Custom design of CMOS read only memories for VLSI residue number system hardware.

Paruvachi Venkatachalam Ramasamy. Raja

University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation

Raja, Paruvachi Venkatachalam Ramasamy., "Custom design of CMOS read only memories for VLSI residue number system hardware." (1988). Electronic Theses and Dissertations. 1598.

https://scholar.uwindsor.ca/etd/1598

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) or by telephone at 519-253-3000ext. 3208.
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.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

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

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 originelles ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.
CUSTOM DESIGN OF CMOS READ ONLY MEMORIES FOR VLSI
RESIDUE NUMBER SYSTEM HARDWARE

by
Paruvachi Venkatachalam Ramasamy Raja

A thesis
submitted to the
Faculty of Graduate Studies and Research
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, Canada

1987
Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

ISBN 0-315-48181-1
ABSTRACT

Re-evaluation of Residue Number System (RNS) arithmetic architectures using components provided by the modern Very Large Scale Integration (VLSI) technology and design techniques such as systolic arrays results in highly modular and efficient architectures for hardware realizations of Digital Signal Processing (DSP) algorithms. Arithmetic operations in the RNS hardware arrays are carried out by basic cells based on Read Only Memories (ROMs) which form the backbone of RNS architectures. It is clear that ROM technology is important for any RNS study, and efficient fast ways of implementing ROMs are important tools in any serious study of RNS arithmetic in a VLSI environment. The goals of this thesis are to design ROMs suitable for RNS applications. CMOS fabrication technology provided by the Canadian Microelectronics Corporation (CMC) is used as a vehicle technology to implement sample test structures.

During initial investigations, a static CMOS ROM cell structure is developed and a 49*3 ROM is designed. The static cell structure is improved and a 24*5 ROM is designed, fabricated, and tested.

In order to simplify the ROM design, dynamic CMOS techniques for ROM are investigated. As a result of initial investigations, a dynamic Domino CMOS decoder is designed, fabricated, and tested. Based on the ideas gathered by a literature survey, a dynamic memory cell structure is developed and a 24*5 ROM is designed, fabricated, and tested. The dynamic ROM cell structure is improved with the help of decoders and a
32x7 dynamic ROM is designed in the recently introduced CMOS Dual In Line Memory process technology. Using this ROM a fault-tolerant systolic cell structure is designed based on a recently introduced technique. Based on this systolic cell a 5-bit pipelined adder is designed which implements the inner product step processor for higher level architectures. The clocks of the ROM are effectively coalesced with the pipeline clocks. Since the predominant feature of the pipelined adder is the 32x7 ROM the area of the ROM is reduced using improved layout of cells. For the purpose of obtaining direct comparisons, a 32x7 static ROM is designed using a developed cell structure. These two ROMs are accurately simulated using SPICE and RELAX simulators. A comparative study reveals that the 32x7 dynamic CMOS ROM is faster, smaller, and dissipates lower amount of power than the 32x7 static CMOS ROM. Using the cell structure of the 32x7 dynamic ROM, ROMs of different sizes can be designed.
ACKNOWLEDGEMENTS

I am very much thankful and indebted to my advisor Dr. G.A. Jullien for his guidance, valuable suggestions, constructive criticisms, constant encouragement, patience, understanding, and consistent support. He is a source of inspiration and stimulation. Working with him is enjoyable and rewarding. My heart-felt appreciation goes to Dr. W.C. Miller for his suggestions, encouragement, and support.

I wish to extend my thanks to other committee members for their constructive comments and criticisms, and the staff members and secretaries of the department of Electrical Engineering for their co-operation.

I would like to express my appreciation to the past and the present VLSI group members for providing an excellent environment for dedicated research, the Canadian Microelectronics Corporation for their service, the NSERC for support, Mr. Donald Chambers of the Leddy library for his timely help, and Mr. Bruce Erickson for his help in testing the fabricated IC designs.

Finally, I would like to thank my parents, brothers, and my sister for their affection, dedication, spiritual support, patience, and understanding during the entire course of my study at the University of Windsor.
TABLE OF CONTENTS

ABSTRACT ................................................................. i
ACKNOWLEDGEMENTS ..................................................... ii
TABLE OF CONTENTS ....................................................... iv
LIST OF FIGURES .......................................................... vii
LIST OF TABLES ............................................................ xii
CHAPTER 1: INTRODUCTION ................................................ 1
  1.1 Overview ............................................................. 1
  1.2 The Residue Number System ....................................... 1
  1.3 ROMs and the RNS hardware ...................................... 3
  1.4 Statement of the objective ....................................... 5
    1.4.1 Systolic arrays and RNS .................................. 7
  1.5 Summary ............................................................. 11
CHAPTER 2: VLSI AND CMOS TECHNOLOGIES ......................... 12
  2.1 Overview ............................................................. 12
  2.2 CMOS technology .................................................. 14
  2.3 CMOS Inverter ...................................................... 14
    2.3.1 Transfer characteristics ................................ 16
  2.4 Dynamic CMOS gates ............................................... 18
  2.5 Switching specifications of CMOS circuits .................... 20
  2.6 Layout techniques for CMOS circuits ............................ 22
    2.6.1 Layout of a CMOS Inverter ................................ 23
    2.6.2 Techniques for better layout of CMOS ICs ............... 23
  2.7 Summary ............................................................. 27
CHAPTER 3: DESIGN OF CMOS READ ONLY MEMORIES ................. 28
  3.1 Overview ............................................................. 28
3.2 Organization of a Read Only Memory ........................................... 28
3.3 An initial design ............................................................................. 31
  3.3.1 Design of a 49*3 ROM .............................................................. 33
3.4 Design of a 24*5 static ROM ......................................................... 38
  3.4.1 Alternative layout of the 24*5 static ROM ............................... 40
3.5 Design of dynamic CMOS ROMs .................................................... 45
  3.5.1 Design of a 24*5 dynamic ROM ............................................... 52
3.6 Comparative study .......................................................................... 59
3.7 Design of 32*7 Read Only Memories .............................................. 62
  3.7.1 Design of a 32*7 dynamic ROM .............................................. 64
3.8 Design of a 5-bit RNS adder .......................................................... 73
3.9 Improved layout of the 32*7 dynamic CMOS ROM ................. 84
3.10 Design of a 32*7 Static ROM ....................................................... 88
3.11 Comparative study ....................................................................... 98
3.12 Summary .................................................................................. 101

CHAPTER 4: SIMULATION ................................................................. 103
4.1 Overview ...................................................................................... 103
4.2 Switch level simulation ................................................................. 104
4.3 Layout level simulation of ROMs .................................................. 105
  4.3.1 SPICE simulation of the 32*7 dynamic ROM .......................... 105
  4.3.2 SPICE simulation of the 32*7 static ROM ............................... 113
4.4 RELAX simulation of ROMs ........................................................ 116
  4.4.1 RELAX simulation of the 32*7 dynamic ROM ....................... 116
  4.4.2 RELAX simulation of the 32*7 static ROM ........................... 117
  4.4.3 RELAX simulation of a fault-tolerant systolic cell .................. 117
4.5 Comparison of simulation results ................................................. 126
4.6 Power dissipation ................................................. 130
  4.6.1 Standby power dissipation ................................... 132
  4.6.2 Dynamic power dissipation .................................. 132
4.7 Comparison of the dynamic CMOS 32*7 ROM and the CMOS ROMs in the literature ...................... 134
4.8 Summary ......................................................... 134

CHAPTER 5: CONCLUSION AND DISCUSSION

  5.1 Overview ...................................................... 136
  5.2 Characteristics of final ROM designs ......................... 137
  5.3 The final conclusions ........................................ 138

REFERENCES ........................................................... 140

APPENDIX 1: Layout of basic components of ROMs .............. 143
APPENDIX 2: SPICE Input deck for the 32*7 dynamic ROM .... 150
APPENDIX 3: SPICE Input deck for the 32*7 static ROM ....... 155
APPENDIX 4: RELAX Input deck for the 32*7 dynamic ROM .... 160
APPENDIX 5: RELAX Input deck for the 32*7 static ROM ...... 165
APPENDIX 6: RELAX Input deck for the Fault-Tolerant Systolic cell ........................................... 170
APPENDIX 7: FORTRAN program for generation of various
  layout options for a ROM .................................... 178
APPENDIX 8: FORTRAN program for optimization of CMOS
  circuits ..................................................... 184
APPENDIX 9: Testing of fabricated relevant designs .......... 198
VITA AUCTORIS .......................................................... 219
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Organization of a typical RNS processor</td>
<td>4</td>
</tr>
<tr>
<td>1.2</td>
<td>Use of ROMs to implement [ a \times b \mid 31 \mid c \times d \mid 31 \mid 31 ]</td>
<td>6</td>
</tr>
<tr>
<td>1.3</td>
<td>A systolic array for DFT computation using RNS arithmetic</td>
<td>9</td>
</tr>
<tr>
<td>2.1</td>
<td>A custom VLSI design flow</td>
<td>13</td>
</tr>
<tr>
<td>2.2(a)</td>
<td>Schematic diagram of a CMOS Inverter</td>
<td>15</td>
</tr>
<tr>
<td>2.2(b)</td>
<td>Transfer characteristics of a CMOS Inverter</td>
<td>15</td>
</tr>
<tr>
<td>2.3</td>
<td>CMOS dynamic logic gate</td>
<td>19</td>
</tr>
<tr>
<td>2.4</td>
<td>Switching specifications of CMOS ICs</td>
<td>21</td>
</tr>
<tr>
<td>2.5</td>
<td>Vertical layout of a CMOS Inverter</td>
<td>24</td>
</tr>
<tr>
<td>2.6</td>
<td>Horizontal layout of a CMOS Inverter</td>
<td>25</td>
</tr>
<tr>
<td>3.1</td>
<td>A general organization of a ROM</td>
<td>30</td>
</tr>
<tr>
<td>3.2</td>
<td>Complementary transistor ROM cell</td>
<td>32</td>
</tr>
<tr>
<td>3.3</td>
<td>A representative 2-word, 3-bit ROM using a static CMOS memory cell</td>
<td>34</td>
</tr>
<tr>
<td>3.4(a)</td>
<td>A 3-to-7 decoder using static AND3 gates</td>
<td>36</td>
</tr>
<tr>
<td>3.4(b)</td>
<td>Schematic of a static CMOS 3-input AND gate</td>
<td>36</td>
</tr>
<tr>
<td>3.5</td>
<td>Layout of the 49*3 static ROM in CMOS3 process</td>
<td>37</td>
</tr>
<tr>
<td>3.6</td>
<td>A representative 3-word, 3-bit ROM using an improved static CMOS ROM cell</td>
<td>39</td>
</tr>
<tr>
<td>3.7(a)</td>
<td>Schematic diagram of a 3-input NAND-Invert gate</td>
<td>41</td>
</tr>
<tr>
<td>3.7(b)</td>
<td>Schematic diagram of a 3-to-6 decoder using 3-input NAND-Invert gate</td>
<td>41</td>
</tr>
<tr>
<td>3.8</td>
<td>Layout organization of the 24*5 static CMOS ROM</td>
<td>42</td>
</tr>
<tr>
<td>3.9</td>
<td>Layout of the 24*5 static CMOS ROM in CMOS3 process</td>
<td>43</td>
</tr>
<tr>
<td>3.10</td>
<td>Improved layout of the 24*5 ROM in CMOS3 process</td>
<td>44</td>
</tr>
</tbody>
</table>
3.11(a) Schematic of a 3-input Domino CMOS AND gate ............ 46
3.11(b) Schematic of a 3-to-7 decoder using
        Domino CMOS gates........................................ 46
3.12  Layout of the 3-to-7 decoder using
        Domino CMOS gates........................................ 47
3.13  A representative 2-word ROM using
        pseudo n-MOS logic........................................ 50
3.14  A representative 2-word, 3-bit dynamic ROM using a
devolved CMOS memory cell ................................. 53
3.15(a) A representative circuit for simulation of the
        24*5 dynamic ROM........................................ 56
3.15(b) Simulation results of the representative circuitry
        of the 24*5 ROM ........................................ 57
3.16  Layout organization of the 24*5 dynamic ROM .......... 58
3.17  Layout of the 24*5 dynamic ROM in CMOS 3-micron
        process.................................................... 60
3.18(a) Schematic of a 3-to-6 decoder using NOR gates ...... 61
3.18(b) Schematic of a 3-to-6 decoder using NOR gates ...... 61
3.19  Schematic of a fault-tolerant ROM based
        systolic cell............................................. 63
3.20  A representative 2-word, 3-bit dynamic CMOS ROM
        using an improved memory technique....................... 65
3.21  Layout organization of the 32*7 dynamic CMOS ROM..... 67
3.22  Schematic of a 3-to-8 dynamic CMOS decoder using AND
gates............................................................ 68
3.23  Layout of the 32*7 dynamic CMOS ROM in CMOS 3-micron
        double level metal technology......................... 69
3.24(a) Simulation circuit of the 32x7 dynamic ROM to test the output logic levels................................. 70
3.24(b) Results of the simulation............................................. 72
3.25(a) Organization of the layout of the 32x7 dynamic ROM based fault-tolerant systolic cell......................... 74
3.25(b) Layout of the 32x7 dynamic ROM based fault-tolerant systolic cell.................................................. 75
3.26(a) Schematic diagram of a 2-input OR gate......................... 76
3.26(b) Schematic diagram of a 2-input exclusive OR gate........... 76
3.26(c) Schematic diagram of a 2-phase latch............................. 77
3.27(a) Two-phase non-overlapping clocks................................. 78
3.27(b) Two-phase non-overlapping clocks and the fault-tolerant systolic cell.......................................... 78
3.28 Waveforms of the fault-tolerant ROM based systolic cell...................................................................... 79
3.29 5-bit RNS addition process............................................. 81
3.30 Schematic of a 5-bit RNS adder using the fault-tolerant ROM based systolic cell................................... 82
3.31 A complete layout of the 5-bit RNS adder.......................... 83
3.32 Improved layout of the 32x7 dynamic ROM.......................... 85
3.33 Organization of the improved layout of the 32x7 dynamic ROM................................................................. 86
3.34 Layout of a word of the 32x7 dynamic ROM........................ 87
3.35 A representative 2-word, 3-bit static ROM using the developed complementary cell technique.................... 89
3.36 Layout organization of the 32x7 static ROM....................... 92
3.37 Layout of the 32x7 static ROM in CMOS3 process................. 94
3.38 A 3-to-8 decoder using 3-input NAND-Invert gates.............. 95
3.39(a) Simulation circuit of the 32*7 static ROM to test the output logic levels.......................... 96
3.39(b) Simulation results of the 32*7 static ROM test circuit.............................................. 97
4.1 LOGCAP simulation of a 24*5 ROM......................... 104
4.2 Estimation of resistance in a layout.................. 107
4.3 Circuit model for the SPICE simulation of the 32*7 dynamic ROM.................................. 110
4.4 Results of the SPICE simulation of the 32*7 dynamic ROM............................................ 112
4.5 Circuit model for the SPICE simulation of the 32*7 static ROM........................................ 114
4.6 Results of the SPICE simulation of the 32*7 static ROM.............................................. 115
4.7(a) Output of the 32*7 dynamic ROM..................... 118
4.7(b) Clock for the 32*7 dynamic ROM.................... 119
4.7(c) Output of the 3-input Domino CMOS AND gate...... 120
4.7(d) Output of the 2-input AND gate....................... 121
4.8(a) Output of the p-channel ROM cell in the 32*7 static ROM........................................... 122
4.8(b) Output of the n-channel ROM cell in the 32*7 static ROM........................................... 123
4.9 Systolic cell for SPICE simulation....................... 124
4.10(a) Output of the 32*7 dynamic ROM.................. 125
4.10(b) Clock CL1 of the two-phase non-overlapping clocks 127
4.10(c) Clock CL2 of the two-phase non-overlapping clock... 128
4.10(d) Output of part A of the latch....................... 129
4.10(e) Output of part B of the latch ................. 130
6.1 Layout of the components of the 49*3 static ROM .... 144
6.2 Layout of the components of the 24*5 static ROM ... 145
6.3 Layout of the components of an improved
    24*5 static ROM ........................................... 146
6.4 Layout of the components of the 24*5 dynamic ROM ... 147
6.5 Layout of the 3-input Domino CMOS gate ........... 148
6.6 Layout of the components of the 32*7 dynamic ROM
    and the fault-tolerant systolic cell ................. 149
8.1 A CMOS Inverter and its input/output waveform pair .. 187
8.2 Area*Time optimization curve .......................... 197
9.1 Test chip for the 24*5 static ROM ................... 201
9.2 Test chip for the 3-to-7 dynamic CMOS decoder ...... 202
9.3 Test chip for the 24*5 dynamic ROM ................. 203
9.4 Bonding diagram of the fabricated 24*5 static ROM
    test chip .................................................. 205
9.5 Bonding diagram of the fabricated 3-to-7 dynamic
    CMOS decoder test chip .................................. 208
9.6 Bonding diagram of the fabricated 24*5 dynamic ROM
    test chip .................................................. 211
9.7 Expected test results of the 24*5 static ROM ....... 212
9.8 Obtained test results of the 24*5 static ROM ....... 214
9.9 Expected test results of the 3-to-7 dynamic
    CMOS decoder ............................................. 215
9.10 Obtained test results of the 3-to-7 dynamic
    CMOS decoder ............................................. 216
9.11 Expected test results of the 24*5 dynamic CMOS ROM .. 217
9.12 Obtained test results of the 24*5 dynamic CMOS ROM .. 218
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>ROM design comparison</td>
<td>59</td>
</tr>
<tr>
<td>3.2</td>
<td>Comparison of 32*7 ROMs laid out in CMOS3DLM</td>
<td>99</td>
</tr>
<tr>
<td>4.1</td>
<td>Sheet resistance of CMOS3DLM layers</td>
<td>106</td>
</tr>
<tr>
<td>4.2</td>
<td>Capacitance of CMOS3DLM layers</td>
<td>109</td>
</tr>
<tr>
<td>4.3</td>
<td>Comparison of simulation results</td>
<td>126</td>
</tr>
<tr>
<td>4.4</td>
<td>Power dissipation</td>
<td>133</td>
</tr>
<tr>
<td>4.5</td>
<td>Comparison of the 32*7 dynamic CMOS ROM and the CMOS ROMs reported in the literature</td>
<td>133</td>
</tr>
<tr>
<td>5.1</td>
<td>Characteristics of final ROM designs</td>
<td>137</td>
</tr>
<tr>
<td>9.1</td>
<td>Test results of the 24*5 static ROM</td>
<td>204</td>
</tr>
<tr>
<td>9.2</td>
<td>Test results of the 3-to-7 dynamic CMOS decoder</td>
<td>207</td>
</tr>
<tr>
<td>9.3</td>
<td>Test results of the 24*5 dynamic ROM</td>
<td>209</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Overview
The ever-increasing demands for performance, sophistication, and real-time digital signal processing strongly indicate the need for tremendous computation capability, in terms of both volume and speed. VLSI component technology has recently opened a floodgate of high performance components that have altered the cost performance criteria for all digital systems. The feasibility of systems consisting of programmable and configurable VLSI chips is exerting its influence on a wide range of new application areas in digital signal processing. This makes feasible ultra-high throughput rates and presages major technological breakthroughs in real-time signal processing applications [32]. In the following sections, we will discuss a type of arithmetic known as residue number system, its use for high speed implementation of digital signal processing algorithms, and the objective of this thesis.

