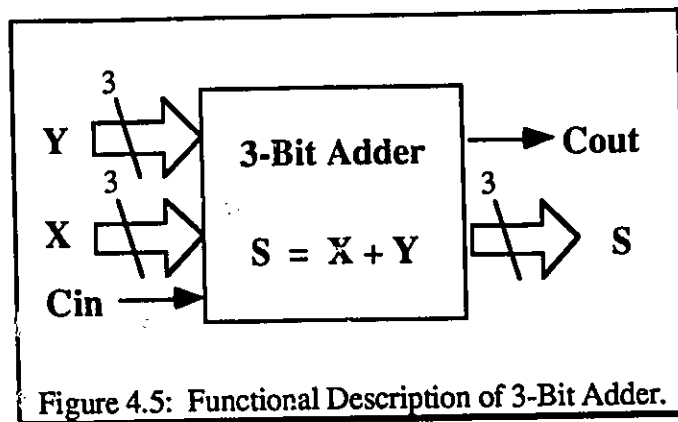
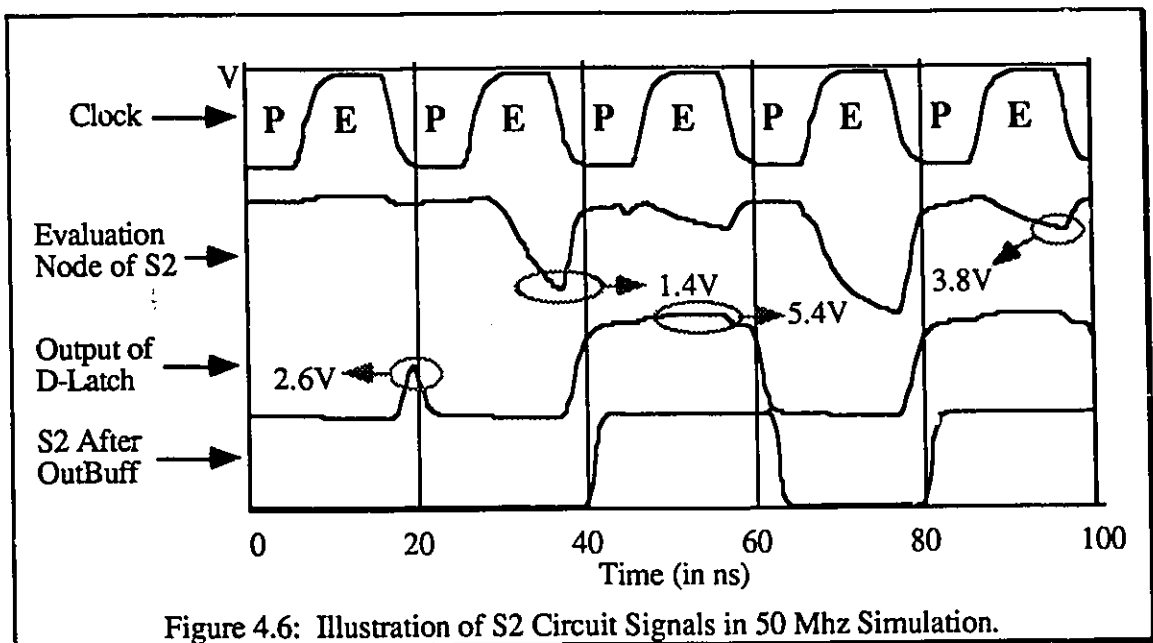


Figure 4.5: Functional Description of 3-Bit Adder.

### 4.3.1 3-BIT ADDER SIMULATION RESULTS

A mask-extracted SPICE-deck was simulated using Mac-Spice 3C1 software. All output nodes were loaded down with 100ff of capacitance. The input signals have modest rise and fall edge times of 3ns. The adder cell was simulated with a clock speed of 50 Mhz. A complete listing of the 3-Bit Adder SPICE-deck can be found in Appendix H.

The wave forms in Figure 4.6 illustrate the functionality of the circuitry associated with the output node 'S2'. The signal at the top of the plot is the clock signal. The letters 'P' and 'E' have been superimposed over the clock signal to indicate the precharge and evaluate portions of the clock cycle. The signal directly below the clock signal is the evaluation node of the S2 ROM structure. The S2 ROM structure has the tallest ROM height in the 3-Bit Adder cell, which is a height of 6 series NTRAN devices including the bottom domino clocked device which controls the ROM evaluation cycle.



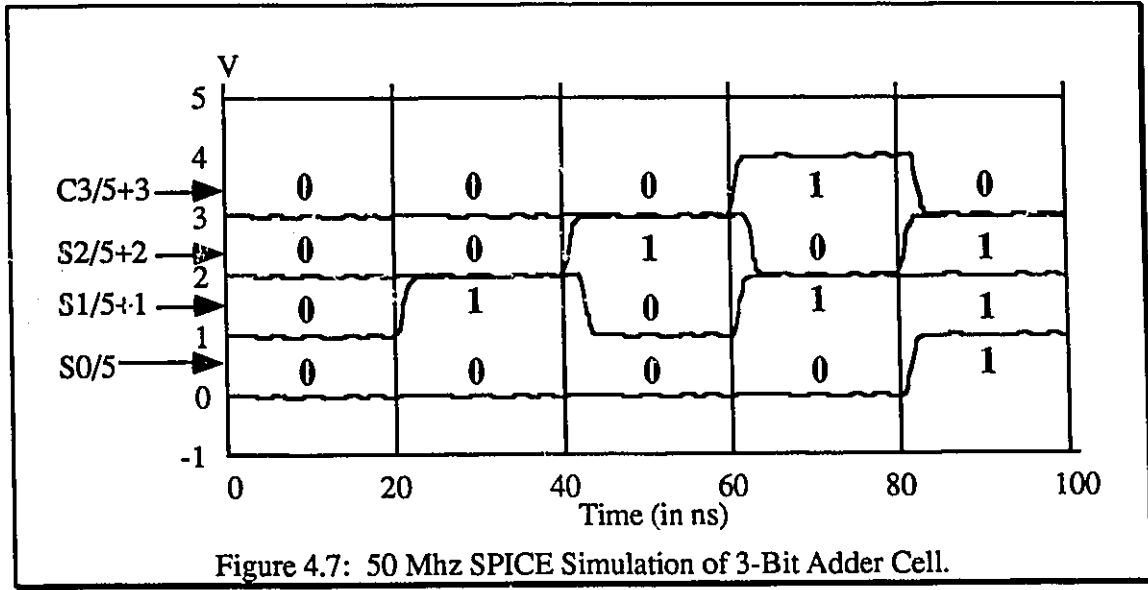
Since the S2 tree is the tallest in the ROM cell it is thus the slowest, and represents the critical evaluation path for the cell. Several voltage levels on the evaluation signal have

been highlighted to indicate the worst case evaluation levels for both high and low logic levels for the ROM cell operating at 50 Mhz.

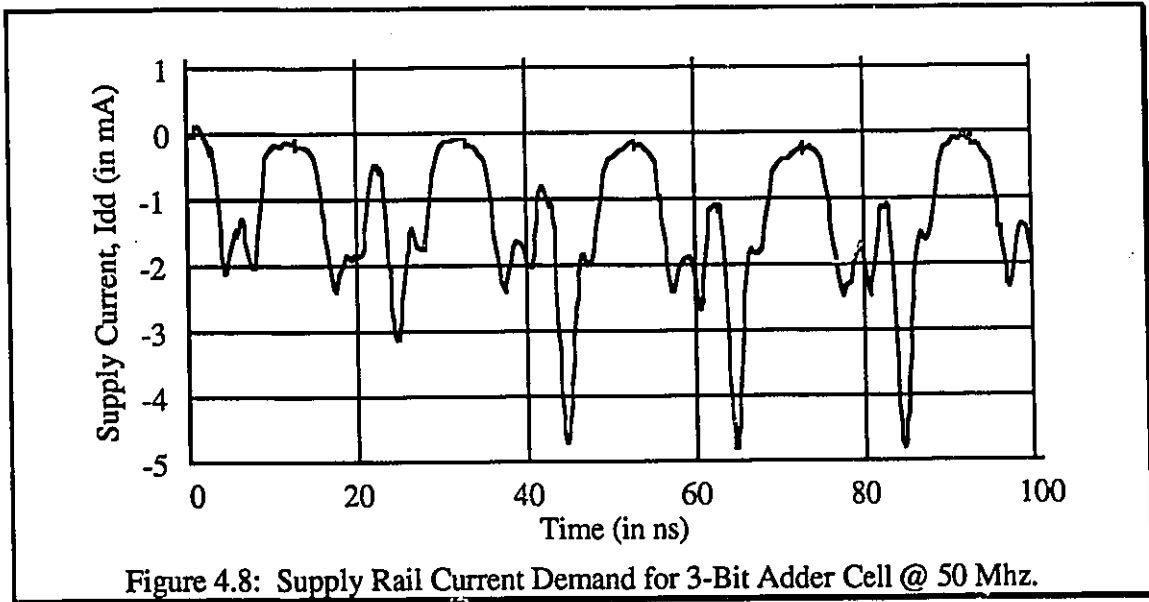
As indicated by Fig. 4.6, the worst-case pull-down voltage of this ROM reaches only 1.4 Volts, which is marginally within the noise margin for a CMOS low level signal. The worst case high evaluation signal level, due to charge sharing and charge redistribution within the internal nodes of the ROM cell, is 3.8 volts. The third signal in Fig. 4.6 is the output signal of the D-latch cell, which inverts the evaluation node value to produce the correct output one clock cycle later. The output of the latch, which holds its value for a full clock cycle, has superimposed dynamic switching noise, as indicated by the highlighted area. The initial 2.6 volt spike on the D-latches output, at the 20 nanosecond timing mark, is due to charge coupled noise associated with the clock signal switching. In essence the clock signal attempts to pull up the internal circuit node of the D-latch which is holding the previous input value presented to the latch. This effect is only transitory as the clocked devices in the latch switch fully on and pull the output node back to ground potential.

The same dynamic switching noise is also found on the transmission of high level signals as indicated by the 5.4 volt 'bump' on the top of the output high level. This is produced by charge coupling of the storage node with the clock signal rising edge. Once this signal passes through the cell output buffer all dynamic noise is removed and 'clean' 0 and 5 volt signals are passed to the next logic block, as illustrated by the bottom waveform in the plot.

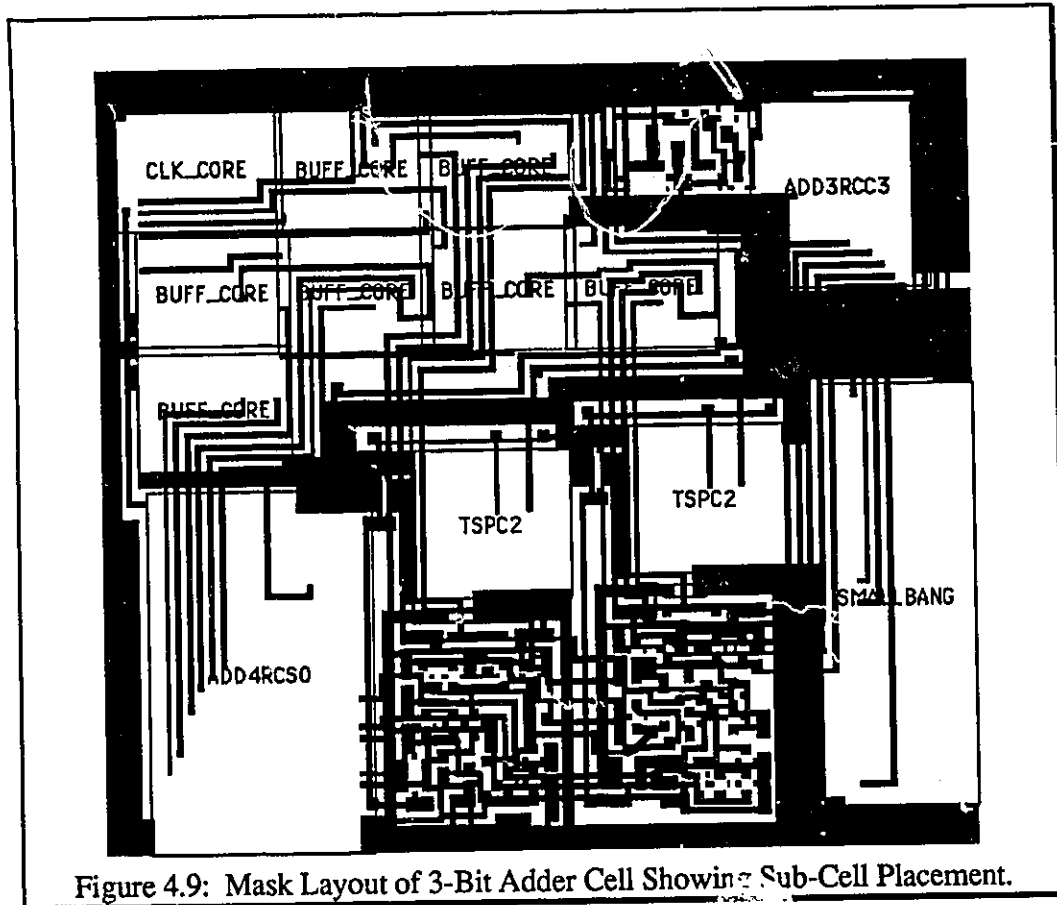
The complete set of output signals for the 3-Bit Adder cell are illustrated in Figure 4.7. Values of zero and one have been superimposed on the plot to indicate the output level of each bit in the output field. The first all zero output value corresponds to the initial values of the output latches prior to evaluating the first set of input data. The next four values show the effect of adding values {2, 4, 10, 7}.



The supply current demand associated with this simulation is illustrated in Figure 4.8. The peak switching currents of the cell are less than 5 milliamperes with an average current demand of approximately 1 milliamperes corresponding to an average power dissipation of 5 milliwatts.

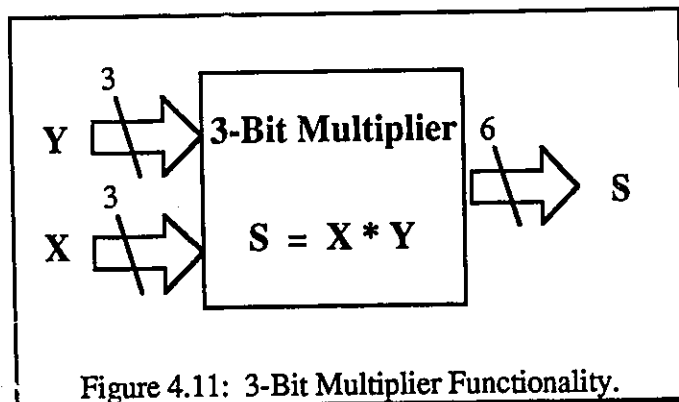


The 3-Bit adder cell measures 780 x 265 microns. The footprint of sub-cells in the 3-Bit Adder cell is shown in Figure 4.9, with a complete layout in Figure 4.10.



#### 4.4 THE 3-BIT MULTIPLIER CELL

The 3-bit multiplier performs a standard 3-bit binary multiplication. The multiplier requires two 3-bit input words and produces a 6-bit output word, as illustrated in Figure 4.11. The 3-bit multiplication is performed



in one clock cycle, and the cell is capable of running at clock speeds up to 35 Mhz. The full schematic realization of the 3-bit multiplier is found in Appendix I.

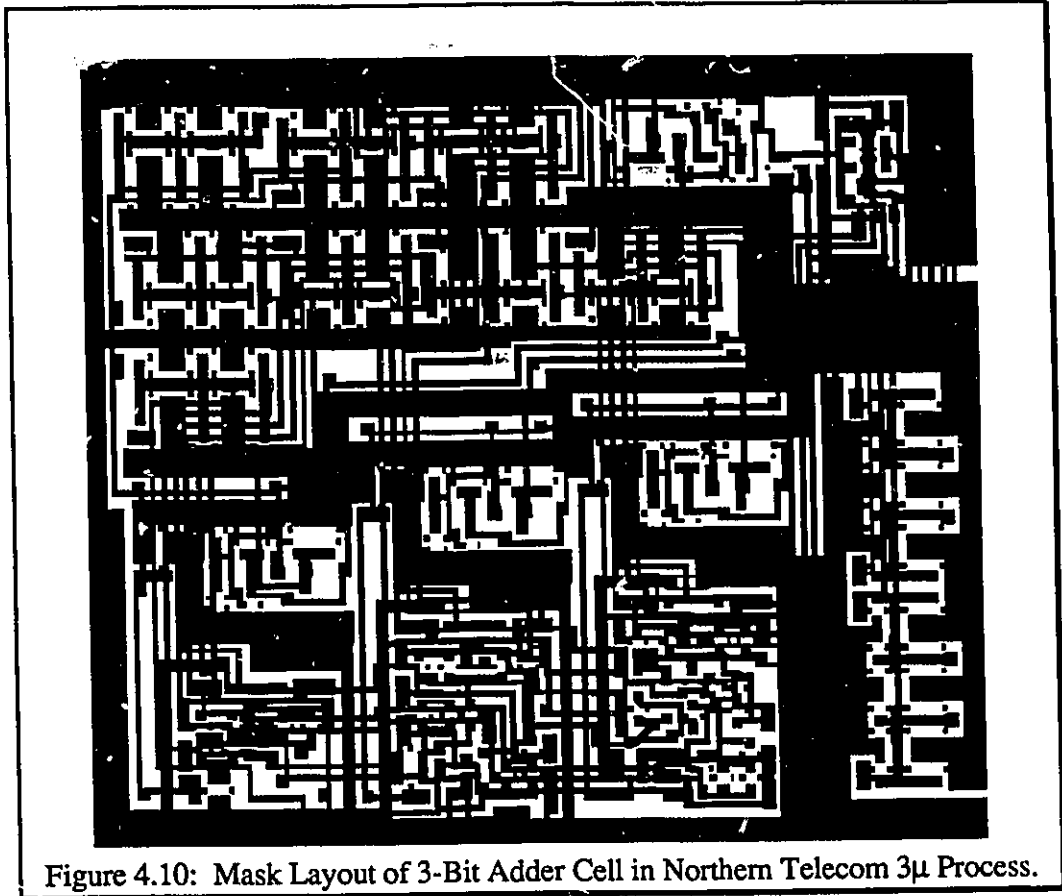


Figure 4.10: Mask Layout of 3-Bit Adder Cell in Northern Telecom 3 $\mu$  Process.

#### 4.4.1 3-BIT MULTIPLIER CELL SIMULATION RESULTS

A mask-extracted SPICE-deck was simulated using Mac-SPICE 3C1 software. Due to the large size of the SPICE-deck, an option card is included to relax the SPICE tolerance parameters in order to attain simulation results. The option card used is:

```
.options reltol = 0.05 abstol = 100u vntol = 1000u.
```

This relaxation of the tolerance levels gives the signals a piecewise linear appearance rather than the smoothed waveforms generated with the default tolerances. Thus some of the signals appear to be sharper or more squared than they would normally be.





The physical layout of the 3-Bit Multiplier cell is shown in Figure 4.14. The cell measures 997 x 268 microns.

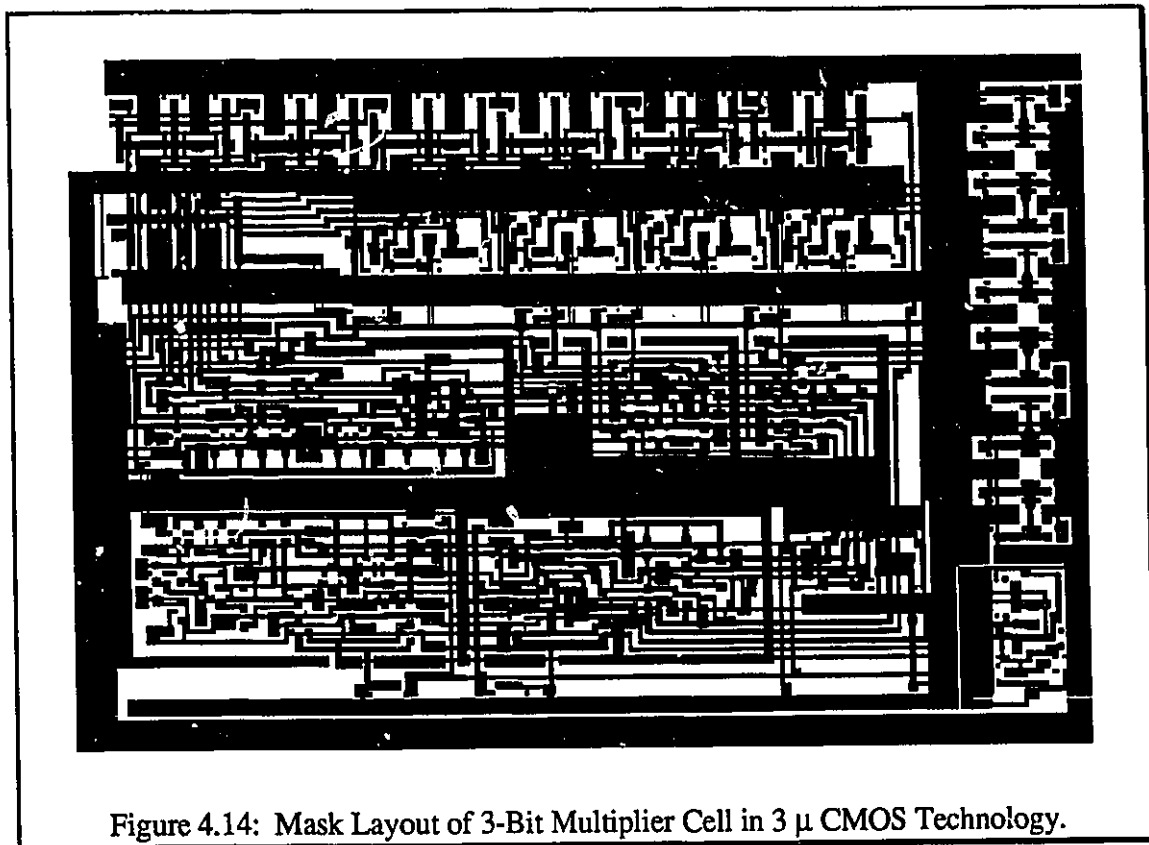


Figure 4.14: Mask Layout of 3-Bit Multiplier Cell in 3  $\mu$  CMOS Technology.

#### 5.4.2 TEST RESULTS ON THE 3-BIT MULTIPLIER CELL

The 3-Bit Multiplier cell was fabricated on the test chip IC3WRAAM. The cell was one of five test cells on the chip. To limit the number of I/O pads all inputs to the cells were wired in parallel in a bus like fashion. Similarly, all power and ground rails were also connected and all cells were driven from a signal clock signal. For this reason no accurate power dissipation reading could be taken. The 3-Bit multiplier cell was tested using the Asix-2 tester. The cell was determined to be fully functional through the use of an exhaustive set of test vectors; these can be generated by a simple program running on the Asix host computer. The cells begin to fail at the upper speed limits of the outpads, which

is approximately 35 Mhz. All five test chips performed identically and proved to be fully functional. The only noticeable problem was the relatively slow rise and fall time of all the output signals compared to other cells tested. This problem may be the fault of an undersized clock buffer cell, which was taken directly from the 4-bit adder cell.

## 4.5 THE 4-BIT ADDER CELL

The 4-bit adder performs a standard binary addition. It requires two 4-bit input words plus a carry-in bit and produces a 5-bit output field consisting of a 4-bit output word plus an overflow-bit; this is illustrated in

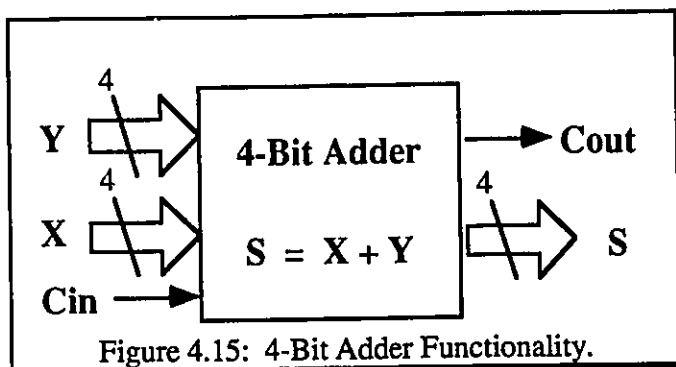
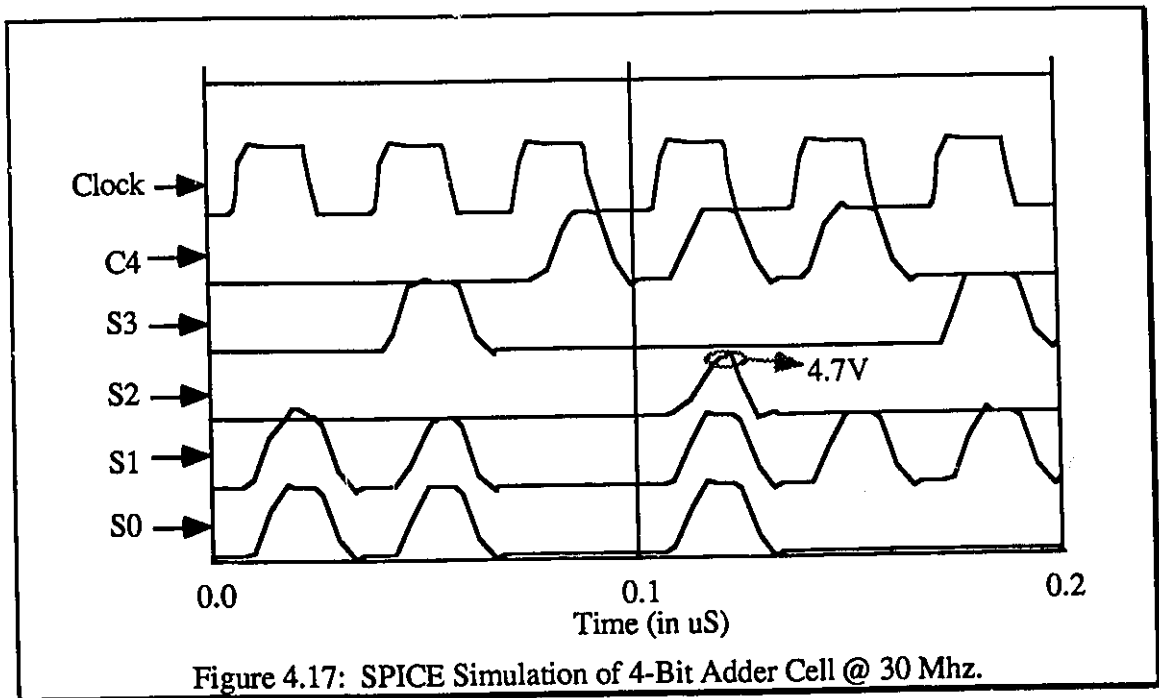
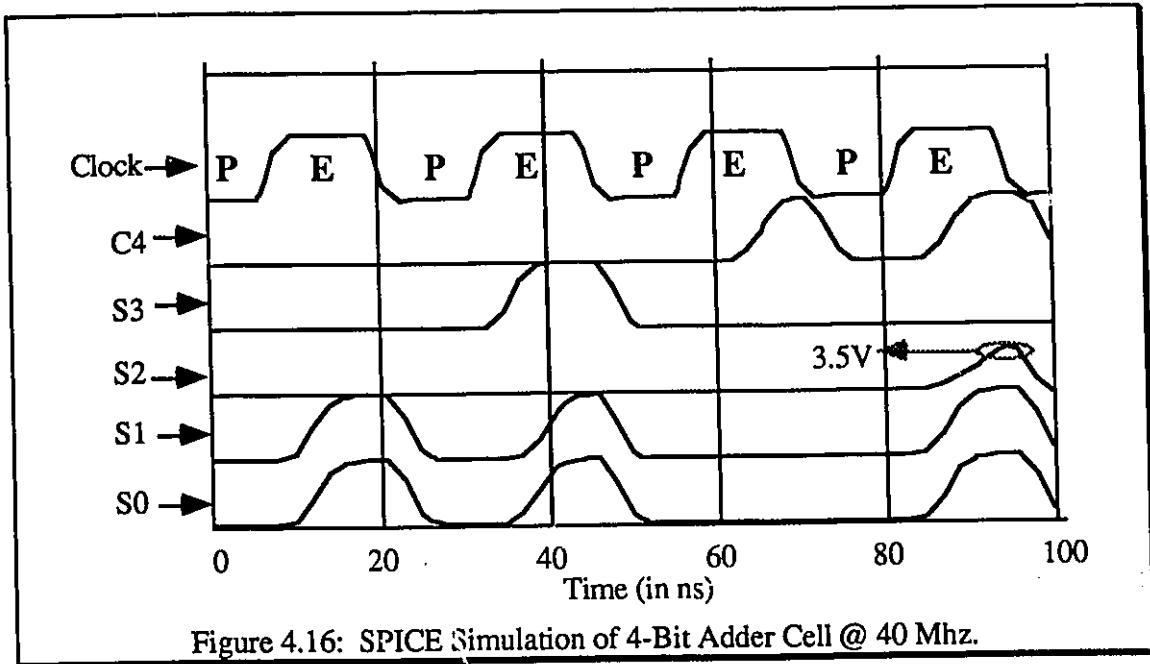


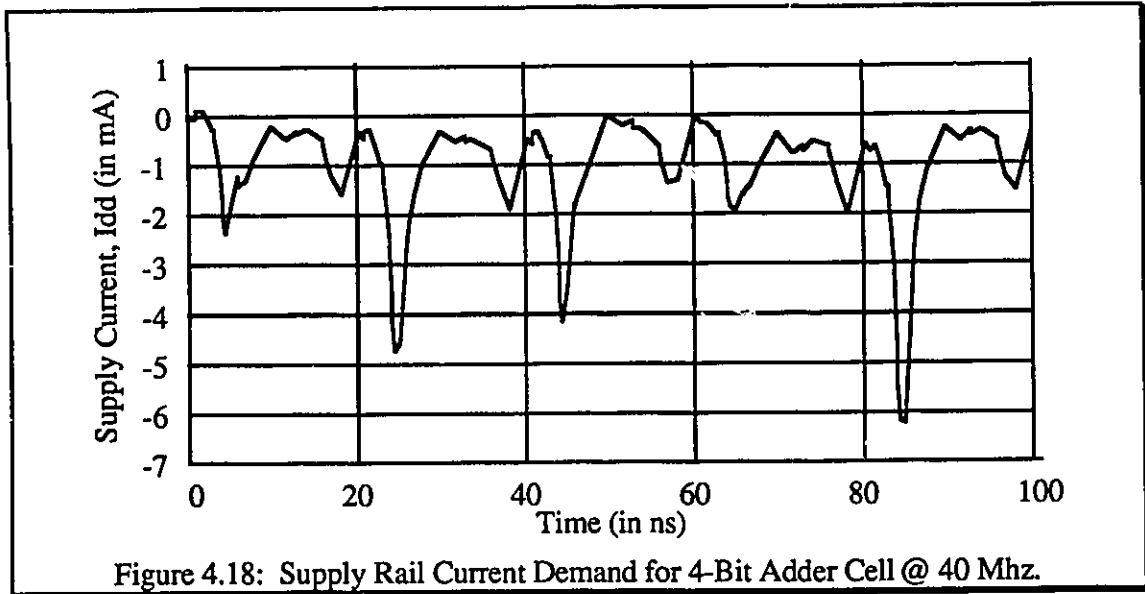
Figure 4.15. The 4-bit addition is performed in one clock cycle, and the cell is capable of running at clock speeds up to 30 Mhz. The 4-Bit Adder cell can be cascaded together to perform larger bit-field additions. For example, a 16 bit addition can be performed with four cells in four clock cycles. The full schematic realization of the 4-Bit Adder can be found in Appendix J.

### 4.5.1 4-BIT ADDER SIMULATION RESULTS

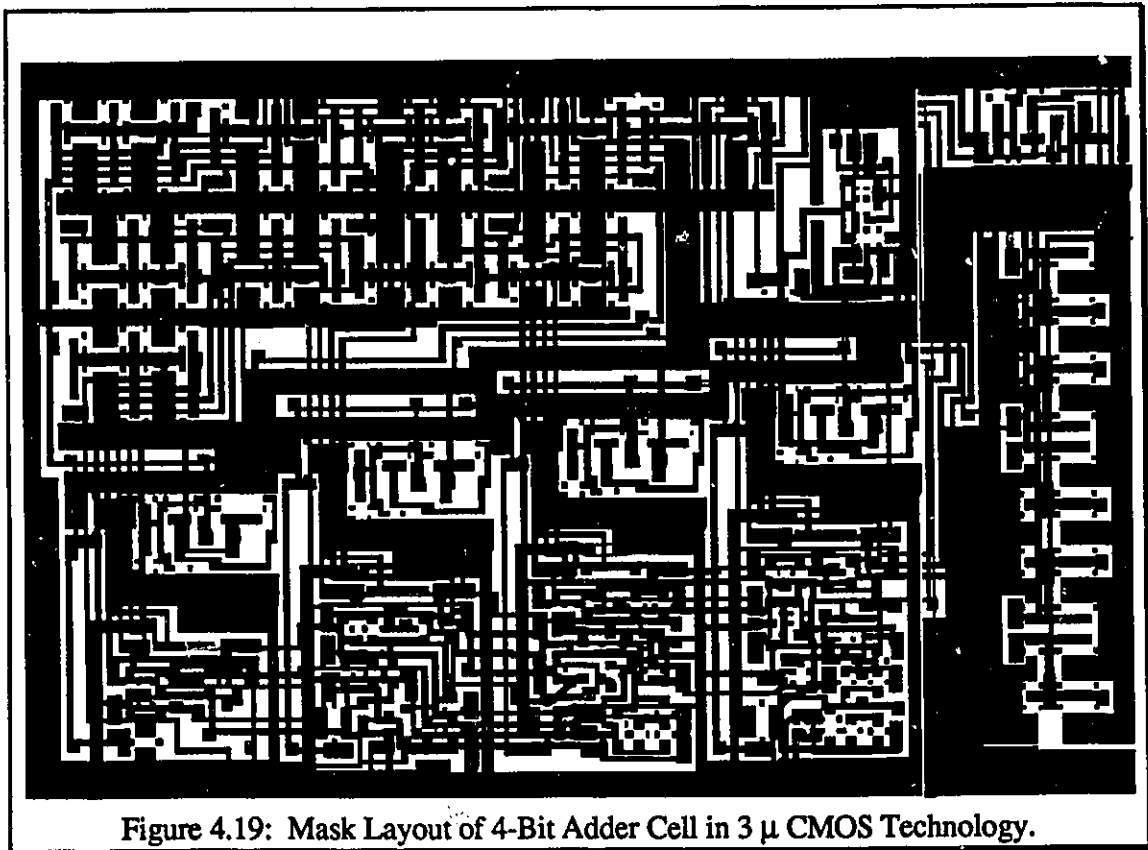
A mask-extracted SPICE-deck simulation was performed using the Mac-SPICE 3C1 software. Simulation results indicate that the cell is capable of running at clock speeds up to 30 Mhz. A 40 Mhz simulation is illustrated in figure 4.16. The output of Signal S2 shows a very poor output high level of only 3.5 volts. The 30 Mhz simulation, illustrated in Figure 4.17, produces output high level very close to the supply rail voltage of 5v. The corresponding supply rail current demand of the 4-Bit Adder is illustrated in Figure 4.18. The peak dynamic switching currents reach approximately 6 milliamps with an average

current demand of approximately 1 milliamp; this leads to an average power dissipation of 5 milliwatts at an operating speed of 40 Mhz.





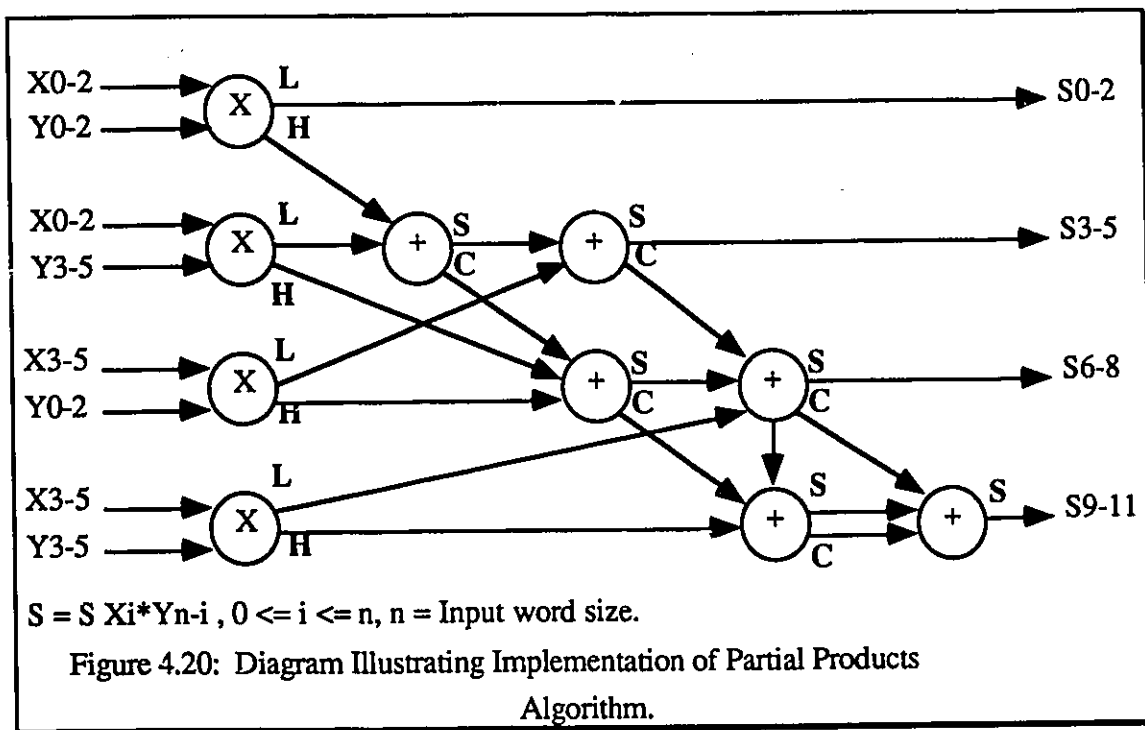
The Physical Realization of the Cell is shown in Figure 4.19. The cell measures 1000 x 270 microns in size.



## 4.5.2 TEST RESULTS ON THE 4-BIT ADDER CELL

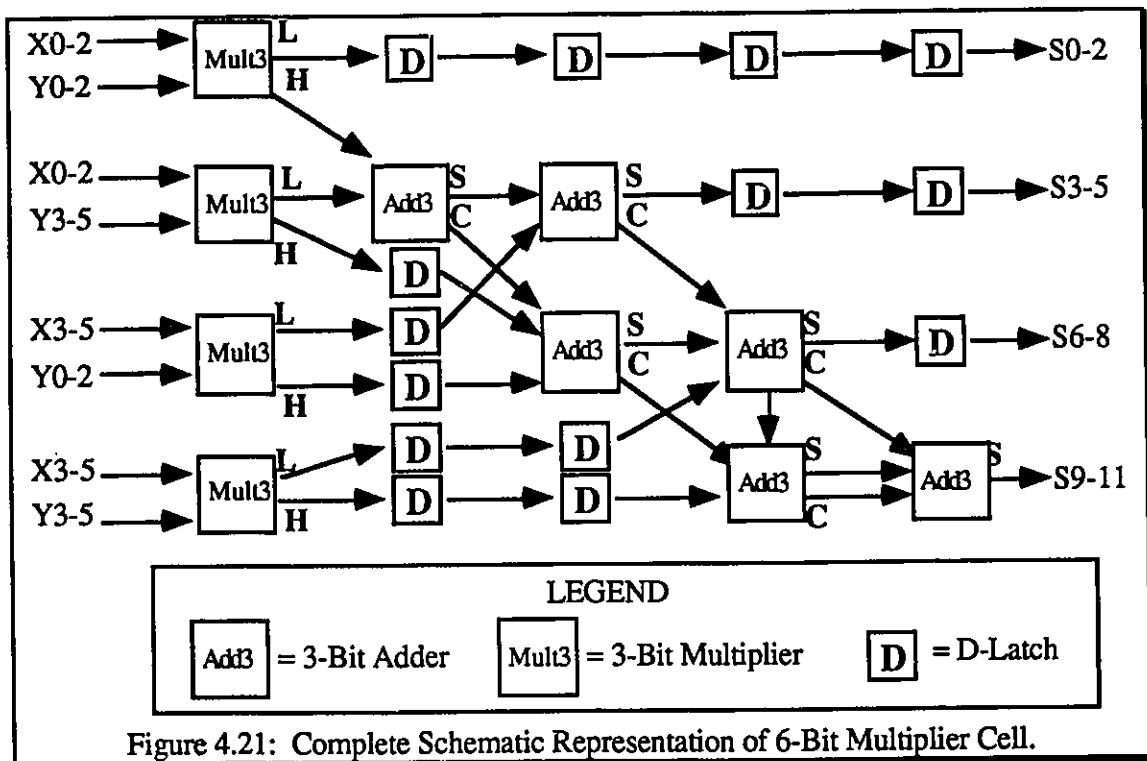
The 4-Bit adder cell was fabricated on the test chip IC3WRAAM. The cell was one of five test cells on the chip. To limit the number of I/O pads all inputs to the cells were wired in parallel in a bus like fashion. Similarly, all power and ground rails were also connected and all cells were driven from a single clock signal. For this reason no accurate power dissipation readings could be taken. The cell was proven to be fully functional through the use of an exhaustive test pattern. The cell performed extremely well and functioned up to the limit of the I/O pads reaching a speed between 35 and 40 Mhz. All five test cells proved to be fully functional and performances were essentially identical.

## 4.6 THE 6-BIT MULTIPLIER CHIP



The 6-bit multiplier performs a standard binary multiplication through the application of the sum of partial products algorithm, as illustrated in Figure 4.20. It requires two 6-bit input words and produces a 12-bit output word. The 6-bit multiplication is performed in four

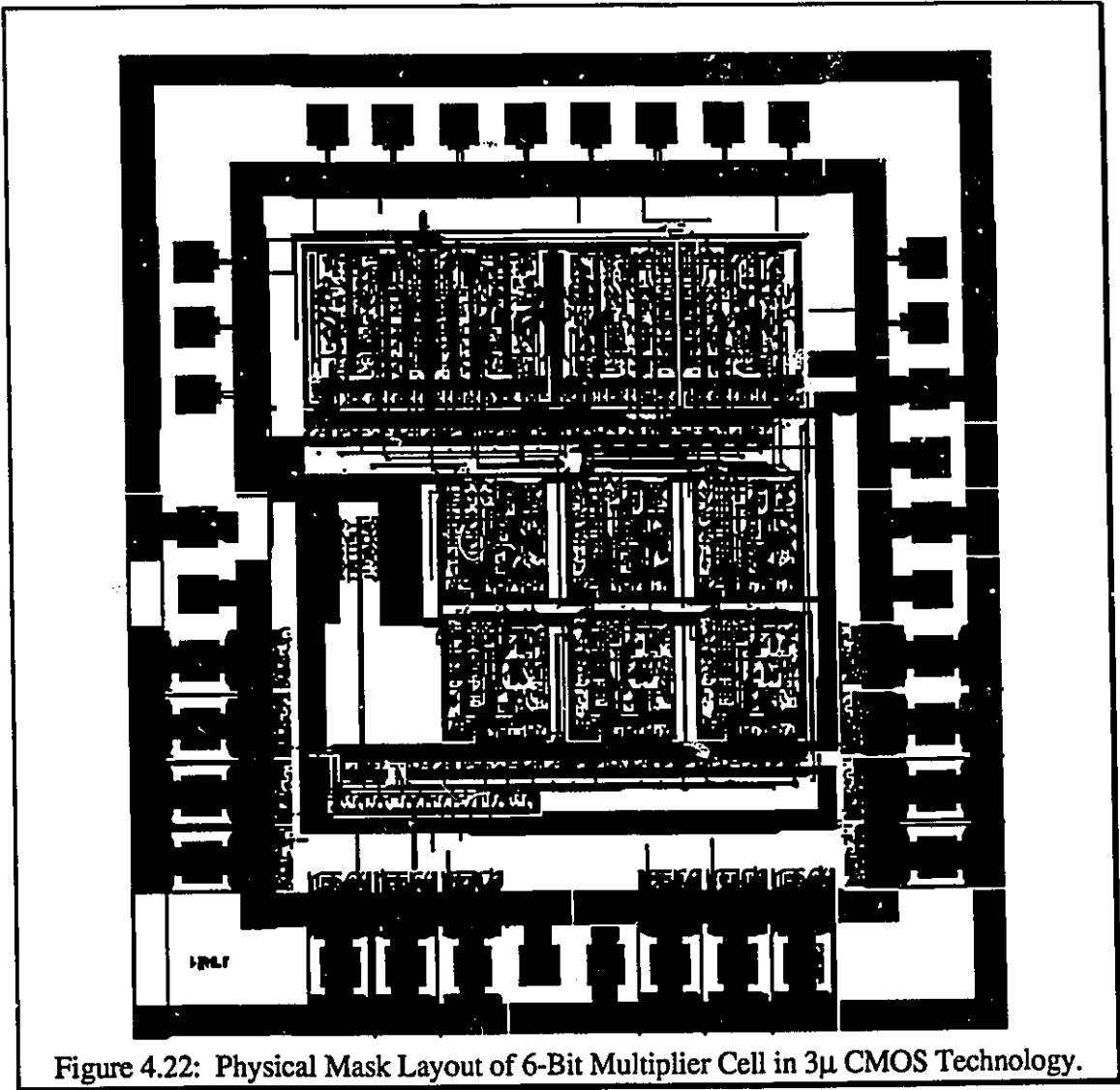
clock cycles. Since the chip is a fully pipelined system it is capable of producing one 6-bit multiplication every clock cycle with a latency of four clock cycles. The 6-bit multiplier cell should be capable of running at clock speeds up to 35 Mhz. The schematic realization of the 6-bit multiplier is illustrated in Figure 4.21. The physical realization of the chip is shown in Figure 4.22. The chip measures 2000 x 1000 microns in design scale, which corresponds to 1200 x 600 microns in actual size. The chip performs the 6-bit multiplication function with an average power consumption of 120 milliwatts, including the output pad driver dissipation.



#### 4.7 THE DEVELOPMENT OF 3 $\mu$ ECL COMPATIBLE PADS

In order to develop and test high-speed CMOS circuitry, which is capable of producing ECL compatible board level signals, it is necessary to become familiar with transmission line theory in a practical sense. Understanding what constitutes a transmission line, how to properly terminate a transmission line, and how to determine the

characteristic impedance of a given signal path are all important subjects which must be understood in order to properly design and test such circuitry. A summary of basic transmission line theory with a practical point of view is covered in Appendix K and can be referenced for any transmission line topic.





## 4.7.1 DEFINING ECL COMPATIBLE TECHNOLOGY

The technology known as ECL, which is an abbreviation for emitter-coupled-logic, is a bipolar logic family well known for its high speed capability, which is accompanied by a relatively high power dissipation. The ECL logic family has a small output voltage swing which offers less noise immunity compared to other bipolar logic families, such as TTL (Transistor Transistor Logic). The CMOS (Complimentary Metal Oxide Semiconductor) logic used for fabrication in this thesis has very large noise immunity in its static configuration. Figure 4.23 provides the 10K ECL family specifications, taken from the Motorola MECL Design Handbook. Strict attention must be paid to ensure accurate output levels for true ECL compatibility.

$-0.960 \text{ volts} < V_{oh} < -0.81 \text{ volts}$	$V_{oh} \text{ (nom)} = -0.90 \text{ volts}$
$-1.850 \text{ volts} < V_{ol} < -1.65 \text{ volts}$	$V_{ol} \text{ (nom)} = -1.75 \text{ volts}$
$R_{load}$	$= 50 \text{ ohms}$
$C_{load}$	$= 0 \text{ to } 100\text{pf}$
Ambient Temp.	$= 0 \text{ to } 85 \text{ }^\circ\text{C}$
Power Supply, ( $V_{ee}$ )	$= -5.2 \text{ volts } \pm 0.10 \text{ volts}$

Figure 4.23: ECL 10K Series Specifications.

The ECL logic family makes extensive use of the well known differential pair circuit, which will be one of the main building blocks for the pads which were designed for this part of the research. Unlike true ECL I/O pads, which take ECL levels from on-chip and produce ECL levels off-chip, the two interface circuits that are developed here take standard CMOS levels on-chip and produce ECL compatible levels off chip, and take off-chip ECL compatible levels and reconstruct CMOS level signals within the chip.

The goal for this design is to attempt to maximize the realizable operating speed of off-chip driver circuitry for standard  $3\mu$  CMOS technology, to aid in the creation of high-speed switching tree architectures. The current maximum off-chip driver speed attainable is approximately 40 Mhz using the existing library drivers. The specifications for the ECL compatible interface circuitry are given in Fig. 4.24. The initial design specifications for these circuits were to be able to transfer data at speeds of 80 to 100 Mhz driving a 50 ohm terminated line with a lumped capacitance equivalent to approximately 40 pf.

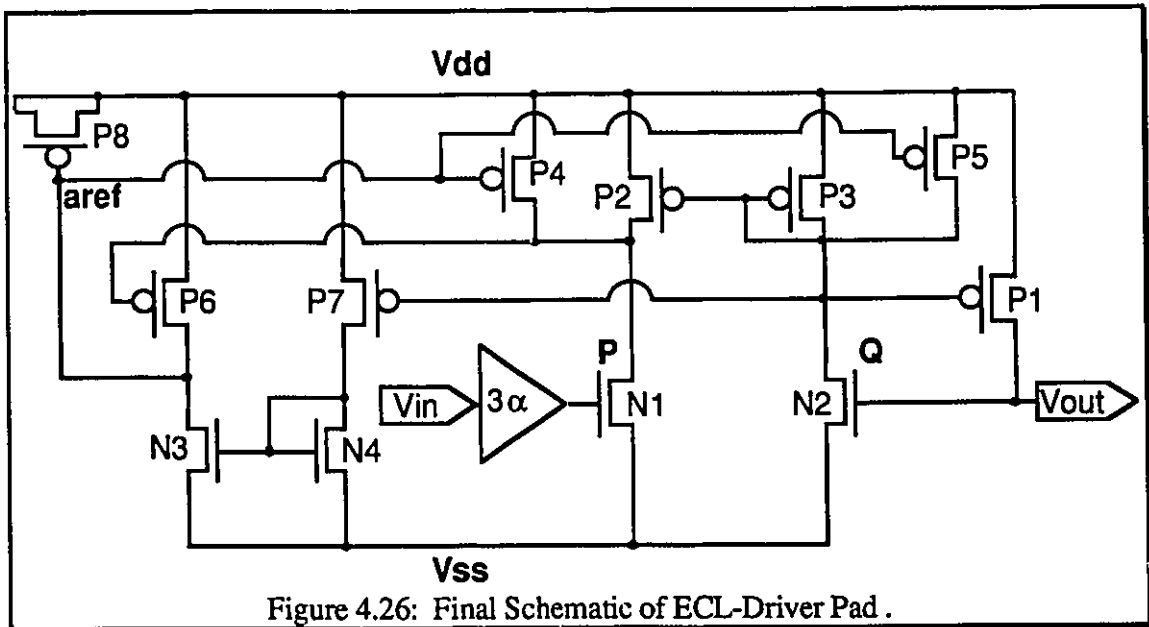
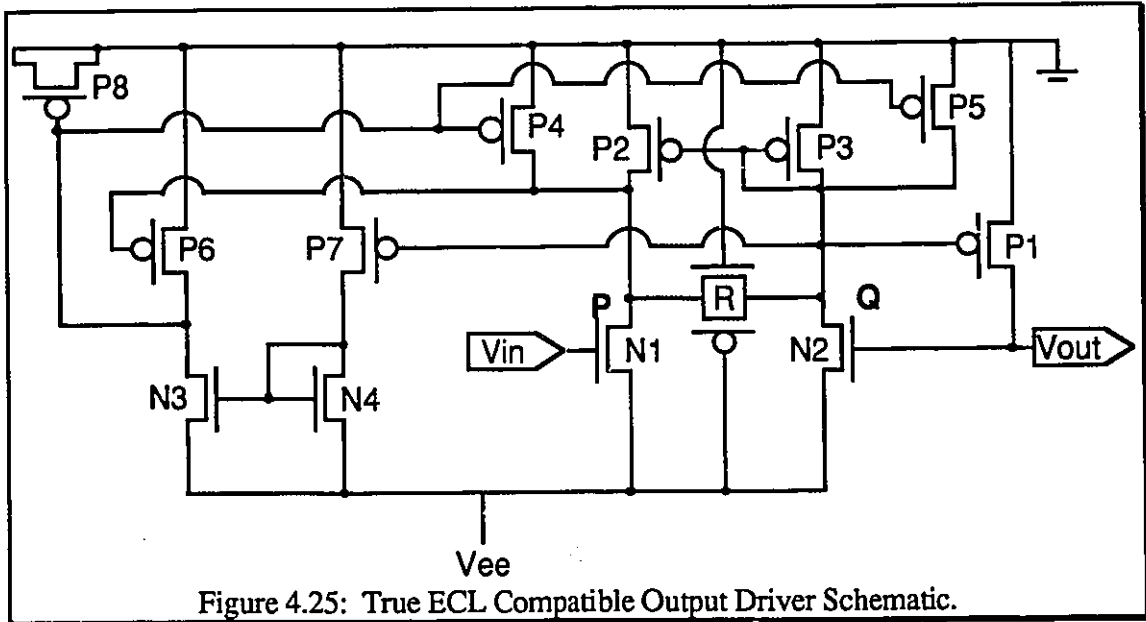
$-0.960 \text{ volts} < V_{oh} < -0.81 \text{ volts}$	$V_{oh} \text{ (nom)} = -0.90 \text{ volts}$
$-1.850 \text{ volts} < V_{ol} < -1.65 \text{ volts}$	$V_{ol} \text{ (nom)} = -1.75 \text{ volts}$
$R_{load}$	$= 50 \text{ ohms ( into } V_{tt} = -2.0 \text{ volts )}$
$C_{load}$	$= 0 \text{ to } 50\text{pf ( lumped capacitance)}$
Ambient Temp.	$= 0 \text{ to } 50 \text{ }^\circ\text{C}$
Power Supply	$= -5.2 \text{ volts } \pm 0.10 \text{ volts}$
Maximum Speed	$= 100 \text{ Mhz}$
Max. Pad Delay	$= 6.0 \text{ ns. @ } 25^\circ\text{C with } 20\text{pf load.}$

Figure 4.24: Specifications for ECL Compatible I/O Pads.

## 4.7.2 DESIGNING THE ECL-PAD DRIVER CIRCUITRY

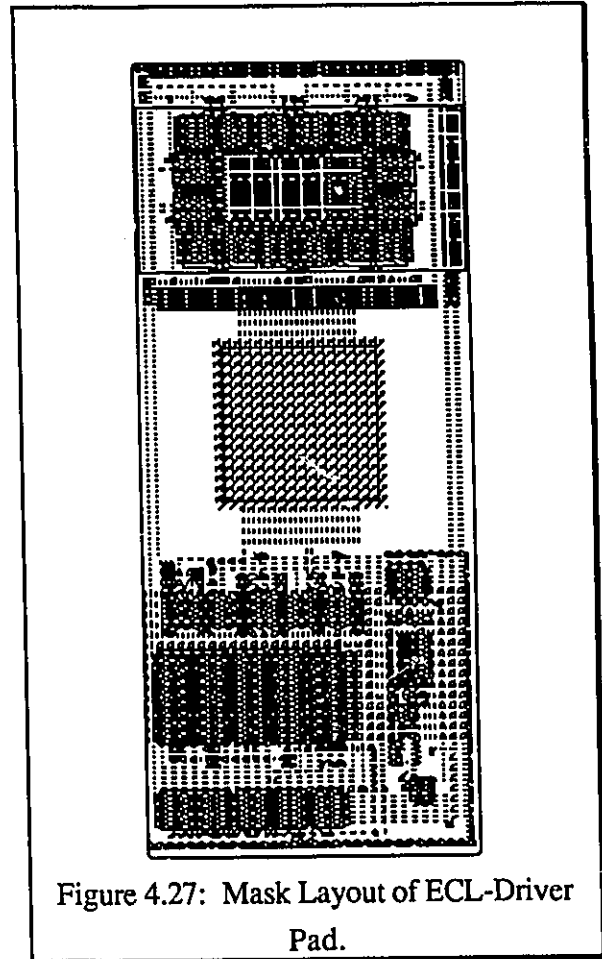
The schematic of the true ECL compatible output driver pad is a modification of a design taken from Schumacher et al.[18], and is shown in Figure 4.25. The original circuit is designed to take true ECL input levels and buffer them off-chip, again producing true ECL levels. This circuit was the starting point for the CMOS to ECL output driver pad designed for this research project. In fact, the final schematic diagram is still essentially the same. The main simplification to this schematic is the removal of the resistive component on the first differential pair, since this was added to the original design to reduce the output gain of the differential pair. This is not necessary for the CMOS to ECL pad developed here. The final version of the ECL output pad driver schematic is shown in Figure 4.26.

The operation of this circuit is quite simple. For simplification we assume that the current mirror devices P2 & P3 and P4 & P5 are matched as well as N1 & N2. The ECL output voltage, which is fed to the gate of N2, produces a voltage on node P and a corresponding current level through current mirror P2 & P3.



If we assume that node Q is lower than node P, then P7 is turned on harder than P6 causing a current flow into N3, effectively turning P4 & P5 on harder. Since P5 sources more current to node Q this will cause node Q to rise resulting in a decrease in current in the diode connected transistor, P3. The current mirror action will cause P2 to also source less current. The decrease of current through P2 is offset by the increase in current through P4, leaving node P relatively unaffected. Hence the end result is that node Q moves closer to node P, reducing the overall differential error of the first differential pair (P6 & P7).

This type of error correction scheme is called differential to common mode feedback. The authors of the original reference claim that this error correction scheme offers good performance over a wide range of temperature as well as changes in supply voltage levels and loading characteristics. This is to be expected since feedback based on differential variables will be relatively unaffected by common mode variations such as described. The ECL-Driver Pad was designed to be driven by on-chip CMOS signals and produce off-chip ECL levels into a 50 ohm terminated line. The terminating resistor is biased at the logic zero output voltage of -2v relative to V<sub>dd</sub>. The mask layout of the ECL-Driver Pad is shown in Figure 4.27. The ECL-Driver Pad measures 277 microns by 402 microns. The portion of the mask above the bonding pad represents the open drain output device, which is capable of sourcing over 50 mA.



### 4.7.3 DESIGNING THE ECL-RECEIVER PAD

The ECL-Receiver pad was designed to take an off-chip differential ECL level signal and reproduce a CMOS level signal capable of driving on-chip loads. Several initial single-stage differential pair circuits, which had been obtained from previous publications, were simulated but found to be inadequate for the specifications that were required [19]. A new two stage differential pair circuit was finally designed. The mask layout of the ECL-Pad Receiver is shown in Figure 4.28. The ECL-Receiver measures 227 microns by 311 microns. The devices directly above the bonding pad protect the input circuit from electrostatic discharge (ESD).

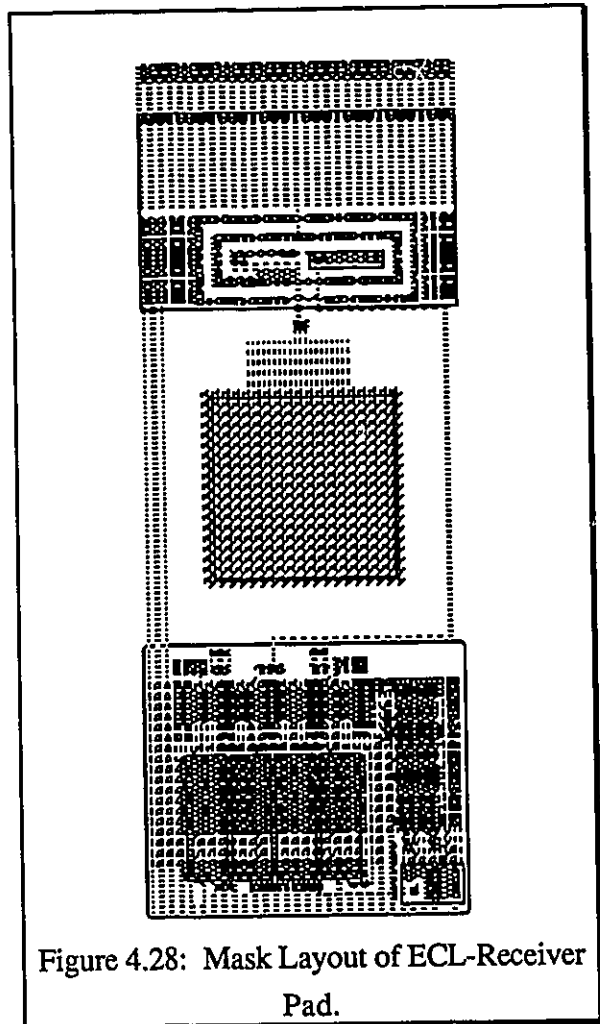
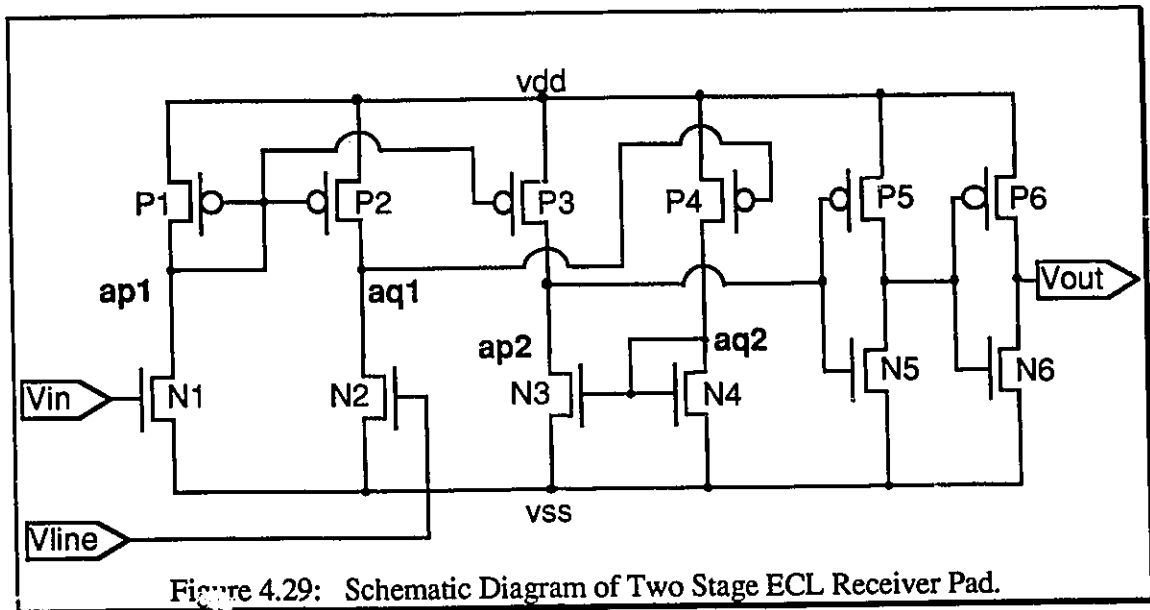


Figure 4.28: Mask Layout of ECL-Receiver Pad.

The ESD protection is a diffused resistor which incorporates a p-n junction forward-bias mechanism plus the distributed RC time constant to absorb steep discharge transients. The ESD device clamps negative and positive going transitions to one diode drop below and above supply levels. The ECL-Receiver pad consists of two differential pairs followed by an output buffer; the schematic diagram is shown in Figure 4.29. The first differential stage is a high gain stage which swings quickly into saturation. The two output voltages from the differential amplifier drive the second stage, which is a minimum gain stage with a high output voltage swing. The active loads of the second stage are purposely unbalanced

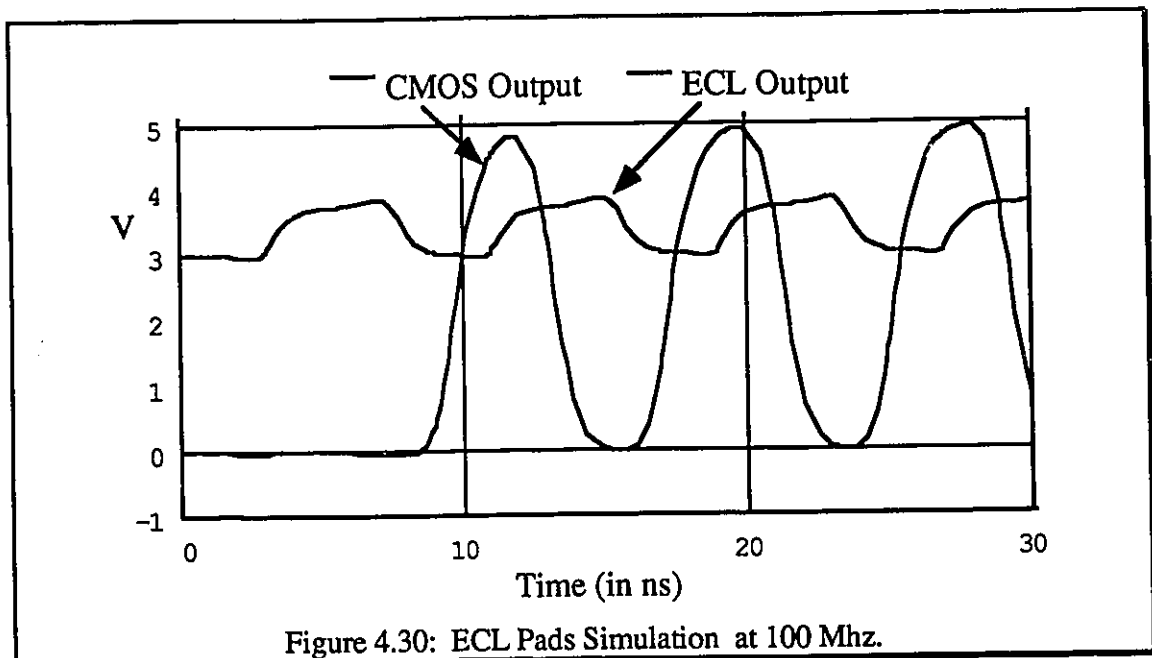
in order to produce a bi-stable device, which causes the output signal AP2 to swing from  $V_{DD}-V_t$  to  $V_{SS}+V_t$ , which is then tapped off by a set of inverters to regenerate a solid rail to rail signal. This pad will thus take an ECL signal and a reference signal,  $V_{tt}$ , and reproduce a corresponding CMOS level signal.



#### 4.7.4 ECL SIMULATION RESULTS

The ECL I/O Pads were simulated using Mac-SPICE 3C1 software. The simulation did not consider device mismatch due to process variations or supply current limitations due to inductive losses. The mask-extracted SPICE-decks indicate that the pads are capable of being driven at speeds up to 100 Mhz, with a maximum simulation speed of 125 Mhz. The simulation used an external 50 ohm termination resistance connected to a termination supply rail,  $V_{tt}$ , of 3.0 volts ( $V_{dd}-2v$ ). For  $V_{dd}=0v$  and  $V_{tt}=-2v$  the resulting output will correspond to ECL logic levels. The simulations also indicate that the gate delay through both devices is approximately 8 ns. The input pad receiver has a propagation delay of 2.1ns while the output pad driver has a propagation delay of 5.4ns. A copy of the extracted SPICE-decks is given in Appendix L.

Simulation results from the mask-extracted SPICE-decks, running at 100 Mhz, are shown in Figure 4.30. The signals in this figure are from the ECL-Driver output and the corresponding ECL-Receiver output. Average currents at these speeds for the internal input and output pad circuitry are 9.1mA and 17mA respectively. Termination sourcing current is dependent on capacitive line loading. For the simulation the termination source current was on the order of 40mA through the 50 ohm termination load to the  $V_{tt}$  rail.



#### 4.7.5 THE TEST CHIP IC3WRECL

A test chip, IC3WRECL, was designed to verify the functionality of the ECL-compatible pads; the mask layout is shown in Figure 4.31. The chip consists of three test structures. The first is a normal CMOS input pad connected to an ECL pad driver. The second test structure consists of an ECL receiver pad connected to a CMOS output pad. The third test structure consists of an ECL receiver pad connected to an ECL pad driver. The first two test structures were placed on the chip to determine the individual functionality and characteristics of each ECL compatible pad. The third test structure,

essentially an ECL buffer, was created to determine the ability of the two pads to function together as well as to test the pads to see if they, in fact, were ECL compatible; this was performed by placing the buffer circuit into a chain of commercial ECL buffers.

#### 4.7.6 TEST RESULTS FROM IC3WRECL

The pads were tested using the Asix-2 tester and a Tektronix digital storage scope. A digital input signal ranging from 1 to 50 Mhz was presented to an ECL-Driver which generated the corresponding ECL pad signal; this signal was then passed to an ECL-Receiver pad which produced a standard on-chip CMOS signal. This signal was connected to

another ECL-Driver pad. Thus the off-chip accessible signals consisted of the initial digital input signal followed by two ECL signals taken from the first and third pad. All ECL output pads drove 50 ohm terminated transmission lines, illustrated in Fig. 4.32.

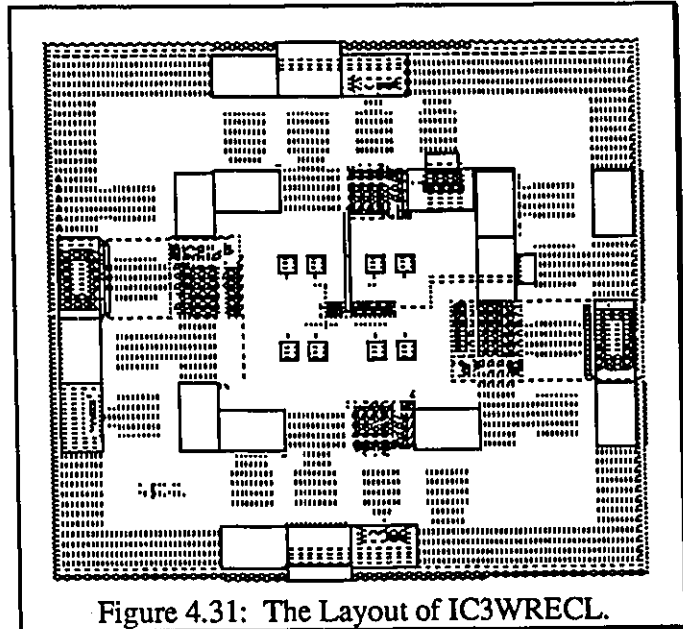


Figure 4.31: The Layout of IC3WRECL.



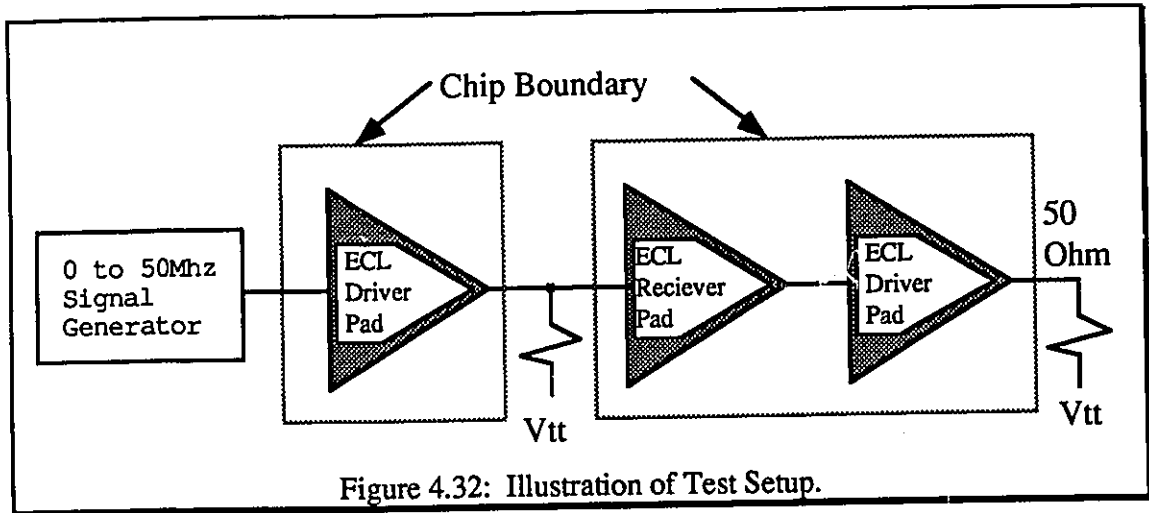


Figure 4.32: Illustration of Test Setup.

The input signal, which was relatively poor in quality due to the inadequate signal apparatus, is paramaterized by:

- mean rise time = 5.51ns, with mean standard deviation = 550 ps
- mean fall time = 5.37ns, with mean standard deviation = 300 ps
- mean pulse width = 6.78ns, with mean standard deviation = 100 ps

The corresponding output signal of the ECL-Driver Pad was:

- mean rise time = 3.06ns, with mean standard deviation = 232 ps
- mean fall time = 1.90ns, with mean standard deviation = 160 ps
- mean pulse width = 7.23ns, with mean standard deviation = 120 ps

$I_{dd}$  @ 50 Mhz = 42ma with approx. 40ff load

The propagation delay for each pad was determined to be:

ECL-Driver gate propagation delay = 6.0ns

ECL-Receiver gate propagation delay = 4.8ns

The pads went under a 24 hour burn in test to determine their reliability and to monitor any heat dissipation problems. Finally, the chip was placed on an ECL board in the middle of a series of ECL buffers made up of AND gates. The ECL pads proved to be

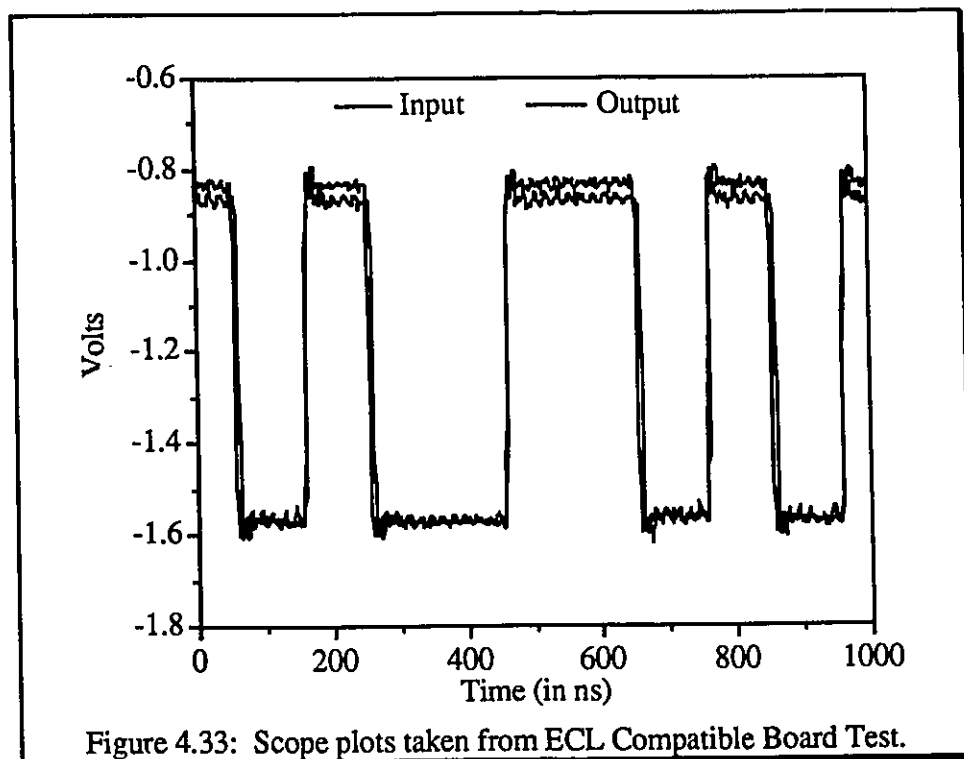
compatible in both receiving true ECL signals as well as driving other ECL devices. A set of scope plots taken during the functional testing of the ECL board is shown in Figures 4.33.

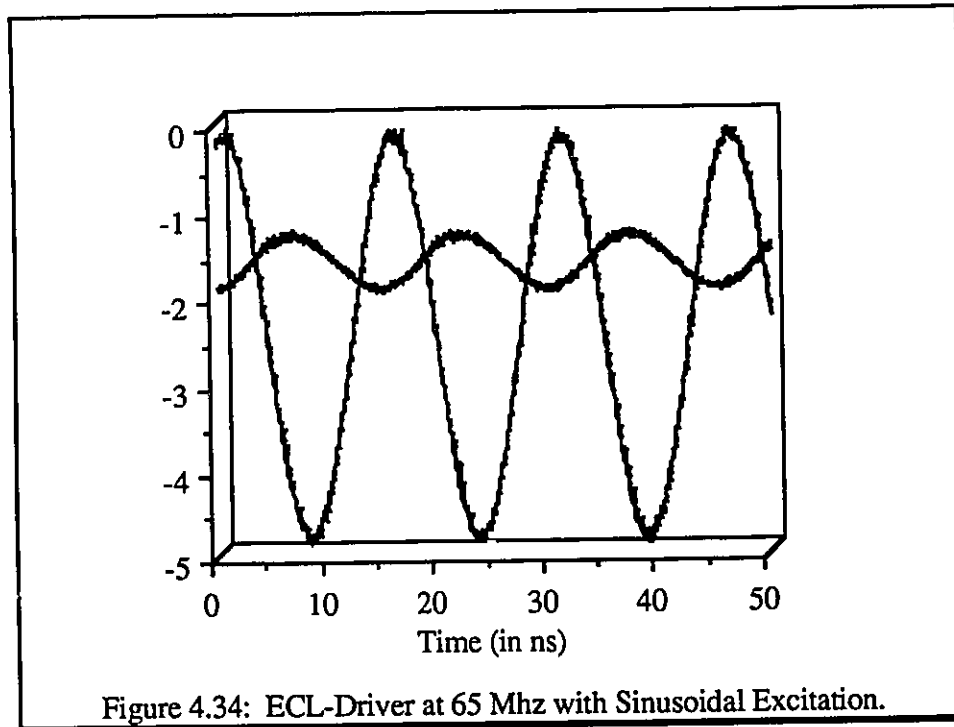
A sinusoidal generator was used to drive output signals that were free of system switching noise and to attempt to determine the maximum speed of the ECL pads. It was set to its maximum frequency of 65 Mhz. The input and output wave forms are shown in Figure 4.34. The following data was observed:

mean low level output = -1.76v

mean high level output = -0.93v

mean peak to peak voltage swing = 830 mv





## 4.8 SUMMARY OF STANDARD CELLS

Several standard test cells were constructed which have been tested and proven to be functional. The test results indicate that the speeds attained match the simulation results and prove to be on the upper-edge of the speed limit for the technology in which they were designed. The cells have been shown to be area efficient as well as power efficient, which makes the new architectures a sound choice for ULSI design. The cells have been assembled into a large chip which has demonstrated their pipelining capabilities as well as ease in construction using the standard sub-cell approach.