1.2 The Residue Number System
The residue number system has been considered by many researchers as an alternative to a weighted number system to provide high speed hardware implementation of modern digital signal processing algorithms. A standard residue number system (RNS) is characterized by a set of L pairwise relatively prime integers
(m_1, m_2, ..., m_L) i.e. (m_i, m_j) = 1 \text{ for } i \neq j \text{ The interval } (0, M), \text{ with } M = \prod_{i=1}^{L} m_i \text{ will be called the total computational range (dynamic range) of the residue number system. Normally, the total range is partitioned into positive and negative regions so that negative quantities can be represented in complement notation. If } M \text{ is odd, the dynamic range of the residue representation is } [-\frac{(M-1)}{2}, \frac{(M-1)}{2}], \text{ if } M \text{ is even the dynamic range is } [-\frac{M}{2}, \frac{M}{2}-1]. \text{ Each natural integer, } X, \text{ in the dynamic range is mapped onto the legitimate range and represented as a sequence of residue digits as shown below.}

X = (X_1, X_2, \ldots, X_L) \quad \text{------------- (1)}

where

\[ X_i = \left| X \right|_{m_i} \quad \text{for } [0, (m_i - 1)/2) \]

\[ X_i = m_i - \left| X \right|_{m_i} \quad \text{for } [-\frac{(m_i - 1)}{2}, 0] \]

where \((m_i)\) is the set of moduli.

The residue operations are carried out as follows:

\((X_1, X_2, \ldots, X_L) \ast (Y_1, Y_2, \ldots, Y_L) = (Z_1, Z_2, \ldots, Z_L)\)

\[ Z_i = (X_i \ast Y_i) \mod m_i, \quad \text{------------- (2)} \]

\[ = ((X_i \ast Y_i) \mod m_i, \ldots, (X_L \ast Y_L) \mod m_L) \]

where \(\ast\) is one of addition, multiplication, and subtraction operators. It should be noted here that \(\text{GCD}(m_i, m_j) = 1\) for \(i \neq j\). It is clear that the suboperations with in each modulus are independent of each other; i.e., no carry information is passed
between the moduli. The absence of carry requirements means that the concept of significance of digit is void. This unique property of RNS arithmetic offers advantages over the conventional fixed-radix or weighted number system, in which the carry information must be passed from digits of lesser significance to those of greater significance; this slows down the arithmetic computations by an amount related to the carry management system used e.g. ripple carry, look-ahead carry, etc. The carry process can be accelerated but only at the expense of additional hardware. The unique property of the RNS overcomes these tradeoffs and allows the design of parallel architectures to process all modular partial sums and products concurrently.

1.3 ROMS and the RNS Hardware

In an RNS processor, the operation starts with decoding the incoming data by residue reduction with respect to a series of prime or relatively prime numbers in the encoder logic. The encoded incoming data are processed by many individual channels. Operation of each channel is controlled by a modulus. The number of moduli or the number of individual processing channels depends on the dynamic range of the RNS processor. The dynamic range \( M \) can be calculated from the product of bases as follows:

\[
M = \prod_{i=1}^{L} m_i
\]

where \( L \) is the number of moduli.

A typical organization of an RNS processor is shown in Fig. 1.1. This processor has \( L \) number of channels. Within each
Fig. 1.1 Organization of a typical RNS processor
channel, the data equivalent to a single base is processed and then sent to the final stage. In the final stage of processing the data is converted back to the original weighted binary number system. The conversion is produced by a mixed radix decoder or a decoder based on the Chinese remainder theorem (CRT). In each stage of this processor, several arithmetic calculations, such as addition, multiplication, and subtraction controlled by a modulus, take place. The most common approach to realize arithmetic operations in RNS hardware is by the use of look-up tables to store all the possible outcomes [1].

This approach is much more viable and efficient for RNS systems than binary systems [40]. Feasibility of saving some hardware is achieved by fixing some of the operands. Constants can be premultiplied or added, and stored along with the binary operation being implemented. Also, the look-up table approach offers the best solution for high speed realization. This is particularly advantageous in multiplication, which becomes as simple and fast as addition. Fig. 1.2 illustrates a residue multiplier for the modulus 31, followed by a residue adder to implement the function \( |a \times b|_{31} + |c \times d|_{31} |_{31}. \) In this pipelined structure, the output of each ROM is stored in a latch and becomes a part of the address for the next ROM. The only control function required is a latch pulse. For every latch pulse new input is accepted and a new output result is generated. The throughput of this array is equal to the inverse of the ROM access time plus latch settling time [40].
Fig. 1.2  Use of ROM's to implement $\lfloor A\times B \rfloor_{31} + \lfloor C\times D \rfloor_{31}$
1.4 Statement of the objective

Implementation vehicles for RNS architectures have traditionally used standard digital components, such as high density ROMs, binary adders, etc [31]. With the capabilities of full custom VLSI design we are able to re-evaluate the implementation architectures in the form of array architectures such as Systolic Arrays. It is clear that Read Only Memory technology is important for any RNS study, and efficient fast ways of implementing ROMs are important tools in any serious study of RNS arithmetic in a VLSI environment. The goals of this thesis are to conduct such a study.

1.4.1 Systolic arrays and RNS

Systolic arrays have been proposed as a cost-effective solution to computation-intensive problems. They consist of simple cells operating synchronously, each communicating only with nearby cell. Their applications range from numeric tasks, such as signal and image processing and matrix arithmetic, to symbolic tasks, such as searching and sorting, relational data bases [34].

The essential features and the potential advantages of systolic architectures are [34]:

1) They are highly modular, limiting the design and layout problem.

2) Interconnections are short and regular, minimizing the time delays.

3) The computation is carried out in a highly parallel manner, with the active regions distributed across the entire array area.
4) Data flows through the array at a uniform rate along fixed paths so that the control overheads are small.

5) Data transfer to and from the array is minimized, giving a high functional throughput without the need for fast memory links.

Interesting structures evolve from the application of the systolic array concept to RNS DSP applications. Several structures have been proposed in the literature. One such structure is the systolic array proposed by Bayoumi, Jullien, and Miller for computing the DFT (Discrete Fourier Transform) [6]. The DFT can be defined as follows:

\[
Y(k)_j = \sum_{k=0}^{N-1} W_j^{n_k} X(n)_j \bmod_{M_1}
\]

for \(k = 0, 1, 2, \ldots, N-1\), where \(X(n)\) and \(Y(k)\) are the residue representation of the \(j\)th residue of the input and output signals, and \(w = e^{-2\pi i/N}\). Fig. 1.3 shows the cell structure and the systolic array. The multiplication and addition operations involved in this cell are performed by ROM based look-up tables.

A new systolic array based RNS filter has been proposed by Bayoumi [7]. This structure uses two computational modules known as CAS-1 and CAS-2. These two modules carry out multiplication and addition operations through the use of ROM based look-up tables. Systolic architectures with simple cell structures can be seen in [8]. In this work, a bit level systolic structure has been implemented for RNS multiplication. The basic cell contains
Fig. 1.3 A systolic array for DFT computation using RNS arithmetic
two small ROMs (16x3, 16x2) to carry out arithmetic operations. This work also compares the distributed ROM array structures with traditional single ROM array structures, with advantages for the VLSI array structures.

One of the recent developments in RNS systolic arrays is the use of a single ROM to build a basic cell [9]. Techniques for realizing arithmetic functions such as additions and multiplications for DSP applications are given using inner product step processors built with ROMs of size 23x5. The same authors have shown that in another work, using 23x5 ROMs, both residue serial and bit serial systolic arrays for implementing FIR filters can be constructed, and these programmable generic cells can be connected to form linear systolic arrays that are 100% efficient in cell utilizations. It is shown that these cells are compact and form a basic building block that can be used for many RNS implementations other than pre-programmed FIR filters.

These developments imply that there is a tremendous amount of potential for RNS systolic array VLSI implementations using inner product step processors based on ROM look-up tables. Due to the distributed use of ROM arrays, several small ROMs (usually of size less than or equal to 32x5) are used instead of a single high density ROM. Hence, there is a need for custom designed small ROMs to be used in fundamental building blocks of RNS systolic arrays for high speed VLSI implementations. Hence, the design objective of this thesis is set to design small, simple, fast, reconfigurable, and pipelinable ROMs suitable for fabrication using the CMOS VLSI fabrication technology offered by
the Canadian Microelectronics Corporation to the University. This CMOS technology is a p-Well CMOS technology, which allows us to create enhancement mode p- and n-channel MOS transistors with fixed threshold voltages [15]. The remaining chapters of this thesis are devoted to the special design of ROMs that form the backbone of RNS arithmetic architectures.

1.5 Summary
The residue number system (RNS) has been reviewed. The RNS provides parallelism and speed to hardware structures through carry free operations. RNS hardware structures make use of Read Only Memories as look-up tables to compute arithmetic operations. A VLSI array design methodology known as systolic arrays has been used to achieve high throughput by pipelining computational elements. It has been shown that these VLSI systolic arrays can be used to carry out high speed RNS computations by the use of distributed, small ROM blocks. The custom design of CMOS ROMs for VLSI systolic RNS structures has been taken as the objective for this thesis work.
2.1 Overview

Current VLSI technology allows the fabrication of many hundreds of thousands of switching devices on a single silicon chip with feature size approaching one micron. VLSI stands for a technology that is capable of creating systems in a single silicon die. Many years of research experience in solid state electronics, circuit design, and processing have been fully utilized in producing high quality integrated circuits for achieving solutions to the complicated problems regarding high functionality, high packing density, low power dissipation, short delay time, minimum cost, maximum reliability, etc.

Custom VLSI chips have acquired a very important position in system design by offering smaller size, higher number of functions, lower cost, higher speed, and higher reliability than using many LSI chips. In the custom IC design flow, reduction of design cycle time is an important concept. Several design methodologies have been proposed to reduce the turn-around time of design and fabrication of IC chips. Fig. 2.1 shows one such design-methodology [10]. Starting at a given specification, the design proceeds from the functional design through logic design, and then layout in which masks are prepared and, finally, wafers are fabricated. In the following sections, we discuss CMOS
Fig. 2.1  A custom VLSI design flow
technology, operation of CMOS circuits, different CMOS logic circuits, switching specifications of CMOS ICs, and layout of CMOS circuits.

2.2 CMOS technology for VLSI circuit design

The CMOS (Complementary Metal Oxide Semiconductor) technology is considered as a suitable technology for VLSI implementation due to the following reasons [35]:

1) low-power density,
2) tolerance to changes in operating conditions,
3) large noise immunity and soft error protection,
4) overall design simplification and layout ease, and
5) flexibility with ability to implement both digital and analog functions on the same chip.

CMOS also has advantages for analog circuits:

1) CMOS switches have no offset voltage,
2) CMOS permits large reduction in area required for an operational amplifier, and
3) CMOS allows symmetric operations around the ground potential.

Therefore, CMOS is the technology now being rapidly pursued for most of the new integrated circuit designs.

2.3 CMOS Inverter

The fundamental logic gate in the CMOS logic family is the Inverter. The schematic diagram of the Inverter is shown in Fig. 2.2(c). In this configuration, there are two MOS transistors, namely, the p-channel transistor, PG, and the n-channel transistor, NG. The p-channel and n-channel transistors’
Fig. 2.2(a) Schematic diagram of a CMOS Inverter

Fig. 2.2(b) Transfer characteristics of a CMOS Inverter
substrates are tied to their respective sources. The input node is common to both the transistors, and the output node is at the point where the drains of both the transistors are connected.

Under static conditions both devices conduct zero current. If we consider the p-channel device to be the load (the source connected to +Vdd), then when Vin = 0 the n-channel is cut off and the p-channel device forms a channel, thus Vout=Vdd. For Vin=Vdd the p-channel device is cut off and the n-channel device forms a channel, thus Vout=Vss.

2.3.1 Transfer characteristics
As a first step, we shall review the operation of MOS devices. The MOS transistors operate primarily in two modes [36]:
1) non-saturated mode
2) saturated mode

Non-saturated mode:
The current \( I_D = \mu \times W/L \times Co \times [(Vgs-Vt) \times Vds - Vds^2/2] \)
The condition is met for \( Vds+Vt \leq Vgs \)

\[ Co = \frac{\epsilon_0 \times Cr}{d0} \]
\[ d0 = \text{gate oxide thickness} \]
\[ \mu = \text{mobility of the carriers in channel} \]
\[ W = \text{channel width}; L = \text{channel length} \]
\[ Vt = \text{The threshold voltage of a MOS transistor} \]
\[ Vgs = \text{The gate to source voltage of a MOS transistor} \]
\[ Vds = \text{The drain to source voltage of a MOS transistor} \]
\[ Vdd = \text{The positive supply voltage} \]
\[ Vss = \text{The ground supply voltage} \]

Saturated mode:
\[ I_D = k \times (Vgs - Vt)^2 \]
where \( k = \mu \times (W/L) \times Co/2 \)
We refer to Fig. 2.2(a) to understand the transfer characteristics [36]. Assume the threshold voltage (magnitude) is the same for both p- and n-channel transistors. (=Vt)

\[ \text{Vout} = \text{Vdd} \text{ for } \text{Vin} \leq \text{Vt}. \text{ Vout} = 0 \text{v for } \text{Vin} \geq \text{Vdd - Vt} \]

The transfer characteristic is shown in Fig. 2.2(b)

Saturation points: the load device is saturated when

\[ \text{Vdp} = \text{Vgp - Vt} \text{ or } \text{Vdd - Vout} = \text{Vdd} - \text{Vin} - \text{Vt} \]

which gives \( \text{Vout} - \text{Vin} = \text{Vt} \).

The driver is saturated for \( \text{Vdn} = \text{Vgn} - \text{Vt} \) or \( \text{Vout} = \text{Vin} - \text{Vt} \)

when both devices are saturated

\[ \text{Id} = K_L(Vgp - Vt)^2 = K_L(Vdd-Vin-Vt)^2 \]

also \( \text{Id} = K_D(Vgn-Vt)^2 = K_D(Vin - Vt)^2 \)

The above-mentioned CMOS Inverter is said to be static because the gate is triggered by the data path signal and does not require the use of an external clock. Based on the above Inverter technique, several CMOS static logic gates are available. They are:

1) AND gates,
2) OR gates,
3) NAND gates,
4) NOR gates,
5) Exclusive NOR gates, and
6) Exclusive OR gates

Detailed descriptions of these gates are available in the literature [11,12]. All these gates, with different numbers of inputs, are used in the thesis. Some of these gates are discussed
at appropriate places in the following chapters.

In this thesis work, several static and dynamic logic techniques for ROMs have been discussed. We have seen the principles behind the design of static CMOS, now we discuss the principles behind dynamic CMOS, particularly Domino CMOS which uses precharge/discharge logic. This precharge/discharge logic has been the basis for operation of many ROMs discussed in this thesis.