The ECL compatible pads were functionally tested at speeds upto 65 Mhz. Based on these test results it would appear that the maximum operating speed of the ECL pads should be approximately 80 Mhz. Unfortunately we have no available test equipment capable of running at those speeds. Those designs, if scaled down into 1.2 micron technology, would probably be capable of reaching 200 Mhz through-put rates.

**CHAPTER**  
**5**  

---

**SUMMARY AND  
FUTURE DIRECTIONS**

## **5.1 SUMMARY AND CONCLUSIONS**

The study into the development of high-speed ROM based cells for the implementation of standard boolean functions has led to the development of the switching tree cell. The switching tree cell implements standard boolean functions through the use of binary ROM trees; synthesis software (Woodchuck [8]) has been developed in a companion research thesis. After the minimization process the resulting switching tree cells have proven to be both area and power efficient. The area efficiency of the switching tree cell is dependent on the functionality that it implements. The area efficiency increases with the increase of the ROM height, as well as the number of ROM cells used in a given cell, as overhead is then amortized over a greater functionality. Switching tree cells with heights of six have proven to be able to implement standard binary multiplication in the same area as is required for addition, for single switching tree cells. This of course is limited by the ROM height limitation to cells which can multiply two three bit variables. However even the adder cell is more efficient over other forms of implementations such as standard gate level implementations using dynamic logic gates. The area efficiency is very dependent on the type of function implemented. Though preliminary estimates have shown efficiencies from 20 to 90% depending on functionality.

The switching tree cell is also very power efficient over other forms of dynamic or even static logic when running at high speeds, because the ROM tree design offers a

minimal amount of circuit nodes which must be cycled through charge and discharge in order to produce a result. The ROM tree itself has only one node per output bit which is precharged for evaluation of the logic function. Secondly, the switching tree cells require no large decoder network and the input circuitry driving the ROMs consists of nothing more than a set of inverters whose circuitry are amortized over several ROM cells. Taking this into account, along with the fact that the clocking scheme incorporated in the cells utilizes only a single clock signal, which minimizes the number of elements which must be clocked, indicates that the amount of power consumed by switching tree cell architectures is very low.

Through the use of standard domino logic techniques, cells of height six have been implemented and proven to be functional up to speeds of 35 Mhz in a  $3\mu$  CMOS technology. Simulations have shown that speeds of up to 50 Mhz are possible if the height of the ROM tree is restricted to five and if minimal sub-tree merging is used. This, however, does not define the maximum operating speed of the switching tree cell. If ROM cells are limited to heights of 3 or 4, then speeds approaching 100 Mhz are possible using standard unscaled ROM devices. Small tree heights make the use of device scaling much easier, which further increases the speed at which the cells can operate. A student working jointly in the study of switching trees has designed a full adder cell which utilizes fully scaled ROM cells with heights of 3 which has simulated to speeds of 150 Mhz in the  $3\mu$  CMOS technology. Of course, as is the case in all of these methods, increasing the speed of the switching tree design is traded-off for increased cell area. This cost however may be justified in the market place where implementations in lower cost technologies can match speeds usually only associated with higher cost technologies.

Due to the overwhelming speed capability of the switching tree cells a companion set of high speed I/O pads was included in the standard cell library. Increasing the maximum output switching speeds of existing pad drivers was essential in order to produce

switching tree systems which could maintain their high speed signal rate at the board level. For this reason a set of ECL compatible I/O pads was developed in the 3 $\mu$  CMOS technology. These pads were designed to be ECL compatible with 10,000 series devices. Mask-extracted simulations indicated that the pads were capable of through-put rates in excess of 100 Mhz. A test chip was fabricated and tested to be functional. The pads were tested to speeds of 65 Mhz which marks the upper limit of our existing testing equipment. Subsequently, the pads were placed in a board level test with ECL 10K series devices. The test proved that the ECL Compatible pads were capable of both receiving signals from and driving ECL devices.

A second set of high speed I/O pads was also developed to increase the I/O speed of the new 1.2 $\mu$  CMOS technology. The initial output pad simulations indicated that speeds of 100 Mhz were possible swinging through full CMOS levels. The initial test chip design, though functional, proved to suffer from excess signal over and under-shoot. The cause of this problem is probably due to insufficient current reaching the large final driver stage due to a combination of inductive losses and narrow power rails. Thus a second test chip was designed which implemented two possible solutions to the problem. First the power and ground rail were both increased in width by 50% and, secondly, diode bootstrapping techniques were employed in an attempt to limit the over and under-shoot to reasonable limits. This chip has been sent for fabrication, but has not returned in time to be tested.

Thus the study into switching tree architectures has produced a new logic implementation scheme which shows great promise. Standard test cell designs have proven functional with good performance characteristics. Several new high speed pads have also been developed as part of the standard cell library making this architecture a true high speed performance driven system.

## 5.2 FUTURE DIRECTIONS

The realm of work which was carried out in this thesis is quite wide. The space from which the work was defined though is much greater. During the course of this research several deficiencies or enhancements were discovered. Most of these discoveries were beyond the scope of this work while others were not able to be investigated within the time frame of this research works. Thus the following suggestions should aid in controlling the direction of future work.

- 1/ The ECL Compatible pads were designed to drive 50 ohm transmission lines. This proves to be quite demanding in terms of supply current, and is only really useful when sending signals over distances greater than several feet. So new ECL driver pads should be designed for driving transmission lines with higher characteristic resistances. This will reduce the current demand which will increase the number of pads which can be fed by a set of power pads, making it a better choice of implementation in the university environment.
- 2/ The ECL driver pads development in the  $3\mu$  CMOS technology should be redesigned within the  $1.2\mu$  CMOS technology. Initial SPICE simulations indicate that such designs would be capable of running at speeds of up to 200 Mhz.
- 3/ The use of precharging the ROM structures as well as the employment of some simple scaling techniques show great promise. These techniques should be further investigated.
- 4/ The clock buffer in the 3-Bit Multiplier and 4-Bit Adder cell were slightly under sized. These buffers should be increased by about 20% to increase the performance of the cells as well as decrease the rise and fall times of the data coming out of the cells.

- 5/ Currently, the switching tree architecture can be classified as a partial automatic synthesis system, because the minimized ROM schematics are produced by software but the ROM cells themselves are still being layed out by hand. If the ROM cells could be designed by a form of silicon compiler, than the turn around time for this system could be greatly decreased making it much more acceptable to industry.
  
- 6/ The main drawback of the switching tree logic implementation is its limited tree height. This limitation, however, is removed when the switching trees are implemented in GaAs technology due to the fact that GaAs devices do not suffer from the parasitic capacitances associated with CMOS devices. Another reason for examining switching trees in GaAs technology is that most GaAs designs are limited to the number of devices on a given chip. Hence switching trees would dramatically increase the functional complexity being designed in GaAs if tree heights of 10 or more were implemented. It is suggested that a research project be started to examine the implementation of switching trees in a commercial GaAs technology, currently available to the university community through the fabrication services of the Canadian Microelectronics Corporation.



## REFERENCES

- [1] Adel S. Sedra and Kenneth C. Smith, MICROELECTRONIC CIRCUITS, Second Edition, pp 862-867, New York, NY 1987.
- [2] J. A. Pretorius Et. Al., "Latched Domino CMOS Logic", IEEE Journal of Solid-State Circuits, pp 514-522, January 1986.
- [3] Niraj K. Jha and Qiao Tong, "Testing of Multiple-Output Domino Logic (MODL) CMOS Circuits", IEEE Journal of Solid-State Circuits, Vol. 25, No.3, pp 800-805, June 1990.
- [4] J. A. Pretorius, A.S. Shubat and C. A. T. Salama "Charge Redistribution and Noise Margins in Domino CMOS Logic", IEEE Trans. on Circuits and Systems, Vol. CAS-33, pp 786-793.
- [5] N. P. Goncalves and H. J. de Man "NORA: a racefree dynamic CMOS technique for pipelined logic structures", IEEE Journal of Solid-State Circuits, Vol. SC-18, pp 261-268, June 1983.
- [6] Masakazu Shoji, CMOS Digital Circuit Technology, Prentice Hall, New Jersey, copyright 1988. (pp 233-240) (621.381'73--dc19)
- [7] Jiren Yuan and Christer Svensson, "High-Speed CMOS Circuit Techniques", IEEE Journal of Solid-State Circuits, Vol. 24:1, pp 62-70, February 1989.
- [8] Rogers Grondin, "WoodChuck, A Switching Tree Compiler Software", Faculty of Graduate Studies, University of Windsor, 1991.
- [9] Bruce Erickson, "Investigation of a Programmable Systolic Cell Using The RNS", Faculty of Graduate Studies, University of Windsor, 1990.
- [10] William R. Blood Jr., "MECL System Design Handbook", fourth edition, Motorola Semiconductor Products Inc., USA, 1983.
- [11] Staff Members, CMC, "Guide to the Integrated Circuit Implementation Services of the Canadian Microelectronics Corporation", Canadian Microelectronics Corporation, Carruthers hall, Queen's University, Kingston, Ontario, Canada, January 1987.
- [12] Thomas E. Dillinger, VLSI Engineering, Prentice Hall, New Jersey, Copyright 1988. (Pg. 420-421)
- [13] Peter D. Bird, "The Application of Multi-Valued Logic To The Implementation of Residue Number System Hardware", Faculty of Graduate Studies, University of Windsor, 1987.
- [14] Dr. G. A. Jullien, personal communication, (July 1990).
- [15] Masakazu Shoji, "FET Scaling in Domino CMOS Gates", IEEE Journal of Solid-State Circuits, Vol. 20:5, pp 1067-1071, October 1985.

- [16] Thomas E. Dillinger, VLSI Engineering, Prentice Hall, New Jersey, Copyright 1988. (Pg. 422)
- [17] John T. Carr, "CMOS VLSI Design Concepts and Practice", Faculty of Graduate Studies, University of Windsor, 1988.
- [18] H. J. Schumacher, J. Dikken, and E. Seevinck, "CMOS Subnanosecond True-ECL Output Buffer", IEEE Journal of Solid-State Circuits, Vol. 25:1, pp 150-154, February 1990.
- [19] Barbara A. Chappel, Terry I. Chappel, Stanley E. Schuster, Herman M. Segmuller, James W. Allan, Robert L. Franch, and Phillip J. Restle, "Fast CMOS ECL Receivers With 100-mV Worst-Case Sensitivity", IEEE Journal of Solid-State Circuits, Vol. 23:1, pp 59-67, February 1988.
- [20] Masakazu Shoji, CMOS Digital Circuit Technology, Prentice Hall, New Jersey, copyright: 1988. (621.381'73--dc19)
- [21] David Royle, "Designer's Guide to Transmission Lines and Interconnections", EDN Electronic Technology for Engineers and Engineering Managers, Vol.13, pp 131-159, June 23, 1988.
- [22] T. Dreher, Fast Pulse Techniques, E-H Research Laboratories, Oakland California, 1969.
- [23] T. Dreher, Fast Pulse Test Fixtures, E-H Research Laboratories, Oakland California, 1969.
- [24] Christopher J. Stanghan and Brian M. MacDonald, "Electrical Characterization of Packages for High-Speed Integrated Circuits", IEEE Transactions on Components, Hybrids, and Manufacturing Technology, Vol. 8:4, pp 468-473, December 1985.

## APPENDIX A

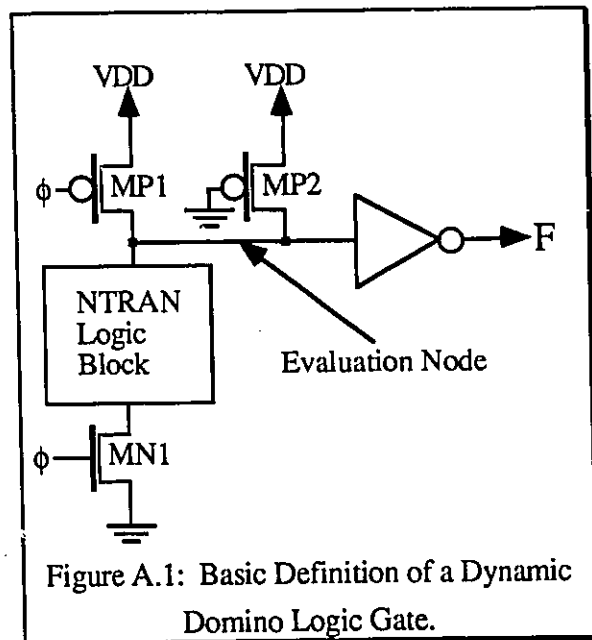
### A SUMMARY ON DYNAMIC DOMINO CMOS LOGIC

## APPENDIX A

### A SUMMARY ON DYNAMIC DOMINO CMOS LOGIC

The following summary of dynamic domino CMOS logic is based on CMOS Digital Circuit Techniques by Shoji [20].

A dynamic domino logic block consists of a dynamic logic gate whose output is inverted, as illustrated in Figure A.1. The dynamic logic gate requires two clocked devices, a PFET pull-up device, which is used to precharging the evaluation node, and an NFET pull-down device, which controls the start of the evaluation process. The role of the clock signal is to precharge the entire logic circuitry at the same time and then control



the start of the evaluation cycle of the first gate. Thus the NFET pull-down device can be removed from subsequent gates. A second high resistance (low W/L) PFET device, MP2, is also connected to the evaluation node. This device supplies current to keep the evaluation node indefinitely high if it is not discharged by the logic block after the start of the cells evaluation begins. This also keeps the output of the domino cell in a low impedance state by ensuring that the input to the inverter does not fall out of the noise margin when the NFET logic chain is off.

The Domino logic chain evaluates as follows. First, the falling edge of the clock signal initiates the start of the precharge cycle, which precharges all of the logic gates in the

series chain. Since the outputs of the domino gates are low in this state, all logic blocks, except the first, will have all transistors off. Then at the rising edge of the clock cycle the first gate in the series chain, illustrated in Figure A.2, is allowed to evaluate its input signals by the activation of the NFET pull-down device. If the logic gate discharges the evaluation node, then the output of the gate will go high which can then trigger the discharging of the second domino gate (assuming the logic block evaluates low). If the second gate also discharges its evaluation node then its inverter output also goes high continuing the chain of NFET logic block discharges. This Domino effect rippling evaluation continues through the domino logic gate chain until the correct output is obtained from the final gate. Hence, domino logic design removes the possibility of producing race conditions in cascaded logic circuits by ensuring a controlled fall of the evaluation nodes.

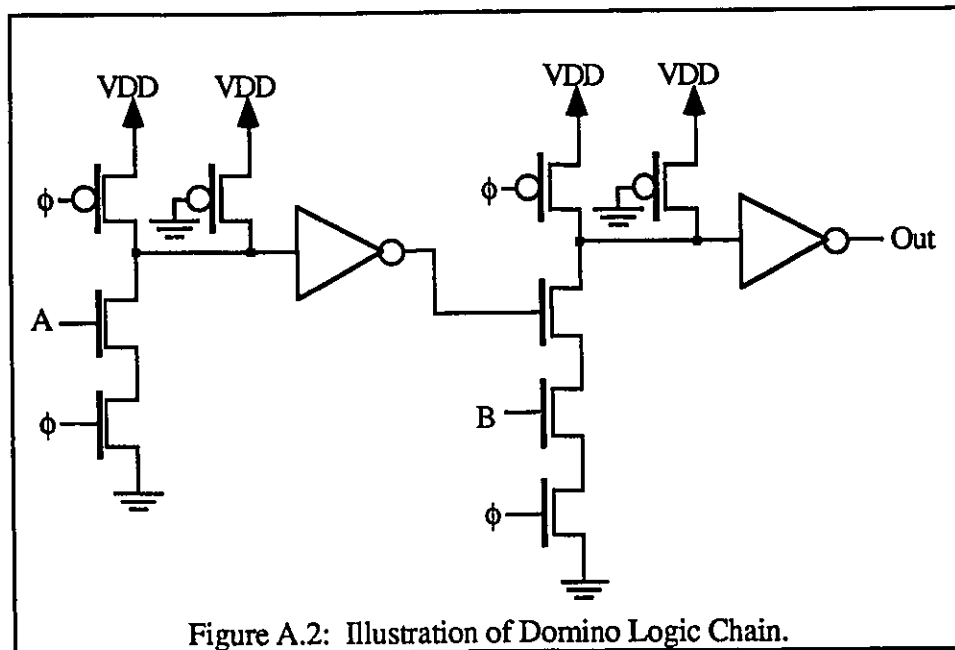


Figure A.2: Illustration of Domino Logic Chain.

One main drawback of the Domino logic gate is that it may only perform noninverting functions. As a consequence of this, not all logic functions can be implemented. For example, a domino logic gate cannot produce a signal and its complement from the same gate, since this would violate the noninverting definition of the

domino logic gate. Other functions may not be implemented if an odd number of inversions exist, as in the circuit shown in Figure A.3. The dotted boxes represent grouped gates which together form noninverting functions which can be implemented by a dynamic domino logic gate. The shaded inverter in the center of the schematic cannot be implemented due to the fact that it represents an odd inversion in the logic chain.

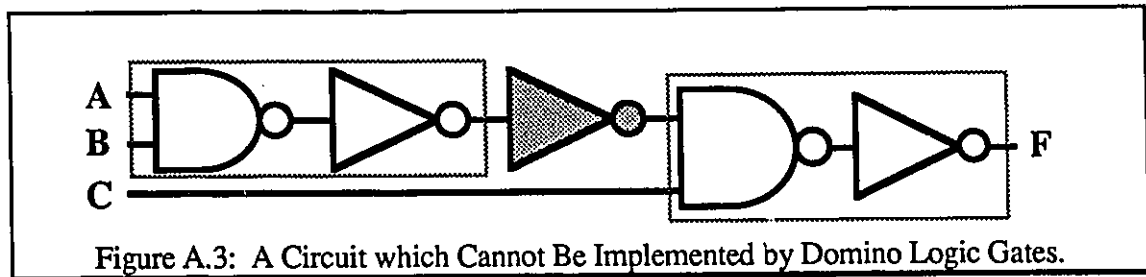


Figure A.3: A Circuit which Cannot Be Implemented by Domino Logic Gates.

Even with this limitation, Domino logic design is a powerful tool in implementing area efficient, fast, combinational logic designs.

## APPENDIX B

### COMPLETE SCHEMATIC DIAGRAM OF ROM5 CELL

## APPENDIX B

### COMPLETE SCHEMATIC DIAGRAM OF ROM5 CELL

The complete schematic representation of the ROM5 cell is illustrated in Figures B.1 through B.7. Figure B.1 represents the clock buffer cell for the

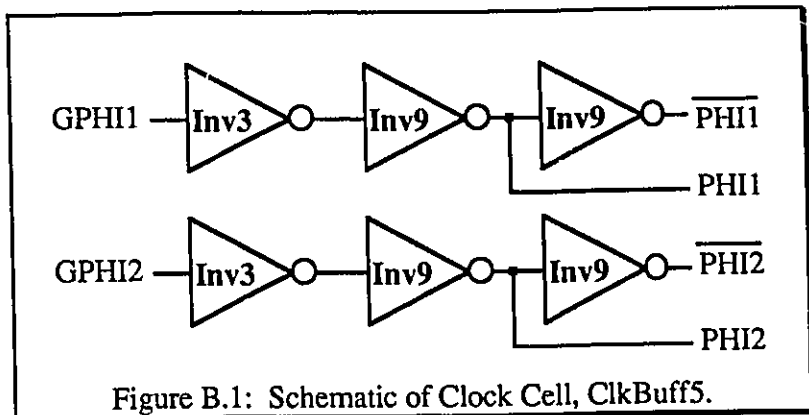


Figure B.1: Schematic of Clock Cell, ClkBuff5.

ROM5 cell. The inverters shown are scaled according to the number associated with their name. For example, the Inv3 inverter has devices which are 3 times minimum size, similarly, the Inv9 inverter has devices 9 times minimum size. GPIH1 and GPIH2 are the global clock signals entering into the cell.

The schematic in Figure B.2 represents a static CMOS design of a 2 to 4 decoder circuit. The four output signals are labelled C0 through C3. The input signals are labelled A0 and A4, along with their complements.

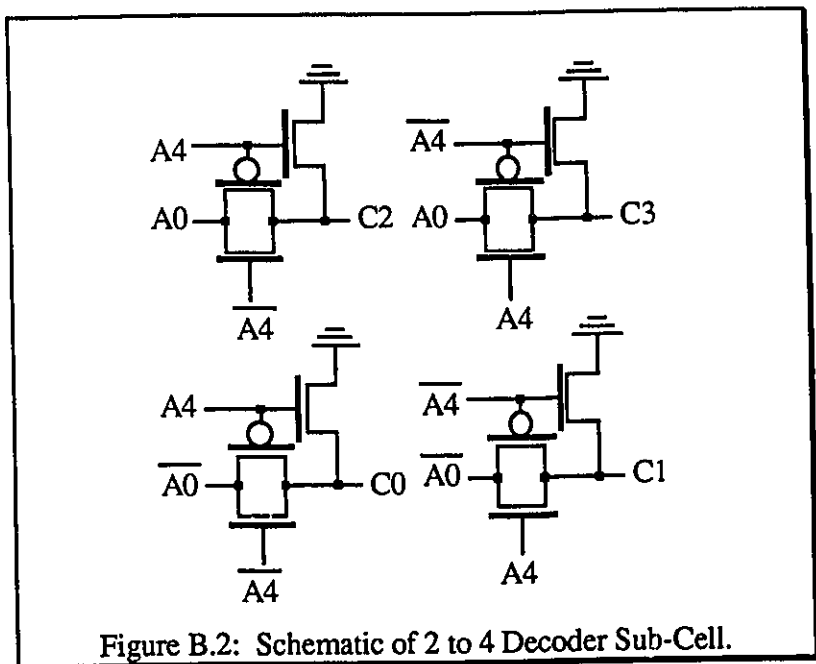


Figure B.2: Schematic of 2 to 4 Decoder Sub-Cell.



The Schematic in Figure B.3 represents a standard CMOS 3 to 8 decoder circuit. All input signals begin with the letter 'A', intermediate signals with the letter 'B', and output signals are labelled R0 through R7.

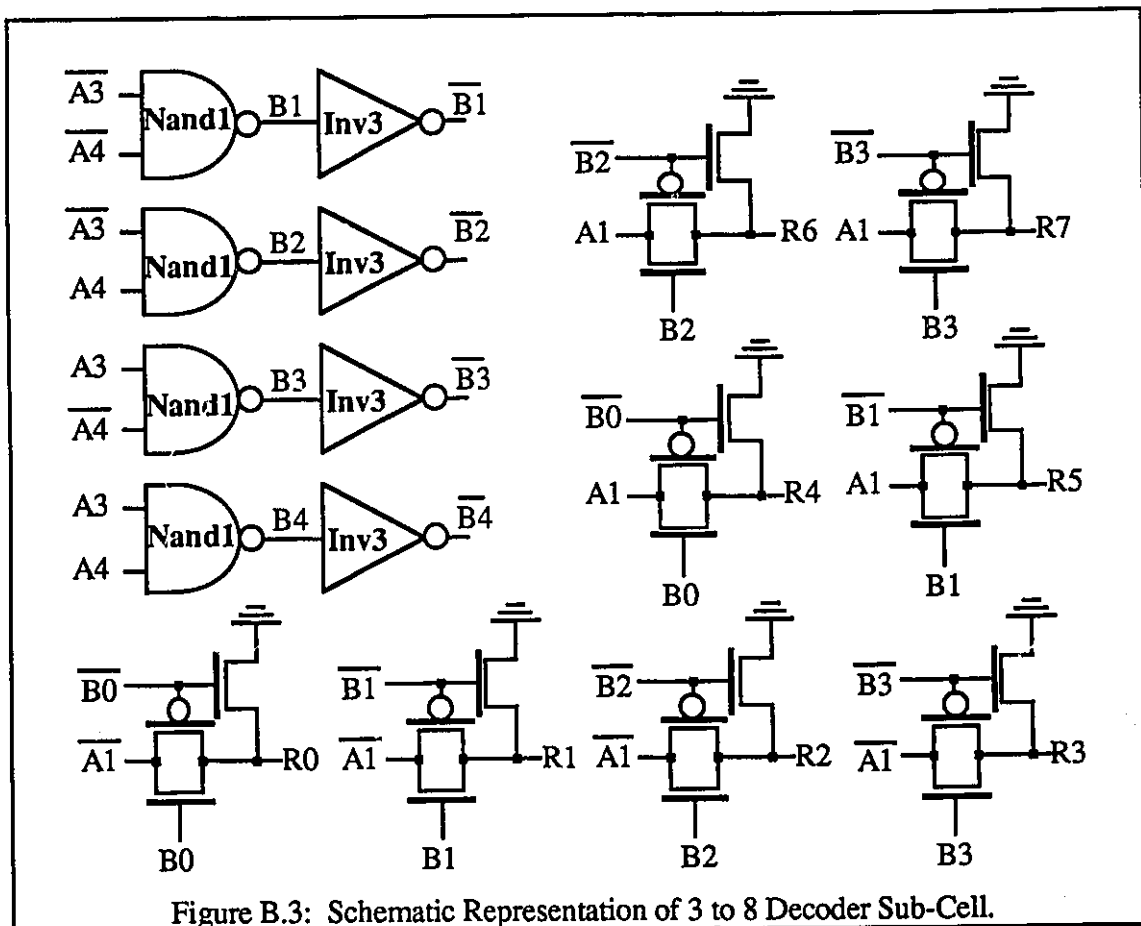


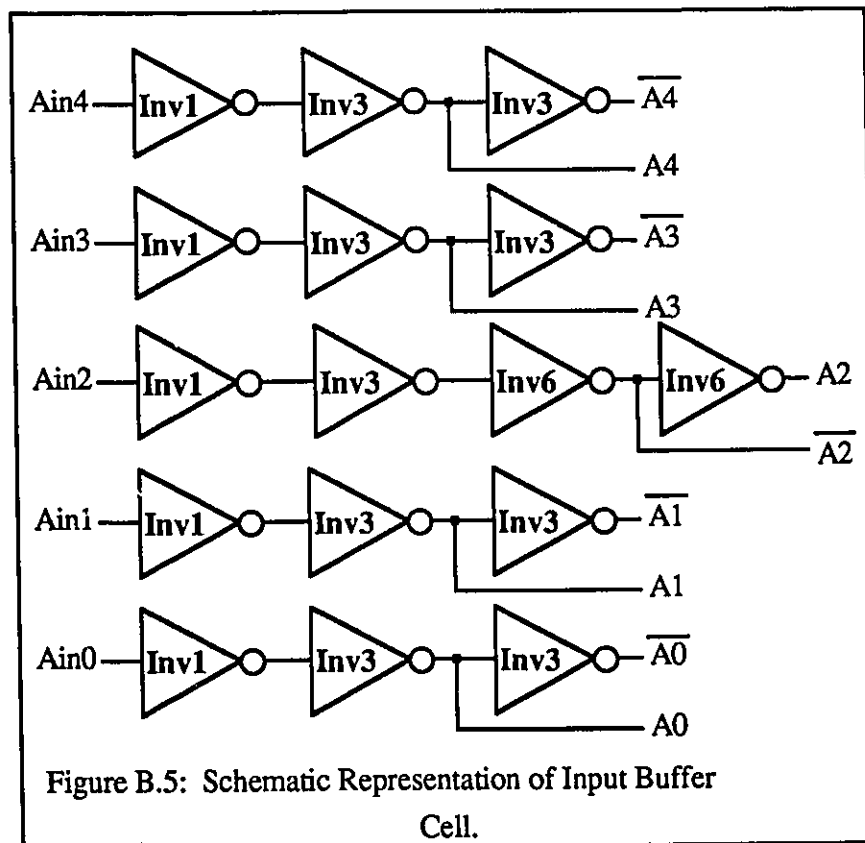
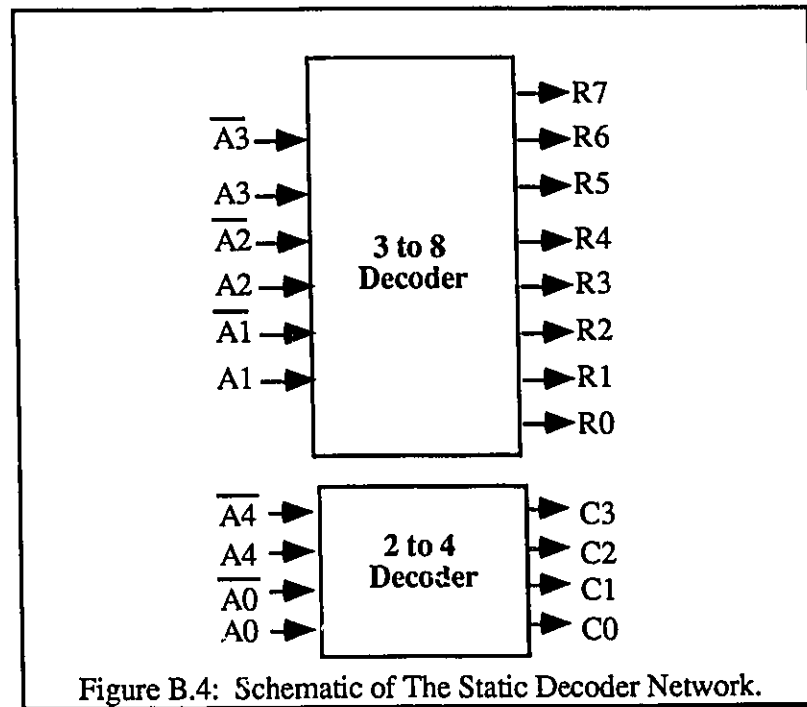
Figure B.3: Schematic Representation of 3 to 8 Decoder Sub-Cell.

The complete static CMOS decoder network used to address the ROM is shown in Figure B.4.

The input buffer circuit for the ROM5 cell is shown in Figure B.5. The global input signals are labelled  $A_{in0}$  through  $A_{in4}$ , while the internal cell signals are labelled  $A_0$  through  $A_4$  along with their complements.

The Structure of the ROM5 storage cell is illustrated in Figure B.6. All ROM row devices have gate signals beginning with the letter 'r', and all ROM column devices begin

with the letter 'c'. The structure consisting of three inverters and two transmission gates at the top of the schematic represents a two phase transmission gate latch structure.



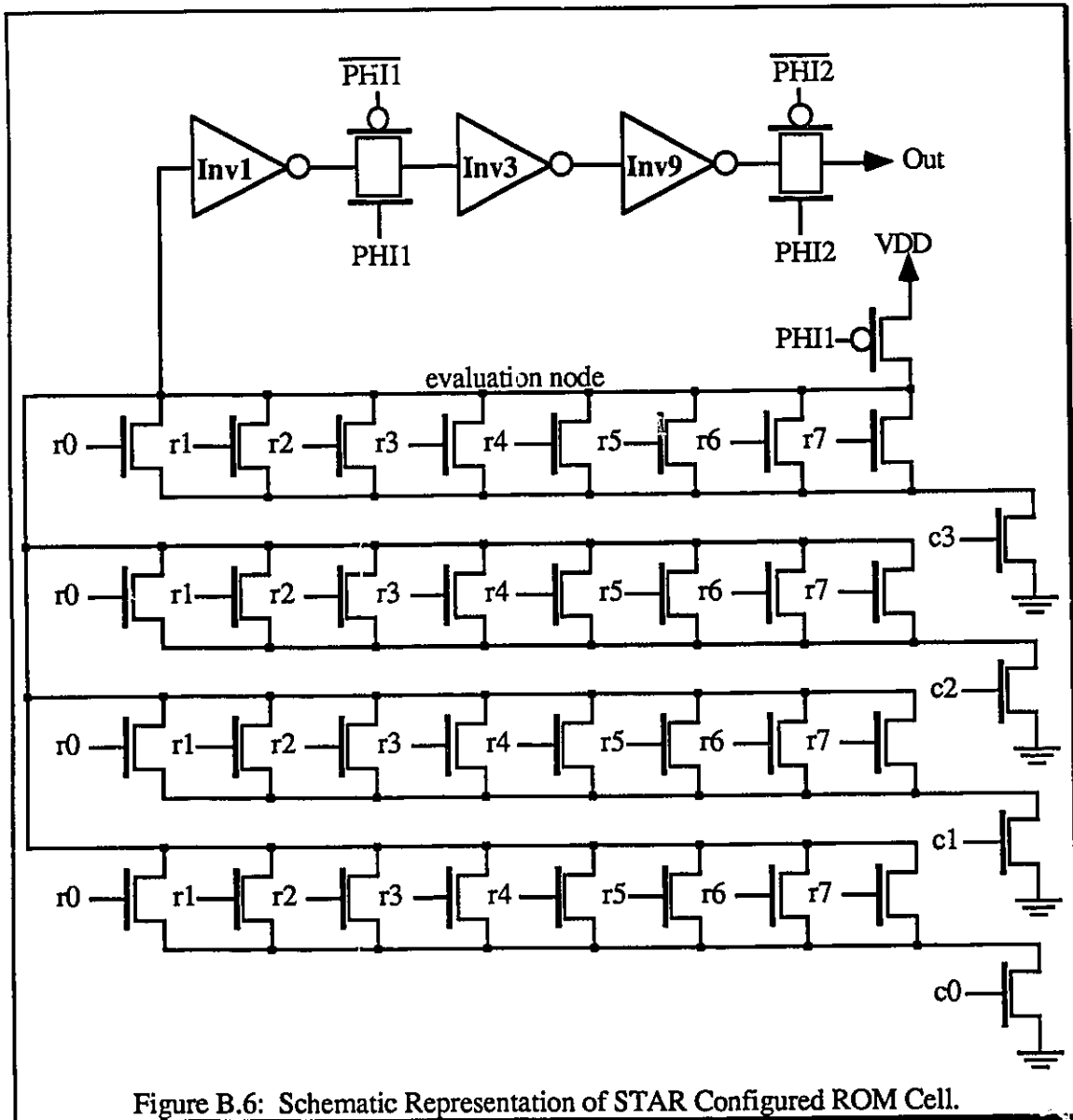


Figure B.6: Schematic Representation of STAR Configured ROM Cell.

The complete Block diagram of the ROM5 cell is illustrated in Figure B.7.

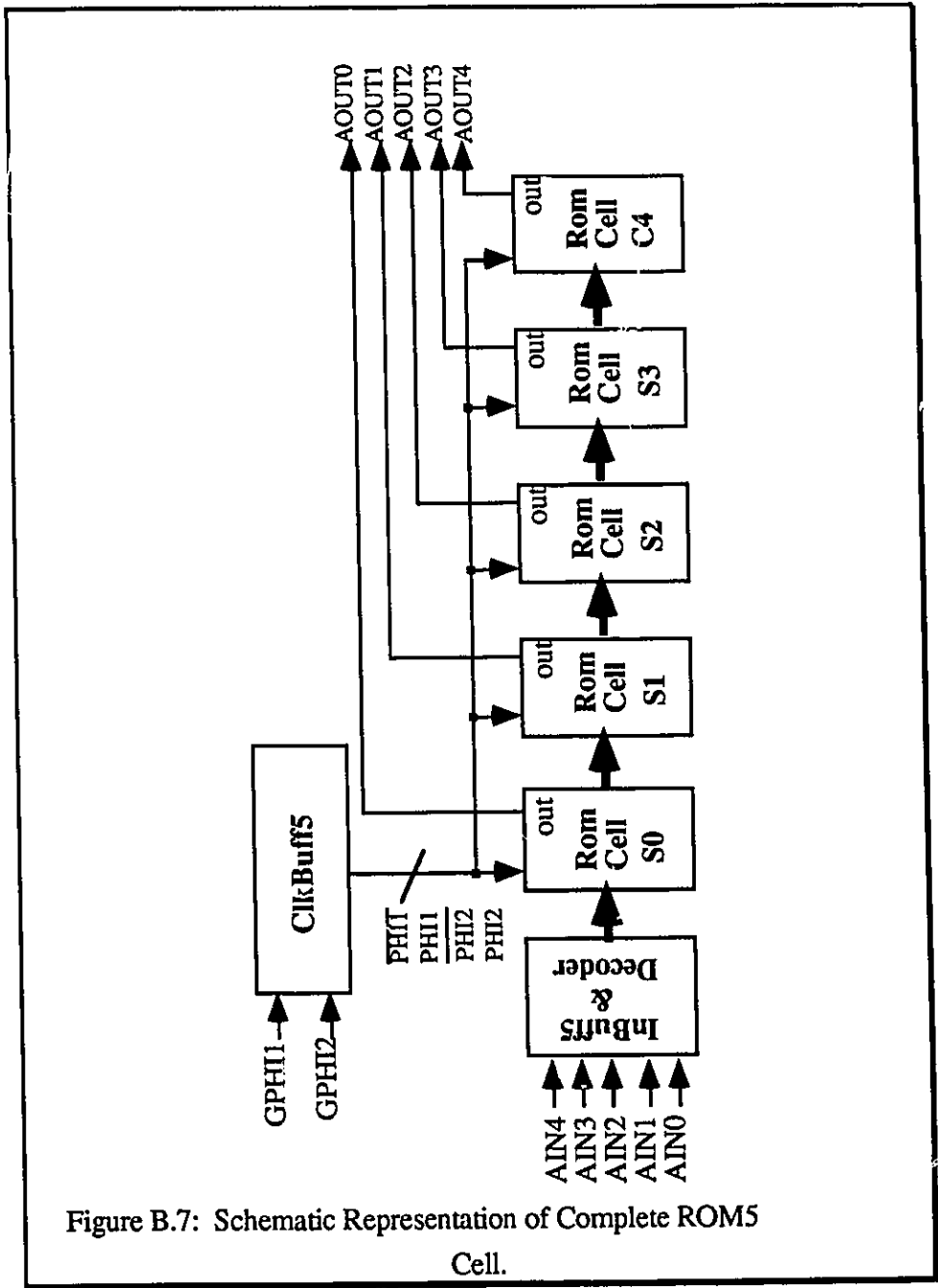


Figure B.7: Schematic Representation of Complete ROM5 Cell.

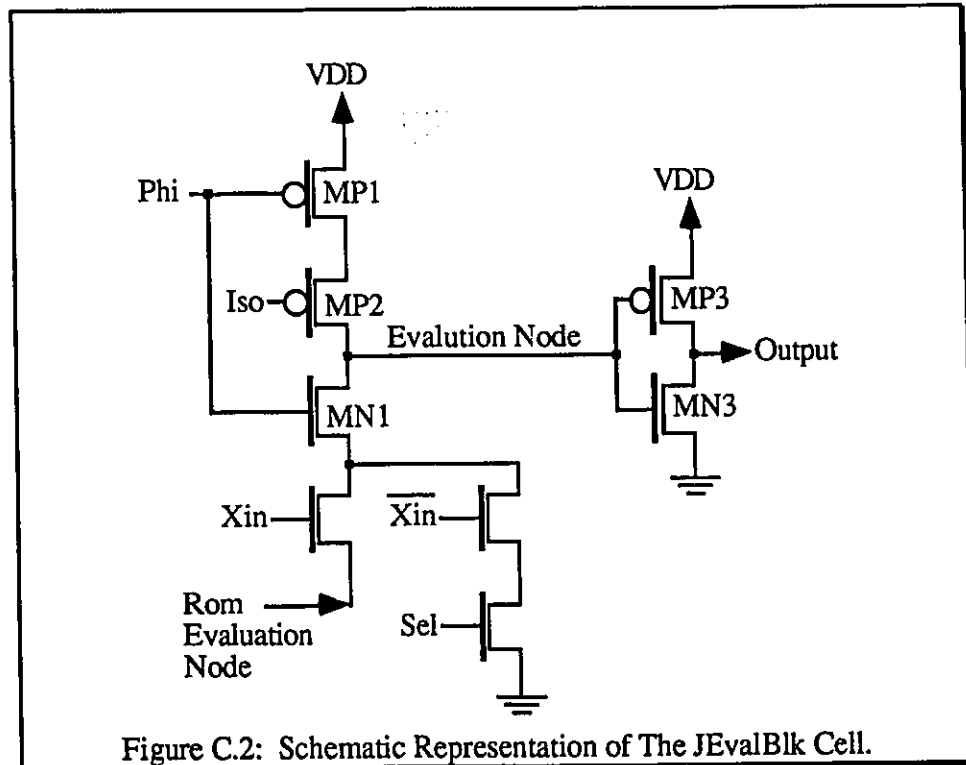
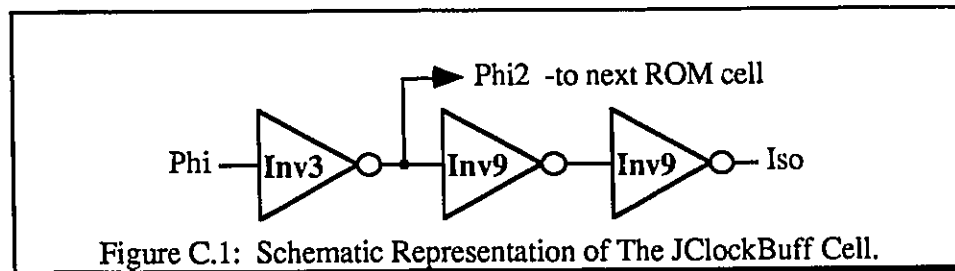
## APPENDIX C

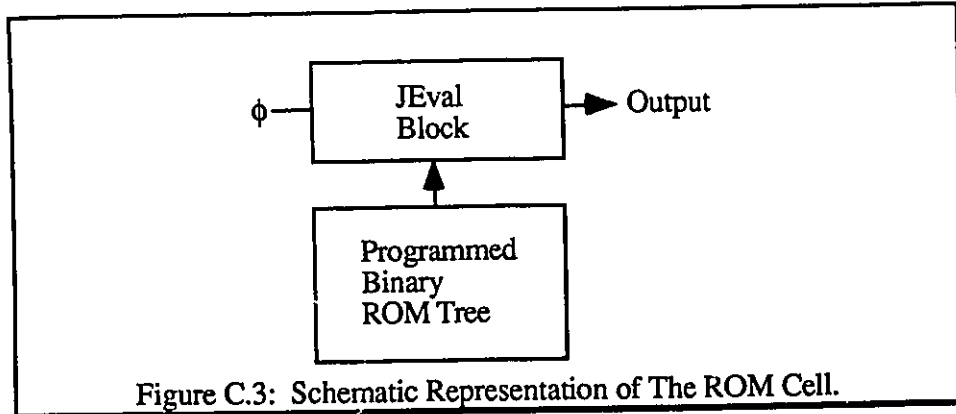
### COMPLETE SCHEMATIC DIAGRAM OF JROM CELL

## APPENDIX C

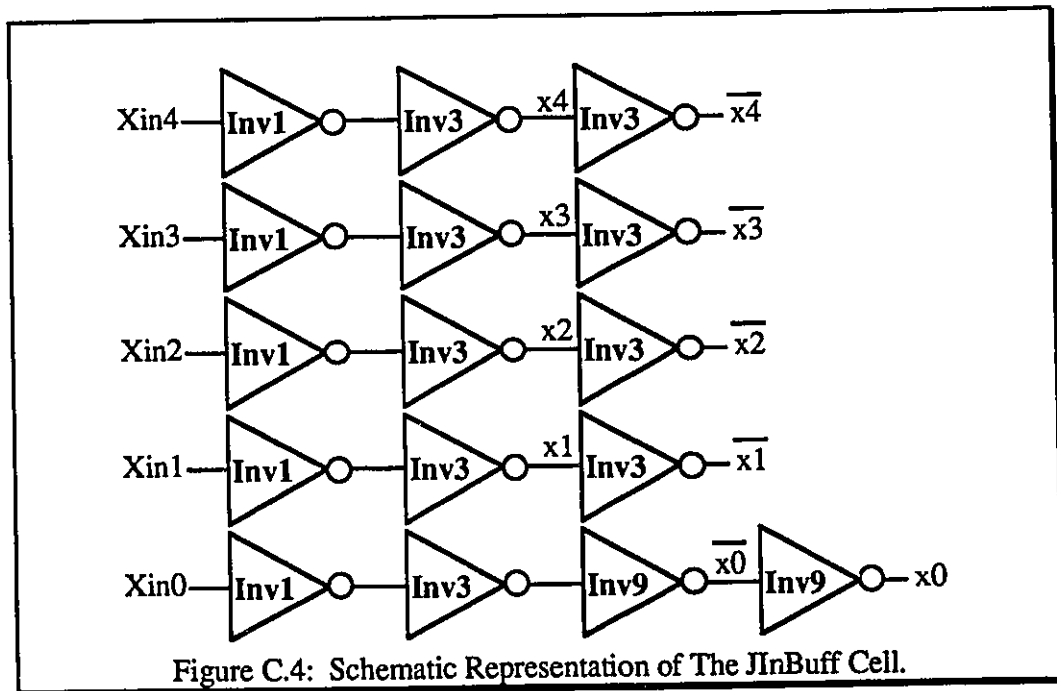
### COMPLETE SCHEMATIC DIAGRAM OF JROM CELL

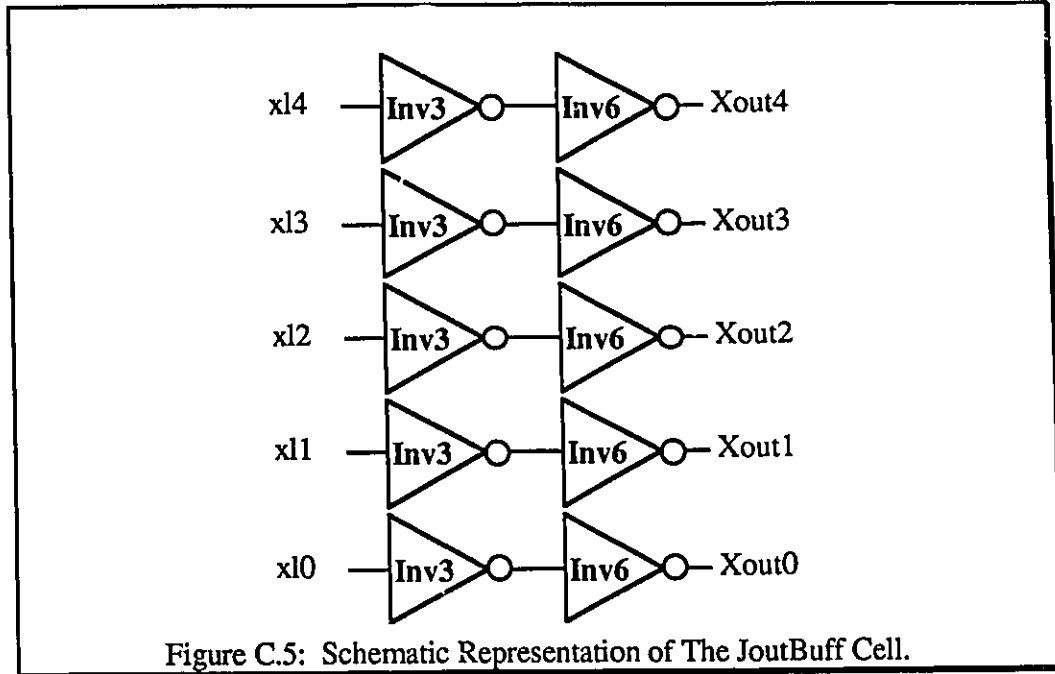
The complete schematic representation of the JROM cell is illustrated in Figures C.1 through C.6 inclusive. The signal, Phi2, becomes the Phi signal for the next ROM cell. The ROM evaluation structure used on the JROM cell is illustrated in Figure C.2. The Xin signal is used for RNS applications and those devices driven by that signal can be removed when implementing standard boolean functions.



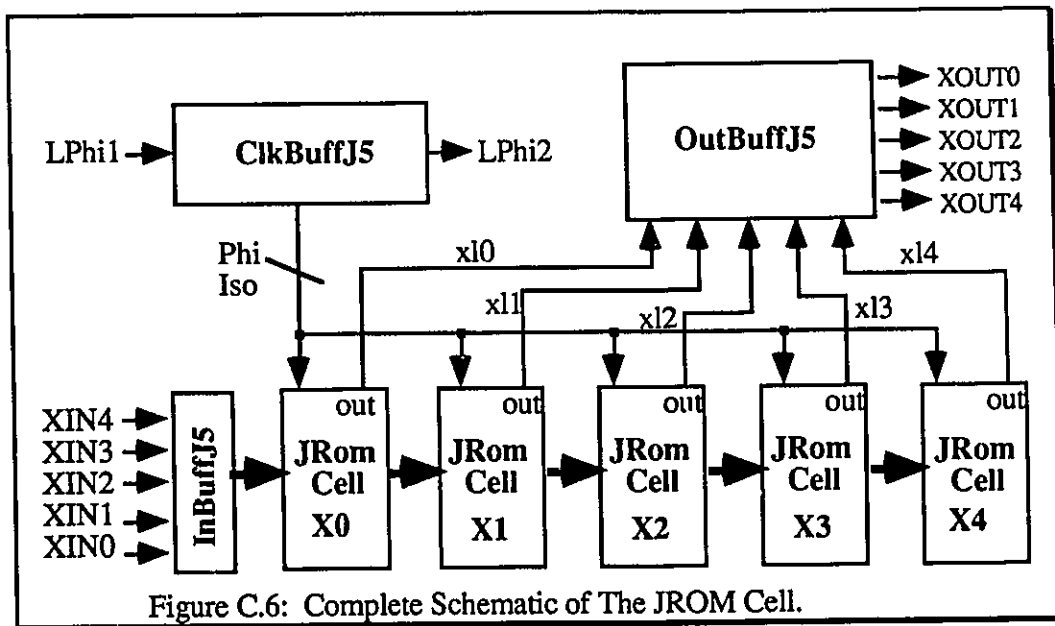


The Block diagram of the ROM cell is illustrated in Figure C.3, while Figure C.4 represents the input signal buffer circuit. The global input signals are labelled  $X_{in0}$  through  $X_{in4}$ . All internal local input signals are labelled  $X_0$  through  $X_4$  including their complements.





The Output Buffer circuit, JoutBuff, is illustrated in Figure C.5. The output signals from the ROM cells are labelled x10 through x14. The complete block diagram of the JROM cell is given in Figure C.6. This schematic represents a 5-bit input/ 5-bit output cell.





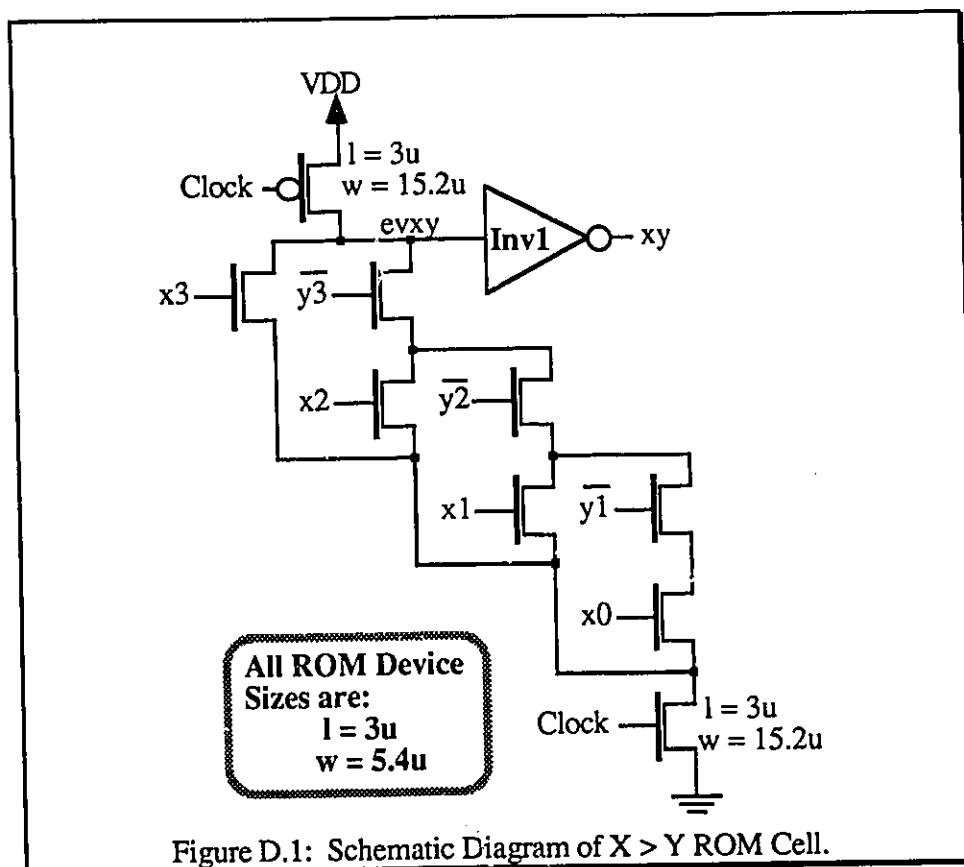
## APPENDIX D

### SCHEMATIC DIAGRAM OF 4-BIT COMPARATOR CELL

## APPENDIX D

### SCHEMATIC DIAGRAM OF 4-BIT COMPARATOR CELL

The dynamic domino CMOS schematics of a 4-bit comparator are illustrated in Figures D.1 through D.3 inclusive. Figure D.1 represents the domino gate implementation of the  $X > Y$  function, while Figure D.2 represents the domino gate implementation of the  $Y > X$  function. The domino gate implementation of the  $X = Y$  function is illustrated in Figure D.3. The three ROM cells comprise the logic core of a 4-bit comparator cell constructed out of dynamic domino logic gates. The input fields are  $x_0$  through  $x_3$  and  $y_0$  through  $y_3$ . The design requires the usual input and output buffering circuitry plus latches to create a fully pipelined structure.





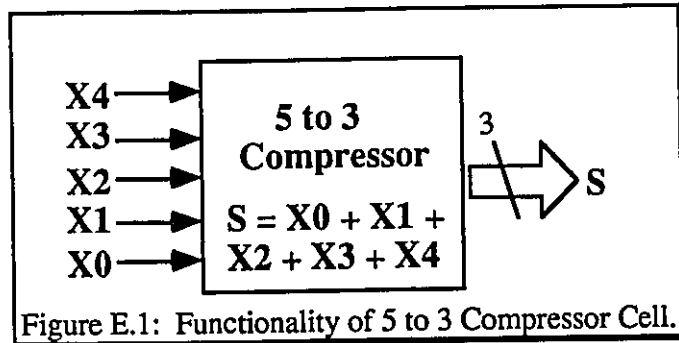
## APPENDIX E

### SCHEMATIC DIAGRAM OF 5 TO 3 COMPRESSOR CELL

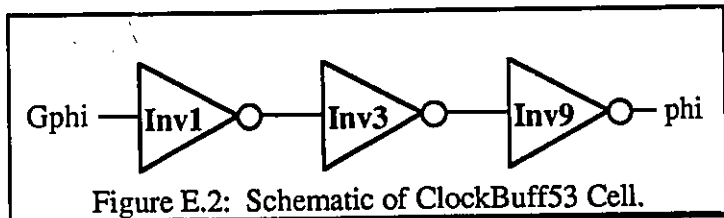
## APPENDIX E

### SCHEMATIC DIAGRAM OF 5 TO 3 COMPRESSOR CELL

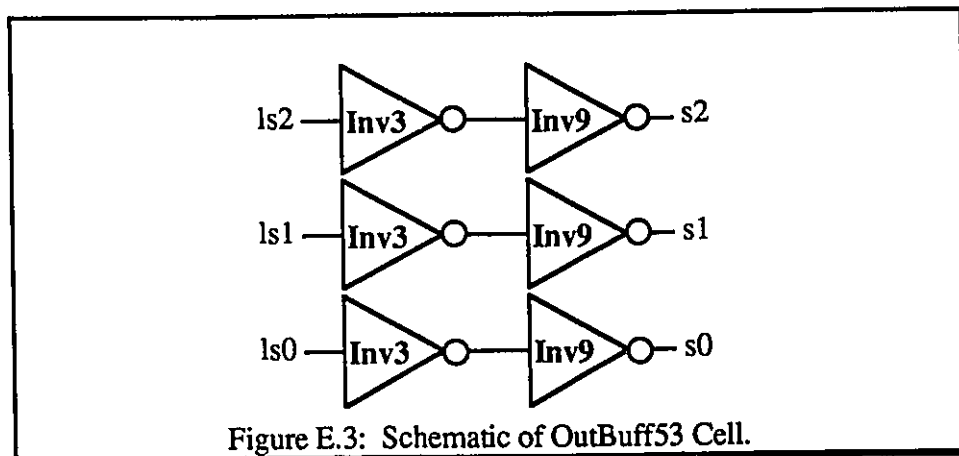
The functionality of the 5 to 3 compressor cell is illustrated in Figure E.1. The 5 to 3 compressor cell essentially takes five binary input signals, X0 through X4 and adds the signals



together to produce a 3-bit result, S. The complete schematic representation of the 5 to 3 compressor cell is



illustrated in Figures E.2 through E.8 inclusive. Figure E.2 represents the clock buffer circuit of the 5 to 3 compressor. The output buffer circuit for the 5 to 3 compressor is illustrated in Figure E.3. The Output of the ROM cells are labelled ls0 through ls2 and the cells output signals are labelled s0 though s2.



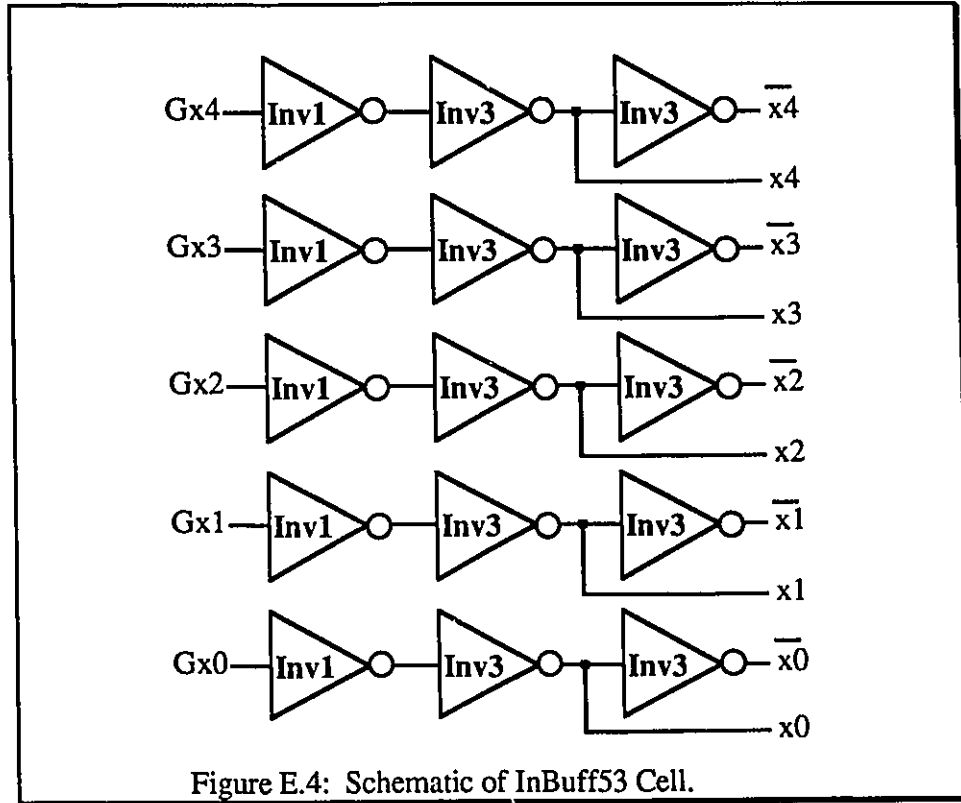
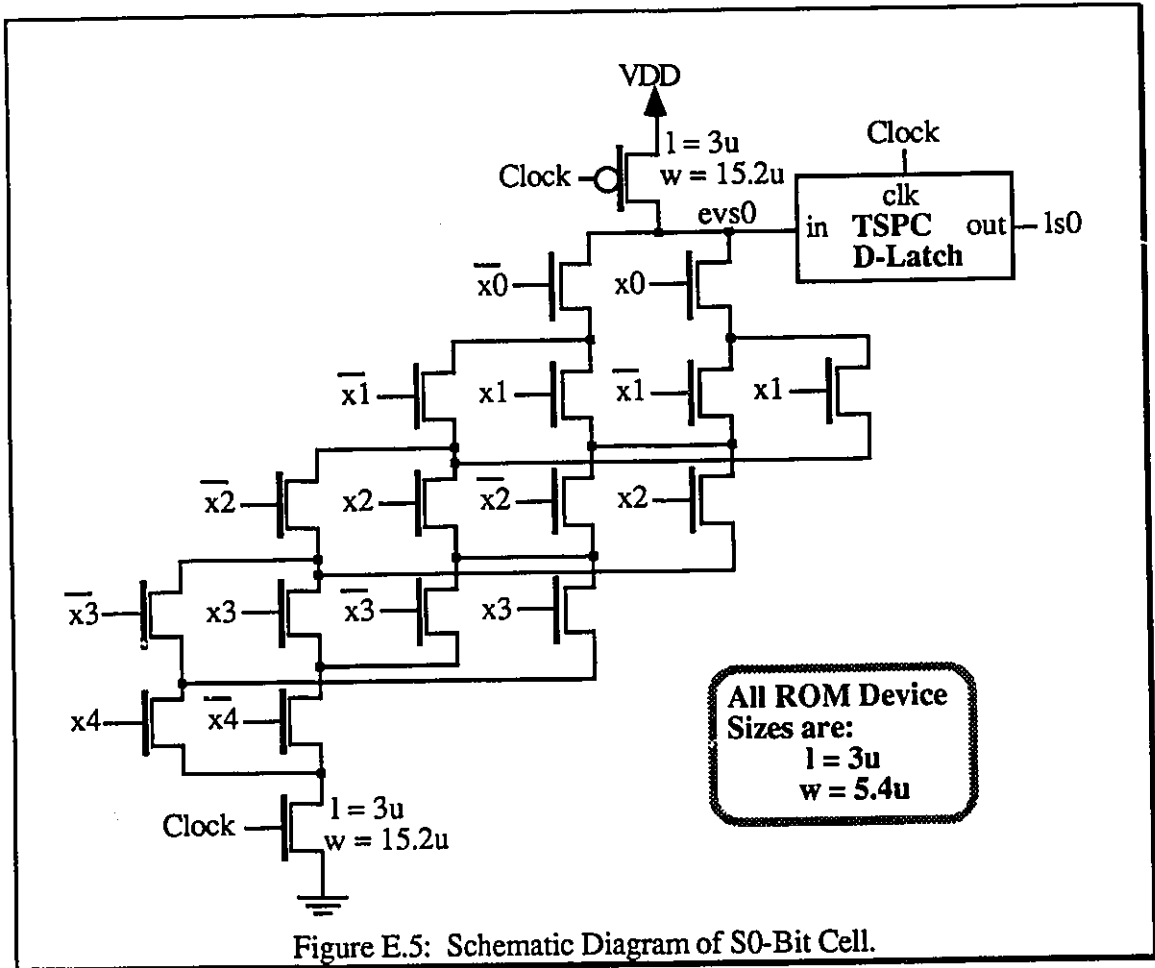
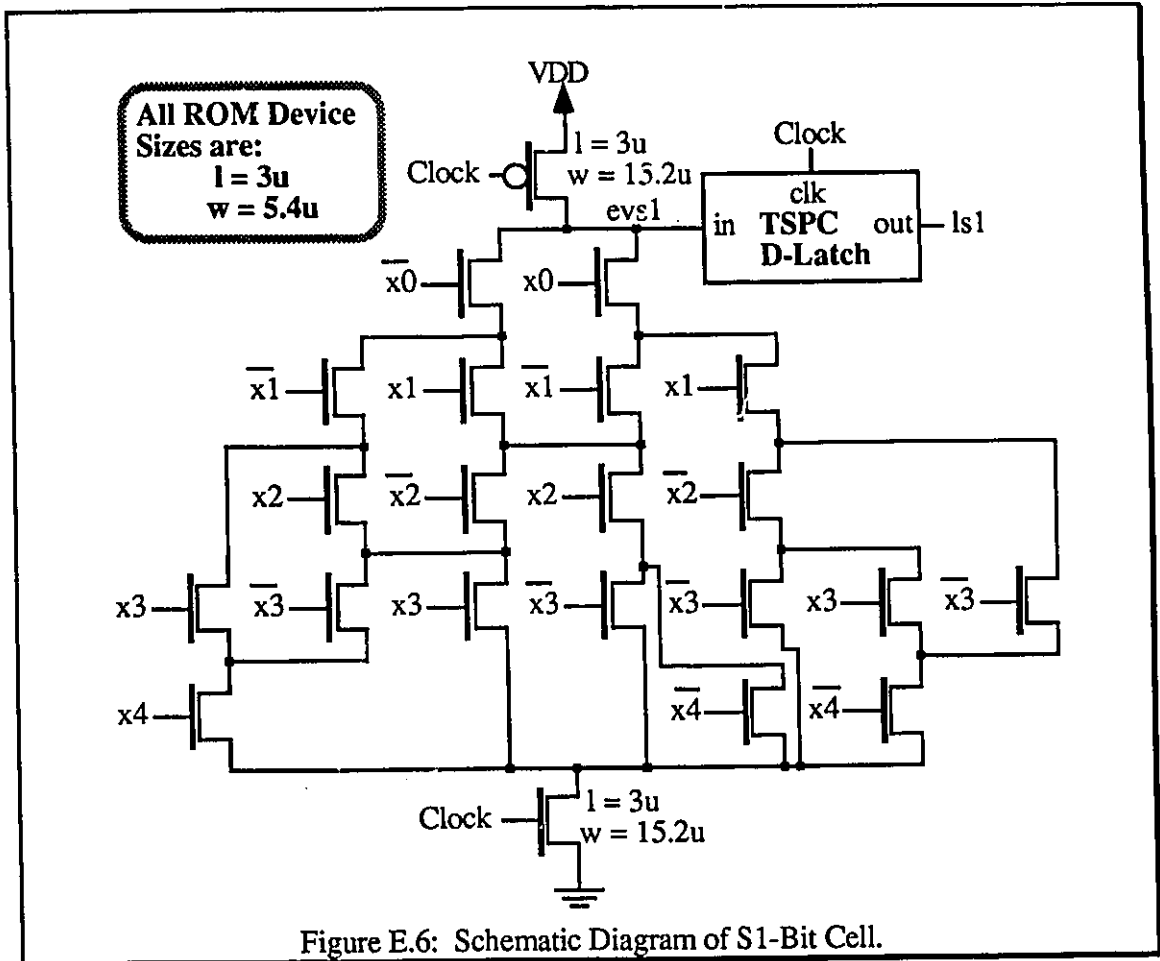


Figure E.4: Schematic of InBuff53 Cell.

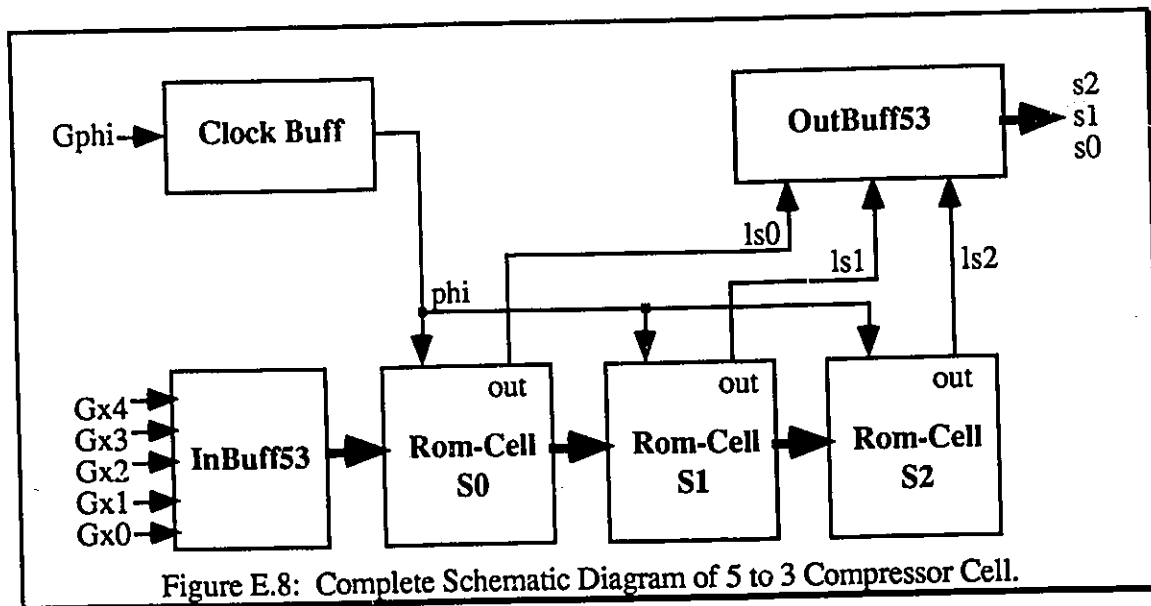
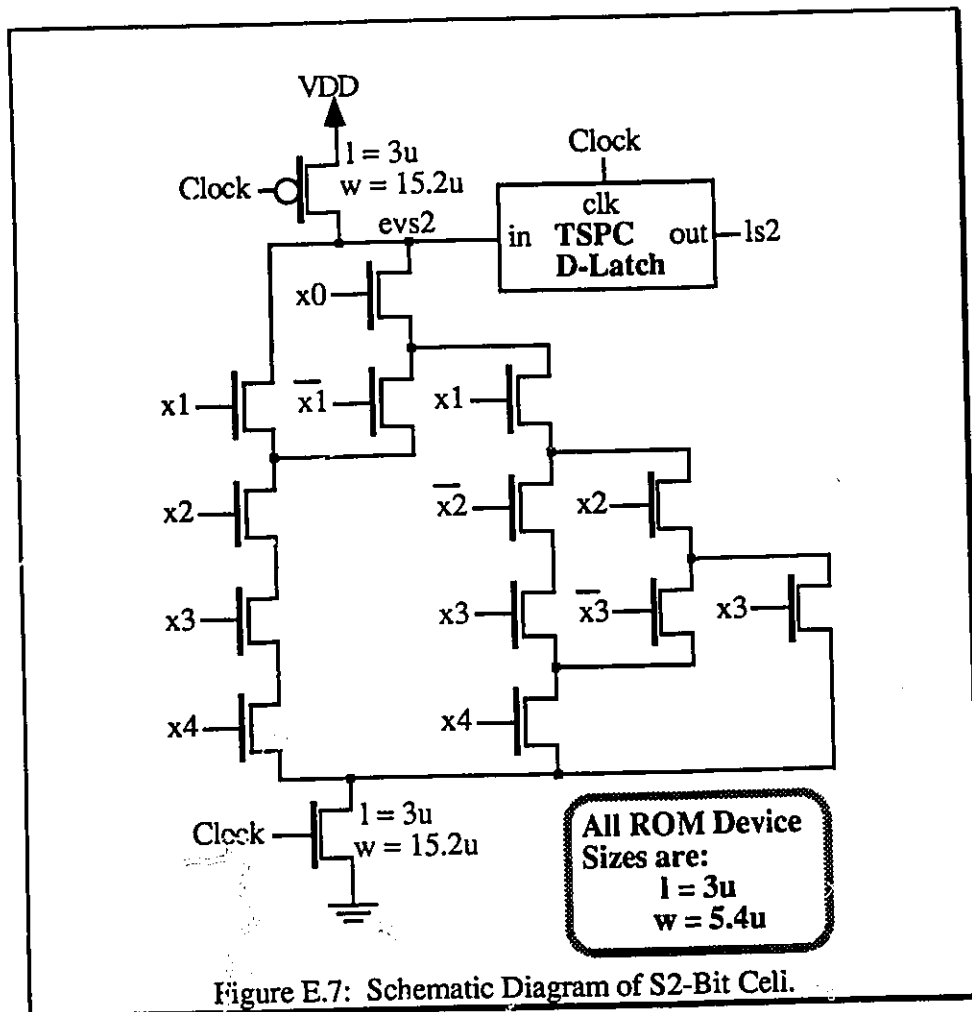
The input buffer cell for the compressor cell is shown in Figure E.4. The global input signals to the cell are labelled Gx0 through Gx4, while the cells local input signals are labelled x0 through x4 and their complements.



Figures E.5 through E.7 represent the three ROM cells for the 5 to 3 compressor cell. The complete Block diagram for the 5 to 3 compressor cell is illustrated in Figure E.8.







## APPENDIX F

### SUMMARY OF DEVELOPMENT OF 1.2 $\mu$ CMOS I/O PADS

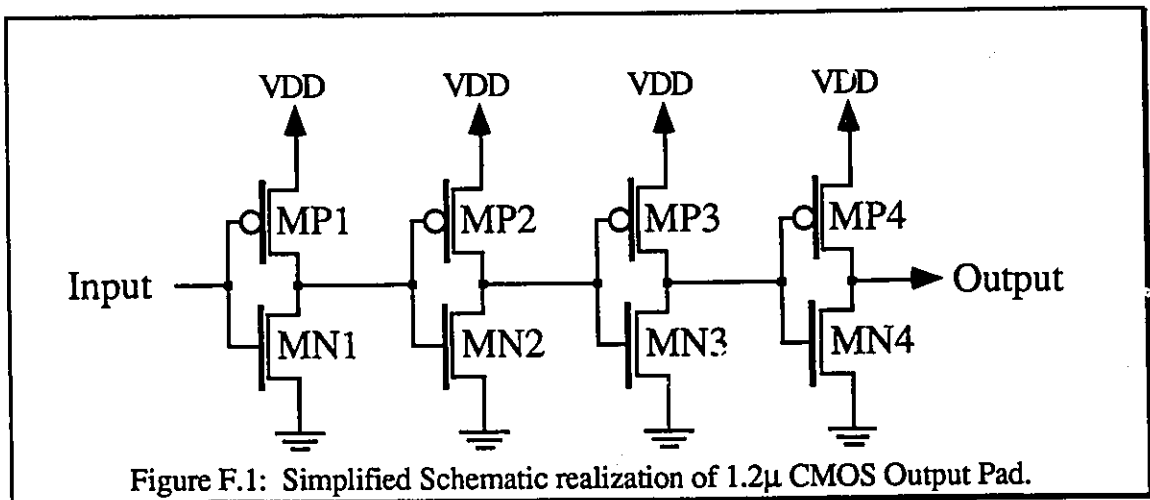
## APPENDIX F

### SUMMARY OF DEVELOPMENT OF 1.2 $\mu$ CMOS I/O PADS

The set of high-speed I/O pads interface circuitry described in this Appendix has been designed in the Northern Telecom 1.2 $\mu$  CMOS technology. The circuitry operates at full CMOS levels. The initial specifications are an output pad driver capable of driving a lumped capacitive load of 40 pf at speeds of up to 200 Mhz with a minimum output voltage swing of 4.3 volts; this meets the acceptable voltage swing level for CMOS devices. The output pad driver is modelled after a similar output pad built by John Carr in the 3 $\mu$  Northern Telecom CMOS technology. The input pad circuitry incorporates an input buffer along with static protection devices to sharpen and add drive capability to signals entering a chip.

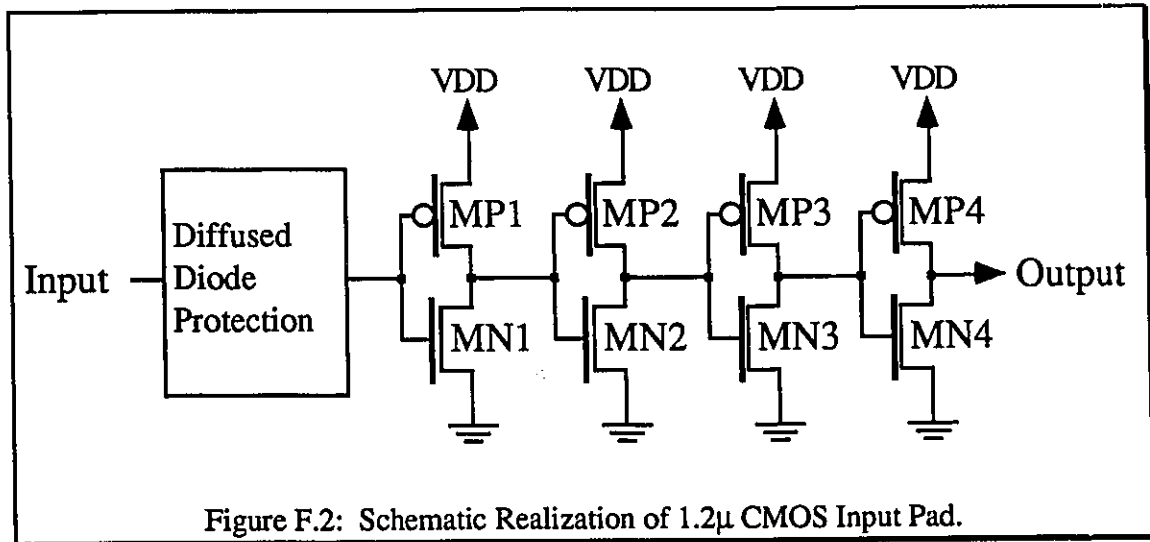
#### F.1 DESIGNING THE 1.2 $\mu$ CMOS OUTPUT PAD

The simplified schematic realization of the CMOS output pads is shown in Figure F.1.



## F.2 DESIGNING THE 1.2 $\mu$ CMOS INPUT PAD

The simplified schematic realization of the CMOS output pads is shown in Figure F.2. The pad utilizes a diffused diode device to provide ESD protection. The device was taken directly off of an 1.2 $\mu$  input pad designed by the University of Alberta as part of a standard pad set for the CMC (Canadian Micro-electronics Corporation) library. Once the input signal has past through the protection device it encounters an input buffer circuit composed of four scaled inverters. The size of each inverter increases by a factor of three, which provides the maximum amount of gain with the least amount of gate delay. The input buffer circuit has been added to the input buffer design to improve the input signal appearance and provide drive capability to drive internal circuitry.



Each device in the schematic is constructed from several parallel devices to reduce the parasitic node capacitance on the signal nodes and to create a composite device with a small 'on' resistance. The final stage of the output pad driver consists of 24 PFET devices with an effective gate width of 50.4 microns, constructed as 12 parallel devices with merged drains, and 24 NFET devices with individual gate widths of 20 microns per gate

constructed as 10 paralleled devices with merged drains. A ratio of three (3) is maintained between adjacent inverter stages to build drive capability.