2.4 Dynamic CMOS gates

The Inverter circuit shown in the Fig. 2.2(a), is the standard complementary MOS structure for logic gates. It requires two transistors for each logical input and provides a redundant realization for the combinational logic function being realized. Krambeck et al. recognized that the standard complementary structure required more area and that it presented approximately twice the load capacitance of comparable NMOS gates [13]. One approach to increasing the performance of CMOS circuits is to use a p-type depletion load transistor in the spirit of n-MOS design principles, but this suffers from the fact that the p-channel pull-up transistor slows the gate response and consumes steady state power. Thus these authors proposed the Domino CMOS circuit of Fig. 2.3 which requires the use of only one p-channel transistor and a simple two phase clocking scheme. When the clock signal is low, the p-transistor is turned on, the path to ground through the n-channel block is opened, and the output node capacitance ND1 is charged to Vdd during what is known as the precharge phase of operation. When the clock becomes high, the logic function represented by the n-channel block is evaluated
Fig. 2.3 CMOS dynamic logic gate
and the capacitor of the node ND1 is conditionally discharged to ground. The output of the gate is connected to a static CMOS buffer which is used to drive other gates in the circuit. NOR logic function can also be realized using the Domino CMOS logic. The first use of Domino logic was in a 32-bit microprocessor where performance speed tradeoff mitigated against the choice of enhancement/depletion NMOS or static CMOS and where area limitations made standard CMOS unfeasible [14].

2.5 Switching specifications of CMOS circuits

**Propagation delay**:

Propagation delay (tPLH) is measured from the 50 percent point of the input pulse to 50 percent of the point of the output pulse as the output goes from a low level to a high level as shown in Fig. 2.4(a).

Propagation delay (tPHL) is measured from the 50 percent of the input to 50 percent of the output as the output goes from a high level to a low level as shown in the Fig. 2.4(a).

**Transition time (tTLH)** is the time required for the output to make the transition from the low state to the high state (n-channel device turns off, p-channel device turns on). This time is measured from the 10 percent point to the 90 percent point of the output pulse as shown in the Fig. 2.4(a).

**Transition time (tTHL)** is the time required for the output to make the transition from the high state to the low state (p-channel device turns off, n-channel device on). This time measured from the 10 percent point to the 90 percent point of the output pulse as shown the Fig. 2.4(a).
Fig. 2.4 Switching specifications of CMOS IC's
Set-up time:
Set-up time ($t_s$) is the time interval during which a signal is applied and maintained at a specified input terminal before the device recognizes the presence of the specified input pulse as described in Fig. 2.4(b).

Pulse width:
Pulse width, $tw$, is defined as the time from the point on the leading edge of the clock pulse curve which is 50 percent of the maximum amplitude to a point on the trailing edge which is 50 percent of the maximum amplitude as shown in Fig. 2.4(d).

Fall time:
Fall time ($t_f$) is the time required for a signal to fall from 90 percent to 10 percent of the maximum amplitude as shown in Fig. 2.4(a).

Rise time:
Rise time ($t_r$) is the time required for a signal to rise from 10 percent to 90 percent of the maximum value as shown in Fig. 2.4(a).

2.6 Layout techniques for CMOS circuits
Designing a VLSI circuit is a process of mapping a given logical behaviour into a circuit laid out on a silicon chip. The objective during the layout process is to find good and/or optimal shapes and positions of the polygons in an efficient manner, subject to the rules of the technology which are specified as geometric constraints. The following sections discuss the layout of a CMOS Inverter and layout techniques for CMOS gates.
2.6.1 Layout of a CMOS Inverter

We have seen the schematic diagram of a CMOS inverter in the Fig 2.2. A layout of a CMOS Inverter is shown in Fig. 2.5. This layout is a direct translation of the inverter schematic into a mask layout. The gates, input port, output port, Vdd port, and Vss port are shown in the Fig. 2.5. The gates are arranged in the vertical direction with Vdd line at the top and Vss line at the bottom of the layout. Fig. 2.6 shows an alternative layout of the Inverter. In this layout, the gates are arranged in the horizontal direction. These layout styles of the CMOS Inverter are the basic approaches for layout of ROMs and other gates in the following chapters. The Figures 2.5 and 2.6 show how CMOS gates can be arranged both vertically and horizontally based on the designer's needs.

The layouts shown in Figures 2.5 and 2.6, follow the rules given by the CMC [13]. The gates PG and NG are formed where the polysilicon layer passes over the diffusion layer (where a gate oxide insulator is not present between the polysilicon layer and diffusion layer). A common polysilicon layer is used to create both gates. Power is provided to the Inverter using split contacts. A complete layout description of this gate can be found in [13].

2.6.2 Techniques for better layout of CMOS ICs

While laying out CMOS circuits, the following points must be considered for good and efficient layouts (extracted from the various literatures such as text-books, technical articles, etc):

1) Metal should interconnect as many logic functions as possible
Fig. 2.5 Vertical layout of a CMOS Inverter
Fig. 2.6 Horizontal layout of a CMOS Inverter
because it is more conductive than polysilicon.

2) Polysilicon underpasses should be avoided except when it is necessary to pass a signal under a power supply or another signal line. Metal lines can be routed directly as if there are no polysilicon underpasses; however, when a signal line is blocked by another, the blocking path can be removed by running the blocking signal through an underpass.

3) Diffusion underpasses can be used, although such a practice is not advised due to the larger capacitance to substrate involved in the diffusion regions.

4) Number of pins in a chip should be minimized.

5) Interconnections should be minimized in order to reduce the delay time.

6) In CMOS, all the n-channel currents are summed into the Vss power line and flow toward the Vss pin. All the p-channel currents are summed into the Vdd power line and flow toward the Vdd pin. Knowing the current flow through various supply lines will dictate any need to widen these lines.

7) Sometimes, it becomes physically impossible to increase a power line to the required width. In this case other alternatives are available such as adding more power supply pins to distribute the current flow or lessen the capacitive load requirements of the design.

8) It is recommended that the pin-out of the part be determined by the layout of the chip rather than matching the chips to a pre-determined pin placement.

9) Input/Output protection schemes should be used to protect chips from electrostatic charges.
10) Drivers should be used to drive large capacitive loads at any part of the layout.

11) Gates connected in series, for example the n-channel gates in an AND gate should be laid out with minimum spacing between them in order to reduce the capacitance between them.

12) Gates connected in parallel, for example the p-channel gates in an AND gate should be laid out to occupy a minimum drain area.

13) Body connections should be maximized in order to control the latch-up problems in a CMOS chip. It is a good practice to short the p-channel transistor source finger to the substrate and the n-channel transistor source finger to the body with metalization along their entire length. This measure effectively prevents these diodes from ever becoming forward biased thus eliminating their participation in latch-up.

2.7 Summary

The need for VLSI technology in a system design is discussed. A VLSI design flow is given. The advantages of CMOS technology are discussed. Operation and the transfer characteristics of a CMOS Inverter are discussed. Dynamic CMOS logic is reviewed. Switching specifications of ICs are given. The layout of a CMOS Inverter is discussed. It is shown that an Inverter can be laid out vertically and horizontally. Techniques for better layout of CMOS circuits are given. Overall, this chapter provides necessary background to understand the discussions in the following chapters.
CHAPTER 3
DESIGN OF CMOS READ ONLY MEMORIES

3.1 Overview
In this chapter, the design of CMOS Read Only Memories (ROMs) is discussed. Based on the available ROMs in the literature, several ROMs have been developed over the period of this research. This chapter discusses ROMs under two categories: 1) static logic ROM structures and 2) dynamic logic ROM structures. Design of both static and dynamic decoders is also discussed along with the discussion on ROMs. The underlying idea of this chapter is to explain how the final designs are achieved based on previous designs.

3.2 Organization of a Read Only Memory
A Read Only Memory is a digital logic storage element used to store permanent information. ROMs have the unique characteristic that they can be read from, but not written into. Thus, the information stored in the memory is introduced into the memory at the time of fabrication. Basically, a ROM is a device with several input and output lines so that for each input value there is a unique output value. Read Only Memories employing this one-to-one correspondence currently operate as converters, as logarithmic and trigonometric function generators, as character generators, and as storage media for computer microprograms. A
ROM may also be regarded as an implementation of a truth table. Each of the bits in the output word can be represented as independent Boolean functions of the address variables. This immediately suggests that ROMs can be used for arithmetic and control logic functions.

A general organization of a ROM is shown in Fig. 3.1. The major elements in the ROM are: 1) the ROM cell array and 2) the decoder. ROM cell array stores the information content of the ROM. It consists of several words. The small boxes shown in Fig. 3.1 represent individual words. A word consists of several storage bits. The length of a word, i.e., the number of bits, usually remains the same throughout the ROM array irrespective of the contents. The decoder provides the addressing mechanism for the ROM cell array. For each valid combination of input signal, the decoder establishes an address and the information stored at a location corresponding to that address is presented to the output circuits. The other important elements in a ROM are: 1) Word lines, 2) bit lines, and 3) address lines. The word lines connect decoders and the ROM cell array. Only one of the word lines is active whenever there is an address present at the input of the ROM. The other lines remain low. The bit lines connect the ROM cells and the output circuits. These lines carry the information stored in a word whenever the word is selected. Address lines connect input circuits and the decoder. For $K$ number of address lines, the number of word lines will be less than or equal to $2^K$. The number of bit lines will be equal to the word length of the memory. The product of the number of word lines and the number of bit lines is the storage capacity of a
Fig. 3.1  A general organization of a ROM
ROM. In the following sections, we discuss the design and development of ROM cells and organization of ROMs.

3.3 An initial design

In order to establish available techniques for ROM design, a 49*3 ROM was investigated, designed, and fabricated as a part of VLSI implementation of structures proposed by Bayoumi [7]. During the literature survey, it was found that an ROM could be designed using complementary (p- and n-channel) cells [16]. Using this complementary cell, a high speed 18K ROM has been built by the designer [16]. The decoder used in this ROM consists of high speed static AND gates. The advantages of using complementary cells are:

1) increased processing speed,
2) elimination of precharge,
3) increased noise margin, and
4) good radiation hardness.

The complementary cell structure is presented in Fig. 3.2. This cell structure has two types of transistors, namely n- and p-channel transistors. A 'logic 1' is programmed by a p-channel transistor connected to the positive supply Vdd, whereas a 'logic 0' is programmed by an n-channel transistor connected to the ground Vss. The connection of both p- and n-channel devices to a single data line requires two word lines for each individual word. A single word line would result in multiple selection of the cells tied to a data line. The operation of this cell can be explained under the assumption that the word line Ni is 'low' in potential and all other word lines are 'high' in potential. The
Fig. 3.2  Complementary transistor ROM cell
'low' voltage on line \( N_i \) turns on the p-channel transistor connected to the data line \( fn \). The inverted signal of the line \( N_i \) turns on the n-channel transistor connected to data line \( fm \). As a result, the potential on data line \( fn \) rises and the potential on data line \( fm \) falls while all other transistors in the matrix are turned off. If the word address changes from \( N_i \) to \( N_j \) at the next address, data line \( fn \) will switch to 'low', and line \( fm \) will be 'low'.

There are some disadvantages discovered during the investigation of this design:

1) Layout of the complementary cells requires two types of body wells, namely p-well and n-well. Layout of these wells requires larger chip area than that required for a single transistor type cell.

2) In order to support the addressing of complementary cells, the decoder requires larger silicon area than that required for a single transistor type cell.

3.3.1 Design of a 49×3 ROM

In order to remove some of the disadvantages of the complementary cell design regarding area, a modified cell structure shown in Fig. 3.3 has been introduced. The cell structure is designed to reduce the ROM size. In this cell structure, there are two n-channel transistors connected in series. In Fig. 3.3, there are 6 ROM cells. A 'logic 1' is programmed by the connection of the source of the upper n-channel transistor to the positive supply, Vdd, whereas a 'logic 0' is programmed by a connection of the
Fig. 3.3  A representative 2-word, 3-bit ROM using a static CMOS memory cell
source of the upper n-channel transistor to the ground Vss. In Fig. 3.3, the cell which consists of n-channel transistors N1 and N4 is programmed for 'logic 0' whereas the second cell, which consists of n-channel transistors N2 and N5, is programmed for 'logic 1'. Word lines for these two cells, WR1 and WC1, are taken from two separate decoders. The decoders are configured as 3-to-7. The decoder which drives the WR1 line is the row decoder and the decoder which drives the WC1 line is the column decoder. The circuit for the 3-to-7 decoder is shown in Fig. 3.4(a), and consists of 7 3-input AND gates. A schematic diagram of a 3-input AND gate is shown in Fig. 3.4(b). The operation of this decoder is similar to a 2-to-4 AND gate decoder discussed in [11]. Operation of this ROM can be explained under an assumption that the ROM is addressed to access the content of the word which consists of transistors N2, N5, N3, N6, N1, and N4. In this case the row word line, WR1, and the column word line, WC1, go high in their potential level and turn on the transistors N1, N4, N2, N5, N3, and N6. Since the drains of transistors N1, N2, and N3 are connected to Vss, Vdd, and Vss respectively, the output of the nodes D0, D1, and D2 will be 'logic 0', 'logic 1', and 'logic 0' respectively.

The layout of the 49×3 ROM in CMOSJ single metal technology is shown in Fig. 3.5. The layout was carried out completely by hand using graph sheets, and hand coded into CIF (Caltech Intermediate Form) code. This ROM consists of 49 locations and each location (word) consists of 3 storage cells. The output nodes of all the 49 words are connected to 3 bit data lines which feed three output buffers. The layout of the basic components of
Fig. 3.4(b) Schematic of a static CMOS 3-input AND gate

Fig. 3.4(a) A 3-to-7 decoder using AND3 gates
Fig. 3.5 Layout of the 49x3 static ROM in CMOS3 process
this ROM is shown in Appendix 1.

3.4 Design of a 24×5 Static ROM

An improved version of the cell structure shown in Fig. 3.3 is shown in Fig. 3.6, which represents a 3-word memory with each word having 3 bits. In this cell structure only one row of column decoding is used, unlike the previous memory in which a column decoding transistor is used in every cell. This arrangement helped reduce the ROM transistor count. Programming the ROM is carried out by connecting the source of the n-channel transistor to Vdd or Vss. In Fig. 3.6, the n-channel transistors, N1, N2, N3 are programmed for 'logic 1', 'logic 0', and 'logic 1' respectively. It is to be noted that both n-channel memory cells, N2 and N5, share a common column decoding transmission gate. CTG1, CTG2, and CTG3 are the column decoding transmission gates. This cell structure produces a 'logic 1' with a value of (Vdd-Vtn) due to the source follower mode operation of the memory cell n-channel transistor.

Operation of this 24×5 static ROM can be explained under the assumption that the ROM is addressed to access the contents of word1 which consists of transistors N1, N2, and N3. In this case the row word line, WR1, from the row decoder and the column word lines, WC1 and WC1', from the column decoder go high in their potential and turn on the transistors N1, N2, N3, P1, P2, P3, N10, N11, and N12. Since the drains of the transistors N1, N2, and N3 are connected to Vdd, Vss, and Vdd respectively, the output values of the nodes D0, D1, and D2 are 'logic 1', 'logic 0', and 'logic 1'.
Fig. 3.6 A representative 3-word, 3-bit ROM using an improved static CMOS cell
Based on above cell structure, a 24*5 memory has been laid out. The row decoder uses 2-input AND gates which activate the single transistor per bit memory cells. The column decoder uses 3-input NAND-Invert gates to activate the column selector set, which has 5 CMOS transmission gates. Fig. 3.7(a) shows the 3-input NAND-Inverter gate and Fig. 3.7(b) shows the 3-to-6 decoder. A schematic diagram which explains the organization of this memory is shown in Fig. 3.8. W1, W2, ..., W24 are the words in this memory. TG1, TG2, ..., TG6 are the column selector sets with each set having 5 p-channel transistors and 5 n-channel transistors. The row decoder output lines WR1', WR2', WR3', WR4' drive the p-channel transistors and the column decoder output lines WC1', WC2', ..., WC6' drive the n-channel transistors of the column selector transmission gates. This memory has 5 address lines A0-A4 and 5 data lines D0-D4. The layout of the memory is shown in Fig. 3.9. The layouts of basic cells are shown in Appendix 1. This design has been fabricated and tested [Appendix 9].

3.4.1 An alternative layout of the 24*5 static ROM

An alternative 24*5 ROM has been laid out based on the above discussed cell structure as shown in Fig. 3.10. This memory is configured as 6 rows by 4 columns. 3-input NOR gates are used in the row decoder in order to reduce the layout area. In the column decoder, 2-input AND-Invert gates are used to activate the CMOS transmission gates in the selector associated with the column decoder. As we can observe, the overall area of this 24*5 ROM layout is smaller than the previous design (assuming the same
Fig. 3.7(a) Schematic diagram of a 3-input NAND-Invert gate

Fig. 3.7(b) Schematic diagram of a 3-to-6 decoder using 3-input NAND-Invert gates
Fig. 3.8 Layout organization of the 24*5 static CMOS ROM
Fig. 3.9 Layout of the 24*5 static CMOS ROM in CMOS3 process
Fig. 3.10  Improved layout of the 24×5 static CMOS ROM
scale on the diagram. The layouts of the basic cells in this ROM are shown in Appendix 1.

Based on the above discussed memory cell, a 16*2 ROM and a 16*3 ROM were designed for a systolic structure proposed in [8]. In these ROMs NOR2 gates are used in the row decoders and NAND2-Invert gates are used in the column decoders.