### **F.3 DESIGNING THE POWER AND GROUND PADS**

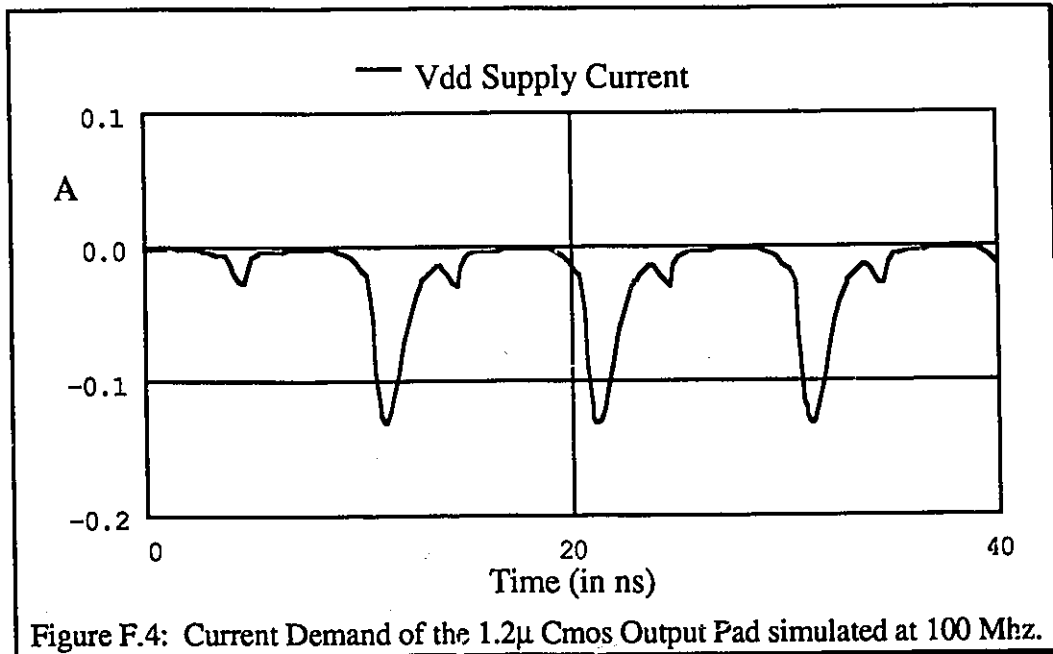
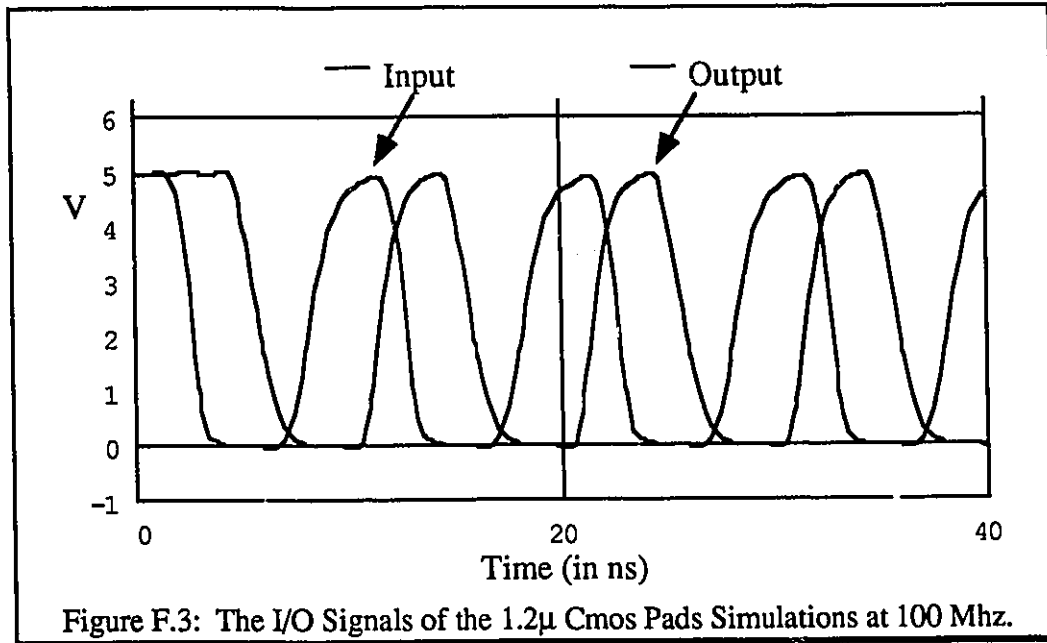
In addition to the input and output pad circuitry power and ground pads were also designed. The ground pad is nothing more than a simple wire connection from the bonding pad to the ground rail. The power pad has an integral reverse biased diode to act as a clamp on the power rail, limiting noise spikes to  $V_{DD}+V_t$  which is approximately 5.6 volts. Note the use of floating regions of isolated wells underneath each bonding pad region. This was an idea borrowed from the standard cell pad interface set generated by the University of Alberta. The floating well insures that if the bonding pad were to be punched through during the bonding process, the signal on the pad would not end up being connected directly to the substrate but rather only to the local area of the floating well region. This reduces the effects of the stray signal on the substrate region and guards against accidental short circuits.

### **F.4 SIMULATION RESULTS OF 1.2 $\mu$ CMOS PADS**

The output driver is designed to drive capacitive loads up to 40pF. The load size is within the load experienced when connecting test chips to the Asix tester. The Asix mother board places approximately 25pf of capacitive loading while the daughter board places an additional 15pf of capacitive loading. This information was obtained from Asix Inc.

A series of four scaled buffers have been designed where the last stage is capable of sufficiently driving this 40pF load. The input/output waveforms from the mask extracted SPICE simulation of the output buffer are shown in Figure F.3, and the corresponding current waveform shown in Figure F.4.

These simulation results use level one Spice parameter models, and no attempt has been made to model the inductive behavior of the power distribution network or output pad connection. The SPICE simulations indicate that the output pad will produce peak switching currents of 130 mA.



## F.5 THE TEST STRUCTURE ICAWRIOT

The complete pad set was connect together to produce a test chip comprised of a pair of buffers where the input pad directly drove the output pad. The name of the test chip was ICAWRIOT. The layout of ICAWRIOT can be seen in Figure F.5, the chip measures 1300 x 200 microns.

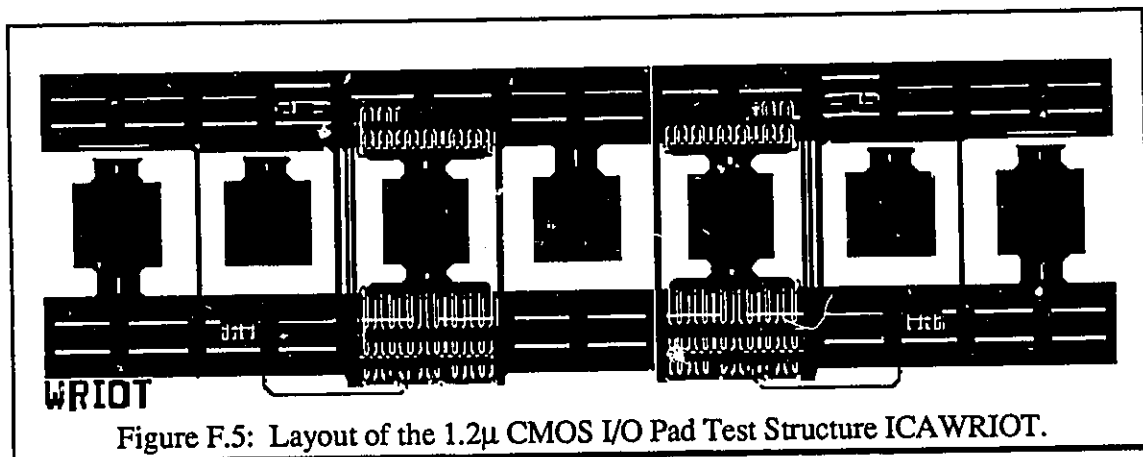


Figure F.5: Layout of the 1.2 $\mu$  CMOS I/O Pad Test Structure ICAWRIOT.

## F.6 TEST RESULTS OF THE TEST STRUCTURE ICAWRIOT

Primary attempts to the the test chip WRIOT using the standard Asix test DUT configuration failed due to poor power supply regulation combined with large amounts of injected switching noise generated by the large buffer stages of the output pads. The Vdd rail exhibited peak to peak noise levels of almost 4 volts. Thus a DUT board was created which utilized tantalum bypass and filter capacitors to insure the minimum amount of power supply noise, and a large lab grade power supply was used in place of the Asix internal supplies. The maximum peak to peak noise on the new DUT board was only 780 millivolts, illustrated in Figure. F.6. The pads were fully functional and test results are given in Table F.1.

The waveform of the output pad suffers from severe overshoot and undershoot during the switching of signal levels. The sequence of scope plots in Figures F.7 through F.9 illustrates the effect of insufficient current sourcing capability of the power rails as the switching speed is increased, which increases the current requirements of the output pad.

Table F.1: Summary of Test Data from Testing of ICAWRIOT	
Pad Delay (for I/O Pair)	= 5.4 ns
I <sub>dd</sub> (per I/O Pair)	= 16.08 mA @ 50 Mhz
	= 13.69 mA @ 25 Mhz
	= 5.67 mA @ 12.5 Mhz
Edge Times (Rise & Fall)	= Approx 2.0 ns/edge under 10.8pf load.

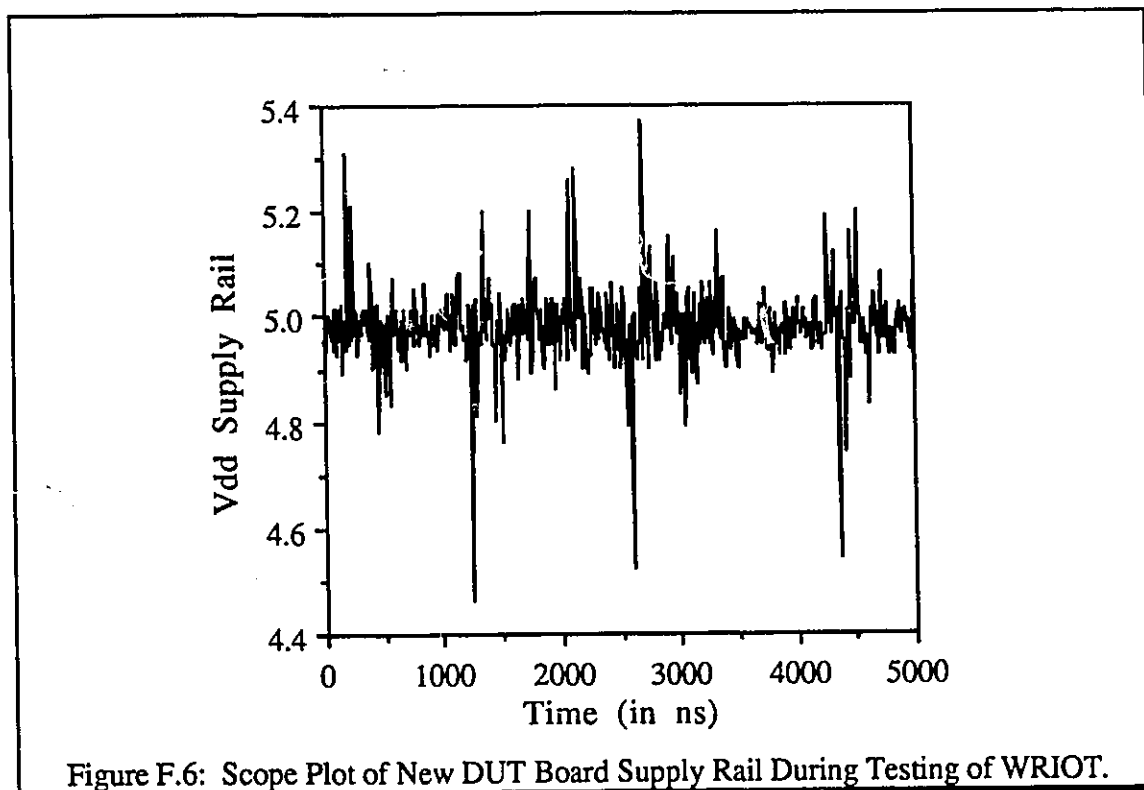
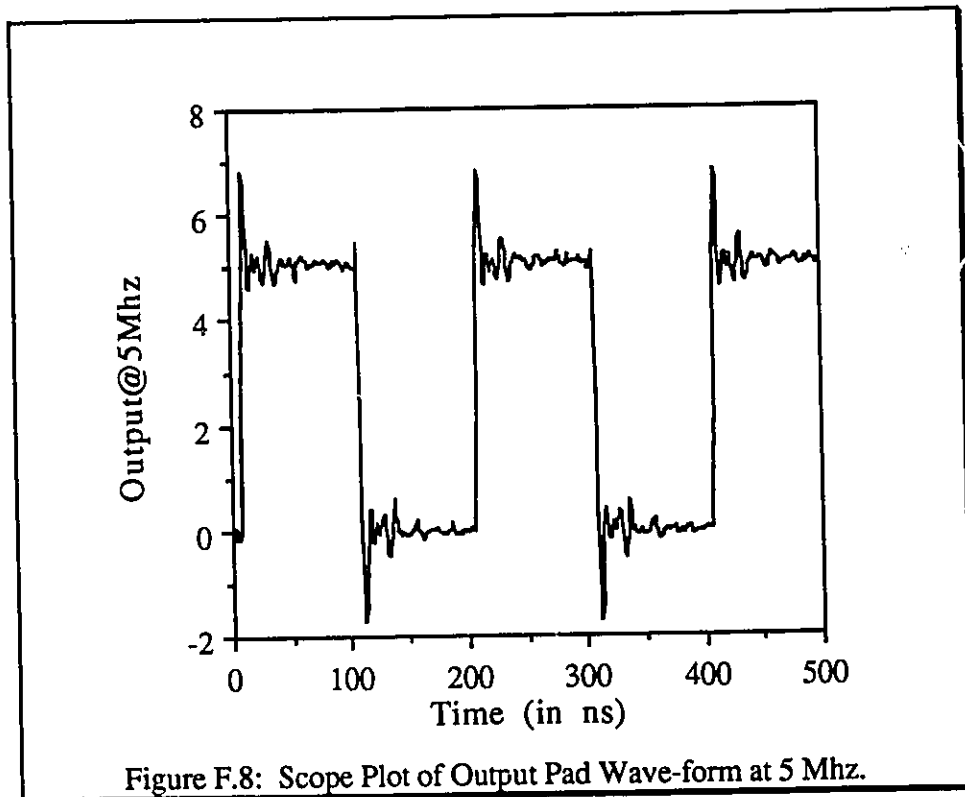
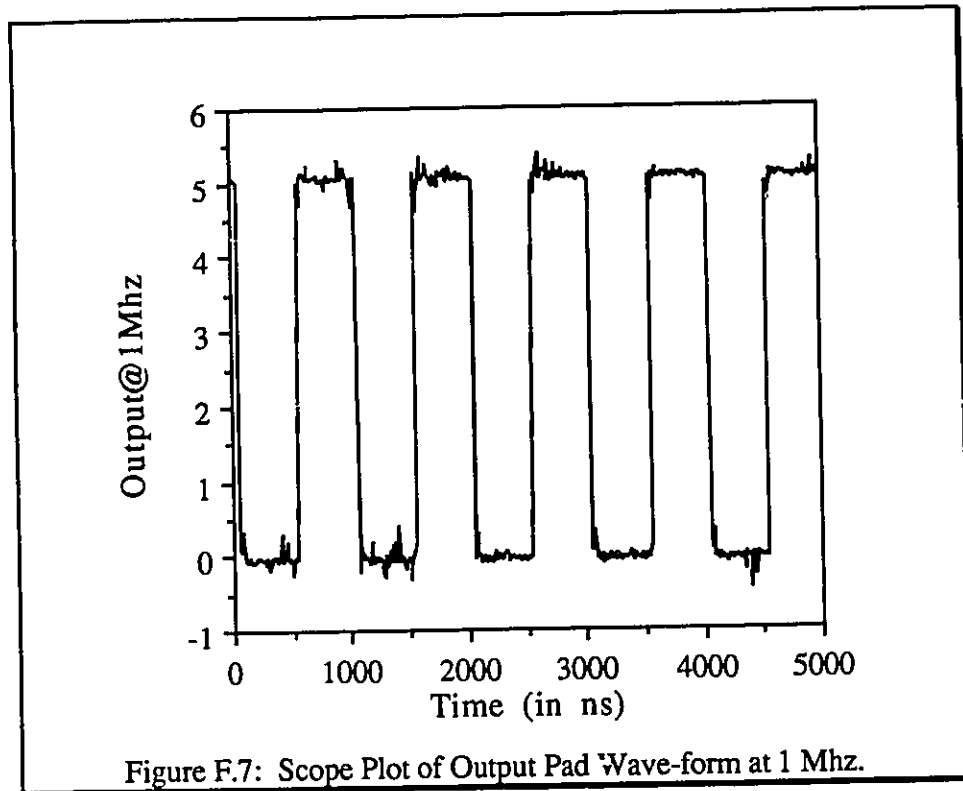
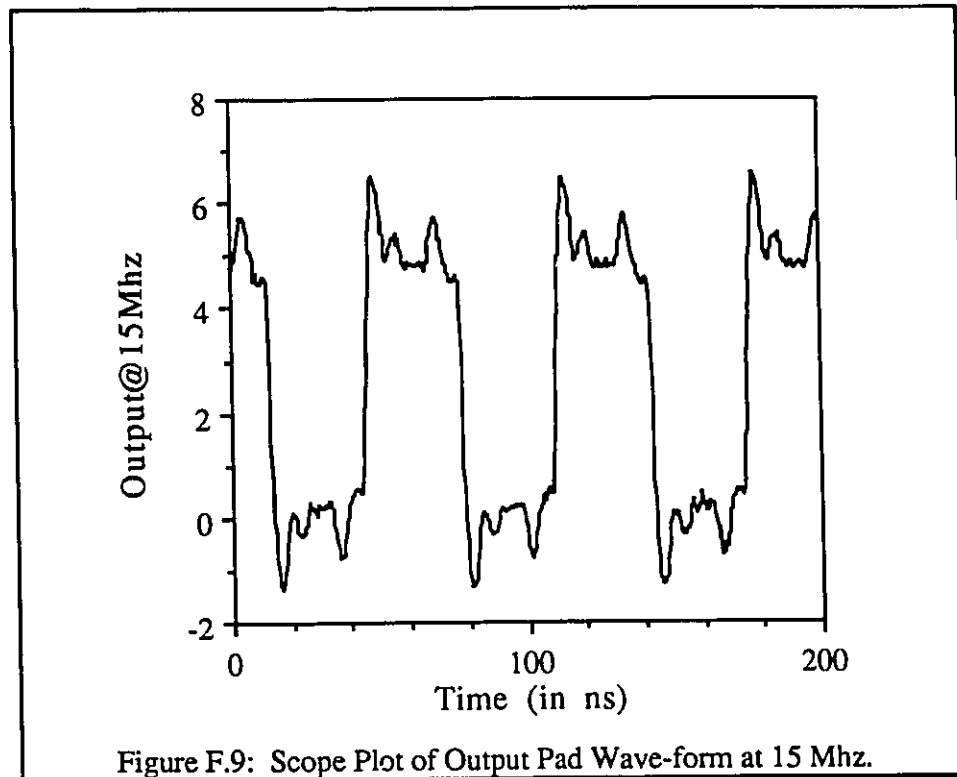


Figure F.6: Scope Plot of New DUT Board Supply Rail During Testing of WRIOT.



The first figure illustrates the output at 1 Mhz, the second at 5 Mhz, and the final at 15 Mhz.





In the  $3\mu$  CMOS pads drivers, designed by John Carr, the power and ground rails measured  $120\mu$  in width which was acceptable for driving approximately 20pf at speeds of up to 40 Mhz. The  $1.2\mu$  CMOS pads have power and ground rails of  $122.4\mu$ , not taking into consideration the slotting of the power rails. In retrospect, these rails do not seem capable of carrying enough current to drive 40pf at speeds of up to 100Mhz.

Thus it was decided to reduce the load requirements of the output pad down to loads of only 20pf which is still sufficient to drive some test equipment or scope probes, since these pads are being designed for academic use only and not to drive a 120pf back plane. Thus a second generation design will incorporate a smaller output stage requiring less switching current, and larger power rails which will be capable of providing sufficient switching current, and secondly a design which utilizes bootstrapping techniques to reduce the over and undershoot to reasonable levels.

## F.7 REDESIGN OF NEW I/O TEST STRUCTURE ICAWRIO2

A second test structure was designed to attempt to correct the poor switching performance of the first design. Since the over and undershoot phenomenon could not be directly modelled using SPICE, two possible design solutions were implemented to attempt to correct the problem. First the power and ground rails were widened by 50% to increase the available current reaching the output driver pad. A second

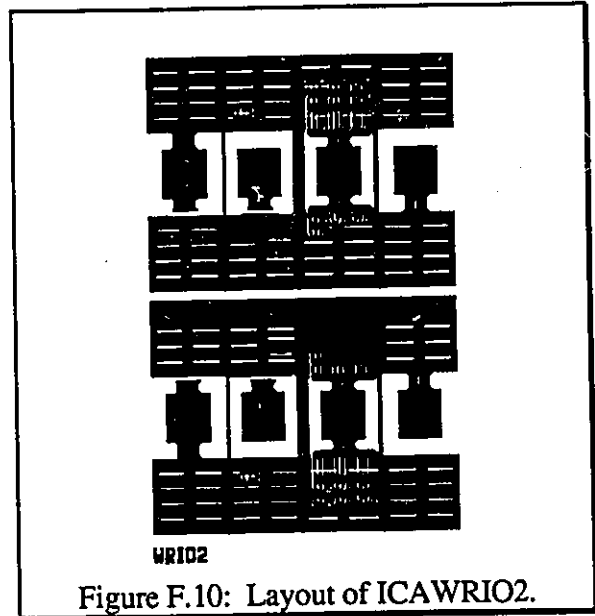


Figure F.10: Layout of ICAWRIO2.

solution, also implemented, utilizes the wider rails in conjunction with a slight decrease in the output pad device size to incorporate diode bootstrapping devices on both the power and ground rail which should, in effect, reduce the over and undershoot of the output signals to approximately 0.6 volts or one  $V_t$  above  $V_{DD}$  and below ground potential. The layout for the design is illustrated in Figure F.10. The design has been submitted for fabrication to CMC, but had not been returned in time to include the test results in this thesis.

## F.8 SUMMARY OF 1.2 $\mu$ CMOS PAD DEVELOPMENT

A set of full level swing CMOS pads were developed for the northern telecom 1.2 $\mu$  CMOS technology. The initial design of the output pad driver, though functional, proved to experience severe signal over and under shoot problems. Several possible solutions were looked at and implemented in an improved design, ICAWRIO2. One or both of

these solutions should improve the performance of these pads to adequate levels. The test structure was submitted for fabrication, but has not returned in time to be tested.

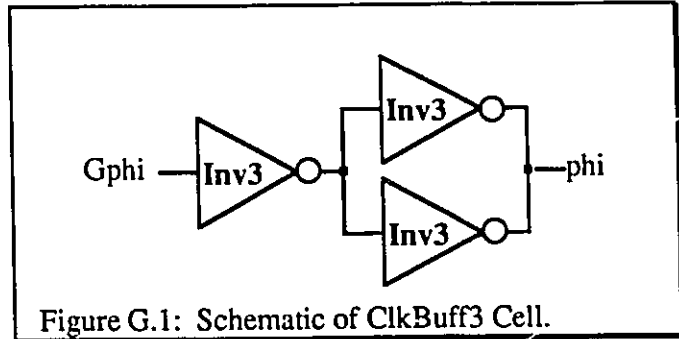
## APPENDIX G

### SCHEMATIC DIAGRAM OF 3-BIT ADDER CELL

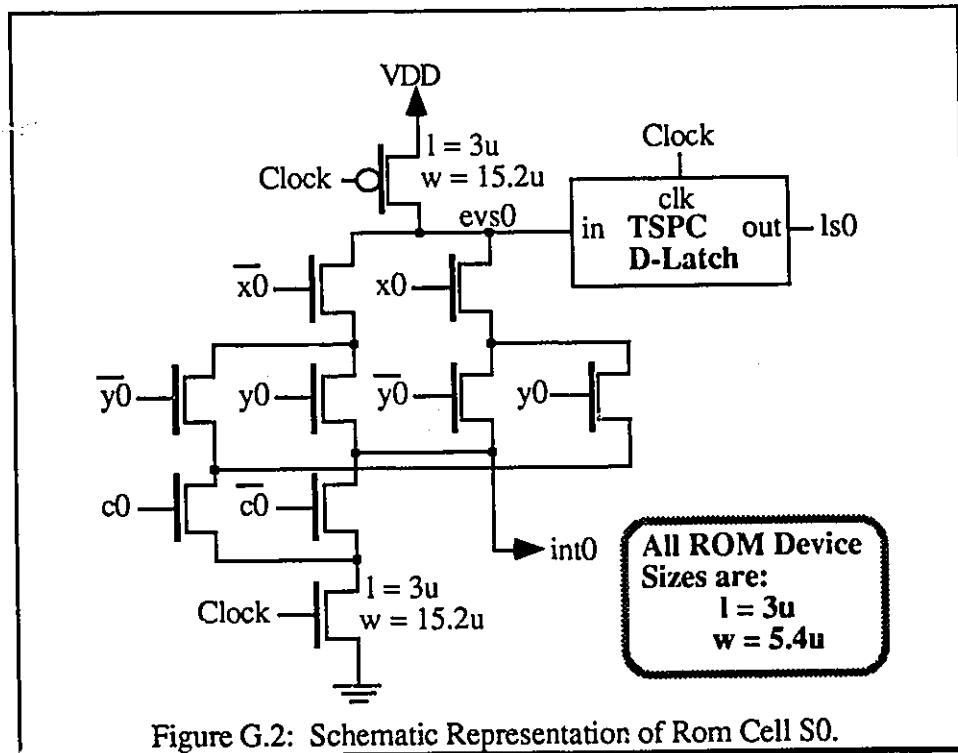
## APPENDIX G

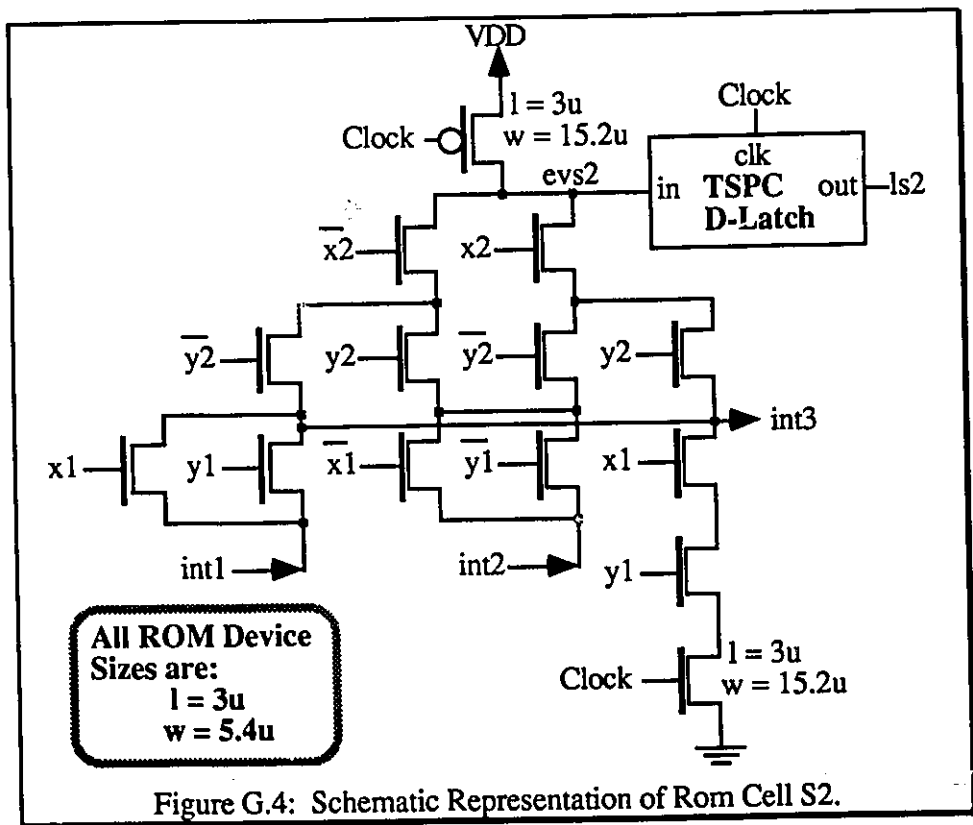
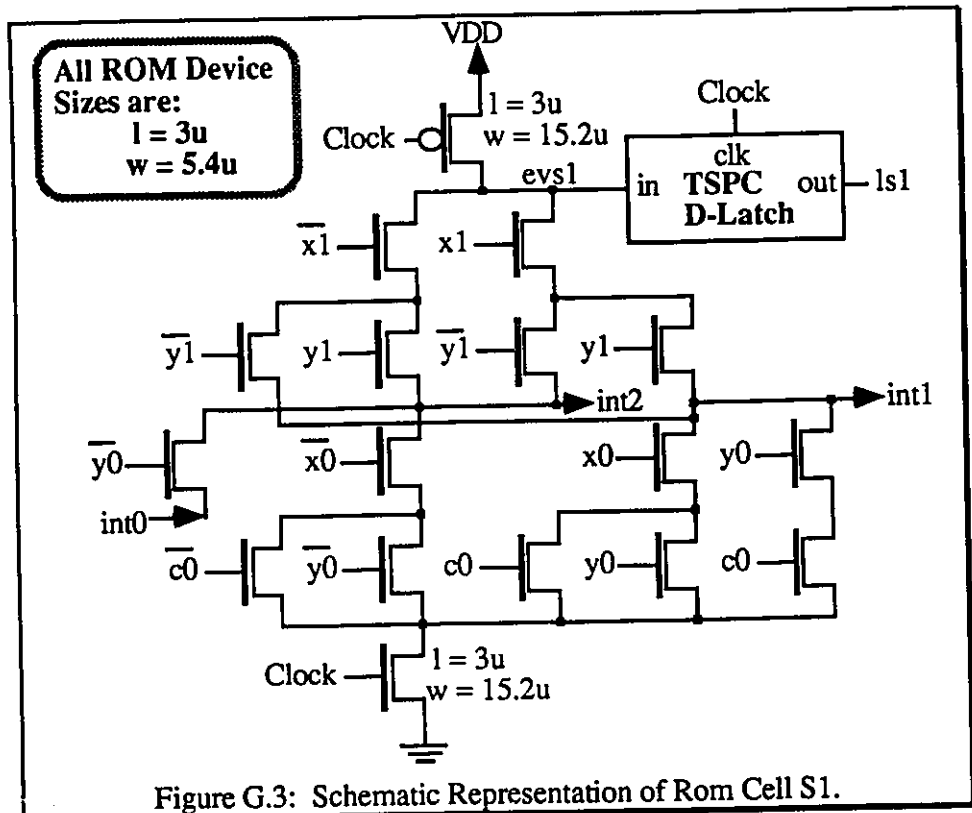
### SCHEMATIC DIAGRAM OF 3-BIT ADDER CELL

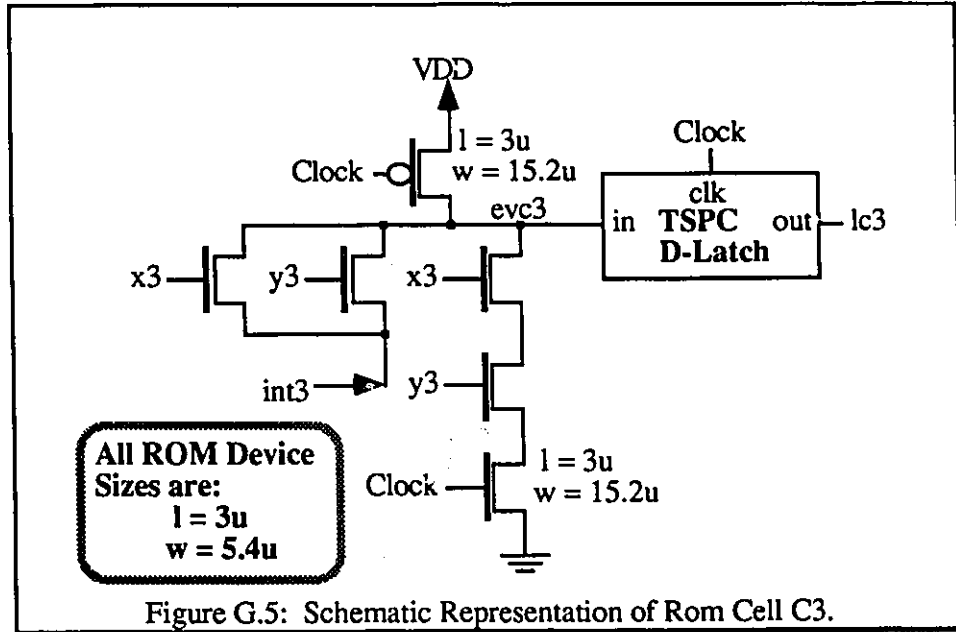
The complete schematic diagram of the 3-bit adder cell is illustrated in Figure G.1 through G.8 inclusive. The clock buffer circuit for the 3-bit adder cell is illustrated in Figure G.1. The



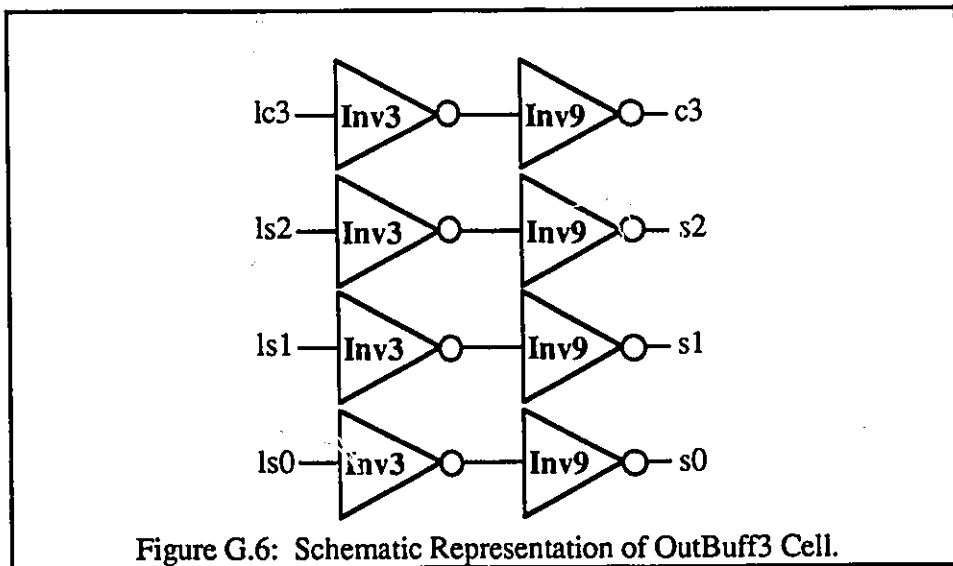
inverters used, Inv3, are designed with devices which are three times minimum size. Figures G.2 through G.5 represent the schematic diagrams of the four ROM cells required for the 3-bit adder cell. Signals of the form 'intx' represent internal connections between the ROM Cells.



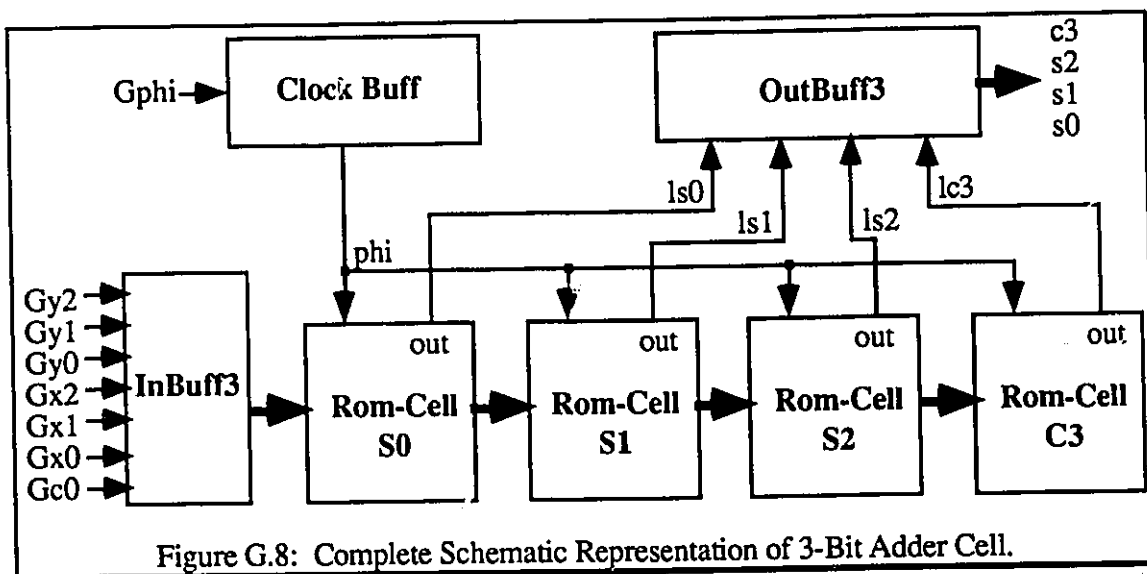
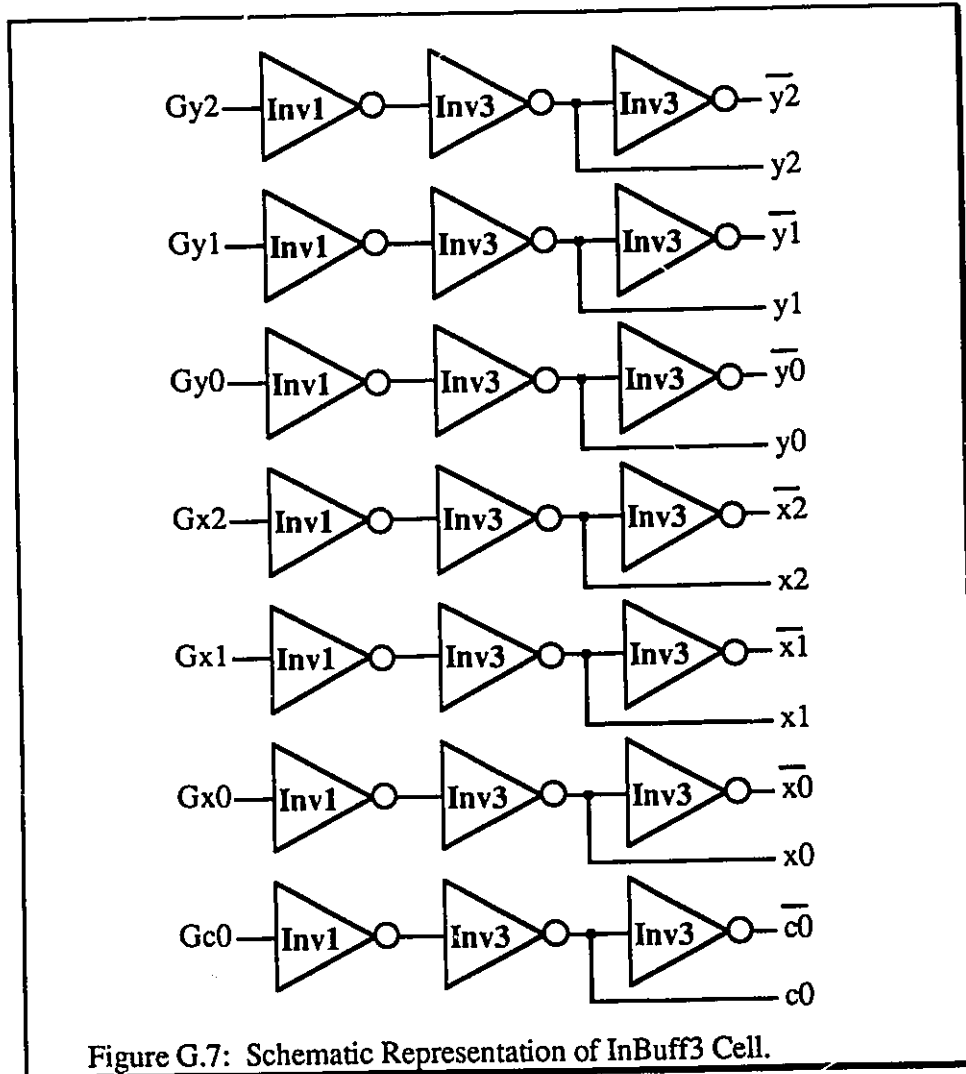




The output buffer of the 3-bit adder cell is illustrated in Figure G.6. The output signals from the ROM cells are labelled 'ls0' through 'ls2' and 'lc3'. The corresponding output signals for the 3-bit adder cell are labelled 's0' through 's2' and 'c3'. The input buffer circuit for the adder is illustrated in Figure G.7. The global input signals to the cell begin with the letter 'G'. The complete block diagram of the 3-bit adder cell is given in Figure G.8.







## APPENDIX H

### SPICE-DECK FOR 3-BIT ADDER CELL

## APPENDIX H

### SPICE-DECK FOR 3-BIT ADDER CELL

Mask Extracted Spice Deck of 3-Bit Adder Cell

\*\*\*\*\*

```
.tran 0.23ns 100ns
.include cmos3d.lib
```

\*\*\*\*\*

```
* SUMMARY OF CELL SIGNALS: *
* *
* -The input signal to the cell is are: gy0,gy1,gy2,gx0,gx1,gx2 and gc0 *
* -The clock signal into the cell is, gphi. *
* -The supply source to the cell is, vdd. *
* -The output nodes of the adder are: nos0,nos1,nos2, and noc3. *
* -All local input signals are preceeded by the letter, 'c'. *
* -The evaluation nodes of the ROM cells are: nv0, nv1, nv2, anv nv3. *
* *
*****
```

```
* Input Signals *
*****
```

```
vdd vdd 0 dc 5.0
vphi gphi 0 pulse(0.05 4.95 3n 3n 3n 7n 20n)
vgc0 gc0 0 pulse(0.05 4.95 1n 3n 3n 17n 40n)
vgx0 gx0 0 pulse(0.05 4.95 21n 3n 3n 37n 80n)
vgx1 gx1 0 pulse(0.05 4.95 61n 3n 3n 17n 100n)
vgx2 gx2 0 pulse(0.05 4.95 41n 3n 3n 17n 40n)
vgy0 gy0 0 pulse(0.05 4.95 1n 3n 3n 37n 60n)
vgy1 gy1 0 pulse(0.05 4.95 21n 3n 3n 17n 60n)
vgy2 gy2 0 pulse(0.05 4.95 41n 3n 3n 37n 100n)
```

\*\*\*\*\*

```
* Output Loading Circuitry *
*****
```

```
cld0 nos0 0 150ff
cld1 nos1 0 150ff
cld2 nos2 0 150ff
cld3 noc3 0 150ff
```

\*\*\*\*\*

```
* Extracted Deck Listing *
*****
```

\*\*\*\*\*

```
* n-channel enhancementmosfets extracted from*
```

\*mask layout follow.\*

\*\*\*\*\*

mn1 4 cy0b 3 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.2592e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.15e-04  
+nrd= 0.88889 nrs= 0.13333e+01

mn2 7 cc0b 6 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.798e-10 as= 0.2592e-10 ps= 0.15e-04 pd= 0.28e-04  
+nrd= 0.1e+01 nrs= 0.88889

mn3 0 cphi 7 0 ntran w= 0.114e-04 l= 0.3e-05  
+ad= 0.7524e-10 as= 0.798e-10 ps= 0.28e-04 pd= 0.36e-04  
+nrd= 0.57895 nrs= 0.1e+01

mn4 7 cc0 4 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.798e-10 as= 0.2592e-10 ps= 0.15e-04 pd= 0.28e-04  
+nrd= 0.1e+01 nrs= 0.88889

mn5 6 cy0 3 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.2592e-10 as= 0.2754e-10 ps= 0.156e-04 pd= 0.15e-04  
+nrd= 0.88889 nrs= 0.94444

mn6 nv0 cx0b 3 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.4536e-10 as= 0.2754e-10 ps= 0.156e-04 pd= 0.276e-04  
+nrd= 0.15556e+01 nrs= 0.94444

mn7 14 cphi 13 0 ntran w= 0.6e-05 l= 0.3e-05  
+ad= 0.9e-11 as= 0.432e-10 ps= 0.264e-04 pd= 0.9e-05  
+nrd= 0.25 nrs= 0.12e+01

mn8 0 nv0 14 0 ntran w= 0.6e-05 l= 0.3e-05  
+ad= 0.396e-10 as= 0.9e-11 ps= 0.9e-05 pd= 0.252e-04  
+nrd= 0.11e+01 nrs= 0.25

mn9 0 gc0 16 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.5652e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.324e-04  
+nrd= 0.2e+01 nrs= 0.13333e+01

mn10 cc0 16 0 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.378e-10 as= 0.5544e-10 ps= 0.3e-04 pd= 0.174e-04  
+nrd= 0.53571 nrs= 0.78571

mn11 0 16 cc0 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.5544e-10 as= 0.378e-10 ps= 0.174e-04 pd= 0.3e-04  
+nrd= 0.78571 nrs= 0.53571

mn12 0 gy0 18 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.5652e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.324e-04  
+nrd= 0.2e+01 nrs= 0.13333e+01

mn13 0 18 cy0 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.5544e-10 as= 0.378e-10 ps= 0.174e-04 pd= 0.3e-04  
+nrd= 0.78571 nrs= 0.53571

mn14 cy0 18 0 0 ntran w=0.84e-05 l=0.3e-05  
 +ad= 0.378e-10 as= 0.5544e-10 ps= 0.3e-04 pd= 0.174e-04  
 +nrd= 0.53571 nrs= 0.78571

mn15 cphi nphi2 0 0 ntran w=0.84e-05 l=0.3e-05  
 +ad= 0.378e-10 as= 0.5544e-10 ps= 0.3e-04 pd= 0.174e-04  
 +nrd= 0.53571 nrs= 0.78571

mn16 0 nphi2 cphi 0 ntran w=0.84e-05 l=0.3e-05  
 +ad= 0.5544e-10 as= 0.378e-10 ps= 0.174e-04 pd= 0.3e-04  
 +nrd= 0.78571 nrs= 0.53571

mn17 cphi nphi2 0 0 ntran w=0.84e-05 l=0.3e-05  
 +ad= 0.6048e-10 as= 0.5544e-10 ps= 0.3e-04 pd= 0.312e-04  
 +nrd= 0.85714 nrs= 0.78571

mn18 6 cy0b 20 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.3888e-10 as= 0.2754e-10 ps= 0.156e-04 pd= 0.252e-04  
 +nrd= 0.13333e+01 nrs= 0.94444

mn19 4 cy0 20 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.3888e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.252e-04  
 +nrd= 0.13333e+01 nrs= 0.13333e+01

mn20 nv0 cx0 20 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.4536e-10 as= 0.2754e-10 ps= 0.156e-04 pd= 0.276e-04  
 +nrd= 0.15556e+01 nrs= 0.94444

mn21 0 cphi 22 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.3564e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.24e-04  
 +nrd= 0.12222e+01 nrs= 0.13333e+01

mn22 0 22 ns0 0 ntran w=0.6e-05 l=0.3e-05  
 +ad= 0.468e-10 as= 0.504e-10 ps= 0.288e-04 pd= 0.276e-04  
 +nrd= 0.13e+01 nrs= 0.14e+01

mn23 0 cc0 cc0b 0 ntran w=0.84e-05 l=0.3e-05  
 +ad= 0.5544e-10 as= 0.6048e-10 ps= 0.312e-04 pd= 0.3e-04  
 +nrd= 0.78571 nrs= 0.85714

mn24 0 cy0 cy0b 0 ntran w=0.84e-05 l=0.3e-05  
 +ad= 0.5544e-10 as= 0.6048e-10 ps= 0.312e-04 pd= 0.3e-04  
 +nrd= 0.78571 nrs= 0.85714

mn25 0 gx0 25 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.5652e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.324e-04  
 +nrd= 0.2e+01 nrs= 0.13333e+01

mn26 0 25 cx0 0 ntran w=0.84e-05 l=0.3e-05  
 +ad= 0.5544e-10 as= 0.378e-10 ps= 0.174e-04 pd= 0.3e-04  
 +nrd= 0.78571 nrs= 0.53571

mn27 nphi2 gphi 0 0 ntran w=0.84e-05 l=0.3e-05

+ad= 0.6048e-10 as= 0.6552e-10 ps= 0.324e-04 pd= 0.312e-04  
+nrd= 0.85714 nrs= 0.92857

mn28 0 gy1 28 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.5652e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.324e-04  
+nrd= 0.2e+01 nrs= 0.13333e+01

mn29 0 28 cy1 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.5544e-10 as= 0.378e-10 ps= 0.174e-04 pd= 0.3e-04  
+nrd= 0.78571 nrs= 0.53571

mn30 6 cy0b 30 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.13333e+01

mn31 32 cy0b 31 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.243e-10 ps= 0.144e-04 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.83333

mn32 32 cc0b 31 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.13333e+01

mn33 0 cphi 32 0 ntran w= 0.174e-04 l= 0.3e-05  
+ad= 0.11484e-09 as= 0.12528e-09 ps= 0.492e-04 pd= 0.48e-04  
+nrd= 0.37931 nrs= 0.41379

mn34 32 cc0 33 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.81e-11 ps= 0.84e-05 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.27778

mn35 33 cy0 34 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.81e-11 as= 0.3888e-10 ps= 0.252e-04 pd= 0.84e-05  
+nrd= 0.27778 nrs= 0.13333e+01

mn36 34 cy1b 36 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.13333e+01

mn37 30 cx0b 31 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.2754e-10 as= 0.243e-10 ps= 0.144e-04 pd= 0.156e-04  
+nrd= 0.94444 nrs= 0.83333

mn38 36 cx1b nv1 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.2754e-10 as= 0.4536e-10 ps= 0.276e-04 pd= 0.156e-04  
+nrd= 0.94444 nrs= 0.15556e+01

mn39 30 cy1 36 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.2754e-10 as= 0.2754e-10 ps= 0.156e-04 pd= 0.156e-04  
+nrd= 0.94444 nrs= 0.94444

mn40 0 cphi 39 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3564e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.24e-04  
+nrd= 0.12222e+01 nrs= 0.13333e+01

mn41 41 cphi 40 0 ntran w= 0.6e-05 l= 0.3e-05  
+ad= 0.9e-11 as= 0.432e-10 ps= 0.264e-04 pd= 0.9e-05  
+nrd= 0.25 nrs= 0.12e+01

mn42 0 nv1 41 0 ntran w= 0.6e-05 l= 0.3e-05  
+ad= 0.396e-10 as= 0.9e-11 ps= 0.9e-05 pd= 0.252e-04  
+nrd= 0.11e+01 nrs= 0.25

mn43 0 cx0 cx0b 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.5544e-10 as= 0.6048e-10 ps= 0.312e-04 pd= 0.3e-04  
+nrd= 0.78571 nrs= 0.85714

mn44 0 25 cx0 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.5544e-10 as= 0.378e-10 ps= 0.174e-04 pd= 0.3e-04  
+nrd= 0.78571 nrs= 0.53571

mn45 0 gy2 43 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.5652e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.324e-04  
+nrd= 0.2e+01 nrs= 0.13333e+01

mn46 0 gx1 45 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.5652e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.324e-04  
+nrd= 0.2e+01 nrs= 0.13333e+01

mn47 0 cyl cy1b 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.5544e-10 as= 0.6048e-10 ps= 0.312e-04 pd= 0.3e-04  
+nrd= 0.78571 nrs= 0.85714

mn48 0 28 cyl 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.5544e-10 as= 0.378e-10 ps= 0.174e-04 pd= 0.3e-04  
+nrd= 0.78571 nrs= 0.53571

mn49 46 cx0 34 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.243e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.144e-04  
+nrd= 0.83333 nrs= 0.13333e+01

mn50 32 cc0 46 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.13333e+01

mn51 32 cy0 46 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.243e-10 ps= 0.144e-04 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.83333

mn52 34 cyl 47 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.13333e+01

mn53 47 cx1 nv1 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.2754e-10 as= 0.4536e-10 ps= 0.276e-04 pd= 0.156e-04  
+nrd= 0.94444 nrs= 0.15556e+01

mn54 30 cy1b 47 0 ntran w= 0.54e-05 l= 0.3e-05

+ad= 0.3888e-10 as= 0.2754e-10 ps= 0.156e-04 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.94444

mn55 0 39 ns1 0 ntran w= 0.6e-05 l= 0.3e-05  
+ad= 0.468e-10 as= 0.504e-10 ps= 0.288e-04 pd= 0.276e-04  
+nrd= 0.13e+01 nrs= 0.14e+01

mn56 51 gx2 0 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.5652e-10 ps= 0.324e-04 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.2e+01

mn57 0 cy2 cy2b 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.5544e-10 as= 0.6048e-10 ps= 0.312e-04 pd= 0.3e-04  
+nrd= 0.78571 nrs= 0.85714

mn58 cy2 43 0 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.378e-10 as= 0.5544e-10 ps= 0.3e-04 pd= 0.174e-04  
+nrd= 0.53571 nrs= 0.78571

mn59 cy2 43 0 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.378e-10 as= 0.5544e-10 ps= 0.3e-04 pd= 0.174e-04  
+nrd= 0.53571 nrs= 0.78571

mn60 0 cx1 cx1b 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.5544e-10 as= 0.6048e-10 ps= 0.312e-04 pd= 0.3e-04  
+nrd= 0.78571 nrs= 0.85714

mn61 cx1 45 0 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.378e-10 as= 0.5544e-10 ps= 0.3e-04 pd= 0.174e-04  
+nrd= 0.53571 nrs= 0.78571

mn62 cx1 45 0 0 ntran w= 0.84e-05 l= 0.3e-05  
+ad= 0.378e-10 as= 0.5544e-10 ps= 0.3e-04 pd= 0.174e-04  
+nrd= 0.53571 nrs= 0.78571

mn63 55 cc0b 54 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.13333e+01

mn64 55 cc0b 56 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.81e-11 ps= 0.84e-05 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.27778

mn65 56 cy0b 57 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.81e-11 as= 0.3888e-10 ps= 0.252e-04 pd= 0.84e-05  
+nrd= 0.27778 nrs= 0.13333e+01

mn66 58 cy1 55 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.243e-10 ps= 0.144e-04 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.83333

mn67 59 cy1 34 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.81e-11 as= 0.4536e-10 ps= 0.276e-04 pd= 0.84e-05  
+nrd= 0.27778 nrs= 0.15556e+01



mn68 34 cy1b 58 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.3888e-10 as= 0.2754e-10 ps= 0.156e-04 pd= 0.252e-04  
 +nrd= 0.13333e+01 nrs= 0.94444

mn69 0 cphi 55 0 ntran w=0.174e-04 l=0.3e-05  
 +ad= 0.11484e-09 as= 0.12528e-09 ps= 0.492e-04 pd= 0.48e-04  
 +nrd= 0.37931 nrs= 0.41379

mn70 61 cy2 60 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.3888e-10 as= 0.2754e-10 ps= 0.156e-04 pd= 0.252e-04  
 +nrd= 0.13333e+01 nrs= 0.94444

mn71 62 cy1b 61 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.81e-11 as= 0.3888e-10 ps= 0.252e-04 pd= 0.84e-05  
 +nrd= 0.27778 nrs= 0.13333e+01

mn72 61 cy2b 63 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.3888e-10 as= 0.2754e-10 ps= 0.156e-04 pd= 0.252e-04  
 +nrd= 0.13333e+01 nrs= 0.94444

mn73 60 cx2b nv2 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.2754e-10 as= 0.4536e-10 ps= 0.276e-04 pd= 0.156e-04  
 +nrd= 0.94444 nrs= 0.15556e+01

mn74 62 cx1b 55 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.81e-11 as= 0.243e-10 ps= 0.144e-04 pd= 0.84e-05  
 +nrd= 0.27778 nrs= 0.83333

mn75 59 cx1b 66 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.81e-11 as= 0.4536e-10 ps= 0.276e-04 pd= 0.84e-05  
 +nrd= 0.27778 nrs= 0.15556e+01

mn76 66 cy2b 60 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.5022e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.24e-04  
 +nrd= 0.17222e+01 nrs= 0.13333e+01

mn77 66 cx1 58 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.5022e-10 as= 0.2754e-10 ps= 0.156e-04 pd= 0.24e-04  
 +nrd= 0.17222e+01 nrs= 0.94444

mn78 63 cx2 nv2 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.2754e-10 as= 0.4536e-10 ps= 0.276e-04 pd= 0.156e-04  
 +nrd= 0.94444 nrs= 0.15556e+01

mn79 0 cphi 68 0 ntran w=0.54e-05 l=0.3e-05  
 +ad= 0.3564e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.24e-04  
 +nrd= 0.12222e+01 nrs= 0.13333e+01

mn80 70 cphi 69 0 ntran w=0.6e-05 l=0.3e-05  
 +ad= 0.9e-11 as= 0.432e-10 ps= 0.264e-04 pd= 0.9e-05  
 +nrd= 0.25 nrs= 0.12e+01

mn81 0 nv2 70 0 ntran w=0.6e-05 l=0.3e-05

+ad= 0.396e-10 as= 0.9e-11 ps= 0.9e-05 pd= 0.252e-04  
 +nrd= 0.11e+01 nrs= 0.25

mn82 0 cx2 cx2b 0 ntran w= 0.84e-05 l= 0.3e-05  
 +ad= 0.5544e-10 as= 0.6048e-10 ps= 0.312e-04 pd= 0.3e-04  
 +nrd= 0.78571 nrs= 0.85714

mn83 0 51 cx2 0 ntran w= 0.84e-05 l= 0.3e-05  
 +ad= 0.5544e-10 as= 0.378e-10 ps= 0.174e-04 pd= 0.3e-04  
 +nrd= 0.78571 nrs= 0.53571

mn84 cx2 51 0 0 ntran w= 0.84e-05 l= 0.3e-05  
 +ad= 0.378e-10 as= 0.5544e-10 ps= 0.3e-04 pd= 0.174e-04  
 +nrd= 0.53571 nrs= 0.78571

mn85 71 cphi 0 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3888e-10 as= 0.3564e-10 ps= 0.24e-04 pd= 0.252e-04  
 +nrd= 0.13333e+01 nrs= 0.12222e+01

mn86 nc3 71 0 0 ntran w= 0.6e-05 l= 0.3e-05  
 +ad= 0.504e-10 as= 0.468e-10 ps= 0.276e-04 pd= 0.288e-04  
 +nrd= 0.14e+01 nrs= 0.13e+01

mn87 54 cx0b 57 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.243e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.144e-04  
 +nrd= 0.83333 nrs= 0.13333e+01

mn88 55 cy0b 54 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3888e-10 as= 0.243e-10 ps= 0.144e-04 pd= 0.252e-04  
 +nrd= 0.13333e+01 nrs= 0.83333

mn89 61 cx1b 57 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3888e-10 as= 0.243e-10 ps= 0.144e-04 pd= 0.252e-04  
 +nrd= 0.13333e+01 nrs= 0.83333

mn90 61 cy1b 57 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3888e-10 as= 0.243e-10 ps= 0.144e-04 pd= 0.252e-04  
 +nrd= 0.13333e+01 nrs= 0.83333

mn91 66 cy2 63 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3888e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.252e-04  
 +nrd= 0.13333e+01 nrs= 0.13333e+01

mn92 0 68 ns2 0 ntran w= 0.6e-05 l= 0.3e-05  
 +ad= 0.468e-10 as= 0.504e-10 ps= 0.288e-04 pd= 0.276e-04  
 +nrd= 0.13e+01 nrs= 0.14e+01

mn93 75 cphi 74 0 ntran w= 0.6e-05 l= 0.3e-05  
 +ad= 0.432e-10 as= 0.9e-11 ps= 0.9e-05 pd= 0.264e-04  
 +nrd= 0.12e+01 nrs= 0.25

mn94 74 nv3 0 0 ntran w= 0.6e-05 l= 0.3e-05  
 +ad= 0.9e-11 as= 0.396e-10 ps= 0.252e-04 pd= 0.9e-05  
 +nrd= 0.25 nrs= 0.11e+01

mn95 nos0 77 0 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.243e-10 as= 0.3564e-10 ps= 0.24e-04 pd= 0.144e-04  
 +nrd= 0.83333 nrs= 0.12222e+01

mn96 0 77 nos0 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3564e-10 as= 0.243e-10 ps= 0.144e-04 pd= 0.24e-04  
 +nrd= 0.12222e+01 nrs= 0.83333

mn97 77 ns0 0 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3888e-10 as= 0.3564e-10 ps= 0.24e-04 pd= 0.252e-04  
 +nrd= 0.13333e+01 nrs= 0.12222e+01

mn98 0 79 nos1 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3564e-10 as= 0.243e-10 ps= 0.144e-04 pd= 0.24e-04  
 +nrd= 0.12222e+01 nrs= 0.83333

mn99 nos1 79 0 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.243e-10 as= 0.3564e-10 ps= 0.24e-04 pd= 0.144e-04  
 +nrd= 0.83333 nrs= 0.12222e+01

mn100 0 ns1 79 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3564e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.24e-04  
 +nrd= 0.12222e+01 nrs= 0.13333e+01

mn101 0 ns2 81 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3564e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.24e-04  
 +nrd= 0.12222e+01 nrs= 0.13333e+01

mn102 0 82 noc3 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3564e-10 as= 0.243e-10 ps= 0.144e-04 pd= 0.24e-04  
 +nrd= 0.12222e+01 nrs= 0.83333

mn103 0 82 noc3 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3564e-10 as= 0.243e-10 ps= 0.144e-04 pd= 0.24e-04  
 +nrd= 0.12222e+01 nrs= 0.83333

mn104 nos2 81 0 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.243e-10 as= 0.3564e-10 ps= 0.24e-04 pd= 0.144e-04  
 +nrd= 0.83333 nrs= 0.12222e+01

mn105 0 81 nos2 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3564e-10 as= 0.243e-10 ps= 0.144e-04 pd= 0.24e-04  
 +nrd= 0.12222e+01 nrs= 0.83333

mn106 0 nc3 82 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.3564e-10 as= 0.3888e-10 ps= 0.252e-04 pd= 0.24e-04  
 +nrd= 0.12222e+01 nrs= 0.13333e+01

mn107 86 cy2 85 0 ntran w= 0.54e-05 l= 0.3e-05  
 +ad= 0.1296e-10 as= 0.1296e-10 ps= 0.102e-04 pd= 0.102e-04  
 +nrd= 0.44444 nrs= 0.44444

mn108 nv3 cx2 86 0 ntran w= 0.54e-05 l= 0.3e-05

+ad= 0.3888e-10 as= 0.1296e-10 ps= 0.102e-04 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.44444

mn109 85 cphi 0 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.1296e-10 as= 0.3564e-10 ps= 0.24e-04 pd= 0.102e-04  
+nrd= 0.44444 nrs= 0.12222e+01

mn110 nv3 cx2b 87 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.81e-11 ps= 0.84e-05 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.27778

mn111 nv3 cx2 88 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.3888e-10 as= 0.81e-11 ps= 0.84e-05 pd= 0.252e-04  
+nrd= 0.13333e+01 nrs= 0.27778

mn112 88 cy2b 66 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.81e-11 as= 0.3888e-10 ps= 0.252e-04 pd= 0.84e-05  
+nrd= 0.27778 nrs= 0.13333e+01

mn113 87 cy2 66 0 ntran w= 0.54e-05 l= 0.3e-05  
+ad= 0.81e-11 as= 0.3888e-10 ps= 0.252e-04 pd= 0.84e-05  
+nrd= 0.27778 nrs= 0.13333e+01

\*\*\*\*\*  
\* p-channel enhancement mosfets extracted from \*  
\* mask layout follow. \*  
\*\*\*\*\*

mp114 vdd cphi nv0 vdd ptran w= 0.174e-04 l= 0.3e-05  
+ad= 0.11484e-09 as= 0.12528e-09 ps= 0.492e-04 pd= 0.48e-04  
+nrd= 0.37931 nrs= 0.41379

mp115 13 nv0 vdd vdd ptran w= 0.6e-05 l= 0.3e-05  
+ad= 0.432e-10 as= 0.396e-10 ps= 0.252e-04 pd= 0.264e-04  
+nrd= 0.12e+01 nrs= 0.11e+01

mp116 16 gc0 vdd vdd ptran w= 0.15e-04 l= 0.3e-05  
+ad= 0.108e-09 as= 0.16308e-09 ps= 0.648e-04 pd= 0.444e-04  
+nrd= 0.48 nrs= 0.116e+01

mp117 vdd 16 cc0 vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.16236e-09 as= 0.1107e-09 ps= 0.336e-04 pd= 0.624e-04  
+nrd= 0.26829 nrs= 0.18293

mp118 cc0 16 vdd vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.1107e-09 as= 0.16236e-09 ps= 0.624e-04 pd= 0.336e-04  
+nrd= 0.18293 nrs= 0.26829

mp119 cphi nphi2 vdd vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.17712e-09 as= 0.16236e-09 ps= 0.624e-04 pd= 0.636e-04  
+nrd= 0.29268 nrs= 0.26829

mp120 vdd nphi2 cphi vdd ptran w= 0.246e-04 l= 0.3e-05

+ad= 0.16236e-09 as= 0.1107e-09 ps= 0.336e-04 pd= 0.624e-04  
+nrd= 0.26829 nrs= 0.18293

mp121 cphi nphi2 vdd vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.1107e-09 as= 0.16236e-09 ps= 0.624e-04 pd= 0.336e-04  
+nrd= 0.18293 nrs= 0.26829

mp122 vdd gy0 18 vdd ptran w= 0.15e-04 l= 0.3e-05  
+ad= 0.16308e-09 as= 0.108e-09 ps= 0.444e-04 pd= 0.648e-04  
+nrd= 0.116e+01 nrs= 0.48

mp123 cy0 18 vdd vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.1107e-09 as= 0.16236e-09 ps= 0.624e-04 pd= 0.336e-04  
+nrd= 0.18293 nrs= 0.26829

mp124 vdd 18 cy0 vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.16236e-09 as= 0.1107e-09 ps= 0.336e-04 pd= 0.624e-04  
+nrd= 0.26829 nrs= 0.18293

mp125 22 13 90 vdd ptran w= 0.126e-04 l= 0.3e-05  
+ad= 0.9072e-10 as= 0.189e-10 ps= 0.156e-04 pd= 0.396e-04  
+nrd= 0.57143 nrs= 0.11905

mp126 91 22 vdd vdd ptran w= 0.216e-04 l= 0.3e-05  
+ad= 0.324e-10 as= 0.14256e-09 ps= 0.564e-04 pd= 0.246e-04  
+nrd= 0.69444e-01 nrs= 0.30556

mp127 ns0 cphi 91 vdd ptran w= 0.216e-04 l= 0.3e-05  
+ad= 0.15552e-09 as= 0.324e-10 ps= 0.246e-04 pd= 0.576e-04  
+nrd= 0.33333 nrs= 0.69444e-01

mp128 90 cphi vdd vdd ptran w= 0.126e-04 l= 0.3e-05  
+ad= 0.189e-10 as= 0.8316e-10 ps= 0.384e-04 pd= 0.156e-04  
+nrd= 0.11905 nrs= 0.52381

mp129 vdd cc0 cc0b vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.16236e-09 as= 0.17712e-09 ps= 0.636e-04 pd= 0.624e-04  
+nrd= 0.26829 nrs= 0.29268

mp130 nphi2 gphi vdd vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.17712e-09 as= 0.19188e-09 ps= 0.648e-04 pd= 0.636e-04  
+nrd= 0.29268 nrs= 0.31707

mp131 vdd cy0 cy0b vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.16236e-09 as= 0.17712e-09 ps= 0.636e-04 pd= 0.624e-04  
+nrd= 0.26829 nrs= 0.29268

mp132 vdd gx0 25 vdd ptran w= 0.15e-04 l= 0.3e-05  
+ad= 0.16308e-09 as= 0.108e-09 ps= 0.444e-04 pd= 0.648e-04  
+nrd= 0.116e+01 nrs= 0.48

mp133 vdd 25 cx0 vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.16236e-09 as= 0.1107e-09 ps= 0.336e-04 pd= 0.624e-04  
+nrd= 0.26829 nrs= 0.18293

mp134 vdd 28 cyl vdd ptran w=0.246e-04 l=0.3e-05  
+ad=0.16236e-09 as=0.1107e-09 ps=0.336e-04 pd=0.624e-04  
+nrd=0.26829 nrs=0.18293

mp135 28 gy1 vdd vdd ptran w=0.15e-04 l=0.3e-05  
+ad=0.108e-09 as=0.16308e-09 ps=0.648e-04 pd=0.444e-04  
+nrd=0.48 nrs=0.116e+01

mp136 vdd cphi nv1 vdd ptran w=0.162e-04 l=0.3e-05  
+ad=0.10692e-09 as=0.11664e-09 ps=0.468e-04 pd=0.456e-04  
+nrd=0.40741 nrs=0.44444

mp137 40 nv1 vdd vdd ptran w=0.6e-05 l=0.3e-05  
+ad=0.432e-10 as=0.396e-10 ps=0.252e-04 pd=0.264e-04  
+nrd=0.12e+01 nrs=0.11e+01

mp138 39 40 92 vdd ptran w=0.126e-04 l=0.3e-05  
+ad=0.9072e-10 as=0.189e-10 ps=0.156e-04 pd=0.396e-04  
+nrd=0.57143 nrs=0.11905

mp139 92 cphi vdd vdd ptran w=0.126e-04 l=0.3e-05  
+ad=0.189e-10 as=0.8316e-10 ps=0.384e-04 pd=0.156e-04  
+nrd=0.11905 nrs=0.52381

mp140 vdd cx0 cx0b vdd ptran w=0.246e-04 l=0.3e-05  
+ad=0.16236e-09 as=0.17712e-09 ps=0.636e-04 pd=0.624e-04  
+nrd=0.26829 nrs=0.29268

mp141 vdd 25 cx0 vdd ptran w=0.246e-04 l=0.3e-05  
+ad=0.16236e-09 as=0.1107e-09 ps=0.336e-04 pd=0.624e-04  
+nrd=0.26829 nrs=0.18293

mp142 vdd gy2 43 vdd ptran w=0.15e-04 l=0.3e-05  
+ad=0.16308e-09 as=0.108e-09 ps=0.444e-04 pd=0.648e-04  
+nrd=0.116e+01 nrs=0.48

mp143 vdd cyl cy1b vdd ptran w=0.246e-04 l=0.3e-05  
+ad=0.16236e-09 as=0.17712e-09 ps=0.636e-04 pd=0.624e-04  
+nrd=0.26829 nrs=0.29268

mp144 vdd 28 cyl vdd ptran w=0.246e-04 l=0.3e-05  
+ad=0.16236e-09 as=0.1107e-09 ps=0.336e-04 pd=0.624e-04  
+nrd=0.26829 nrs=0.18293

mp145 45 gx1 vdd vdd ptran w=0.15e-04 l=0.3e-05  
+ad=0.108e-09 as=0.16308e-09 ps=0.648e-04 pd=0.444e-04  
+nrd=0.48 nrs=0.116e+01

mp146 nv2 cphi vdd vdd ptran w=0.174e-04 l=0.3e-05  
+ad=0.12528e-09 as=0.11484e-09 ps=0.48e-04 pd=0.492e-04  
+nrd=0.41379 nrs=0.37931

mp147 ns1 cphi 93 vdd ptran w=0.216e-04 l=0.3e-05

+ad= 0.15552e-09 as= 0.324e-10 ps= 0.246e-04 pd= 0.576e-04  
+nrd= 0.33333 nrs= 0.69444e-01

mp148 93 39 vdd vdd ptran w= 0.216e-04 l= 0.3e-05  
+ad= 0.324e-10 as= 0.14256e-09 ps= 0.564e-04 pd= 0.246e-04  
+nrd= 0.69444e-01 nrs= 0.30556

mp149 vdd cx1 cx1b vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.16236e-09 as= 0.17712e-09 ps= 0.636e-04 pd= 0.624e-04  
+nrd= 0.26829 nrs= 0.29268

mp150 cx1 45 vdd vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.1107e-09 as= 0.16236e-09 ps= 0.624e-04 pd= 0.336e-04  
+nrd= 0.18293 nrs= 0.26829

mp151 cx1 45 vdd vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.1107e-09 as= 0.16236e-09 ps= 0.624e-04 pd= 0.336e-04  
+nrd= 0.18293 nrs= 0.26829

mp152 vdd gx2 51 vdd ptran w= 0.15e-04 l= 0.3e-05  
+ad= 0.16308e-09 as= 0.108e-09 ps= 0.444e-04 pd= 0.648e-04  
+nrd= 0.116e+01 nrs= 0.48

mp153 vdd cy2 cy2b vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.16236e-09 as= 0.17712e-09 ps= 0.636e-04 pd= 0.624e-04  
+nrd= 0.26829 nrs= 0.29268

mp154 cy2 43 vdd vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.1107e-09 as= 0.16236e-09 ps= 0.624e-04 pd= 0.336e-04  
+nrd= 0.18293 nrs= 0.26829

mp155 cy2 43 vdd vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.1107e-09 as= 0.16236e-09 ps= 0.624e-04 pd= 0.336e-04  
+nrd= 0.18293 nrs= 0.26829

mp156 vdd nv2 69 vdd ptran w= 0.6e-05 l= 0.3e-05  
+ad= 0.396e-10 as= 0.432e-10 ps= 0.264e-04 pd= 0.252e-04  
+nrd= 0.11e+01 nrs= 0.12e+01

mp157 68 69 94 vdd ptran w= 0.126e-04 l= 0.3e-05  
+ad= 0.9072e-10 as= 0.189e-10 ps= 0.156e-04 pd= 0.396e-04  
+nrd= 0.57143 nrs= 0.11905

mp158 vdd cphi 94 vdd ptran w= 0.126e-04 l= 0.3e-05  
+ad= 0.8316e-10 as= 0.189e-10 ps= 0.156e-04 pd= 0.384e-04  
+nrd= 0.52381 nrs= 0.11905

mp159 vdd cx2 cx2b vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.16236e-09 as= 0.17712e-09 ps= 0.636e-04 pd= 0.624e-04  
+nrd= 0.26829 nrs= 0.29268

mp160 cx2 51 vdd vdd ptran w= 0.246e-04 l= 0.3e-05  
+ad= 0.1107e-09 as= 0.16236e-09 ps= 0.624e-04 pd= 0.336e-04  
+nrd= 0.18293 nrs= 0.26829

mp161 vdd 51 cx2 vdd ptran w=0.246e-04 l=0.3e-05  
+ad= 0.16236e-09 as= 0.1107e-09 ps= 0.336e-04 pd= 0.624e-04  
+nrd= 0.26829 nrs= 0.18293

mp162 95 75 71 vdd ptran w=0.126e-04 l=0.3e-05  
+ad= 0.189e-10 as= 0.9072e-10 ps= 0.396e-04 pd= 0.156e-04  
+nrd= 0.11905 nrs= 0.57143

mp163 95 cphi vdd vdd ptran w=0.126e-04 l=0.3e-05  
+ad= 0.189e-10 as= 0.8316e-10 ps= 0.384e-04 pd= 0.156e-04  
+nrd= 0.11905 nrs= 0.52381

mp164 96 cphi nc3 vdd ptran w=0.216e-04 l=0.3e-05  
+ad= 0.324e-10 as= 0.15552e-09 ps= 0.576e-04 pd= 0.246e-04  
+nrd= 0.69444e-01 nrs= 0.33333

mp165 96 71 vdd vdd ptran w=0.216e-04 l=0.3e-05  
+ad= 0.324e-10 as= 0.14256e-09 ps= 0.564e-04 pd= 0.246e-04  
+nrd= 0.69444e-01 nrs= 0.30556

mp166 ns2 cphi 97 vdd ptran w=0.216e-04 l=0.3e-05  
+ad= 0.15552e-09 as= 0.324e-10 ps= 0.246e-04 pd= 0.576e-04  
+nrd= 0.33333 nrs= 0.69444e-01

mp167 vdd 68 97 vdd ptran w=0.216e-04 l=0.3e-05  
+ad= 0.14256e-09 as= 0.324e-10 ps= 0.246e-04 pd= 0.564e-04  
+nrd= 0.30556 nrs= 0.69444e-01

mp168 75 nv3 vdd vdd ptran w=0.6e-05 l=0.3e-05  
+ad= 0.432e-10 as= 0.396e-10 ps= 0.252e-04 pd= 0.264e-04  
+nrd= 0.12e+01 nrs= 0.11e+01

mp169 vdd cphi nv3 vdd ptran w=0.114e-04 l=0.3e-05  
+ad= 0.7524e-10 as= 0.8208e-10 ps= 0.372e-04 pd= 0.36e-04  
+nrd= 0.57895 nrs= 0.63158

mp170 vdd 77 nos0 vdd ptran w=0.162e-04 l=0.3e-05  
+ad= 0.10692e-09 as= 0.729e-10 ps= 0.252e-04 pd= 0.456e-04  
+nrd= 0.40741 nrs= 0.27778

mp171 nos0 77 vdd vdd ptran w=0.162e-04 l=0.3e-05  
+ad= 0.729e-10 as= 0.10692e-09 ps= 0.456e-04 pd= 0.252e-04  
+nrd= 0.27778 nrs= 0.40741

mp172 77 ns0 vdd vdd ptran w=0.162e-04 l=0.3e-05  
+ad= 0.11664e-09 as= 0.10692e-09 ps= 0.456e-04 pd= 0.468e-04  
+nrd= 0.44444 nrs= 0.40741

mp173 vdd 79 nos1 vdd ptran w=0.162e-04 l=0.3e-05  
+ad= 0.10692e-09 as= 0.729e-10 ps= 0.252e-04 pd= 0.456e-04  
+nrd= 0.40741 nrs= 0.27778

mp174 nos1 79 vdd vdd ptran w=0.162e-04 l=0.3e-05



+ad= 0.729e-10 as= 0.10692e-09 ps= 0.456e-04 pd= 0.252e-04  
+nrd= 0.27778 nrs= 0.40741

mp175 vdd ns1 79 vdd ptran w= 0.162e-04 l= 0.3e-05  
+ad= 0.10692e-09 as= 0.11664e-09 ps= 0.468e-04 pd= 0.456e-04  
+nrd= 0.40741 nrs= 0.44444

mp176 vdd ns2 81 vdd ptran w= 0.162e-04 l= 0.3e-05  
+ad= 0.10692e-09 as= 0.11664e-09 ps= 0.468e-04 pd= 0.456e-04  
+nrd= 0.40741 nrs= 0.44444

mp177 vdd 82 noc3 vdd ptran w= 0.162e-04 l= 0.3e-05  
+ad= 0.10692e-09 as= 0.729e-10 ps= 0.252e-04 pd= 0.456e-04  
+nrd= 0.40741 nrs= 0.27778

mp178 vdd 82 noc3 vdd ptran w= 0.162e-04 l= 0.3e-05  
+ad= 0.10692e-09 as= 0.729e-10 ps= 0.252e-04 pd= 0.456e-04  
+nrd= 0.40741 nrs= 0.27778

mp179 vdd 81 nos2 vdd ptran w= 0.162e-04 l= 0.3e-05  
+ad= 0.10692e-09 as= 0.729e-10 ps= 0.252e-04 pd= 0.456e-04  
+nrd= 0.40741 nrs= 0.27778

mp180 nos2 81 vdd vdd ptran w= 0.162e-04 l= 0.3e-05  
+ad= 0.729e-10 as= 0.10692e-09 ps= 0.456e-04 pd= 0.252e-04  
+nrd= 0.27778 nrs= 0.40741

mp181 vdd nc3 82 vdd ptran w= 0.162e-04 l= 0.3e-05  
+ad= 0.10692e-09 as= 0.11664e-09 ps= 0.468e-04 pd= 0.456e-04  
+nrd= 0.40741 nrs= 0.44444