3.5 Design of Dynamic CMOS ROMs

At this stage of the research, it was recognized that dynamic CMOS techniques could be used to realize simple ROMs with good noise margins. Initial investigations were carried out, and a 3-to-7 decoder using Domino CMOS AND3 gates was designed and fabricated. This chip performed the assigned job well during the test [Appendix 9]. The circuit schematic of a Domino AND3 gate is shown in Fig. 3.11(a). This gate consists of 7 transistors as compared to a static AND3 gate which contains 8 transistors. A schematic diagram of a 3-to-7 decoder based on the Domino 3-input gate is shown in Fig. 3.11(b). A0, A1, and A2 are the three address lines to the decoder and D1, D2, ..., D7 are the output lines of the decoder. Layout of this gate is shown in Appendix 1. The layout is simpler than the static AND3 gate layout, which implies that the total circuit capacitance of the Domino AND3 gate is lower than the total circuit capacitance of the static AND3 gate. The layout of the Domino CMOS decoder is shown in Fig. 3.12. A layout of the Domino CMOS 3-input AND gate is shown in Appendix 1.

A literature survey was carried out on the design of dynamic ROMs. An early work on the design of dynamic ROMs is presented
Fig. 3.11(a) Schematic of a 3-input Domino CMOS AND gate

Fig. 3.11(b) Schematic of a 3-to-7 decoder using Domino CMOS gates
Fig. 3.12 Layout of the 3-to-7 decoder using Domino CMOS gates
Stewart [17]. In this work, an 8K memory structure is discussed. This memory operates on an externally generated clock and uses special circuits to generate and control the duration of the internal precharge cycle. These circuits include several different types of gates, a power-on reset generator, a transition detector, etc. The area of the memory is 170*206 sq.mils and the read access time is 700 nanoseconds.

A dynamic 1-Mbit ROM is presented by Davis [18]. This design, like the previous design, uses a precharge/discharge scheme for its operation, and two types of n-channel memory cells, namely, n-channel transistors with threshold voltage equal to 0.5 volts and n-channel transistors with threshold voltage greater than 5.0 volts. Column decoding is carried out by p-channel gates. This memory is laid out in an n-well CMOS, double metal technology. This memory achieves an access time of 70 nanoseconds.

A 256K CMOS ROM is presented by Kamuro et al. [37]. This ROM uses a serial-parallel cell structure. This ROM array consists of \(n \times m\) MOSFETs transistors containing storage data and \(m\) parallel enhancement mode MOSFET's to select one chain of \(n\) serial MOSFETs. ROM data are stored in the form of E-MOSFET (enhancement mode MOSFET) and D-MOSFET (depletion mode MOSFET). This memory uses a precharge/discharge scheme for its operation. The access time of this ROM is 370 nanoseconds.

A complementary cell dynamic ROM technique is given by Takahashi [11]. The cell structure has a p-MOS memory cell and an n-MOS memory cell. The p-MOS memory cell is used to program a
'logic 1' and the n-MOS memory cell is used to program a 'logic 0'. The column decoding is performed by CMOS transmission gates. This structure is included in a gate array. The internal access time of this 6K memory is 36NS.

A CMOS ROM technique is presented by Mukherjee [38]. This ROM uses an external 2-phase clock for its operation and p-channel transistors as basic memory cells. The load transistors are n-channel transistors which discharge the output bit lines during the first phase of the clock. Programming is carried out by placement of the p-channel transistors at appropriate locations. If a p-channel transistor is connected between Vdd and the bit line, then that particular bit will represent a 'logic 1'. Absence of a p-channel transistor between a bit line and Vdd will keep the bit line discharged representing 'logic 0'. During the second phase of the clock, the word decoder selects a location of the memory corresponding to the input address. The bit lines which are connected to the p-channel transistors will rise to a potential of Vdd; other lines will remain discharged.

A simple dynamic ROM structure is presented by Miyahara et al. as part of a composite gate array [20]. A block diagram of the ROM is shown in Fig. 3.13. The memory cell of the ROM consists of a single n-channel transistor for programming. Data "0" is programmed by the connection of the n-channel memory cell transistor to the bit line and data "1" is programmed by no connection to the bit line or absence of the n-channel memory cell. In Fig. 3.13, memory cell N1 is programmed to represent 'logic 0' and memory cell N2 is programmed to represent 'logic 1'. The gates of p-channel transistors, P1 and P2, are
Fig. 3.13 A representative 2-word ROM using pseudo n-MOS logic
permanently connected to ground. The column decoding transistors N4 and N5 are single n-channel transistors, which are connected between the drains of the p-channel transistors and the drains of memory cells. The output is taken between the drain of p-channel transistor and the drain of n-channel column decoding transistors. The access time of this 2K*8-bit ROM, which is fabricated in a 1.6 micron n-well double metal technology, is 30NS.

The ROM techniques proposed by Stewart [17] do not match our requirements since this ROM requires a large amount of associative circuitry, such as power on reset generator, transition detector, etc. This memory is much more suitable for single design per chip-type implementations. The techniques given by Davis [18] for the 1Mbit ROM are not compatible with the available technology. The reason for this incompatibility is that there are only single value threshold voltages available (about 0.7 volts) for both n- and p-channel transistors in the available fabrication processes. The other disadvantage comes from the use of p-channel transistors as pass gates. P-channel gates could not pass a 'logic 0' perfectly, since a value of Vt (threshold voltage of a p-channel transistor) would stay in the output nodes of the pass gates.

The ROM technique given by Kamuro et al.[37] requires both enhancement mode and depletion mode n-channel transistors. Using the technology available to the University it is possible to create only n-channel enhancement mode MOSFETs. Hence, this ROM technique is not useful for our requirements.
Use of the complementary cells given by Takahashi et al. [19] would result in a large memory, which does not seem to support our objective. Moreover, the use of p-channel transistor memory cells would result in a slower access time. Another disadvantage of this memory cell is the requirement of a CMOS transmission gate for column decoding which occupies a larger layout area.

The ROM technique presented by Mukherjee [38] uses p-channel transistors. The p-channel transistors are slower and larger than the n-channel transistors. A memory which employs the p-channel transistors will be larger than a memory which employs n-channel transistors.

The memory cell structure given by Miyahara et al. [20] is very simple in concept. The only disadvantage of this memory cell structure is the permanently grounded p-channel transistors that lead to an increase in the power dissipation of the ROM. We can reduce the power dissipation by connecting the p-channel transistor with an external clock and operating the memory like a precharge/discharge circuit. Based on this idea, a cell structure for a dynamic ROM has been developed. In the following sections we discuss the developed cell structure.

3.5 Design of a 24×5 dynamic ROM

Fig. 3.14 shows a memory schematic based on the developed cell. In this ROM there are two words. Each word consists of 3-bits. The cell uses precharge/discharge logic for operation. A set of p-channel transistors P1, P2, and P3 are used to precharge the bit lines; the n-channel transistors N1, N2, N3 are used for
Fig. 3.14 A representative 2-word, 3-bit dynamic ROM using a developed CMOS memory cell
column decoding. The n-channel transistors N4, N5, N6, and N7 are programmed to represent 'logic 0'. The p-channel transistors P4, P5, and P6 are used to precharge the internal node ND2 between N1 and N6 to eliminate the problem of charge sharing [11] between the output node ND1 and the internal node ND2.

During a read operation, the word decoder is first disabled and the bit lines are precharged to Vdd by the p-channel transistors P1, P2, and P3. Simultaneously, the nodes ND2 are also precharged by the p-channel transistors P4, P5, and P6. Then the word decoder (row decoder) is enabled and the word line's potential is raised to Vdd. If a memory cell n-channel transistor is present at the location determined by the word line, and if it is connected between ground and the node ND2, then the node ND2 is discharged. If the column decoding transistor is turned on, the precharged node ND1 is also discharged and memory output is 'logic 0'. On the other hand, if N6 is not present at the location selected by the row decoder, the node ND1 remains precharged and the output is 'logic 1'. Thus, it is possible to generate 5-volt 'logic 1' and 0-volt 'logic 0' with this cell structure.

The operation of this ROM can be more precisely understood under the assumption that the ROM is addressed to access the contents of word1 which consists of n-channel transistors N4 and N5, and all the bit lines and internal lines have been precharged by the p-channel transistors P1, P2, P3, P4, P5, and P6. In this case, the row word line WR1 and the column row word line WC1 go high in their potential and turn on the transistors N1, N2, N3, N4, and N5. Since the sources of the transistors N4 and N5 are
connected to Vss, these transistors discharge the nodes, ND2s, of their respective internal bit lines. Since the transistors N1, N2, and N3 are already turned on, the bit lines D1 and D2 will be discharged and the output values of the lines D0, D1, and D3 will be 'logic 1', 'logic 0', and 'logic 0'.

A representative circuitry required to retrieve a bit of information from a memory cell programmed to a 'logic 0' is shown in Fig. 3.15(a). The NOR2 gate represents the column decoder, the NOR3 gate represents the row decoder, the p-channel transistor P1 represents the precharge transistors, the n-channel transistor N2 represents the column decoder-selector transistors, the n-channel transistor N6 represents the ROM cells, and the p-channel transistor P2 represents the internal precharge transistors of the memory array. Fig. 3.15(b) shows the simulation results of this circuitry. In this Figure, the curve V(2) shows the potential of address lines, curve V(26) shows the potential of the output node of the memory, the curve V(27) shows the potential of the clock lines, and the curve V(28) shows the output node voltage of an Inverter (which is not shown) connected to node 26. We can observe that the output node precharges to 5-volts and discharges to 0 volts.

Based on the above discussed memory cell structure, a 24*5 memory has been laid out. A schematic diagram, which explains the organization of the memory is shown in Fig. 3.16. This memory is configured as 6 rows and 4 columns. W1, W2, ..., W24 are the 24 5-bit words of the memory. PB7, PB8, ..., PB12 are the bit line precharge transistor sets. Each of these sets consists of 5 p-
Fig. 3.15(a) A representative circuit for simulation of the 24*5 dynamic ROM.
Fig. 3.15(b) Simulation results of the representative circuitry of the 24*5 ROM
Fig. 3.16  Layout organization of the 24*5 dynamic ROM
channel transistors. PB1, PB2, ..., PB6 are the internal precharge transistor sets. Each of these sets consists of 5 p-channel transistors. CT1, CT2, ..., CT6 are the column selector n-channel transistor sets. Each of these sets consists of 5 n-channel transistors. This memory has 5 address lines A0 - A4, and 5 output lines D0 - D4. The layout of this memory is shown in Fig. 3.17. The ROW decoder uses a different 3-input NOR gate shown in Fig. 3.18(a). A schematic diagram of the 3-to-6 decoder based on the NOR gate is shown in Fig. 3.18(b). The 2-to-4 column decoder uses a 2-input version of these gates. The layout of this memory is regular and compact, and transistors are placed with minimum spacing. In this memory cell array, there are 20 bit line precharge transistors, and 20 internal node precharge transistors. Word and column lines are laid out in metal in order to reduce the access time to all locations in the memory. This memory has been fabricated and tested [Appendix 9].

3.6 Comparative study

In this section, we compare the characteristics of the ROMs discussed above. The following table provides these comparisons.

<table>
<thead>
<tr>
<th>Type</th>
<th>Configuration rows</th>
<th>Process</th>
<th>size sq.mm</th>
<th>logic levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>24*5 static</td>
<td>4</td>
<td>CMOS3</td>
<td>0.515</td>
<td>Vdd-Vtn VSS</td>
</tr>
<tr>
<td>24*5 static</td>
<td>6</td>
<td>CMOS3</td>
<td>0.331</td>
<td>Vdd-Vtn VSS</td>
</tr>
<tr>
<td>24*5 dynamic</td>
<td>6</td>
<td>CMOS3</td>
<td>0.400</td>
<td>Vdd VSS</td>
</tr>
</tbody>
</table>
Fig. 3.17 Layout of the 24x5 dynamic ROM in CMOS 3-micron process
Fig. 3.18(a) Schematic of a 3-input NOR gate

Fig. 3.18(b) Schematic of a 3-to-6 decoder using NOR gates
From Table 3.1, we can observe the following points:

1) Static ROMs suffer from reduced high level noise margins.

2) Although the dynamic memory seems to be slightly larger than a static memory, the number of transistors in the dynamic memory may vary between 0 or 120; whereas in the static memory, the number of transistors is fixed. The dynamic memory is slightly larger than the static ROM due to the use of metal word lines.

The other advantages of dynamic ROMs over static ROMs include the simplicity involved in programming and the requirement for only one power supply rail (Vss) for programming connections. The output voltage level of 'logic 1' of static memories can be improved from (Vdd-Vtn) to Vdd by the use of a bootstrap technique but relatively large capacitors are required which increase the area requirements.

3.7 Design of 32*7 Read Only Memories

Taheri, Jullien, and Miller proposed a fault tolerant systolic cell structure for RNS applications [23]. The cell structure is shown in Fig. 3.19. This cell requires a 32*7 ROM for the purpose of achieving fault tolerance. The word length of this memory is 7 bits; two of these bits are reserved for parity codes, and the remaining 5 bits are reserved for storage of look-up table contents. Of the two bits required for parity, one is reserved for address parity and the other is reserved for content parity. The following section describes the design of a 32*7 ROM.
Fig. 3.19 Schematic of a fault-tolerant ROM based systolic cell
3.7.1 Design of a 32*7 dynamic ROM

A 32*7 dynamic ROM was laid out in the recently available CMOS 3-micron double metal technology from CMC. In this design, an improved version of the cell structure presented in Fig. 3.14 is used. A schematic diagram of a ROM with this cell structure is shown in Fig. 3.20. This cell structure is an improved version of the cell structure presented in Fig. 3.14. We can observe that the internal precharge generating transistors P4, P5, and P6 shown in Fig. 3.14 are absent in the cell structure shown in Fig. 3.20. The Figure shows a 2-word memory with each word having 3 bits. The programming technique is the same for both memory cell structures. The row decoder uses an external clock controlled dynamic logic CMOS gates. The column decoder uses static logic gates.

Before we discuss the operation of this cell structure, we will make the following assumptions:

1) The input address is already present on the address lines; this implies that one of the gates in the column decoder is turned on.

2) The clock, which precharges the bit lines is low; this implies that the precharge generating p-channel transistor P1 is on and the nodes ND1 and ND2 are precharged to Vdd.

3) The clock, which controls the row decoder, is low; this implies that the row decoder is disabled.

When the precharge clock CL1 and the row decoder clock CL2 go high, the p-channel transistor P1 turns off and the potential of the row line WR1 rises to Vdd. The n-channel transistor N1 turns
Fig. 3.20 A representative 2-word, 3-bit dynamic CMOS ROM using an improved memory technique
on and discharges the node ND2. Assuming that the input signal has turned on N2, then the output node ND1 will be discharged; therefore, the output of the memory will be 'logic 0'. If the transistor N1 is not present in the location selected by the word decoder, the output of the memory will be 'logic 1', a reflection of the charged node, ND2, through N2. Since the node ND2 is also precharged during the regular precharge cycle, the problem of charge sharing between the nodes ND1 and ND2 is prevented; thus the static AND gates help prevent the charge sharing problem. The use of Domino AND3 gates in the row decoder is justified by not allowing the word decoder to turn on during the pre-charge period of the ROM.

A schematic diagram which explains the layout organization of this memory is shown in Fig. 3.21. This memory is configured as 8 rows by 4 columns. There are 32 words and each word has 7 bits. COL-TRAN1, COL-TRAN2, COL-TRAN3, and COL-TRAN4 are the column decoding circuit sets, each containing 7 n-channel transistors. A0-A4 are the five input address lines to the memory and D0-D6 are the seven data lines of the memory. The seven bit-line precharging p-channel transistors are shown at the right side top of the Figure. The word and bit lines are metal thus providing reduced access time for all locations.

The word decoder is configured as a 3-to-8 decoder and is realized by 8 3-input Domino AND gates. The column decoder is configured as a 2-to-4 decoder and is realized by 2-input AND gates. The circuit schematic of the Domino AND3 gate decoder is shown in Fig. 3.22. The layout of this memory is shown in Fig.
Fig. 3.21 Layout organization of the 32×7 dynamic CMOS ROM
3-Input Domino CMOS AND gate

Fig. 3.22 Schematic of a 3-to-8 dynamic CMOS decoder using AND gates
Fig. 3.23 Layout of the 32*7 dynamic CMOS ROM in CMOS 3-micron double level metal technology
Fig. 3.24(a) Simulation circuit of the 32*7 dynamic ROM to test the output logic levels
3.23. The layouts of the basic elements in this ROM are shown in Appendix 1. Drive capability is given to all decoder gates. Fig. 3.24(a) shows how a representative circuitry of the ROM retrieves a bit of information from a cell programmed to a logic '0'. The AND2 gate represents the column decoder, the DAND3 (Domino AND3) gate represents the row decoder, the n-channel transistor N1 represents the memory cells, and the n-channel transistor N2 represents the column decoder-selector transistors. Fig. 3.24(b) shows the simulation results of the circuitry. In this Figure, the curve V(2) shows the potential of the address lines, the curve V(27) shows the potential of the clock line, and the curve V(26) shows the potential of the output node of the memory. We can observe that the output node precharges to Vdd and discharges to Vss.

The operation of this entire ROM can be explained based on Fig. 3.21. When the precharge clock CL1 is high and the row decoder clock CL2 is low, the bit lines BL1-BL7 are precharged to Vdd and the word decoder (row decoder) is disabled. When both the clocks change their states, the process of precharge generation is stopped and the word decoder is enabled. The enabled word decoder allows only one of the word lines to go high. That word line selects an entire row of memory cells. The internal lines, which are connected to the drains of the memory cell transistors are discharged. The other internal lines remain precharged. The column decoders either discharge or leave the output bit lines charged corresponding to the state of the lines A3 and A4 by either activating or not activating the column selector transistors which are connected between the output lines
Fig. 3.24(b) Results of the simulation
and the internal lines.

Based on the above dynamic memory, layout of the fault tolerant systolic cell (shown in Fig. 3.19) is designed. This cell has two sets of inputs, A(A0-A4) and B(B0-B4). The B input lines are connected to the ROM address lines. The layout organization of the cell is shown in Fig. 3.25(a). The layout of this systolic cell is shown in Fig. 3.25(b) and consists of a 32*7 ROM, a 2-input Exclusive OR gate, a 2-input OR gate, and 12 latches. The schematic diagrams of the OR2 gate, Ex-OR2 gate, and latch are shown in Figures 3.26(a), 3.26(b), and 3.26(c) respectively.

3.8 Design of a 5-bit RNS adder

Based on the systolic cell structure shown in Fig.3.24, a 5-bit pipelined adder is designed [24]. This adder has 5 stages, each stage consists of a fault-tolerant systolic cell. A new clocking scheme is devised for this pipelined adder using two-phase non-overlapping clocks. Fig.3.27(a) shows the two phase non-overlapping clocks with finite rise and fall times. These clocks are applied to the systolic cell as shown in Fig.3.27(b). The timing diagram of the systolic cell, with respect to the clocks CL1 and CL2, is shown in Fig.3.28. During the active region of Clock CL1 (state A) the second portion of latch, LB, is activated and the bit lines of the ROM are precharged to Vdd. The inactive state of clock CL2 (state C) keeps the word decoder and the first portion of latch LA disabled. When the clock CL2 becomes ‘high’ (state D), the word decoder and the latch section LA are enabled. The word decoder selects a location and the output of the ROM is
Fig. 3.25(a) Organization of the layout of the 32×7 dynamic ROM based fault-tolerant systolic cell
Fig. 3.25(b) Layout of the 32×7 dynamic ROM based fault-tolerant systolic cell
Fig. 3.26(a) Schematic diagram of a 2-input OR gate

Fig. 3.26(b) Schematic diagram of a 2-input exclusive OR gate
Fig. 3.26(c) Schematic diagram of a 2-phase latch
Fig. 3.27(b) Two-phase non-overlapping clocks and the Fault-tolerant systolic cell

Fig. 3.27(a) Two-phase non-overlapping clocks
Fig. 3.28 Waveforms of the fault-tolerant ROM based systolic cell.
stored in the input nodes of latch LB. During the ROM operation, the low clock phase of CL1 (state B) keeps the precharge transistors and the latch section LB disabled. Thus, the two phase non-overlapping clocks are coalesced with the precharge clock and row decoder clock of the ROM.

The basic principles behind the bit level addition of 2 numbers in the RNS medium are simple in concept [24]. Assume that the addition of two 5-bit numbers, A and B, is required for a modulus M. The addition operation \((A+B) \mod M\) can be expressed in the bit level as follows [24]:

\[(A+B) \mod M = 2^4A_4+2^3A_3+2^2A_2+2^1A_1+2^0A_0+B) \mod M\]

The above operation can be performed in 5-stages as follows:

\[\text{SUM1=} \quad (2^0A+B) \mod M\]

\[\text{SUM2=} \quad (2^1A_1+\text{SUM1}) \mod M\]

\[\text{SUM3=} \quad (2^2A_2+\text{SUM2}) \mod M\]

\[\text{SUM4=} \quad (2^3A_3+\text{SUM3}) \mod M\]

\[\text{SUM5=} \quad (2^4A_4+\text{SUM4}) \mod M\]

\[(A+B) \mod M = \text{SUM5}\]

A schematic diagram which explains the addition scheme is shown in Fig. 3.29.

A complete schematic diagram of the 5-bit adder using the two non-overlapping clocks, CL1 and CL2, is shown in Fig. 3.30. A final layout of the 5-bit RNS fault tolerant pipelined adder is shown in Fig. 3.31. The chip is complete and has been checked for
Fig. 3.29 5-bit RNS addition process
Fig. 3.30 Schematic of a 5-bit RNS adder using the fault-tolerant ROM based systolic cell
Fig. 3.31 A complete layout of the 5-bit RNS adder
design rule errors. We can see that there are 5 systolic cell blocks in the layout. The clocks and the input lines are fed in through protected input pins and input buffers which protect the circuitry inside from electrostatic discharge. In the layouts of the systolic cell and adder, an attempt has been made to make interconnections with second level metal wherever possible and minimize the use of polysilicon lines in order to reduce the delay and increase the throughput rate.

3.9 Improved layout of the 32*7 dynamic CMOS ROM

Since the predominant feature of the systolic cell is the ROM, we will concentrate on improving its construction. A first step is to improve the area of the ROM. By the use of modified layout techniques the area of the ROM has been reduced. The improved layout of the ROM is shown in Fig. 3.32. A schematic diagram which explains the organization of the layout is shown in Fig. 3.33. The reduction in area occurs mainly due to an improved layout of the basic memory cell structure shown in Fig. 3.34. The area of the ROM is reduced from 0.316 sq.mm to 0.200 sq.mm. The advantages gained by the reduction are:

1) Reduced capacitance on the word lines,

2) Reduced capacitance on the bit lines,

3) Reduced drain and source regions of the memory cells and corresponding reduction of parasitic capacitance, and

4) Reduced silicon area required.

An accurate simulation of this improved ROM using SPICE and RELAX simulation programs is discussed in the following chapter.
Fig. 3.32 Improved layout of the 32x7 dynamic ROM
Fig. 3.33 Organization of the improved layout of the 32x7 dynamic ROM
Fig. 3.34 Layout of a word of the 32×7 dynamic ROM
3.10 Design of a 32×7 Static ROM

For the purpose of obtaining direct comparisons between dynamic and static ROMs, a static ROM cell structure has been developed. Fig. 3.35 shows the schematic diagram of a 2-word memory structure based on the newly developed complementary cell structure. The well known two dimensional decoder which consists of a row decoder and a column decoder is used for decoding. The transistors P1, P2, P3, and P4 are the p-type memory cells and the transistors N1 and N2 are the n-channel memory cells. The sources of the p-channel memory cells are connected to Vdd and the drains of the p-channel memory cells are connected to the column decoding transistors. The sources of the n-channel memory cells are connected to Vss and the drains of the n-channel memory cells are connected to the column decoding n-channel memory cells. P5, P6, and P7 are the column decoding p-channel transistors and N3, N4, and N5 are the column decoding n-channel transistors. The difference between this memory and the complementary cell memory structures discussed earlier in section 3.2 is the use of the two dimensional decoding scheme which is feasible due to the addition of column decoding transistors such as P5 and N3. This two dimensional decoding scheme reduces the decoder area significantly. For example, for a 32×7 memory, the use of complementary cells shown in Fig. 3.2 would require 32 6 input AND gates for row decoder, whereas the complementary cell structure shown in Fig. 3.35 would require only 8 3-input AND gates and 4 2-input AND gates. Hence, an enormous amount of silicon area can be saved using the two dimensional decoding scheme and the developed complementary cell structure.
Fig. 3.35 A representative 2-word, 3-bit static ROM using the developed complementary cell technique.
Programming for each bit is carried out by placement of either a p-channel transistor or an n-channel transistor. If the p-channel transistor is present in a selected location and the corresponding n-channel transistor is absent, then the memory bit is programmed to a 'logic 1'. If the n-channel cell is present in the selected location and the corresponding p-channel transistor is absent, then the memory is programmed to a 'logic 0'. In Fig. 3.35, the word1 which consists of p-channel transistors P1, P2, and n-channel transistor N1 is programmed to '1 0 1'. The word, which consists of p-channel transistors P3, P4, and n-channel transistor N2, is programmed to '1 1 0'.

During a read operation, the word and column decoder select one of the row word lines, Wi, and one of the column word lines, Cj, where i,j are the functions of the input address and raise the potential to Vdd. Simultaneously, the corresponding row complementary word line Wi' and the column word line Cj' are lowered to the potential of Vss. The column word lines select the column decoding transistors and turn them on. If the cell is programmed to a 'logic 1', the 'low' word line Wi selects the memory cell p-channel transistor; and the 'high' word line does not select any transistor. If the cell is programmed to a 'logic 0', the 'high' word line selects the memory cell n-channel transistor; the 'low' word line does not select any transistor. Thus it is possible to retrieve a 'logic 1' and a 'logic 0' from this memory. At this point, we can observe that the cell structure for 'logic 0' and the cell structure for 'logic 1' of this memory are similar to the cell structure of the 32*7 dynamic
ROM discussed earlier.

The operation of the ROM can be more precisely understood under the assumption that the ROM is addressed to access the contents of word1 which consists of the p-channel transistors P1 and P2, and the n-channel transistor N1. In this case, the row word line W1 goes high, the row word line W1' goes low, the column word line C1 goes low, and the column word C1 line goes high. These lines turn on the transistors P1, P2, P5, P6, P7, N1, and N4. Since the sources of the transistors P1, P2, and N1 are connected to Vdd, Vdd, and Vss lines respectively, the value of the output nodes D0, D1, and D2 become 'logic 1', 'logic 0', and 'logic 1' respectively.

Based on the complementary memory cells, a 32*7 ROM has been laid out. Fig. 3.36 shows the schematic block diagram of the layout organization of this memory. We can see that the memory cell array consists of two major blocks: 1) a p-channel cell block and 2) an n-channel cell block. We can also observe that the row decoder is constructed at the left side of the memory arrays, and the column decoder is placed in between the memory cell blocks in order to reduce the word line and bit line capacitance. The row and column decoder use complementary row and column word lines to activate the complementary transistors in the memory. COL-NT1, COL-NT2, COL-NT3, and COL-NT4 are the column selector n-channel transistor sets and COL-PT1, COL-PT2, COL-PT3, and COL-PT4 are the column selector p-channel transistor sets. This memory has 5 input address lines A0 - A4 and 7 output data lines D0 - D6.
A layout of this 32*7 static ROM is shown in Fig. 3.37 where we can see that the layout is regular and uses metal lines for wiring and interconnections. Use of poly silicon is restricted to the creation of gates. We can also observe an important point. The memory arrays use the same layout techniques of the 32*7 dynamic ROM discussed due to the similarities in the cell structures of both ROMs.

The row decoder uses NAND3-Inverter gates for decoding in order to generate complementary word line signals. A schematic which shows the row decoder is shown in Fig. 3.38. The column decoder uses NAND2-Inverter gates in order to generate complementary signals to operate the column decoding complementary transistors. Drive capabilities are added to these gates to drive the capacitive word lines. The area of this memory is 0.371 sq.mm. Fig.3.39(a) shows the representative circuitry to retrieve a bit of information from a cell programmed to a 'logic 1' and a 'logic 0'. The NAND2-Inverter gate represents the column decoder, the NAND3-Inverter gate represents the row decoder, the p-channel transistor P1 represents the p-channel memory cells, and the p-channel transistor P2 represents the column decoder-selector transistors, the n-channel transistor N1 represents the n-channel memory cells, and the n-channel transistor N2 represents the column decoder-selector memory cells. In this circuitry, as previously mentioned, the memory is programmed to a 'logic 0' and to a 'logic 1'; however, in a real case, either P1 will be present or N1 will be present as shown in Fig. 3.35. Fig. 3.39(b) shows the simulation results of this circuitry. V(39) is the output of the memory cell programmed to a
Fig. 3.36 Layout organization of the 32x7 static ROM
Fig. 3.37 Layout of the 32*7 static ROM in CMOS3 process
A CMOS 3-input NAND-Invert gate

Fig. 3.38 A 3-to-8 decoder using 3-input NAND-Invert gates.
Fig. 3.39(a) Simulation circuit of the 32x7 static ROM to test the output logic levels
Fig. 3.39(b) Simulation results of the 32x7 static ROM test circuit
'logic 1' and V(38) is the output of the memory cell programmed to a 'logic 0'. We can observe that the V(39) goes to Vdd (5 volts) and the V(38) curve goes to Vss (0 volt). Accurate simulation of this ROM using SPICE and RELAX simulators is discussed in the next chapter.

For the same storage capacity, the above discussed static ROM cells offer the following advantages over the static ROM cells discussed in Sections 3.3 and 3.4:

1) These cells provide excellent output voltage swings (Vdd and Vss) because all the transistors operate in the common source mode.

2) Layout of the memory cells are simpler and regular due to the use of only one type of programming line (either Vdd or Vss supply rails in an array).

3) Elimination of CMOS transmission gates that require a larger layout area and complicated wire routing.

3.11 Comparative study

This section compares the characteristics of the 32×7 ROMs discussed in this chapter. Table 3.2 provides the comparison based on various parameters. From Table 3.2 we can observe the following points:

1) A significant amount of reduction in the layout area of the dynamic memory has been achieved.

2) The dynamic memory is smaller than the static memory due to the obvious difference in the number of transistors between the dynamic memory cell array and the static memory array.

3) Neither of the memory designs suffer from reduced noise
margin problems.

**Table 3.2**

<table>
<thead>
<tr>
<th>Logic</th>
<th>#.trans per cel</th>
<th>area sq.mm</th>
<th>Process</th>
<th>output voltage logic 1</th>
<th>logic 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic</td>
<td>-1</td>
<td>0.278</td>
<td>CMOS3DLM</td>
<td>Vdd</td>
<td>Vss</td>
</tr>
<tr>
<td>Dynamic (improved)</td>
<td>1</td>
<td>0.200</td>
<td>CMOS3DLM</td>
<td>Vdd</td>
<td>Vss</td>
</tr>
<tr>
<td>Static</td>
<td>1</td>
<td>0.371</td>
<td>CMOS3DLM</td>
<td>Vdd</td>
<td>Vss</td>
</tr>
</tbody>
</table>

Comparison of speed between these two memories is discussed in the following chapter. Before we discuss the simulation of ROMs, we will discuss how the 32*7 dynamic ROM and the 32*7 static ROM compare with the ROMs reported in terms of circuit techniques and layout techniques.

Very few articles have been published on CMOS ROMs over the past few years. Most of them give importance to only circuit techniques. Layout techniques have been discussed in a very few articles and without unveiling much information which is attributed to the fact that this area is completely industry oriented. Detailed descriptions of the designs are missing in the technical articles on ROMs which make it hard to compare and select the best circuit and layout techniques, therefore it was necessary to develop good circuit and layout techniques for the designs discussed in this thesis.
We have discussed both static and dynamic logic CMOS memories. A static memory (available in the literature) is discussed in Section 3.3. As we have already discussed, this memory requires one decoder gate to address one memory cell location. The technique used in the design of 32*7 ROM in Chapter 3 cuts down the requirement significantly due to the use of a two-dimensional decoding structure which reduces the decoder gate count. Hence the significant difference between the reported memory technique and the technique presented in the thesis for a static ROM is the two dimensional decoding scheme. The layout techniques for the static ROM were developed in this thesis.

There are a quite few articles available (in the literature) for the design of dynamic ROMs as we have discussed in Section 3.5. These articles, reported by industry researchers, focus on the circuit techniques. The designs reported in these articles were developed in order to meet the business market requirements. The dynamic ROM developed in this thesis follow the techniques involved in these designs; however, there are quite a few significant difference between the designs and the 32*7 dynamic ROM.

The 32*7 dynamic ROM uses gated row decoders using Domino CMOS AND gates (a technique that has not been used in the reported designs). The column decoder uses static AND gates which prevent the problem of charge re-distribution in the ROM with the help of Domino CMOS gate row decoders as we have discussed in Section 3.7. This technique was developed for this thesis. The Domino CMOS gate decoders help allow the use of two phase non-overlapping clocks which are very useful in the operation of
latches in conjunction with the precharge/discharge operations of the dynamic ROMs. Also, layout techniques for the design of 32\*7 ROMs have been discussed in detail. The major difference, between the techniques reported in the literature and the techniques developed in this thesis, is that the techniques presented in the literature were developed to design one ROM per chip as supportive components for standard digital hardware with pre-determined specifications, and the techniques presented for the 32\*7 dynamic ROM were developed to build multi-ROM based pipelined arrays where the parameters size, speed, and power consumption are critical.

3.12 Summary
The basic organization of a Read Only Memory has been discussed. A 49\*3 static ROM is designed using two series connected n-channel transistor cells. The cell structure is improved and a 24\*5 ROM has been designed based on the improved cell structure. This cell structure still suffers from noise margin problems. In order to design a simple, perfect ROM, the design of dynamic memories is considered. Based on a literature survey, a cell structure for a dynamic ROM has been developed. Based on this cell structure, a 24\*5 ROM was laid out. This cell structure offers full output drive voltages. The cell structure of this ROM is improved through use of decoders and a 32\*7 ROM based on this dynamic cell structure is designed. Using this ROM, a 5-bit pipelined aider is designed. The area of the 32\*7 dynamic ROM is further reduced. For comparison purposes a static memory is laid out in CMOS3DLM technology using a developed complementary
structure in this thesis. This memory does not suffer from noise margin problems and uses compact layout techniques. It can be concluded that the final dynamic ROM and the final static ROM are the best in their categories among the memory structures presented in this thesis. Accurate simulation and comparison of these memories are discussed in the following chapter.
CHAPTER 4
SIMULATION

4.1 Overview
Simulation plays an important role in the design of high speed VLSI systems assisting a designer in circuit and layout verification, and selection of efficient circuit and layout techniques. Simulation tells how a circuit behaves for certain states of inputs. Based on the behaviour of the circuit, the designer can modify the circuit. These simulations are usually carried out under identical conditions of the real world.