\*\*\*\*\*  
\* non-parasitic capacitors extracted from \*  
\* mask layout follow. \*  
\*\*\*\*\*

\*\*\*\*\*  
\* parasitic node capacitors follow \*  
\*\*\*\*\*

cn1 cy0b vdd 0.12301e-13  
cn2 0 vdd 0.99507e-12  
cn3 cc0b vdd 0.74172e-14  
cn5 cphi vdd 0.26736e-12  
cn6 cc0 vdd 0.14798e-13  
cn7 cy0 vdd 0.21979e-13  
cn8 cx0b vdd 0.12813e-13  
cn9 nv0 vdd 0.27412e-13  
cn10 13 vdd 0.98208e-14  
cn11 gc0 vdd 0.81316e-14  
cn12 16 vdd 0.15626e-13  
cn13 gy0 vdd 0.81316e-14

cn14 18 vdd 0.15626e-13  
cn15 nphi2 vdd 0.22371e-13  
cn16 cx0 vdd 0.13292e-13  
cn17 22 vdd 0.15803e-13  
cn18 ns0 vdd 0.66668e-13  
cn19 gx0 vdd 0.8782e-14  
cn20 25 vdd 0.15626e-13  
cn21 gphi vdd 0.92521e-14  
cn22 gy1 vdd 0.83776e-14  
cn23 28 vdd 0.15626e-13  
cn24 cy1 vdd 0.21642e-13  
cn26 cy1b vdd 0.10591e-13  
cn27 cx1b vdd 0.94457e-14  
cn28 nv1 vdd 0.27441e-13  
cn29 39 vdd 0.15803e-13  
cn30 40 vdd 0.98208e-14  
cn31 gy2 vdd 0.11416e-13  
cn32 43 vdd 0.15626e-13  
cn33 gx1 vdd 0.83776e-14  
cn34 45 vdd 0.15626e-13  
cn35 cx1 vdd 0.21985e-13  
cn36 ns1 vdd 0.37254e-13  
cn37 gx2 vdd 0.13885e-13  
cn38 51 vdd 0.15626e-13  
cn39 cy2 vdd 0.25859e-13  
cn40 cy2b vdd 0.9791e-14  
cn41 cx2b vdd 0.13628e-13  
cn42 nv2 vdd 0.2775e-13  
cn43 66 vdd 0.41328e-14  
cn44 cx2 vdd 0.17585e-13  
cn45 68 vdd 0.15803e-13  
cn46 69 vdd 0.98208e-14  
cn47 71 vdd 0.15358e-13  
cn48 nc3 vdd 0.10975e-13  
cn49 ns2 vdd 0.13804e-13  
cn50 75 vdd 0.99202e-14  
cn51 nv3 vdd 0.18913e-13  
cn52 77 vdd 0.15337e-13  
cn53 nos0 vdd 0.32354e-14  
cn54 79 vdd 0.15337e-13  
cn55 nos1 vdd 0.32354e-14  
cn56 81 vdd 0.15337e-13  
cn57 82 vdd 0.15337e-13  
cn58 noc3 vdd 0.32354e-14  
cn59 nos2 vdd 0.32354e-14

cp1 cy0b 0 0.57804e-13  
cp2 0 3 0.17388e-14  
cp3 0 4 0.75562e-14  
cp4 0 cc0b 0.36038e-13  
cp5 0 6 0.16704e-13  
cp6 0 cphi 0.65301e-13  
cp7 0 cc0 0.2754e-13  
cp8 0 cy0 0.51953e-13

cp9 0 cx0b 0.43014e-13  
cp10 0 nv0 0.10758e-13  
cp12 0 gc0 0.1644e-14  
cp13 0 16 0.58042e-14  
cp14 0 gy0 0.1644e-14  
cp15 0 18 0.58042e-14  
cp16 0 nphi2 0.69095e-14  
cp17 0 20 0.1449e-14  
cp18 0 cx0 0.32249e-13  
cp19 0 22 0.79027e-14  
cp20 0 ns0 0.58164e-14  
cp21 0 gx0 0.1644e-14  
cp22 0 25 0.58042e-14  
cp23 0 gphi 0.51254e-14  
cp24 0 gy1 0.1644e-14  
cp25 0 28 0.58042e-14  
cp26 0 cy1 0.488e-13  
cp27 0 30 0.48066e-14  
cp28 0 31 0.20812e-14  
cp29 0 32 0.15218e-13  
cp30 0 34 0.2166e-13  
cp31 0 cy1b 0.4145e-13  
cp32 0 36 0.17388e-14  
cp33 0 cx1b 0.36747e-13  
cp34 0 nv1 0.10758e-13  
cp35 0 39 0.79027e-14  
cp37 0 gy2 0.1644e-14  
cp38 0 43 0.58042e-14  
cp39 0 gx1 0.1644e-14  
cp40 0 45 0.58042e-14  
cp41 0 46 0.1449e-14  
cp42 0 47 0.1449e-14  
cp43 0 cx1 0.2803e-13  
cp44 0 ns1 0.95803e-14  
cp45 0 gx2 0.189e-14  
cp46 0 51 0.58042e-14  
cp47 0 cy2 0.46133e-13  
cp48 0 cy2b 0.41487e-13  
cp49 0 54 0.1449e-14  
cp50 0 55 0.16305e-13  
cp51 0 57 0.83316e-14  
cp52 0 58 0.10632e-14  
cp53 0 60 0.17388e-14  
cp54 0 61 0.10185e-13  
cp55 0 63 0.1449e-14  
cp56 0 cx2b 0.28076e-13  
cp57 0 nv2 0.11387e-13  
cp58 0 66 0.17048e-13  
cp59 0 cx2 0.28288e-13  
cp60 0 68 0.79027e-14  
cp62 0 71 0.67709e-14  
cp63 0 nc3 0.36605e-14  
cp64 0 ns2 0.39115e-14  
cp65 0 75 0.10015e-14

```

cp66 0 nv3 0.88268e-14
cp67 0 77 0.49852e-14
cp69 0 79 0.49852e-14
cp71 0 81 0.49852e-14
cp72 0 82 0.49852e-14

```

```

*****
* parasiti      ccouplingcapacitors follow*
*****

```

```

cc1  cy0b  0  0.40914e-14
cc4  cy0b  cc0b 0.19854e-14
cc7  cy0b  cc0  0.10746e-14
cc8  cy0b  cy0  0.43668e-14
cc9  cy0b  cx0b 0.41283e-14
cc10 cy0b  nv0  0.1305e-14
cc16 cy0b  55  0.17514e-14
cc17 cy0b  vdd 0.45792e-14
cc18 0 cc0b 0.5985e-14
cc19 0 6 0.15714e-14
cc20 0 cphi 0.3492e-14
cc21 0 cc0 0.8019e-14
cc22 0 cy0 0.91685e-14
cc23 0 cx0b 0.38394e-14
cc25 0 cx0 0.4554e-14
cc26 0 ns0 0.23715e-13
cc27 0 cy1 0.88952e-14
cc29 0 cy1b 0.51797e-14
cc30 0 cx1b 0.594e-14
cc32 0 cx1 0.17459e-13
cc33 0 ns1 0.21735e-13
cc34 0 gx2 0.10845e-13
cc35 0 cy2 0.97679e-14
cc36 0 cy2b 0.15494e-13
cc37 0 cx2b 0.9225e-14
cc38 0 nv2 0.1071e-14
cc39 0 66 0.14085e-13
cc40 0 cx2 0.5697e-14
cc41 0 nc3 0.17559e-13
cc42 0 ns2 0.23423e-13
cc43 0 nv3 0.1548e-14
cc44 3 cx0b 0.1395e-14
cc45 4 6 0.10328e-14
cc48 cc0b cphi 0.33966e-14
cc49 cc0b cc0 0.18576e-14
cc50 cc0b cy0 0.14472e-14
cc54 cc0b 32 0.26446e-14
cc56 cc0b vdd 0.54792e-14
cc59 6 32 0.12924e-14
cc60 7 cc0 0.1314e-14
cc64 cphi 13 0.162e-14
cc65 cphi nphi2 0.4446e-14
cc70 cphi cy1b 0.14778e-14
cc72 cphi 40 0.162e-14

```

cc78 cphi cy2b 0.4203e-14  
cc79 cphi cx2b 0.1125e-14  
cc81 cphi 69 0.162e-14  
cc83 cphi 75 0.162e-14  
cc84 cphi vdd 0.27359e-13  
cc85 cphi 90 0.189e-14  
cc86 cphi 92 0.189e-14  
cc87 cphi 94 0.189e-14  
cc88 cphi 95 0.189e-14  
cc89 cc0 cy0 C 1188e-14  
cc91 cc0 gc0 0.1467e-14  
cc92 cc0 16 0.3726e-14  
cc94 cc0 vdd 0.53978e-14  
cc95 cy0 cx0b 0.23155e-14  
cc96 cy0 nv0 0.1611e-14  
cc98 cy0 18 0.3366e-14  
cc100 cy0 cx0 0.4892e-14  
cc101 cy0 gx0 0.11538e-14  
cc102 cy0 25 0.17064e-14  
cc103 cy0 32 0.153e-14  
cc104 cy0 vdd 0.10109e-13  
cc105 cx0b nv0 0.1395e-14  
cc108 cx0b cx0 0.17154e-14  
cc110 cx0b 25 0.20938e-14  
cc116 cx0b 55 0.11304e-14  
cc118 cx0b vdd 0.58982e-14  
cc119 nv0 cx0 0.1791e-14  
cc120 13 cx0 0.14987e-14  
cc126 nphi2 gphi 0.297e-14  
cc127 nphi2 gy2 0.10314e-14  
cc128 nphi2 gx2 0.11754e-14  
cc129 cx0 25 0.4626e-14  
cc135 cx0 vdd 0.70988e-14  
cc147 ns0 77 0.35798e-14  
cc149 ns0 79 0.2376e-14  
cc151 ns0 81 0.2376e-14  
cc152 ns0 82 0.2376e-14  
cc155 ns0 vdd 0.17266e-13  
cc157 gx0 vdd 0.28069e-14  
cc163 gy1 vdd 0.603e-14  
cc164 28 cy1 0.3366e-14  
cc166 28 cx1 0.126e-14  
cc169 cy1 cy1b 0.2817e-14  
cc170 cy1 cx1b 0.21348e-14  
cc171 cy1 nv1 0.162e-14  
cc176 cy1 cx1 0.58806e-14  
cc177 cy1 gx2 0.25164e-14  
cc178 cy1 58 0.3609e-14  
cc179 cy1 vdd 0.73771e-14  
cc180 30 34 0.1152e-14  
cc182 cy1b 36 0.10757e-14  
cc183 cy1b cx1b 0.79614e-14  
cc187 cy1b 45 0.1125e-14  
cc189 cy1b cx1 0.26341e-14

cc192 cy1b vdd 0.585e-14  
cc193 cx1b nv1 0.2349e-14  
cc197 cx1b cx1 0.24732e-14  
cc198 cx1b 57 0.11329e-14  
cc199 cx1b 61 0.16682e-14  
cc201 cx1b vdd 0.85842e-14  
cc202 nv1 cx1 0.1836e-14  
cc207 gy2 vdd 0.56214e-14  
cc208 43 cx1 0.15048e-14  
cc209 43 cy2 0.3366e-14  
cc212 gx1 vdd 0.12285e-13  
cc213 45 cx1 0.47545e-14  
cc215 cx1 vdd 0.4837e-14  
cc221 ns1 71 0.2025e-14  
cc224 ns1 79 0.117e-14  
cc229 ns1 vdd 0.17065e-13  
cc233 gx2 vdd 0.51786e-14  
cc234 51 cy2 0.21564e-14  
cc235 51 cy2b 0.108e-14  
cc237 51 cx2 0.4131e-14  
cc238 cy2 cy2b 0.49662e-14  
cc240 cy2 cx2b 0.25272e-14  
cc241 cy2 nv2 0.14468e-14  
cc242 cy2 cx2 0.51048e-14  
cc247 cy2 vdd 0.12485e-13  
cc250 cy2b cx2b 0.47556e-14  
cc256 cy2b vdd 0.5355e-14  
cc258 60 cx2b 0.13478e-14  
cc259 61 66 0.1764e-14  
cc261 cx2b nv2 0.1305e-14  
cc262 cx2b cx2 0.12978e-14  
cc263 cx2b nv3 0.19786e-14  
cc264 cx2b vdd 0.10024e-13  
cc265 nv2 cx2 0.3159e-14  
cc266 66 vdd 0.3465e-14  
cc267 cx2 vdd 0.1017e-13  
cc273 nc3 vdd 0.5985e-14  
cc274 ns2 81 0.12564e-14  
cc278 ns2 vdd 0.21548e-13  
cc281 77 nos0 0.1782e-14  
cc282 79 nos1 0.1782e-14  
cc283 81 nos2 0.1782e-14  
cc284 82 noc3 0.1782e-14

\*\*\*\*\*  
.end

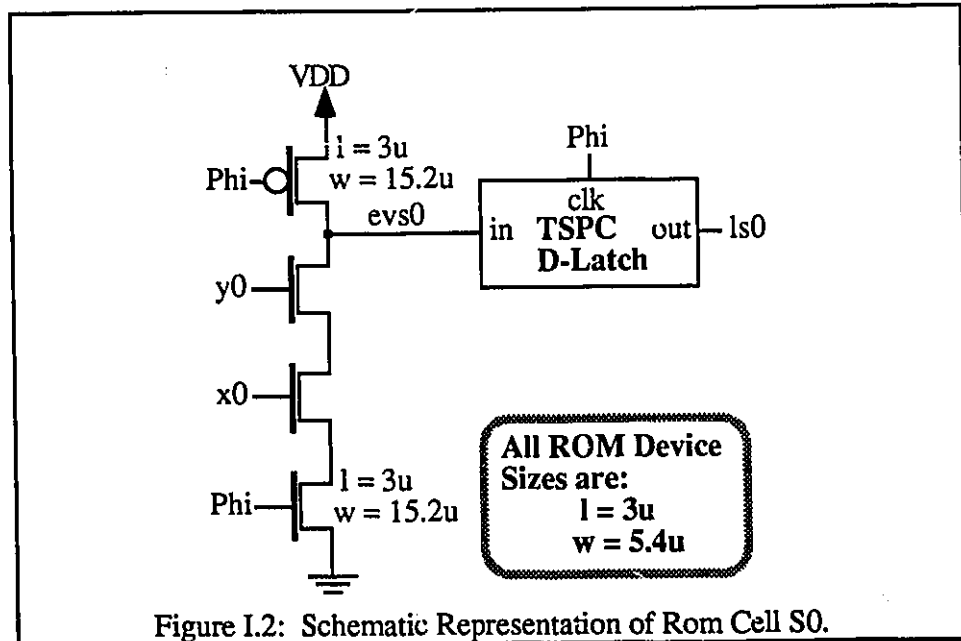
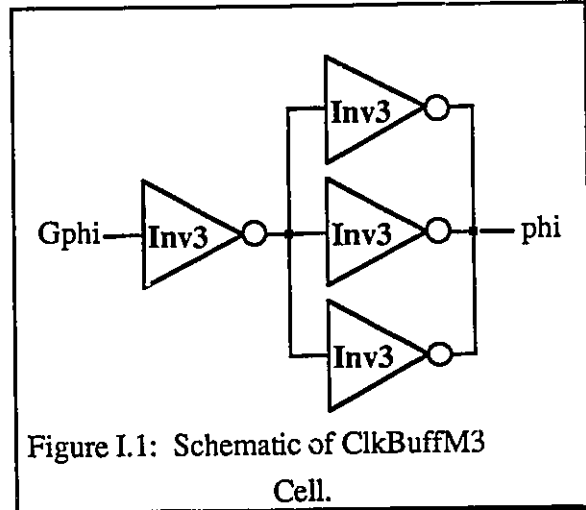
## APPENDIX I

### SCHEMATIC DIAGRAM OF 3-BIT MULTIPLIER CELL

## APPENDIX I

### SCHEMATIC DIAGRAM OF 3-BIT MULTIPLIER CELL

The Complete schematic listing of the 3-bit multiplier cell is given in Figures I.1 through I.8 inclusive. Figure I.1 represents the clock buffer circuit of the 3-bit multiplier cell. Gphi is the global input clocking signal to the cell, while phi is the local clock signal in the cell. The ROM cells that produce the 'S0' through 'S5' output are given in Figures I.2 through I.7. The input and output buffer of the 3-bit multiplier cell are shown in Figures I.8 and I.9, while the complete block schematic of the 3-bit multiplier cell is given in Figure I.10.





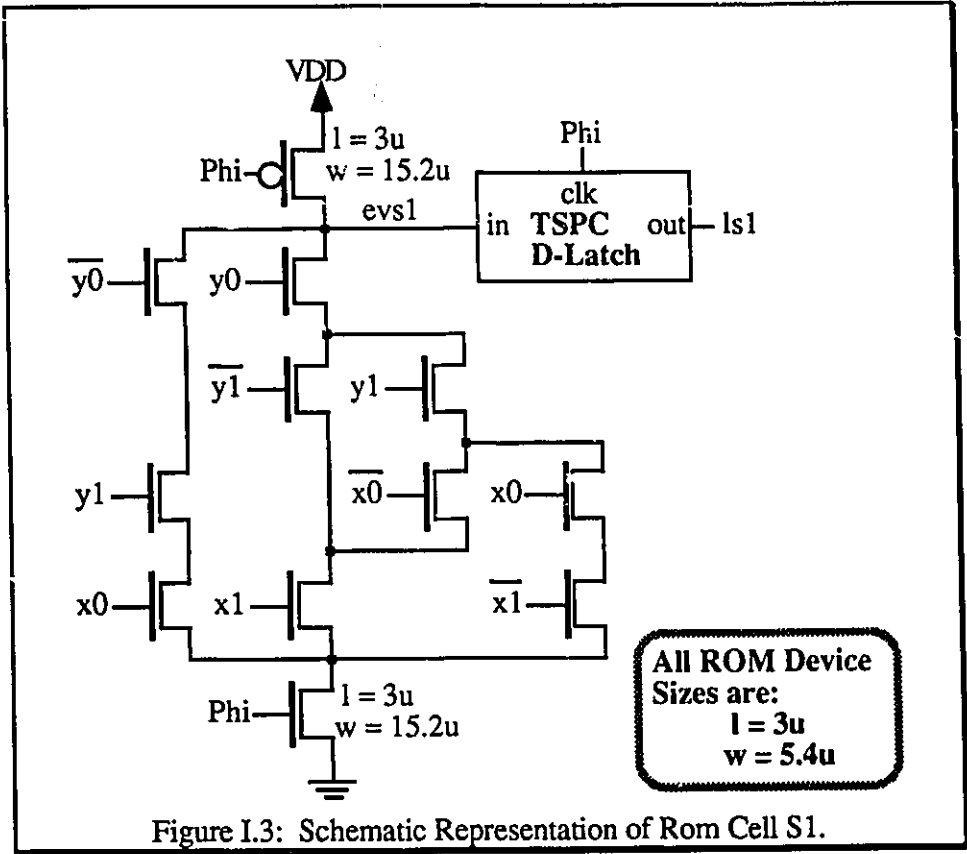


Figure I.3: Schematic Representation of Rom Cell S1.



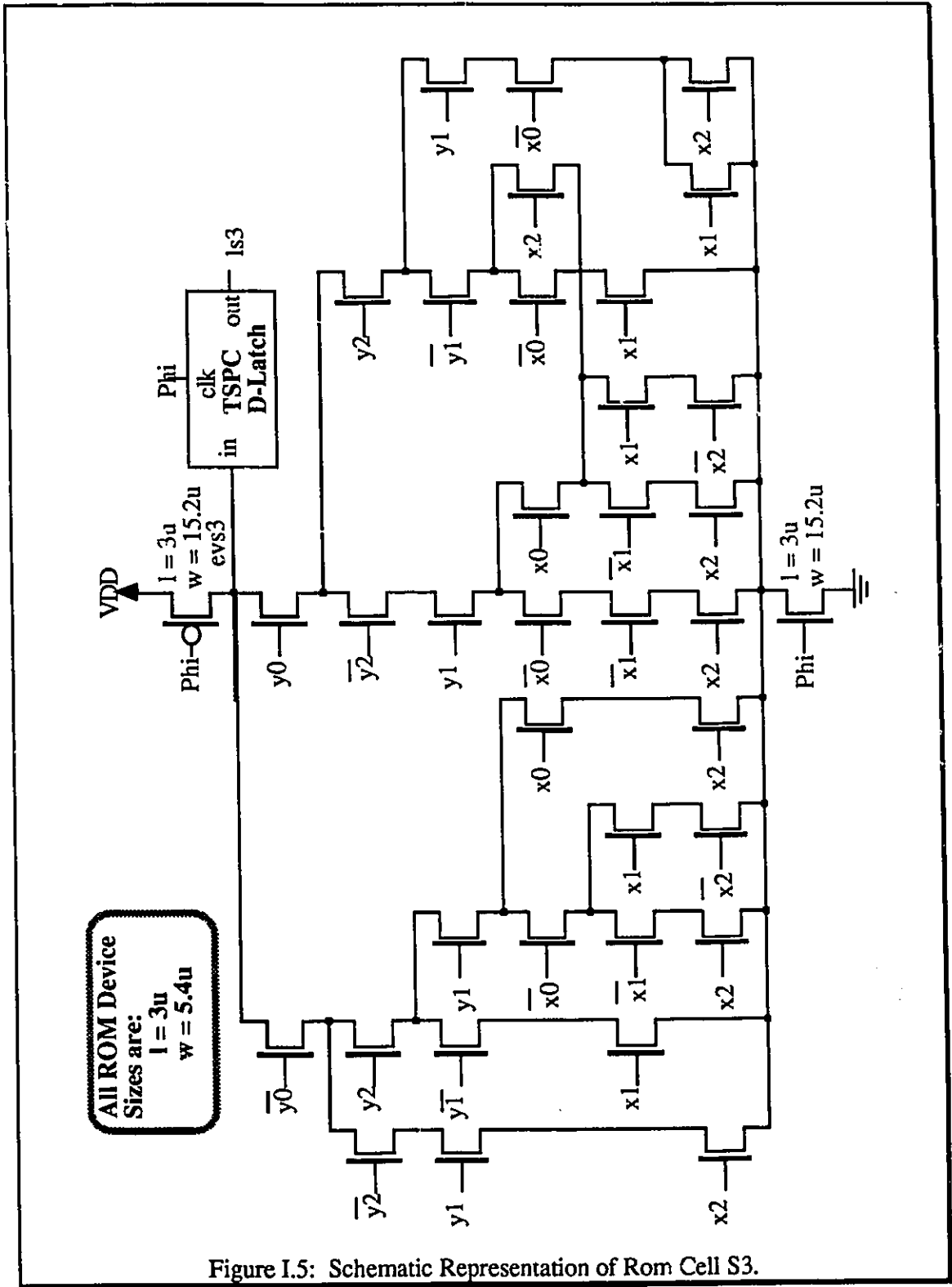
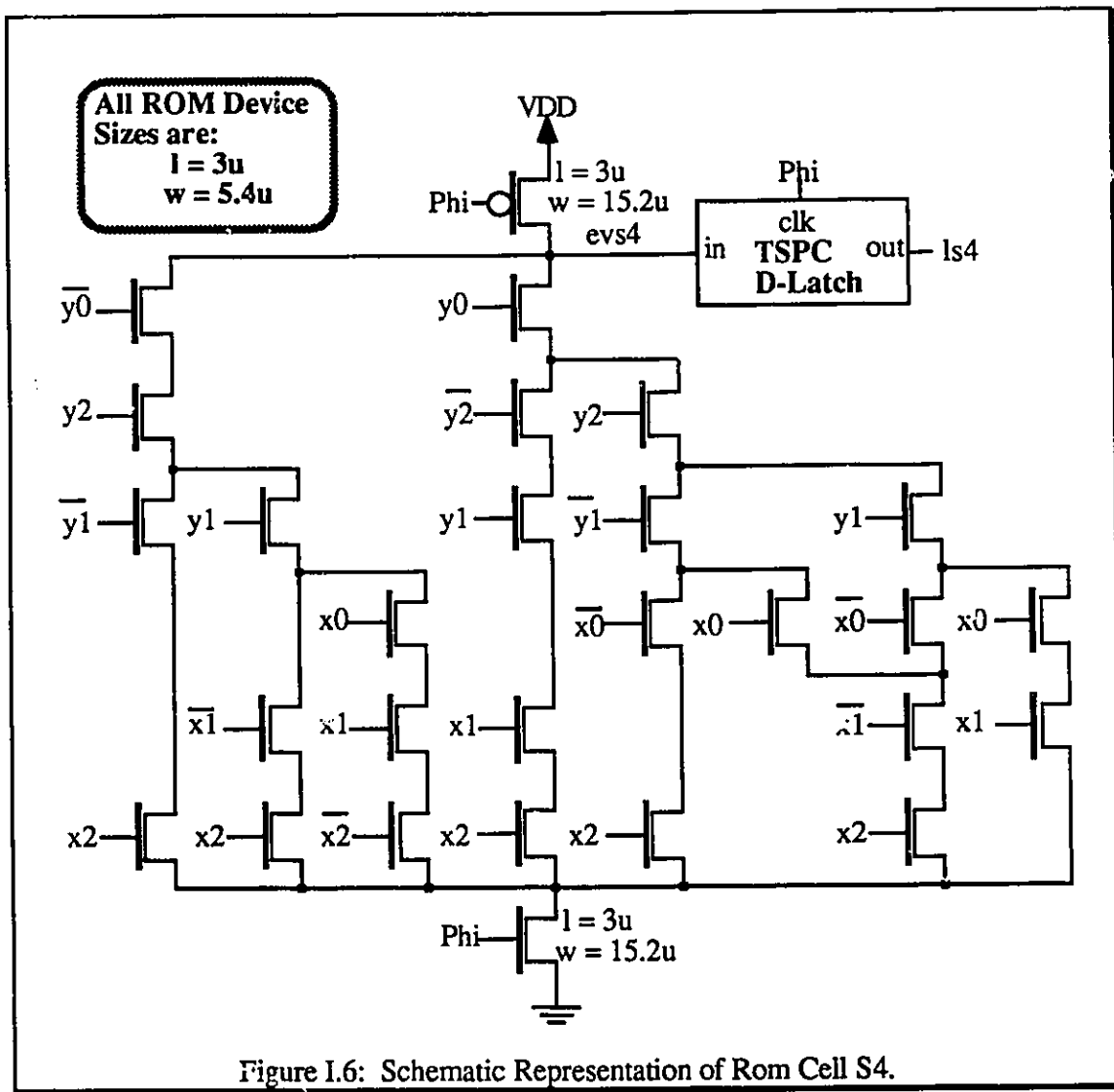


Figure I.5: Schematic Representation of Rom Cell S3.



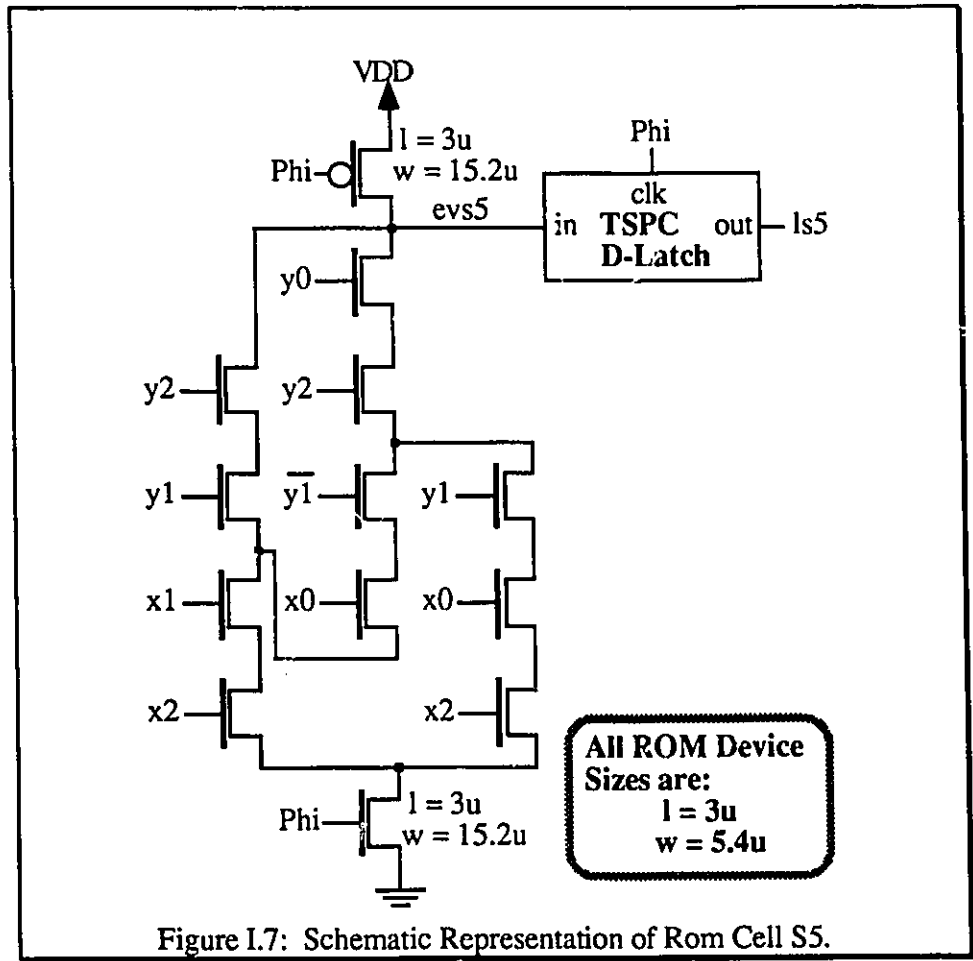


Figure I.7: Schematic Representation of Rom Cell S5.

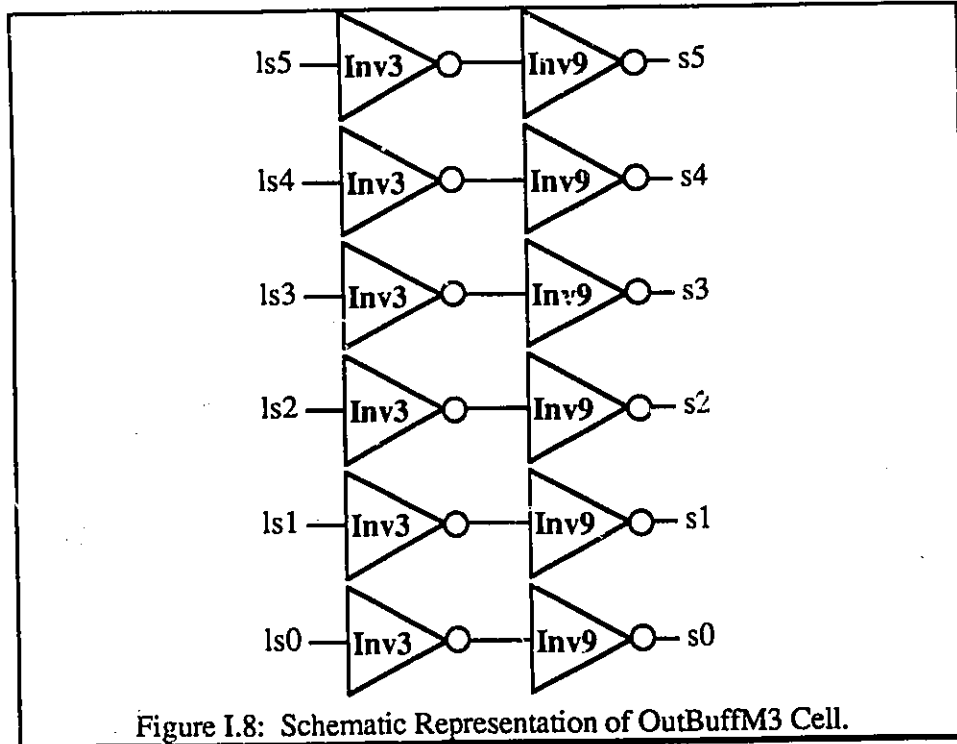


Figure I.8: Schematic Representation of OutBuffM3 Cell.

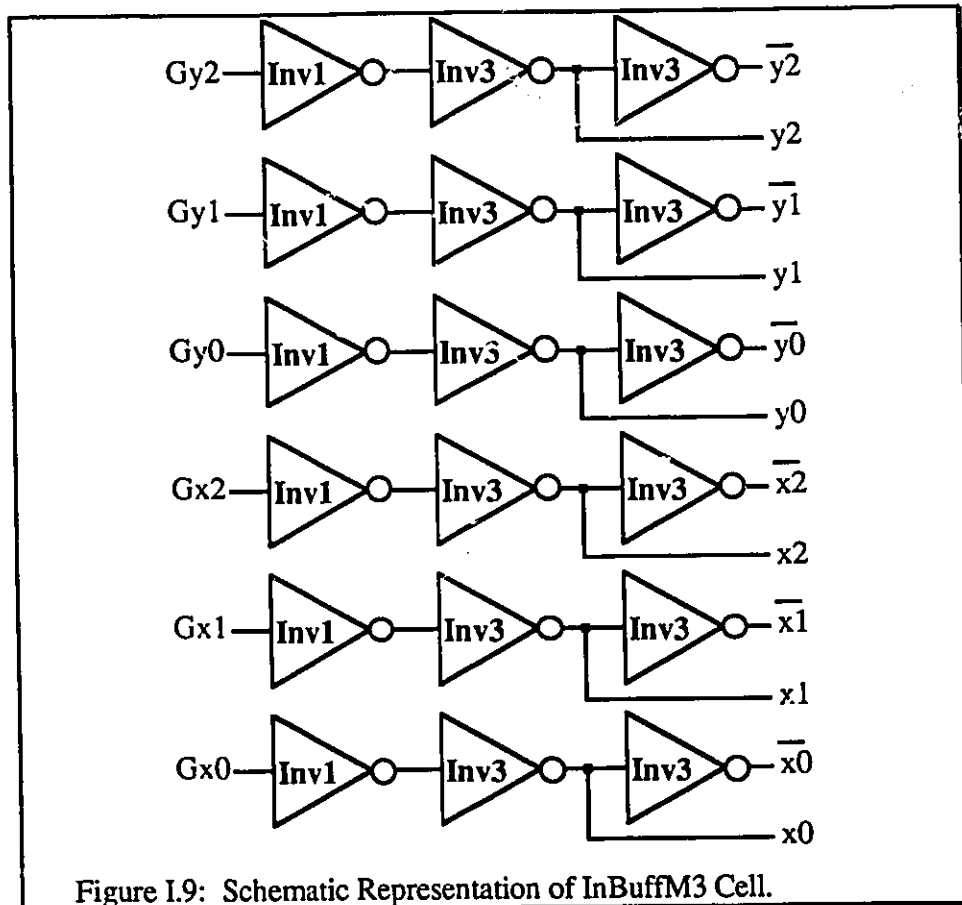
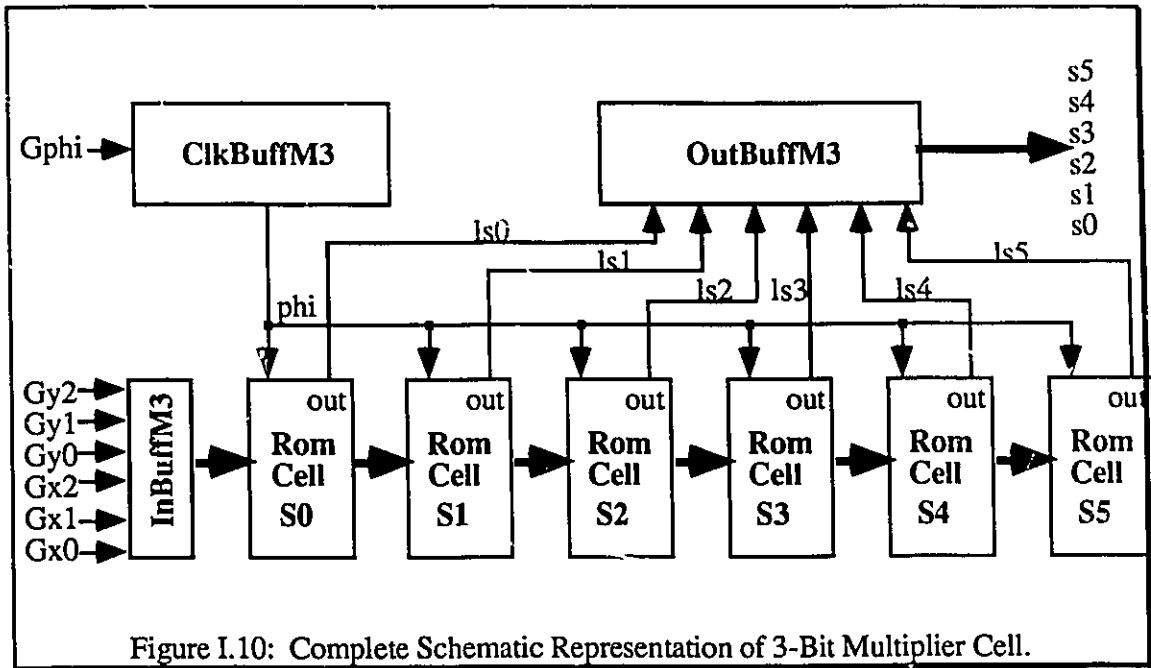


Figure I.9: Schematic Representation of InBuffM3 Cell.