Usually there are two types of simulation:
1) Switch level simulation and
2) Layout level simulation.

In the following sections, we will discuss both level of simulation techniques for the ROMs discussed in the previous chapter.

4.2 Switch level simulation
In switch level simulation, the transistors are used as switches. The simulation results will be logical '0' or logical '1'. Electrical characteristics of process layers, such as capacitance, resistance, and inductance effects are not usually considered. These types of simulations are usually carried out to verify the logic of a circuit. Circuits checked by this type simulation can be readily laid out.
Fig. 4.1 LOGCAP simulation of a 24×5 ROM
A CMOS ROM shown in Fig. 3.8 was verified using the LOGCAP switch level simulator available on the VAX 11/750 at the University of Windsor CAD/CAM centre; LOGCAP is a module in Phoenix Data System's VLSI design software package ICAP/MASKAP. As a first step, the circuit schematic of the ROM was entered into the system using the component library. This schematic represents the memory cell array for the 24*5 ROM which was later sent for fabrication. For a set of input pattern, the contents of all the 24*5 ROM locations (a Mod31 multiplication look-up table) were retrieved. The entered schematic is shown in Fig. 4.1. In this Figure, the input waveforms for all the address lines and the output waveforms of the output lines are displayed.

4.3 Layout level simulation of ROMs

SPICE is the most widely used program for layout level simulation. In SPICE, elements such as AND, OR, NAND gates are entered at the transistor level. The program models these components using linear and non-linear differential equations and solves them using matrix operations. Resistors and capacitors can also be included in the simulation [27].

4.3.1 SPICE simulation of the 32*7 dynamic ROM

In this section the simulation of the ROM, discussed in section 3.9, is presented. Circuit level SPICE simulation is presented in Section 3.7.1. This simulation was carried out to check the output logic levels of the ROM. In order to know the exact delay time of the ROM a layout level simulation is carried out. This layout level simulation is an extension of the circuit level simulation with extracted resistances and capacitances from the
32x7 ROM layout shown in Fig. 3.32.

The extraction of resistances and capacitors are explained in detail in [11]. The resistance of a uniform slab of conducting material may be expressed as

$$R = R_s (l/w) \text{ ohms}$$

where $R_s$ is the sheet resistance having units of ohm/square, $l$ is the length of the slab and $w$ is the width of the slab. Thus to obtain the resistance of a layer one would simply multiply the sheet resistance $R_s$ by the ratio of the length to width of the conductor. The resistance of various non-rectangular shapes can be estimated as shown in Fig. 4.2. For right angle bends in which one of the legs is wider than the other, an approximation for the resistance of the corner rectangle can be expressed as follows [25]:

$$R_{\text{corner}} = (0.46 + 0.1C) R_s$$

where $C$ is the ratio of wide to narrow widths and $R_s$ is the sheet resistance of the layer. The following table provides the resistance values associated with the CMOS3 DLM process [26]:

<table>
<thead>
<tr>
<th>Layer</th>
<th>Resistances (ohms/sq.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N+ Diffusion</td>
<td>25</td>
</tr>
<tr>
<td>P+ Diffusion</td>
<td>80</td>
</tr>
<tr>
<td>Poly</td>
<td>30</td>
</tr>
<tr>
<td>Capacitor P+</td>
<td>300</td>
</tr>
<tr>
<td>p-well</td>
<td>4k</td>
</tr>
<tr>
<td>Metal1</td>
<td>0.035</td>
</tr>
<tr>
<td>Metal2</td>
<td>0.030</td>
</tr>
</tbody>
</table>

**TABLE 4.1**

*Sheet resistance of CMOS3DLM layers*
Fig. 4.2 Estimation of resistance in a layout
The capacitance of a wire, which has metals, polysilicon, diffusion, interconnects, and feeds many MOS gates can be expressed as follows:

\[ C = (L_{\text{poly}} \times W_{\text{poly}}) \times C_{\text{paf}} + (2 \times L_{\text{poly}}) \times C_{\text{pe}} + (L_{\text{metal}} \times W_{\text{Metal}}) \times C_{\text{maf}} + 2 \times (L_{\text{metal}}) \times C_{\text{me}} + N \times C_{\text{int}} + (L_{\text{diff}} \times W_{\text{diff}}) \times C_{\text{daf}} + 2 \times (L_{\text{diff}}) \times C_{\text{de}} + M \times (W_{n} \times L_{n} + W_{p} \times L_{p}) \times C_{g} \]

where

- \( L_{\text{poly}} \) - total length of polysilicon wires
- \( W_{\text{poly}} \) - width of poly silicon wires
- \( L_{\text{metal}} \) - total length of metal wires (for both metals)
- \( W_{\text{Metal}} \) - width of metal wires (for both metals)
- \( L_{\text{diff}} \) - length of diffusion wires
- \( W_{\text{diff}} \) - width of diffusion wires
- \( C_{\text{int}} \) - capacitance of interconnect
- \( N \) - number of interconnects
- \( W_{n} \) - width of n-channel MOS transistor
- \( L_{n} \) - length of n-channel MOS transistor
- \( W_{p} \) - width of p-channel MOS transistor
- \( L_{p} \) - length of p-channel MOS transistor
- \( C_{g} \) - gate capacitance
- \( C_{\text{paf}} \) - area component of polysilicon to field capacitance
- \( C_{\text{pe}} \) - edge component of polysilicon to field capacitance
- \( C_{\text{maf}} \) - area component of metal to field capacitance
- \( C_{\text{me}} \) - edge component of metal to field capacitance
- \( M \) - number of identical gates connected to the wire

The values of capacitances for the CMOS3 DLM process are given in Table 4.2.
Table 4.2  
Capacitance of CMOS DLM layers

<table>
<thead>
<tr>
<th>Type of capacitance</th>
<th>Area Component (pF/um²)</th>
<th>Edge component (pF/um)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gate</td>
<td>6.9E-4</td>
<td>0.5E-4</td>
</tr>
<tr>
<td>Metall1-Field</td>
<td>2.7E-5</td>
<td>0.4E-4</td>
</tr>
<tr>
<td>Metall1-Poly</td>
<td>5.0E-5</td>
<td></td>
</tr>
<tr>
<td>Metall1-Diffusion</td>
<td>5.0E-5</td>
<td></td>
</tr>
<tr>
<td>Poly - Field</td>
<td>6.0E-5</td>
<td>0.2E-4</td>
</tr>
<tr>
<td>Metal2 - Diffusion</td>
<td>1.4E-5</td>
<td>2.0E-5</td>
</tr>
<tr>
<td>Metal2 - Poly</td>
<td>2.0E-5</td>
<td></td>
</tr>
<tr>
<td>Metal2 - Metall</td>
<td>2.5E-5</td>
<td></td>
</tr>
<tr>
<td>Capacitor P+ - Poly</td>
<td>6.9E-4</td>
<td>0.5E-4</td>
</tr>
</tbody>
</table>

A complete ROM circuit model, which includes the capacitances and resistances, is shown in Fig. 4.3. This schematic represents the circuit required to operate the outermost bit of the 32×7 ROM array. The simulation of the outermost bit was selected in order to find out the maximum delay time of the ROM.

The SPICE input deck for the 32×7 ROM is shown in Appendix 2. This deck includes all the resistances and capacitances of the layout in the form of distributed capacitors and resistors. The transistor drain and source areas, and drain and source perimeters are calculated as shown in [28]. Node numbers given in this file correspond with the same node numbers shown in Fig. 4.3. Results of this simulation are shown in Fig. 4.4. Like any other dynamic circuits, the operation of this memory is also
Fig. 4.3 Circuit model for the SPICE simulation of the 32×7 dynamic ROM
based on 2 types of operation: 1) Precharge, during which the clock precharges the capacitors associated with the bit lines, and 2) evaluation of the bit lines, which evaluates the bit lines. Fig. 4.4 shows the simulation results of the 32*7 dynamic CMOS ROM. Curves V(34), V(15), V(25), V(28), and V(27) show the clock, the output of the 3-input Domino CMOS 3-input AND gate in the row decoder, the output of the 2-input AND gate in the column decoder, the value of the internal node between the memory cell and the column selector n-channel transistor, and the output of the ROM respectively. The memory cell in the circuit is programmed to produce a 'logic 0' output value. Curve V(27) shows the precharge and evaluation phases of this dynamic ROM. The time difference between the starting point of the curve V(27) and the rising edge of the clock curve V(34) is the time required to precharge the bit lines of the ROM and the time difference between the rising edge of the clock curve V(34) and the end point of the ROM output curve V(27) is the time required to evaluate the bit lines of the ROM. The time required to precharge the bit lines (the set-up time) is 7.0 NS (nanoseconds). The time required for evaluation of the ROM bitlines is 6.3 NS. Hence, we can conclude that the access time of the memory is 13.3 NS which is the sum of the set-up time and the time required for evaluation. The curve V(28) shows the value of the parasitic capacitor between the column selector n-channel transistor, N2, and the memory cell n-channel transistor, N1, during the operation. The high voltage on this capacitor helps prevent a possible charge-sharing problem between the output node of the
Fig. 4.4 Results of the SPICE simulation of the 32*7 dynamic ROM
ROM and the node between the transistors N1 and N2.

4.3.2 SPICE simulation of the 32*7 Static ROM

The layout model of the 32*7 ROM is shown in Fig. 4.5. This figure represents the circuit required to operate the outermost bit of the 32*7 ROM array. A SPICE deck for the 32*7 static ROM is shown in Appendix 3. Node numbers given in this file are the same node numbers shown in Fig. 4.5. The resistances and capacitances extracted from the layout, shown in Fig. 3.37, are included as distributed resistors and capacitors. The results of the simulation are shown in Fig. 4.6. The curves V(2), V(15), V(27), V(32), and V(40) show the input value, the output value of the row decoder 3-input NAND-Invert gate, the output value of the column decoder 2-input NAND-Invert gate, the output value of the ROM cell programmed to 'logic 1', and the output value of the ROM cell programmed to 'logic 0'. The time difference between the starting point and the end point of the curve V(32) is the time required to generate a 5-volt 'logic 1', and the time difference between the starting point and the end point of the curve V(40) is the time required to generate a 0-volt 'logic 0'. The time required to generate a 'logic 1' is 21.7 NS. The time required to generate a 'logic 0' is 13.4 NS. We observe that the time required to generate a 'logic 1' is larger than that required to generate 'logic 0' and can be accounted as the access time of the ROM. The significant time difference between the curves V(15) and V(27) is mainly due to the fact that the address and word lines of the row decoder are longer than that of the column decoder.
Fig. 4.5 Circuit model for the SPICE simulation of the 32x7 static ROM
Fig. 4.6 Results of the SPICE simulation of the 32x7 static ROM
4.4 RELAX simulation of ROMs

RELAX is an integrated circuit simulation program similar to SPICE. A RELAX simulation of a circuit is claimed to be faster than a SPICE simulation [29]. This is due to the use of a new numerical method called Waveform Relaxation algorithm. The basic idea here is to apply relaxation directly to the system of nonlinear algebraic differential equations corresponding to decoupled dynamical sub-circuits. Each decoupled circuit is then analyzed for the entire simulation time interval by means of standard simulation techniques, i.e. stiffly stable integration iteration method and Newton-Raphson iteration method. The decomposition achieved allows latency to be exploited in the most natural way [29].

4.4.1 RELAX simulation of the 32*7 Dynamic ROM

The RELAX program can use the model shown in Fig. 4.3 for SPICE analysis. The RELAX input deck for the 32*7 dynamic ROM is given in Appendix 4. The node numbers specified in this file are obtained by adding the node numbers in the SPICE deck for 32*7 dynamic ROM to the number 100. Hence the node number 127 in the RELAX input deck represents the node 27 in Fig. 4.3. The output voltage curve of the ROM (V(127)), the clock ((V(134)), the output voltage of the 3-input Domino CMOS AND gate (V(115)), and the output voltage of the 2-input AND gate (V(125)) are shown in Figures 4.7(a), 4.7(b), 4.7(c), and 4.7(d) respectively. Four cycles of operation are shown here. The access time can be calculated using the Figures 4.7(a) and 4.7(b). The time difference between the starting point of the ROM output curve
(V(127)) and the rising edge of the clock (V(134)) is the time required to precharge the ROM bit lines which is 6.5 NS. The time difference between the rising edge of the clock and the end point of the output voltage curve is the time required to evaluate the ROM bit lines which is 7.1 NS. Hence, the access time (sum of both the precharge time and the evaluation time) is 13.6 NS.

4.4.2 RELAX simulation of the 32*7 Static ROM

The RELAX input file deck for the simulation of the 32*7 static ROM is shown in Appendix 5. This deck uses the model of the ROM as shown in Fig. 4.5. The node numbers given in this file deck are obtained by adding the node numbers in the SPICE deck for the 32*7 static ROM to the number 100. For example, the node number 132 corresponds to the node number 32 in Fig. 4.5. The output results (potential of the nodes 132 and 140) of this simulation are shown in Figures 4.8(a) and 4.8(b). The time required to generate a 5-volt 'logic 1' is 23 NS and the time required to generate a 0-volt 'logic 0' is 11.6 NS.

4.4.3 RELAX simulation of a fault-tolerant systolic cell.

The RELAX simulation of the systolic cell shown in Fig. 4.9 is discussed in this section. The RELAX input deck for simulation of the systolic cell is shown in Appendix 6. The node numbers given in this file are the same node numbers shown in Fig. 4.9. The major elements of this cell are: 1) a 32*7 ROM, 2) a 2-input Exclusive OR gate, 3) a 2-input OR gate, and 4) a 2-phase Latch. Fig. 4.10(a) shows the output of the 32*7 dynamic ROM. Two cycles of operation are shown which means that the ROM bit lines are
Fig. 4.7(b) Clock for the 32x7 dynamic ROM
Fig. 4.7(c) Output of the 3-input Domino CMOS AND gate
Fig. 4.8(a) Output of the p-channel ROM cell in the 32x7 static ROM
Fig. 4.8(b) Output of the n-channel ROM cell in the 32x7 static ROM
Fig. 4.9 Systolic cell for SPICE simulation
Fig. 4.10(a) Output of the 32x7 dynamic ROM
twice precharged and evaluated using the two-phase non-overlapping clocks shown in Figures 4.10(b) and 4.10(c). The output of part A of the latch, which is stored in the node 158, is shown in Fig. 4.10(d). The output of part B of the latch, which is stored in the node 159, is shown in Fig. 4.10(e). The time required to generate a 5-volt output signal is 26.7 NS. Since the output of the ROM is a 'logic 0' signal and the input signals to the EX-OR2 gate and the OR2 gate are 'logic 1' signals, the output of the systolic cell is a 'logic 1' signal.

4.5 Comparison of simulation results

The simulation results given by SPICE and RELAX can be tabulated as follows:

<table>
<thead>
<tr>
<th>Type of design</th>
<th>Process</th>
<th>Delay time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SPICE</td>
</tr>
<tr>
<td>32*7 Dynamic ROM</td>
<td>CMOS3DLM</td>
<td>13.3 NS</td>
</tr>
<tr>
<td>32*7 Static ROM</td>
<td>CMOS3DLM</td>
<td>21.7 NS</td>
</tr>
<tr>
<td>Fault Tolerant systolic cell</td>
<td>CMOS3DLM</td>
<td>----</td>
</tr>
</tbody>
</table>

From Table 4.3, we can observe the following points:

1) The results given by SPICE and RELAX agree with each other within acceptable error bounds. The difference in the results of these programs is due to the difference in number of parameters used in the MOSFET models. We will use the results given by the
Fig. 4.10(b) Clock CLl of the two-phase non-overlapping clocks
Fig. 4.10(c) Clock CL2 of the two-phase non-overlapping clock
Fig. 4.10(d) Output of part A of the latch
Fig. 4.10(e) Output of part B of the latch
SPICE program because it is more popular and accurate than the RELAX program.

2) Both the static and dynamic ROMs have a small access time. These memories can be employed in a hardware system whose operational frequency is 40 MHz. The major factor for the high speed operation of these memories is due to the use of metal word lines and bit lines, the restricted use of poly-silicon interconnects, and the regular and dense layout of memory cells.

3) The 32*7 dynamic ROM is faster than the 32*7 static ROM. This is the major conclusion of this chapter. The major reasons for the higher speed operation of the dynamic memory are:
   a) The dynamic memory cell has fewer number of p-channel transistors than the static cell.
   b) The ROW decoder gates of the dynamic ROM are Domino CMOS gates which have lower number of gates than the static AND3 gates used in the row decoder of the static ROM.
   c) The word lines of the static ROM are longer than the word lines of the dynamic ROM. The longer the line length of wires, the slower the rise and fall times.

4) The 32*7 ROM based systolic fault-tolerant cell operates at a high speed. The major cause for this high speed is fast operation of the dynamic ROM. The perfect operation of the systolic cells suggests that the clocks of the ROM can be effectively coalesced with the pipeline clocks.

4.6 Power dissipation

One of the important parameters that characterizes an IC chip is the power dissipation since it directly relates to the time delay
and reliability of the chip. Power dissipation of a design consists of two major components:
1) Standby power dissipation and
2) Dynamic power dissipation.

4.6.1 Standby power dissipation
We have seen that both the static and dynamic memories do not operate in the pseudo-nMOS logic mode. The pseudo-nMOS operation is prevented in the static ROM by using complete CMOS static circuits, and in the dynamic ROM by not allowing the decoder to turn on during the precharge period, and by not allowing the precharge p-channel transistors to turn on during the evaluation period. These measures result in negligible standby power for both memories. It can be concluded that none of these memories sacrifice any of the constraints on static power dissipation for the quest of speed.

4.6.2 Dynamic power dissipation
The dynamic power dissipation is the power dissipation that occurs due to switching of the devices. The power dissipation can be calculated as follows:

\[ P_{\text{Dynamic}} = C \ast V^2 \ast f \]

where
\[ f = \text{the machine cycle frequency} \]
\[ C = \text{the total capacitance of the design} \]
\[ V = \text{the logic voltage swing (usually Vdd)} \]

The following table provides the values of power dissipation
for the static and dynamic ROMs:

### Table 4.4

**Power Dissipation (critical path)**

<table>
<thead>
<tr>
<th>Type of ROM logic</th>
<th>Total capacitance</th>
<th>Voltage swing</th>
<th>Switching frequency</th>
<th>Power dissipation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic</td>
<td>2.37</td>
<td>5</td>
<td>40</td>
<td>2.37</td>
</tr>
<tr>
<td>Static</td>
<td>3.17</td>
<td>5</td>
<td>40</td>
<td>3.17</td>
</tr>
</tbody>
</table>

From Table 4.4 we can observe that the power dissipation of the 32*7 dynamic memory is lower than the power dissipation of the 32*7 static ROM. This is due to the lower capacitance of the dynamic ROM than the static ROM.

### Table 4.5

**Comparison of the 32*7 Dynamic CMOS ROM and the CMOS ROMs available in the literature.**

<table>
<thead>
<tr>
<th>Type of ROM</th>
<th>Logic</th>
<th>#Trans/cell</th>
<th>Access time</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>256*16</td>
<td>dynamic</td>
<td>1</td>
<td>36NS</td>
<td>1.8 micron N-well CMOS3DLM</td>
</tr>
<tr>
<td>(Takahashi, Fujitsu, '85)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2K*8</td>
<td>dynamic</td>
<td>1</td>
<td>30NS</td>
<td>1.8 micron N-well CMOS3DLM</td>
</tr>
<tr>
<td>(Miyahara, NTT)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32*7</td>
<td>dynamic</td>
<td>1</td>
<td>13.6NS</td>
<td>3.0 micron P-well CMOS3DLM</td>
</tr>
</tbody>
</table>
4.7 Comparison of the dynamic 32x7 ROM and the ROMs in the literature

Let us try to compare the characteristics of the dynamic ROM with the CMOS ROMs (fairly recent ones) available in the literature. The recent CMOS ROMs are the 256x16 ROM developed by Takahashi et al. of Fujitsu as a part of a CMOS array and a 1Kx8 CMOS ROM developed by Miyahara et al. of NTT as a part of a composite CMOS gate array. These two memories are reviewed in Chapter 3. The characteristics of these ROMs are given in Table 4.5.

From Table 4.5 it is clear that access time of the 32x7 dynamic ROM could be lower if designed with much smaller feature sizes. The access time increases with the increase of size. More precisely, the increase in access time is due to the increase in length of the word lines and bit lines; however, the RC product will remain the same despite scaling unless there is a significant improvement in the process. Comparison of area is not possible with memories available in the literature because of different process technologies with different feature sizes. It should be noted at this point that in this thesis importance is given to the development of circuit techniques rather than to the layout techniques. With a good memory cell technique, we can layout a ROM with different process technologies.

4.8 Summary

Simulation of the 32x7 dynamic ROM, the 32x7 static ROM, and the 32x7 ROM based fault-tolerant systolic cell are carried out using the simulators SPICE and RELAX. These simulations include capacitances and resistances extracted from the layouts of the
ROMs. Results given by both the simulators agree with each other in certain error bounds. It has been recognized that the 32*7 dynamic ROM is faster than the 32*7 static ROM. Studies on the power dissipation of ROMs reveal that the 32*7 static ROM consumes more power than the 32*7 dynamic ROM.
5.1 Overview

The major objective of this thesis is the custom design of CMOS ROMs which form the backbone for RNS VLSI architectures, particularly for systolic arrays using RNS arithmetic. Chapter 1 discussed the principles of RNS arithmetic, systolic arrays, and the need for small and fast ROM arrays. The principles behind the custom VLSI design and the CMOS IC design were reviewed in Chapter 2. In that chapter, static and dynamic CMOS logic gates were reviewed and techniques for better layout of CMOS circuits were given.

In Chapter 3, the heart of this thesis, the design of CMOS ROMs was discussed. It was demonstrated that how complete designs of a dynamic 32*7 ROM and a static 32*7 ROM were developed. Through literature survey several circuit techniques for the design of ROMs were gathered and based on these circuits techniques, several static and dynamic CMOS ROMs were designed. These techniques were further refined and a 32*7 dynamic CMOS ROM and a 32*7 static CMOS ROM were developed. Both the memories offer good output voltage swings and operate perfectly without any serious problems and use only one transistor per bit memory cells. The layout techniques for the cell structure of both memories are the same. The area comparison reveals that the 32*7 dynamic ROM is smaller than the 32*7 static ROM. A pipelined
fault-tolerant 5-bit RNS adder has been laid out using 32*7
dynamic ROMs.

In Chapter 4, a layout level simulation of both the memories
was discussed. Both the memories were simulated using SPICE and
RELAX simulators. These simulations revealed that the 32*7
dynamic CMOS ROM is faster than the 32*7 static CMOS ROM. Based
on the layouts, power dissipation for both the memories was
calculated. This calculation shows that the dynamic CMOS 32*7 ROM
dissipates a lower amount of power than the static CMOS 32*7 ROM.
The 5-bit adder design based on 32*7 ROMs operates at a high
speed.

5.2 Characteristics of final ROM designs

We can summarize the characteristics of the 32*7 static and
dynamic ROMs in the following tables. We will use the parameters
area, time, and power dissipation for comparisons. The area of
the ROM is obtained by the product of its length and width. The
access times of the ROMs are obtained by the SPICE and RELAX
simulations discussed in the previous chapter. Power dissipation
calculations of these memories can be found in Chapter 4.

<table>
<thead>
<tr>
<th>Type of ROM</th>
<th>Area (sq. microns)</th>
<th>Access time (nanoseconds)</th>
<th>Power dissipation (mWatts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32*7 dynamic CMOS ROM</td>
<td>0.200</td>
<td>13.3</td>
<td>2.37 @ 40 MHz</td>
</tr>
<tr>
<td>32*7 static CMOS ROM</td>
<td>0.371</td>
<td>21.7</td>
<td>3.17 @ 40 MHz</td>
</tr>
</tbody>
</table>
5.3 The final conclusions and discussions

We can summarize the previous conclusions and discussion in all four chapters as follows:

Read only memories can be designed in both static and dynamic logic CMOS. The major difference between these two types of memories is the type of transistors employed as memory cells. A static ROM requires both the p-channel and n-channel transistors as memory cells to be used extensively for proper output logic levels. A dynamic ROM requires only one type of transistor as memory cells to be used for proper output logic levels.

Dynamic read only memories are smaller and faster than the static read only memories under conditions such as equal logic swing voltage magnitudes, use of non pseudo-nMOS logic mode of operation, regular layout of ROM cells, common source mode operation of memory cells, the same layout rules, the same fabrication processes, use of two dimensional decoding scheme, etc. Performance of these memories could be improved if designed using smaller feature size advanced fabrication processes.

Dynamic read only memories can be used to design high throughput pipelined systolic RNS structures such as adders and filters. Use of a simplified clocking scheme such as a two phase non-overlapping clocking allows simplify the design of a pipelined element using dynamic ROMs and latches.

The circuit techniques developed for the 32*7 dynamic and 32*7 static ROM can be applied to the design of large ROMs. The size, speed, and the power dissipation will be different for
these ROMs.

The developed complementary cell static ROM techniques are better than any other techniques available for static ROMs due to the simplification achieved by the use of well known two dimensional decoding scheme. This decoding scheme helps reduce transistor counts in the memory cell array and decoders.

Layout techniques for dynamic memory cells can be used to layout static memory cells since both the memory cells use only one transistor to store a bit of information.

Extensive use of the two metals of the CMOS3DLM process and restricted use of polysilicon help reduce the time delay of an ROM. The metals have lower capacitance and resistance per unit area than polysilicon layers.

The gate matrix-type layout approach for decoders and ROM cell arrays provides standard cell look-alike ROMs with defined input and output connection points, which help abutt ROMs with each other.

Decoders can be designed using static and dynamic logic; however, dynamic logic decoders can be used to realize gated decoders which are useful in the design of pipelined structures since they can be operated with the pipeline clocks.
REFERENCES


[25]. L.A. Glasser and D.N. Doppherhul, The design and analysis of VLSI circuits, Addison-Wesley, 1985, Reading,MA p.125


[27]. A. Vladimirescu, et al., SPICE version 2G.6 Users Guide, Department of Electrical Engineering, University of
California, Berkley, 1981


[38]. Amar Mukherjee, *nMOS and CMOS VLSI system design*, Prentice-Hall, Englewood-Cliffs, New Jersey.


APPENDIX 1

Layout of basic components of ROMs
Fig. 6.1 Layout of the components of the 49*3 static ROM
Fig. 6.2 Layout of the components of the 24x5 static ROM
Fig. 6.3 Layout of the components of an improved 24*5 static ROM
Fig. 6.4 Layout of the components of the 24×5 dynamic ROM
Fig. 6.5 Layout of the 3-input Domino CMOS gate
Fig. 6.6 Layout of the components of the 32x7 dynamic ROM and the Fault-tolerant systolic cell
APPENDIX 2

SPICE Input deck for the 32*7 dynamic ROM
TEST OF 32x7 DYNAMIC ROM
*SPICE SIMULATION PROGRAM FOR ANALYSIS FOR A DYNAMIC CMOS ROM
*WRITTEN BY: P.V.R. RAJA FOR HIS M.A.Sc THESIS
*DYNAMIC CMOS ROM
.OPT ACCT LIST NODE LIMPTS=500000 LVLCOD=2 ITL5=0
.TRAN 0.INS 0NS 0NS

/*********************************************************/

/*SINGLE INVERTER
.SUBCKT INV1 1 2 3
M1 3 2 1 1 MODP L=3.0U W=5.4U AS=64.8P PS=34.8U AD=42.12P PD=26.4U
+NRS=2.2 NRD=1.2
M2 4 2 0 0 MODN L=3.0U W=3U AS=57.6P PS=34.8U AD=36.6P PD=26.4U
+NRS=2.2 NRD=1.5
.ENDS INV1

/*********************************************************/

/*DOMINO AND3 GATE
*FOR ROW DECODER
.SUBCKT DYAND3 1 2 4 6 8 30 10
M1 3 2 1 1 MODP L=3U W=5.4U AS=46.8P PS=34.5U AD=42.12P PD=26.4U
+NRS=2.2 NRD=1.5
M2 3 4 5 0 MODN L=3U W=5.4U AD=42.12P PD=26.4U AS=8.1P PS=13.8U
+NRS=2.2 NRD=0.27
M3 5 6 7 0 MODN L=3U W=5.4U AD=8.1P PD=13.8U AS=8.1P PS=13.8U
+NRD=0.27 NRS=0.27
M4 7 8 9 0 MODN L=3U W=5.4U AS=8.1P PS=13.8U AD=8.1P PD=13.8U
+NRD=0.27 NRS=0.27
M5 9 2 0 0 MODN L=3U W=5.4U AD=8.1P PD=13.8U AS=64.8P PS=34.8U
+NRD=0.27 NRS=2.2
R66 3 30 0.250K
C66 30 0 0.023PF
M6 10 30 1 1 MODP L=3U W=16.2U AS=194.46P PS=56.4U AD=126.36P PD=48.0U
+NRD=1.5 NRS=2.2
M7 10 30 0 0 MODN L=3U W=5.4U AS=42.12P PD=26.4U AD=64.8P PD=34.8U
+NRD=1.5 NRS=2.2
.ENDS DYAND3

/*********************************************************/

/*STATIC AND2 GATE FOR COLUMN DECODER
.SUBCKT AND2 1 2 3 6
M1 4 2 1 1 MODP L=3U W=5.4U AS=64.6P PS=34.8U AD=8.1P PD=13.8U
+NRD=1.5 NRS=2.2
M2 4 3 1 1 MODP L=3U W=5.4U AS=42.12P PD=26.4U AD=55.08P PD=31.2U
+NRD=0.27 NRS=1.5
M3 4 3 5 0 MODN L=3U W=5.4U AS=55.08P PS=31.2U AD=42.12P PD=26.4U
+NRD=1.5 NRS=1.44
M4 5 2 0 0 MODN L=3U W=5.4U AS=64.8P PS=34.0U AD=8.1P PD=13.8U
+NRS=2.2 NRD=0.27
R77 4 40 0.210K
C77 40 0 0.0214PF
M5 6 40 1 1 MODP L=3U W=16.2U AS=194.46P PS=56.4U AD=64.8P PD=48.0U
+NRD=1.5 NRS=2.2
M6 6 40 0 0 MODN L=3U W=5.4U AS=42.12P PS=26.4U AD=64.8P PD=34.8U
+NRD=1.5 NRS=2.2
.ENDS AND2

/*********************************************************/
*SINGLE TRANSISTOR MEMORY CELL
.SUBCKT ROM_CELL 3 2
M1 2 3 0 0 MODN L=3u W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
+NRD=1.5 NRS=1.5
.ENDS ROM_CELL

*COLUMN DECODING TRANSISTOR
.SUBCKT COL_TRANS 2 3 4
M1 2 3 4 0 MODN L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
+NRD=1.5 NRS=1.5
.ENDS COL_TRANS

*P CHANNEL TRANSISTOR FOR PRECHARGE GENERATION
.SUBCKT P_TRANS 1 2 3
M1 3 2 1 1 MODP L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
+NRD=1.5 NRS=1.5
.ENDS P_TRANS

******************************************************************************
.MODEL MODN NMOS (TOX=5.0E-8 NSUB=1.7E16 XJ=6E-7 RSH=25
  + VTO=0.7 UO=775.0 GAMMA=1.1
  + VMAX=1.0E5 PB=0.7 JS=10.0E-6 LD=3.5E-7 TPG=1 LAMBDA=0.01
  + RS=40.0 RD=40.0 CJ=4.4E-4 MJ=0.5 CJSW=4.0E-10 PHI=0.6
  + MJSW=0.3 CGSO=3.0E-10 CGDO=3.0E-10 CGBO=5.0E-10 KP=50E-6
  + LEVEL=2)
.MODEL MODP PMOS (TOX=5.0E-8 NSUB=5.0E15 XJ=5E-7 RSH=80
  + VTO=-0.8 UO=250.0 GAMMA=0.6
  + VMAX=7.0E4 PB=0.6 JS=10.0E-6 LD=2.5E-7 TPG=1 LAMBDA=0.03
  + RS=100.0 RD=100.0 CJ=1.5E-4 MJ=0.6 CJSW=4.0E-10 PHI=0.6
  + MJSW=0.6 CGSO=2.5E-10 CGDO=2.5E-10 CGBO=5.0E-10 KP=16E-6
  + LEVEL=2)
******************************************************************************
*DESCRIPTION OF THE ROM CIRCUIT
*ROW DECODER ADDRESS DESCRIPTION
*INPUT ADDRESS A0-A2 IN NODES 2,3,& 4

VDD 1 0 DC 5
R1 2 5 0.150K
R2 3 6 0.150K
R3 4 7 0.150K
C1 5 0 0.17826PF
C2 6 0 0.17826PF
C3 7 0 0.17826PF

------------------
X11 1 5 8 INV1
X12 1 6 9 INV1
X13 1 7 10 INV1
R4 8 11 0.330K
R5 9 12 0.330K
R6 10 13 0.330K
R7 34 14 0.180K
C4 11 0 0.2101PF
C5 12 0 0.2101PF
C6 13 0 0.2101PF
C7 14 0 0.1248PF

X14 1 14 11 12 13 80 15 DYAND3

R8 15 16 0.720K
C8 16 0 0.110406PF

X15 16 18 ROM_CELL

*DESCRIPTION FOR COLUMN DECODER |

*INPUT ADDRESS A3,A4 IN NODES 19 & 20
R10 19 21 0.150K
R11 20 22 0.150K
C10 21 0 0.092164PF
C11 22 0 0.092164PF

X16 1 21 31 INV1
X17 1 22 32 INV1
R12 31 23 0.350K
R13 32 24 0.350K
C12 23 0 0.060121PF
C13 24 0 0.060121PF

*---------------------------
X18 1 23 24 25 AND2
*---------------------------
R14 25 26 0.720k
C14 26 0 0.02PF
X19 27 26 28 COL_TRANS
R15 18 28 0.003k
C15 28 0 0.016036PF
R16 29 27 0.004k
C16 27 0 0.0711984PF
R17 34 35 0.720K
C17 35 0 0.04PF