## APPENDIX J

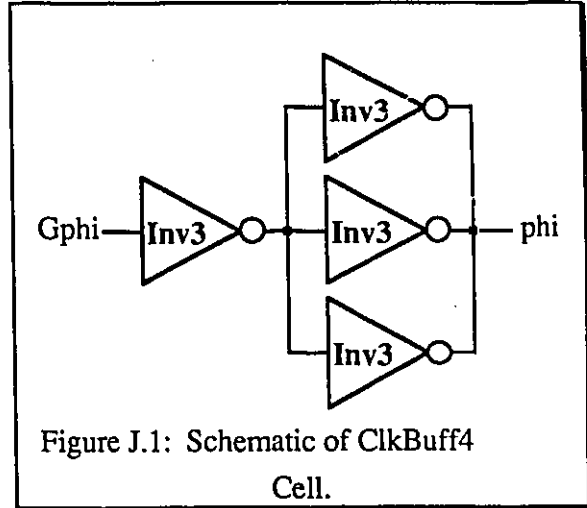
### SCHEMATIC DIAGRAM OF 4-BIT ADDER CELL



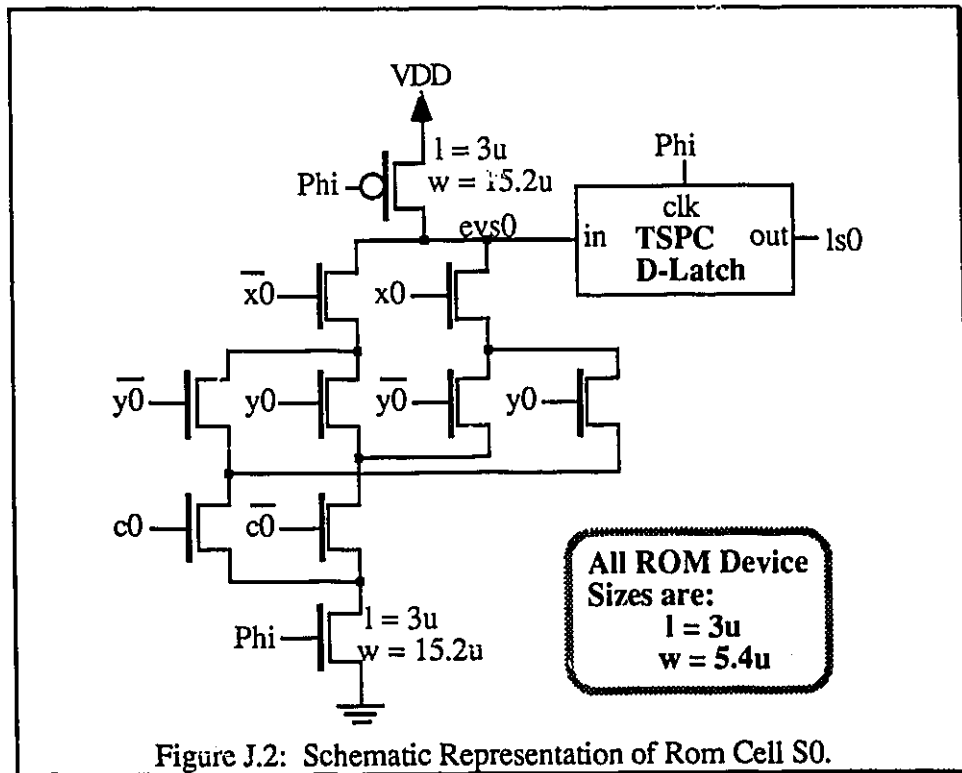
## APPENDIX J

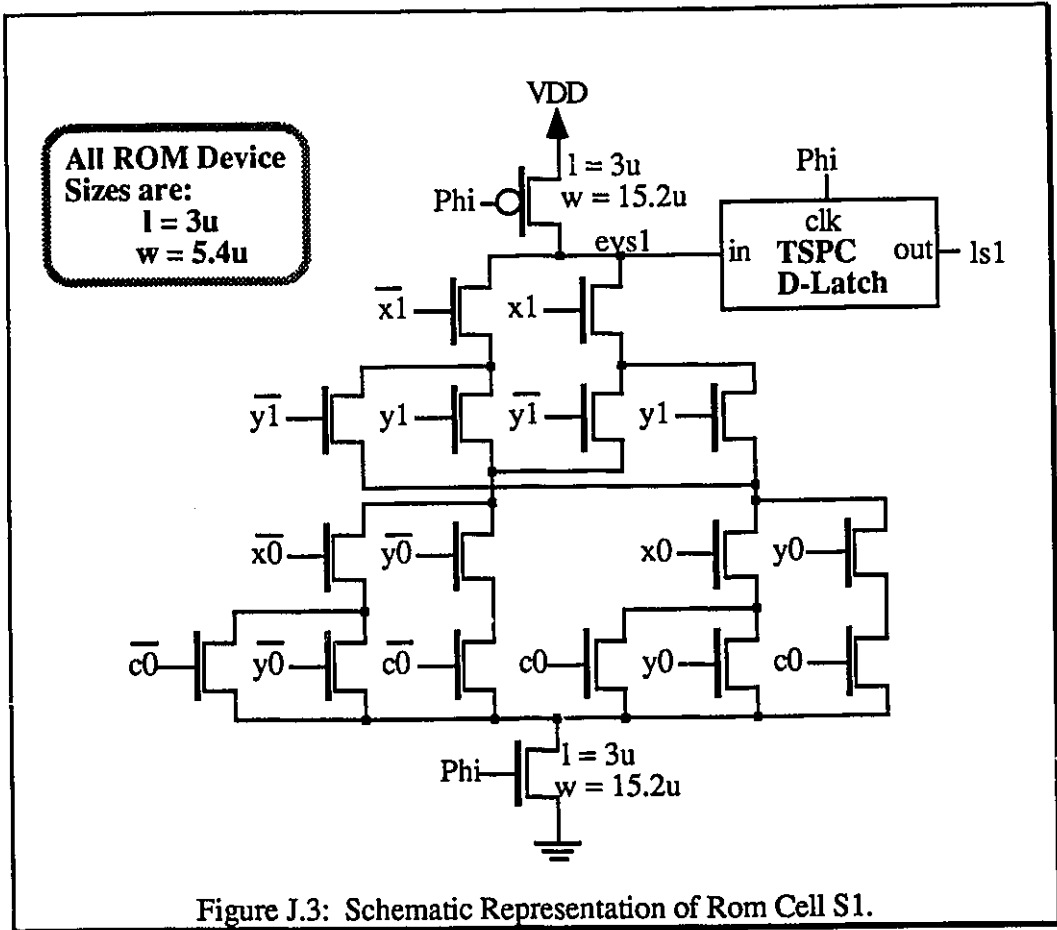
### SCHEMATIC DIAGRAM OF 4-BIT ADDER CELL

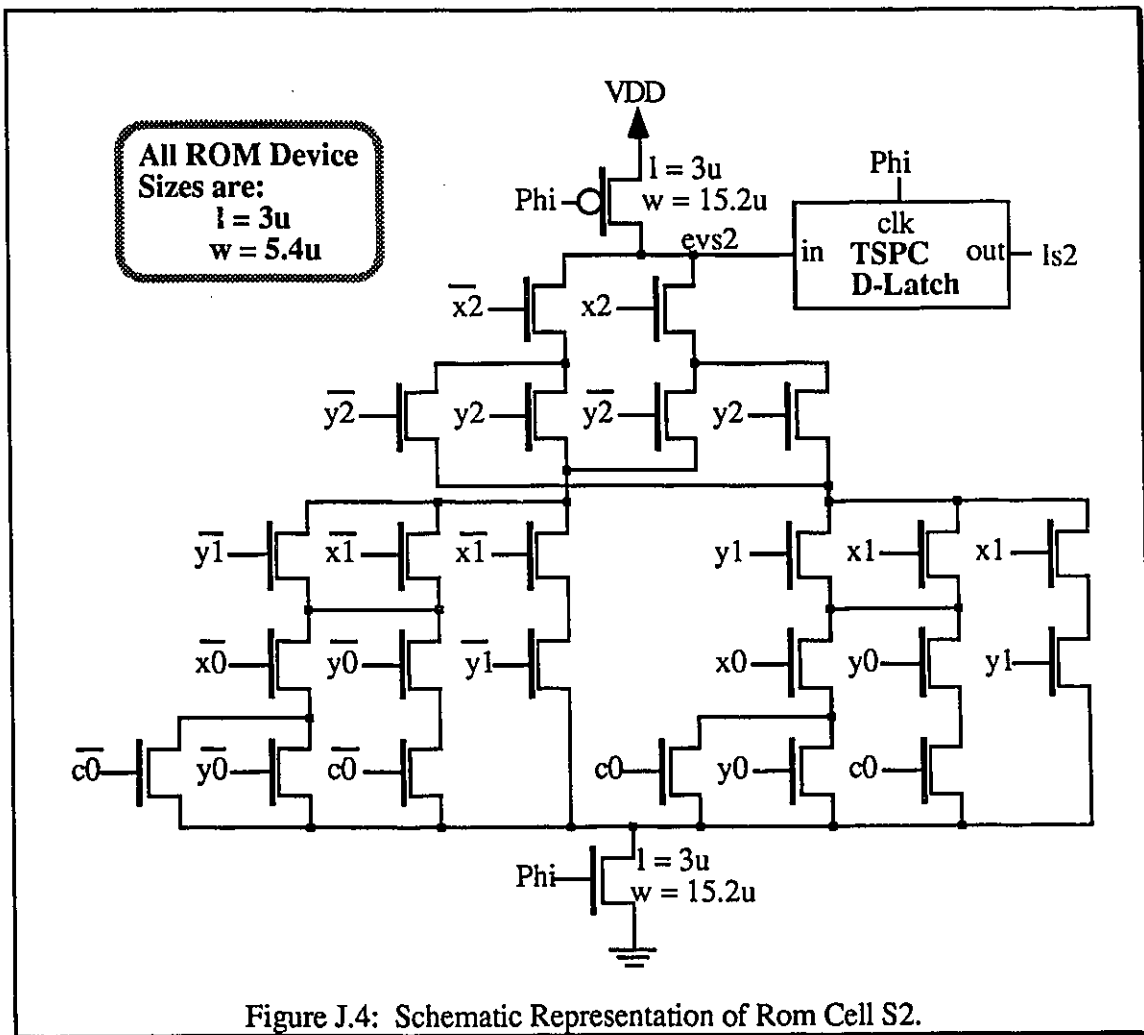
The complete 4-bit adder schematic is given in Figures J.1 through J.9 inclusive. Figure J.1 represents the clock buffer circuit for the 4-bit adder cell. Gphi is the global input clock signal, while phi is the local clock signal of the cell. Figures J.2 through J.6 represent the ROM cells which produce

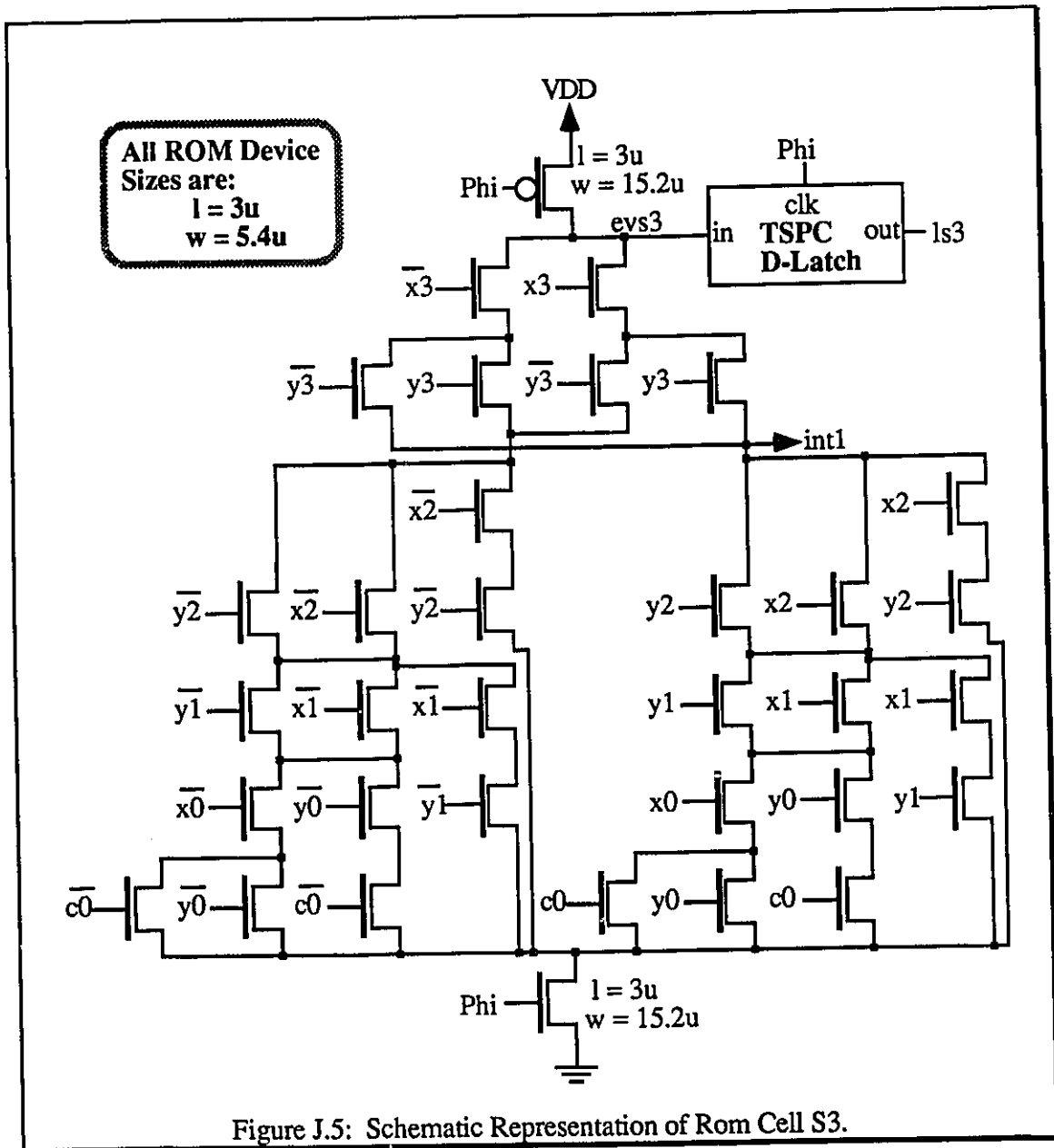


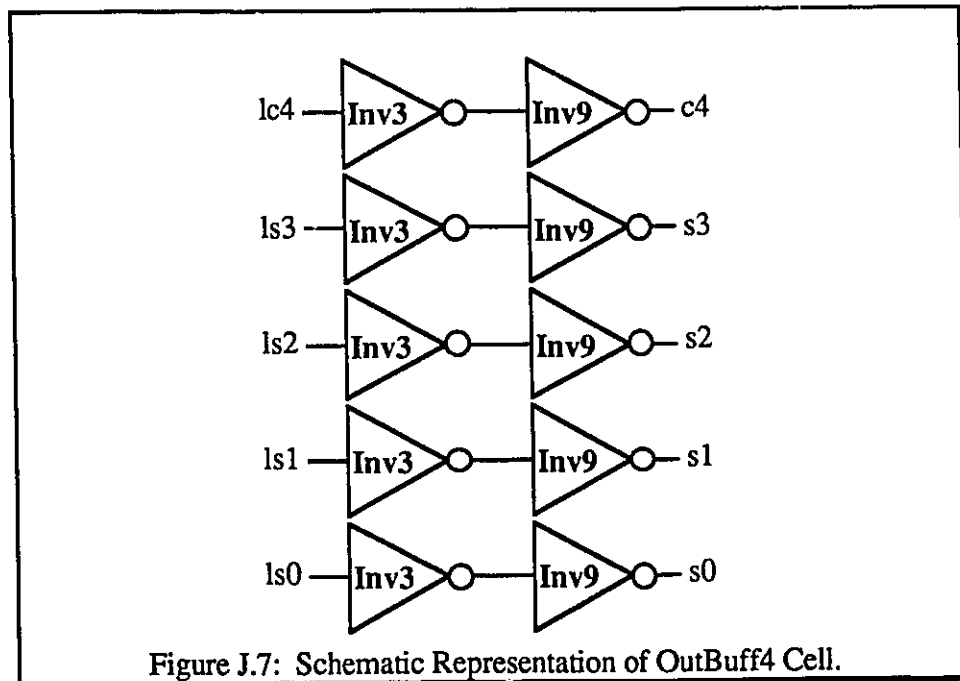
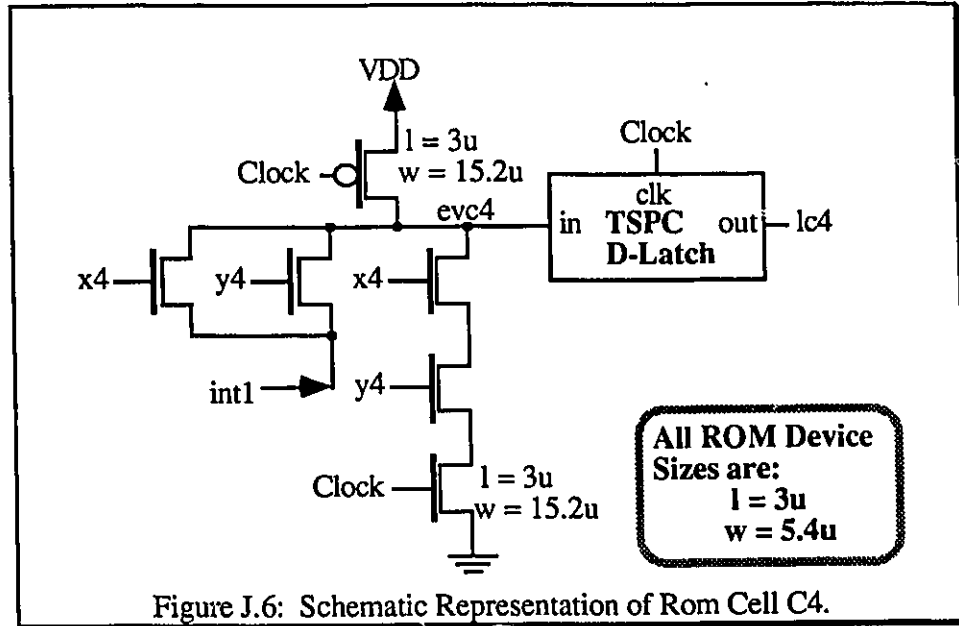
the 'S0' through 'S3', and 'C4' output signals. The input and output buffer cells for the 4-bit adder cell are shown in Figure J.7 and J.8, while the complete block diagram of the 4-bit adder cell is given in Figure J.9.











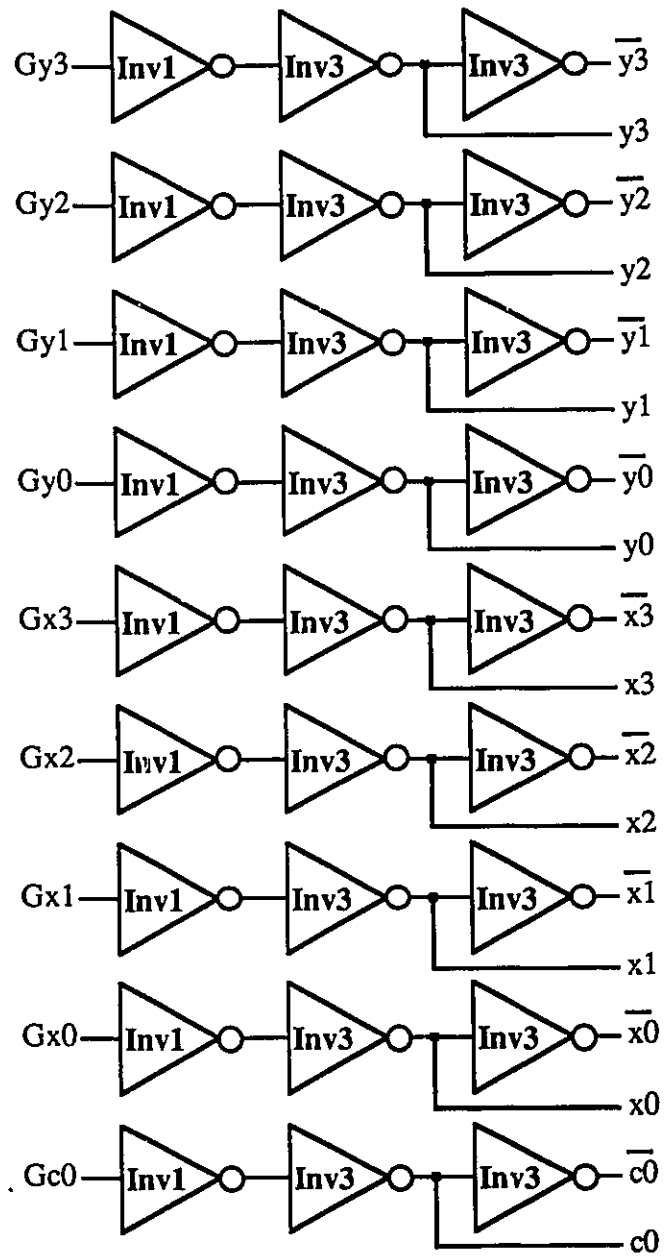
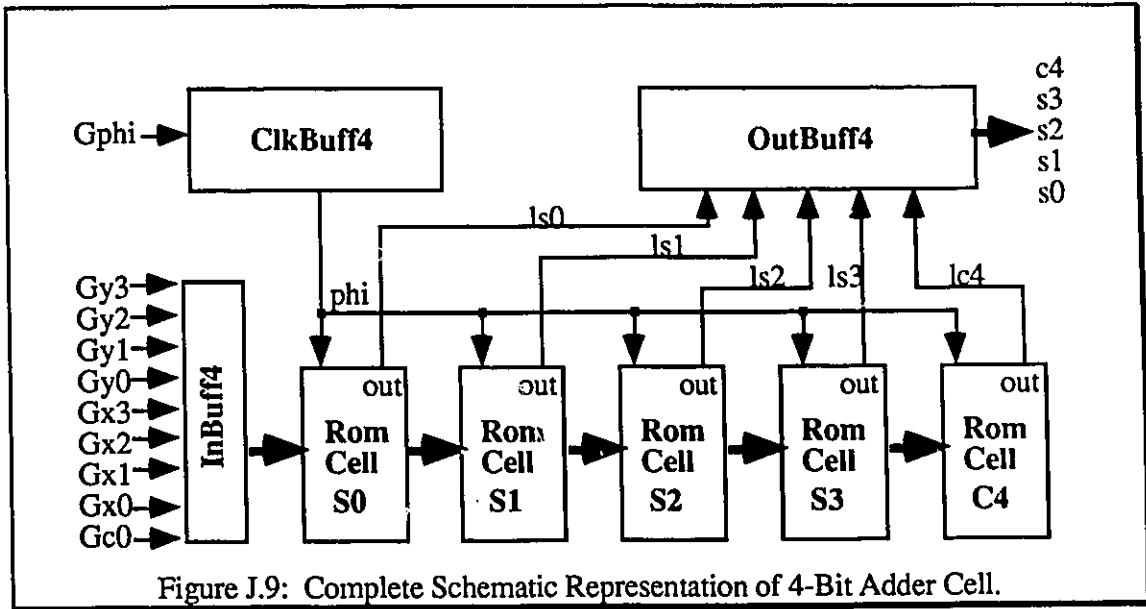


Figure J.8: Schematic Representation of InBuff4 Cell.



## APPENDIX K

### BASIC TRANSMISSION LINE THEORY



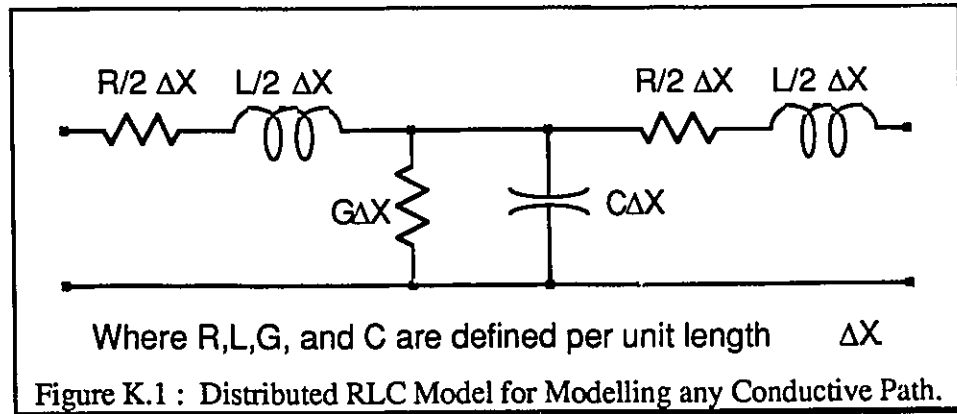
## APPENDIX K

### BASIC TRANSMISSION LINE THEORY

The following practical theory was taken from several sources dealing with fast pulse techniques as well as a designers guide to transmission lines and interconnections. [21][22][23][24]

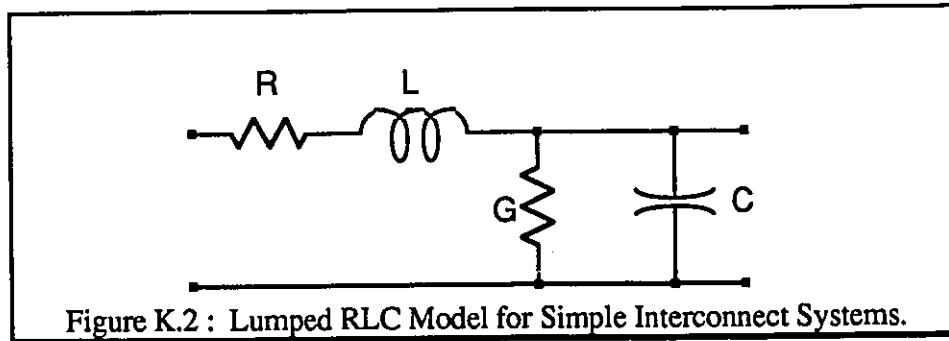
#### K.1 WHAT CONSTITUTES A TRANSMISSION LINE

The physical reality of any conductive path eliminates the possibility for a signal to propagate instantaneously down that conductor. A conductive path is normally modelled by a network of distributed capacitive, resistive, and inductive elements, whose magnitudes are governed by the physical make-up of the conductor and its surroundings, as illustrated in Fig K.1.



Systems whose physical elements contribute insignificant delays to the propagation of signal information down the conductor path may be treated as "lumped" component systems, where the conductor common components are lumped from a simpler interconnection realization, as illustrated in Fig K.2. Systems which are comprised solely of lumped elements obey Kirchoff's circuit laws, which fail for distributed element

interconnection systems. Transmission lines are conductive paths whose physical elements significantly affect the propagation of the signal information. In the case of transmission lines the physical elements that comprise the interconnect system must be treated as distributed elements, which requires a more involved method of analysis.



## K.2 DETERMINING POSSIBLE TRANSMISSION LINE SYSTEMS

There are two approaches to determine if a conductor may be affected by transmission line effects. The two approaches are based in: i/ the frequency domain, and ii/ the time domain.

### K.2.1 FREQUENCY DOMAIN APPROACH

If the length of interconnect in question is greater than 1/15 of the wavelength of the maximum frequency component,  $f_{max}$ , of the transmitted signal, then the conductor may be modelled as a simple lumped component system.

Procedure:

- 1/ Determine the highest frequency component present in the signal,  $f=f_{max}$ .
- 2/ Determine the velocity of propagation through the signal path,  $v$ .
- 3/ Determine the length of the signal path,  $l$ .

- 4/ Calculate the signal wavelength,  $\lambda = v/f_{max}$ .
- 5/ Calculate the wavelength-to-line-length ratio,  $\lambda/l$ .
- 6/ If  $\lambda/l$  is greater than or equal to 15 then consider the signal path to be a lumped element system, not a distributed system.

## K.2.2 TIME DOMAIN APPROACH

If the time required for a signal to travel the length of the interconnection is greater than 1/8 of the signal rise time, then consider the interconnection system to be a transmission line. Note that the rise time is equal to either the rising or falling edge time, whichever is smaller. The exact edge time corresponds to the linear-ramp amplitude from 0 to 100% of the signal magnitude. It is also important to keep in mind that digital circuits can tolerate transmission line effects better than analog circuits, and thus the ratio of signal rise time to one-way travel time,  $t_r/t$  can be  $\geq 4$  when examining interconnect in digital circuitry.

A simple correlation between the frequency and time domain proceeds as follows:

The bandwidth,  $f_{max}$ , of a pulse with rise time,  $t_r$ , is

$$f_{max} \approx 0.56/t_r.$$

Hence,  $\lambda = v/f_{max} = v t_r / 0.56$

The ratio of wavelength to line length is  $\lambda/l = v t_r/0.56l$

Thus the one way time delay is,  $t = l/v$ ,

$$\text{then } \lambda/l = t_r/0.56t = 1.8(t_r/t)$$

so, if  $t_r/t = 8$  then,  $\lambda/l = 1.8*8 = 14.4 \approx 15$ .

Procedure:

- 1/ Find  $t_r$ .
- 2/ Determine velocity of propagation,  $v$ .
- 3/ Determine the signal path length,  $l$ .
- 4/ Calculate the one-way propagation delay,  $t = l/v$ .
- 5/ Calculate the ratio of rise time to propagation delay,  $t_r/t$ .
- 6/ If  $t_r/t$  is greater or equal to 8, then the signal path is not a transmission line, and one can use lumped element analysis methods.

### B.2.3 LUMPED ELEMENT CALCULATIONS

Once it has been determined that lumped element analysis can be used, the following procedure can be used to determine the correct lumped model to be used.

- 1/ Determine the characteristic line impedance,  $Z_0$ , and the supply and load resistance,  $R_s$  and  $R_l$ .
- 2/ If  $R_s > Z_0$ , then use  $C_t = 1/(v \times Z_0)$
- 3/ If  $R_l < Z_0$ , then use  $L_t = Z_0/v$
- 4/ If Both  $R_s > Z_0$  and  $R_l < Z_0$ , then use only  $L_t = Z_0/v$ , as in 3/ above.
- 5/ If neither  $R_s > Z_0$  or  $R_l < Z_0$ , then no lumped elements can be used.

## APPENDIX L

### ECL RECIEVER AND DRIVER SPICE-DECKS

## APPENDIX L

### ECL RECIEVER AND DRIVER SPICE-DECKS

The following listing constitutes the mask-extracted SPICE-deck for the ECL Driver pad. The SPICE-deck is a back-anotated deck generated from Design Works Software running on a Macintosh system.

**outpad-07-12-90 Monday, June 3, 1991 10:52 AM**

\*\*\*\*\*

```
.include cmos3d.lib
.tran 0.35ns 30ns
```

```
vdd vdd 0 dc 5.0
vin in 0 pulse(0.05 4.95 1ns 2ns 2ns 8ns 20ns)
```

\*\*\*\*\*

```
* Termination Network *
*****
```

```
rline aline aout 50
vload aline 0 dc 3.0
cload aout aline 10000ff
```

\*\*\*\*\*

```
* Input Buffer Circuitry *
*****
```

```
mn8 S001 in 0 0 ntran l=3u w=27u pd=43.8u ad=226.8p
mn9 ain S001 0 0 ntran l=3u w=27u pd=43.8u ad=226.8p
mp24 S001 in vdd vdd ptran l=3u w=81u pd=97.8u ad=680.4p
mp25 ain S001 vdd vdd ptran l=3u w=81u pd=97.8u ad=680.4p
```

\*\*\*\*\*

```
* Main Circuit Description *
*****
```

```
ct1 vdd aref 200ff
mn1 aref aint1 0 0 ntran l=3u w=20u pd=14.4u ad=72p
mn2 aint1 aint1 0 0 ntran l=3u w=20u pd=14.4u ad=72p
mn3 ap ain 0 0 ntran l=3u w=50u pd=14.4u ad=180p
mn4 ap ain 0 0 ntran l=3u w=50u pd=14.4u ad=180p
mn5 aq aout 0 0 ntran l=3u w=50u pd=14.4u ad=180p
mn6 aq aout 0 0 ntran l=3u w=50u pd=14.4u ad=180p
mn7 ap ain 0 0 ntran l=3u w=50u pd=14.4u ad=180p
```

```

mp1 aref ap vdd vdd ptran l=3u w=60u pd=14.4u ad=130p
mp2 aint1 aq vdd vdd ptran l=3u w=60u pd=14.4u ad=130p
mp3 ap aref vdd vdd ptran l=3u w=150u pd=14.4u ad=540p
mp4 aq aref vdd vdd ptran l=3u w=150u pd=14.4u ad=540p
mp5 ap aq vdd vdd ptran l=3u w=150u pd=14.4u ad=540p
mp6 aq aq vdd vdd ptran l=3u w=150u pd=14.4u ad=540p
mp7 ap aq vdd vdd ptran l=3u w=150u pd=14.4u ad=540p
mp8 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp9 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp10 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp11 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp12 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp13 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp14 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp15 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp16 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp17 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp18 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp19 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp20 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp21 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp22 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p
mp23 aout ap vdd vdd ptran l=3u w=53.4u pd=56.4u ad=172.8p

```

```

*****
* End of Main Circuit Description *
*****
.end

```

The following listing constitutes the mask-extracted SPICE-deck for the ECL Receiver pad. The SPICE-deck is a back-anotated deck generated from Design Works Software running on a Machintosh system.

**Inpad-05-21-90 Monday, June 3, 1991 9:21 PM**

```
*****
```

```

.include cmos3d.lib
.tran 0.35ns 30ns

vdd vdd 0 dc 5.0
vdd2 vdd2 0 dc 5.0
vin ainb 0 pulse(0.05 4.95 1ns 2ns 2ns 8ns 20ns)

```

```
*****
```

```

* Termination Network *
*****

```

```

rline aload aline 50
cline aline aload 10000ff

```

vload aload 0 dc 3.0

\*\*\*\*\*  
\* Main Circuit Description \*  
\*\*\*\*\*

mn00001 ap1 aline 0 0 ntran l=3u w=21.6u pd=31u ad=97p nrd=0.21  
mn00002 ap1 aline 0 0 ntran l=3u w=21.6u pd=31u ad=97p nrd=0.21  
mn00003 ap1 aline 0 0 ntran l=3u w=21.6u pd=31u ad=97p nrd=0.21  
mn00004 ap1 aline 0 0 ntran l=3u w=21.6u pd=31u ad=97p nrd=0.21  
mn00005 aq1 aload 0 0 ntran l=3u w=21.6u pd=31u ad=97p nrd=0.21  
mn00006 aq1 aload 0 0 ntran l=3u w=21.6u pd=31u ad=97p nrd=0.21  
mn00007 aq1 aload 0 0 ntran l=3u w=21.6u pd=31u ad=97p nrd=0.21  
mn00008 aq1 aload 0 0 ntran l=3u w=21.6u pd=31u ad=97p nrd=0.21  
mn00009 ap2 aq2 0 0 ntran l=3u w=30u pd=39u ad=135p nrd=0.15  
mn00010 ap2 aq2 0 0 ntran l=3u w=30u pd=39u ad=135p nrd=0.15  
mn00011 ap2 aq2 0 0 ntran l=3u w=30u pd=39u ad=135p nrd=0.15  
mn00012 ap2 aq2 0 0 ntran l=3u w=30u pd=39u ad=135p nrd=0.15  
mn00013 aq2 aq2 0 0 ntran l=3u w=30u pd=39u ad=135p nrd=0.15  
mn00014 aq2 aq2 0 0 ntran l=3u w=30u pd=39u ad=135p nrd=0.15  
mn00015 aoutbar ap2 0 0 ntran l=3u w=3u pd=27.6u ad=38.2p nrd=3.6  
mn00016 aout aoutbar 0 0 ntran l=3u w=7.8u pd=32.4u ad=65.5p nrd=1.08  
mn00017 ain ainb 0 0 ntran l=3u w=27u pd=43.8u ad=226.8p  
mn00018 ain ainb 0 0 ntran l=3u w=27u pd=43.8u ad=226.8p  
mn00019 ain ainb 0 0 ntran l=3u w=27u pd=43.8u ad=226.8p  
mn00020 ain ainb 0 0 ntran l=3u w=27u pd=43.8u ad=226.8p  
mn00021 ain ainb 0 0 ntran l=3u w=27u pd=43.8u ad=226.8p  
mn00022 ain ainb 0 0 ntran l=3u w=27u pd=43.8u ad=226.8p  
mp1 aline ain vdd2 vdd2 ptran l=3u w=500u pd=22.8u ad=2850p  
mp2 aline ain vdd2 vdd2 ptran l=3u w=500u pd=22.8u ad=2850p  
mp3 aline ain vdd2 vdd2 ptran l=3u w=500u pd=22.8u ad=2850p  
mp4 aline ain vdd2 vdd2 ptran l=3u w=500u pd=22.8u ad=2850p  
mp00005 ap1 ap1 vdd vdd ptran l=3u w=55.2u pd=64.2u ad=248p nrd=0.082  
mp00006 ap1 ap1 vdd vdd ptran l=3u w=55.2u pd=64.2u ad=248p nrd=0.082  
mp00007 ap1 ap1 vdd vdd ptran l=3u w=55.2u pd=64.2u ad=248p nrd=0.082  
mp00008 ap1 ap1 vdd vdd ptran l=3u w=55.2u pd=64.2u ad=248p nrd=0.082  
mp00009 aq1 ap1 vdd vdd ptran l=3u w=55.2u pd=64.2u ad=248p nrd=0.082  
mp00010 aq1 ap1 vdd vdd ptran l=3u w=55.2u pd=64.2u ad=248p nrd=0.082  
mp00011 aq1 ap1 vdd vdd ptran l=3u w=55.2u pd=64.2u ad=248p nrd=0.082  
mp00012 aq1 ap1 vdd vdd ptran l=3u w=55.2u pd=64.2u ad=248p nrd=0.082  
mp00013 ap2 ap1 vdd vdd ptran l=3u w=13.2u pd=22.2u ad=59.4p nrd=0.34  
mp00014 ap2 ap1 vdd vdd ptran l=3u w=13.2u pd=22.2u ad=59.4p nrd=0.34  
mp00015 ap2 ap1 vdd vdd ptran l=3u w=13.2u pd=22.2u ad=59.4p nrd=0.34  
mp00016 ap2 ap1 vdd vdd ptran l=3u w=13.2u pd=22.2u ad=59.4p nrd=0.34  
mp00017 aq2 aq1 vdd vdd ptran l=3u w=13.2u pd=22.2u ad=59.4p nrd=0.34  
mp00018 aq2 aq1 vdd vdd ptran l=3u w=13.2u pd=22.2u ad=59.4p nrd=0.34  
mp00019 aq2 aq1 vdd vdd ptran l=3u w=13.2u pd=22.2u ad=59.4p nrd=0.34  
mp00020 aq2 aq1 vdd vdd ptran l=3u w=13.2u pd=22.2u ad=59.4p nrd=0.34  
mp00021 aoutbar ap2 vdd vdd ptran l=3u w=9.6u pd=36u ad=81p nrd=0.88  
mp00022 aout aoutbar vdd vdd ptran l=3u w=23.4u pd=61.2u ad=168p nrd=0.31  
mp00023 ain ainb vdd2 vdd2 ptran l=3u w=81u pd=97.8u ad=680.4p  
mp00024 ain ainb vdd2 vdd2 ptran l=3u w=81u pd=97.8u ad=680.4p  
mp00025 ain ainb vdd2 vdd2 ptran l=3u w=81u pd=97.8u ad=680.4p  
mp00026 ain ainb vdd2 vdd2 ptran l=3u w=81u pd=97.8u ad=680.4p



mp00027 ain ainb vdd2 vdd2 ptran l=3u w=81u pd=97.8u ad=680.4p  
mp00028 ain ainb vdd2 vdd2 ptran l=3u w=81u pd=97.8u ad=680.4p

\*\*\*\*\*

\* End of Main Circuit Description \*

\*\*\*\*\*

\* Summary Listing of Extracted Parasitic Capacitances \*

\*\*\*\*\*

c01 vdd aline 1457ff  
c02 vdd ap1 82ff  
c03 ap1 0 84ff  
c04 vdd aq1 77ff  
c05 aq1 0 52ff  
c06 aq1 ap1 5ff  
c07 aline 0 349ff  
c08 vdd aload 1366ff  
c09 aload 0 10ff  
c10 vdd aq2 46ff  
c11 aq2 0 80ff  
c12 vdd ap2 34ff  
c13 ap2 0 35ff  
c14 ap2 aq2 13ff  
c15 vdd aoutbar 14ff  
c16 aout 0 150ff

\*\*\*\*\*

.end

## VITA AUCTORIS

Lino Del Pup was born in 1965 in Windsor, Ontario. He graduated from Assumption High School in 1985. From there he went on to the University of Windsor where he obtained a B.A.Sc. in Electrical Engineering in 1989. He has been a member of the VLSI group at the University of Windsor since 1987. This thesis marks the completion of his requirements for the M.A.Sc. in Electrical Engineering specializing in the development of Very-Large-Scale-Integrated Circuits.