*---------------------------
X20 1 35 29 P_TRANS
*---------------------------
VA 2 0 PULSE(0 5 15NS 0 0 0NS 15NS)
VB 3 0 PULSE(0 5 15NS 0 0 0NS 15NS)
VC 4 0 PULSE(0 5 15NS 0 0 0NS 15NS)
VD 19 0 PULSE(0 5 15NS 0 0 0NS 15NS)
VE 20 0 PULSE(0 5 15NS 0 0 0NS 15NS)
VCL 34 0 PULSE(0 5 7NS 0 0 7NS 15NS)

*These initial conditions are set for convergence
*All the internal nodes are set to 'zero'
.IC V(5)=0 V(6)=0 V(7)=0 V(8)=0 V(9)=0
.IC V(10)=0 V(11)=0 V(12)=0 V(13)=0 V(14)=0 V(15)=0 V(16)=0
.IC V(18)=0 V(28)=0 V(27)=0 V(29)=0 V(26)=0 V(25)=0 V(80)=5
.IC V(23)=0 V(24)=0 V(31)=0 V(32)=0 V(21)=0 V(22)=0
.PRINT TRAN V(2) V(34) V(15) V(25) V(28) V(27)
*Print the input wave and the output
.END
APPENDIX 3

SPICE Input deck for the 32*7 static ROM
TEST OF 32X7 STATIC ROM
*SIMULATION OF A STATIC CMOS ROM
*WRITTEN BY: P.V.R. RAJA FOR HIS M.A.SC THESIS
.OPT ACCT LIST NODE LIMPTS=500000 LVLCCD=2 ITL5=0
.TRAN 0.1NS 0NS

* Single Inverter circuit
*-----------------------------------------------------
.SUBCKT INV1 1 2 3
M1 3 2 1 1 MODP L=3.0U W=5.4U AS=64.8P PS=34.8U AD=42.12P PD=26.4U
+NRS=2.2 NRD=1.4
M2 3 2 0 0 MODN L=3.0U W=3U AS=57.6P PS=34.8U AD=36.6P PD=26.4U
+NRS=2.2 NRD=1.4
.ENDS INV1

* 2-Input NAND2-Inverter circuit
*-----------------------------------------------------
.SUBCKT ANAND2 1 2 3 4 6
M1 4 2 1 1 MODP L=3U W=5.4U AS=64.6P PS=34.8U AD=12.96P PD=15.6P
+NRR=0.44 NRS=2.2
M2 4 3 1 1 MODP L=3U W=5.4U AS=42.12P PS=26.4U AD=55.08P PD=31.2U
+NRR=1.4 NRS=1
M3 4 3 5 0 MODN L=3U W=5.4U AS=25.92P PS=21.6U AD=42.12P PD=26.4U
+NRR=1 NRS=1
M4 5 2 0 0 MODN L=3U W=5.4U AS=64.8P PS=34.0U AD=25.92P PD=21.6U
+NRR=1.4 NRS=2.2
R1 4 10 0.300K
C1 40 0 0.02214PF
M5 6 40 1 1 MODP L=3U W=16.2U AS=194.46P PS=56.4U AD=64.8P PD=48.0U
+NRR=2.2 NRS=1.4
M6 6 40 0 0 MODN L=3U W=5.4U AS=42.12P PS=26.4U AD=64.8P PD=34.8U
+NRR=2.2 NRS=1.4
.ENDS ANAND2

* 3-Input NAND-Inverter circuit
*-----------------------------------------------------
.SUBCKT ANAND3 1 2 4 6 7 10
M1 7 2 1 1 MODP L=3.0U W=5.4U AS=64.8P PS=34.8U AD=12.96P PD=15.6U
+NRR=0.44 NRS=2.2
M2 7 4 1 1 MODP L=3.0U W=5.4U AS=55.08P PS=30.0U AD=12.96P PD=15.6U
+NRR=0.4 NRS=1
M3 7 6 1 1 MODP L=3.0U W=5.4U AS=55.08P PS=30.0U AD=42.12P PD=26.4U
+NRR=1.2 NRS=1
M4 7 6 8 0 MODN L=3.0U W=5.4U AD=42.12P PD=26.4U AS=29.16P PS=21.6U
+NRR=1.2 NRS=1.0
M5 8 4 9 0 MODN L=3.0U W=5.4U AD=25.92P PD=20.4U AS=29.16P PS=21.6U
+NRR=1 NRS=1.0
M6 9 2 0 0 MODN L=3.0U W=5.4U AS=64.8P PS=34.8U AD=25.92P PD=20.4U
+NRR=1 NRS=2.2
R1 7 70 0.350K
C1 70 0 0.02284PF
M7 10 70 1 1 MODP L=3.0U W=16.2U AS=194.46P PS=56.4U AD=64.8P PD=48.0U
+NRR=2.2 NRS=1.4
M8 10 70 0 0 MODN L=3.0U W=5.4U AS=42.12P PS=26.4U AD=64.8P PD=34.8U

+NRS=2.2 NRD=1.4
.ENDS ANAND3

* Column decoding N-channel transistor
*------------------------------
.SUBCKT COL_NTRAN 2 3 4
M1 4 3 2 0 MODN L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
+NRS=1.4 NRD=1.4
.ENDS COL_NTRAN

* Column decoding P-channel transistor
*------------------------------
.SUBCKT COL_PTRAN 1 2 3 4
M1 4 3 2 1 MODP L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
+NRS=1.4 NRD=1.4
.ENDS COL_PTRAN

* N-channel memory cell
*-------------------------------
.SUBCKT N_CELL 3 2
M1 2 3 0 0 MODN L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
+NRD=1.4 NRS=1.4
.ENDS N_CELL

* P-channel memory cell
*-------------------------------
.SUBCKT P_CELL 1 3 2
M1 2 3 1 1 MODP L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
+NRD=1.4 NRS=1.4
.ENDS P_CELL

******************************************************************************
** MODEL MODN NMOS(TOX=5.0E-8 NSUB=1.7E16 XJ=6E-7 RSH=25
+ VTO=0.7 UO=775.0 GAMMA=1.1
+ VMAX=1.0E5 PB=0.7 JS=10.0E-6 LD=3.5E-7 TFG=1 LAMBDA=0.01
+ RS=1.0E4 RD=1.0E-10 CJ=4.4E-4 MJ=0.5 CJSW=4.0E-10 PHI=0.6
+ MJSW=0.3 CGSO=3.0E-10 CGDO=3.0E-10 CGBO=5.0E-10 KP=50E-6
+ LEVEL=2)
** MODEL MODP PMOS(TOX=5.0E-8 NSUB=5.0E15 XJ=5E-7 RSH=80
+ VTO=0.8 UO=250.0 GAMMA=0.6
+ VMAX=7.0E4 PB=0.6 JS=10.0E-6 LD=2.5E-7 TFG=1 LAMBDA=0.03
+ RS=100.0 RD=100.0 CJ=1.5E-4 MJ=0.6 CJSW=4.0E-10 PHI=0.6
+ MJSW=0.6 CGSO=2.5E-10 CGDO=2.5E-10 CGBO=5.0E-10 KP=16E-6
+ LEVEL=2)
******************************************************************************

*************** TOP LEVEL CALLS ***************
*Description of the ROM circuit
*------------------------------
*ROM decoder circuit
*Input Address A0-A2 at nodes 2,3, & 4

VDD 1 0 DC 5

R1 2 5 0.300K
R2 3 6 0.300K
R3 4 7 0.300K
C1 5 0 0.23267PF
C2 6 0 0.23267PF
C3 7 0 0.23267PF
X11 1 5 8 INV1
X12 1 6 9 INV1
X13 1 7 10 INV1
R4 8 11 0.450K
R5 9 12 0.450K
R6 10 13 0.450K
C4 11 0 0.243964PF
C5 12 0 0.243964PF
C6 13 0 0.243964PF
X14 1 11 12 13 14 15 ANAND3
R7 14 16 0.720K
C7 16 0 0.1270672PF
R8 15 17 0.720K
C8 17 0 0.176448PF
*-----------------------------------------------
* Column decoder address at nodes 18 & 19
*-----------------------------------------------
R9 18 20 0.450K
R10 19 21 0.450K
C9 20 0 0.03894PF
C10 21 0 0.03894PF
X15 1 20 22 INV1
X16 1 21 23 INV1
R11 22 24 0.450K
R12 23 25 0.450K
C11 24 0 0.048PF
C12 25 0 0.048PF
X17 1 24 25 26 27 ANAND2
R13 26 28 0.720K
C13 28 0 0.040616PF
R14 27 29 0.720K
C14 29 0 0.047402PF
X18 17 30 N_CELL
R18 30 31 0.003K
C18 31 0 0.017933PF
X19 31 29 38 COL_NTRAN
R30 38 40 0.003K
C30 40 0 0.132924PF
C19 32 0 0.132924PF
R19 33 32 0.003K
X20 1 34 28 33 COL_PTRAN
R20 35 34 0.003K
C20 34 0 0.017933PF
X21 1 16 35 P_CELL

*****************************************************************************
VA 2 0 PULSE(0 5 20NS 0 0 0 8NS)
VB 3 0 PULSE(0 5 20NS 0 0 0 8NS)
VC 4 0 PULSE(0 5 20NS 0 0 0 8NS)
VD 18 0 PULSE(0 5 20NS 0 0 0 8NS)
VE 19 0 PULSE(0 5 20NS 0 0 0 8NS)
*****************************************************************************
.PRINT TRAN V(2) V(15) V(27) V(32) V(40)
. IC V(8)=0 V(9)=0 V(10)=0 V(11)=0 V(12)=0 V(13)=0 V(14)=5
. IC V(15)=0 V(16)=5 V(17)=0 V(22)=0 V(23)=0 V(26)=5 V(27)=0
. IC V(28)=5 V(33)=0 V(32)=0 V(34)=0 V(40)=5 V(24)=5 V(29)=0
.END
APPENDIX 4

RELAX Input deck for the 32*7 dynamic ROM
; TEST OF 32x7 DYNAMIC ROM
; RELAX INPUT DECK
; WRITTEN BY P. V. R. RAJA FOR HIS M.A.Sc THESIS
GLOBAL 0
; ----------------------------------------
; Power definition
VDD 100 0 DC V=5
; ----------------------------------------
; Description of the circuit
; ---------------------------
; ROW decode circuitry description
; ---------------------------
RA1 102 105 R R=0.150K
RA2 103 106 R R=0.150K
RA3 104 107 R R=0.150K
CA1 105 0 C C=0.17826PF
CA2 106 0 C C=0.17826PF
CA3 107 0 C C=0.17826PF
; ---------------------------
X11 100 105 108 INV1
X12 100 106 109 INV1
X13 100 107 110 INV1
; ---------------------------
RA4 108 111 R R=0.330K
RA5 109 112 R R=0.330K
RA6 110 113 R R=0.330K
RA7 134 114 R R=0.180K
CA4 111 0 C C=0.2101PF
CA5 112 0 C C=0.2101PF
CA6 113 0 C C=0.2101PF
CA7 114 0 C C=0.1248PF
; ---------------------------
X14 100 114 111 112 113 190 115 DYAND3
; ---------------------------
RA8 115 116 R R=0.720K
CA8 116 0 C C=0.110406PF
X15 116 118 ROMCELL

;* Column decoder circuitry description

RB0 119 121 R R=0.150K
RB1 120 122 R R=0.150K
CB0 121 0 C C=0.092164PF
CB1 122 0 C C=0.092164PF

X16 100 121 131 INV1
X17 100 122 132 INV1

RB2 131 123 R R=0.350K
RB3 132 124 R R=0.350K
CB2 123 0 C C=0.060121PF
CB3 124 0 C C=0.060121PF

X18 100 123 124 125 AND2

RB4 125 126 R R=0.720K
CB4 126 0 C C=0.02PF

X19 127 126 128 COLTRANS

RB5 118 128 R R=0.003K
CB5 128 0 C C=0.016036PF

RB6 129 127 R R=0.004K
CB6 127 0 C C=0.0711984PF

RB7 134 135 R R=0.720K
CB7 135 0 C C=0.04PF

X20 100 135 129 PTRANS

; CONSTRUCTED COMPLEX MODEL.

MODEL NMOS NMOS VTO=0.7 KP=50U GAMMA=1.1 PHI=0.6 LAMBDA=0.01 \ CGSO=3.0E-10 CGDO=3.0E-10 CGBO=5.0E-10 TOX=500E-10

MODEL NPDMOD D IS=3E-9 CJ0=0.73E-10 M=0.33
MODEL NADMOD D IS=3E-9 CJ0=0.77E-4 M=0.5

DEFINE NQSHARE ( D G S B )
PARAMETERS W=10U L=3U AS=1.0 AD=1.0 PS=1.0 PD=1.0
DEVICE D G S B NMOD W=W L=L
DPS B S NPDMS MOD AREA=PS
DPD B D NPDMS MOD AREA=PD
DPS B S NADMOD AREA=AS
DPD B D NADMOD AREA=AD
END NQSHARE

; CONSTRUCTED COMPLEX MODEL.
MODEL PMOD PMOS VT0=-0.8 KP=16E-6 GAMMA=0.6 PHI=0.6 LAMBDA=0.03 \CGS0=2.5E-10 CGDO=2.5E-10 CGBO=5.0E-10 TOX=500E-10

MODEL PPDMD MOD D IS=1E-9 CJ0=2.2E-10 M=0.33
MODEL PADMOD D IS=3E-9 CJ0=1.4E-4 M=0.5

DEFINE PSHARE ( D G S B )
PARAMETERS W=10U L=3U AS=1.0 AD=1.0 PS=1.0 PD=1.0
DEVICE D G S B PMOD W=W L=L
DPS S B PPDMD MOD AREA=PS
DPD D B PPDMD MOD AREA=PS
DPS S B PADMD MOD AREA=AS
DPD D B PADMD MOD AREA=AD
END PSHARE

; Definition of a single inverter
DEFINE INV1 ( 1 2 3 )
M1 3 2 1 1 PSHARE L=3.0U W=5.4U AS=64.8P PS=34.8U AD=42.12P PD=26.4U
M2 3 2 0 0 NSHARE L=3.0U W=3U AS=57.6P PS=34.8U AD=36.6P PD=26.4U
END INV1

; Definition of Domino AND3 gate
DEFINE DYAND3 ( 1 2 4 5 8 30 10 )
M1 3 2 1 1 PSHARE L=3U W=5.4U AS=46.8P PS=34.5U AD=42.12P PD=26.4U
M2 3 4 5 0 NSHARE L=3U W=5.4U AD=42.12P PD=56.4U AS=8.1P PS=13.8U
M3 5 6 7 0 NSHARE L=3U W=5.4U AD=8.1P PS=13.8U AS=13.8U
M4 7 8 9 0 NSHARE L=3U W=5.4U AS=8.1P PS=13.8U AD=8.1P PD=13.8U
M5 9 2 0 0 NSHARE L=3U W=5.4U AD=8.1P PD=13.8U AS=64.8P PS=34.8U
RAE 3 30 R=0.250K
CAE 30 0 0 C=0.023PF
M6 10 30 1 1 NSHARE L=3U W=16.2U AS=194.46P PS=56.8U AD=126.36P PD=48.0U
M7 10 30 0 0 NSHARE L=3U W=5.4U AD=42.12P PD=26.4U AD=55.08P PD=31.2U
END DYAND3

; Definition of a STATIC 2-Input AND gate
DEFINE AND2 ( 1 2 3 6 )
M1 4 2 1 1 PSHARE L=3U W=5.4U AS=64.6P PS=34.8U AD=8.1P PD=13.8U
M2 4 3 1 1 PSHARE L=3U W=5.4U AS=42.12P PS=26.4U AD=55.08P PD=31.2U
M3 4 3 5 0 NSHARE L=3U W=5.4U AS=55.08P PS=31.2U AD=42.12P PD=26.4U
M4 5 2 0 0 NQSHARE L=3U W=5.4U AS=64.8P PS=34.0U AD=8.1P PD=13.8U
RA7 4 40 R R=0.210K
CA7 40 0 C C=0.0214PF
M5 6 40 1 1 PQSHARE L=3U W=16.2U AS=194.46P PS=56.4U AD=64.8P PD=48.0U
M6 6 40 0 0 NQSHARE L=3U W=5.4U AS=42.12P PS=26.4U AD=64.8P PD=34.8U
END AND2
;----------------------------------------
;Definition of a ROM cell
;------------------------
DEFINE ROMCELL ( 3 2 )
M1 2 3 0 0 NQSHARE L=3U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
END ROMCELL
;----------------------------------------
;Definition of a Column decoding transistor
;------------------------
DEFINE COLTRANS ( 2 3 4 )
M1 2 3 4 0 NQSHARE L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
END COLTRANS
;----------------------------------------
;Definition of a P-channel precharge generator
;------------------------
DEFINE PTRANS ( 1 2 3 )
M1 3 2 1 1 PQSHARE L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
END PTRANS
;----------------------------------------
V1A 102 0 PWL PERIOD=30NS T0=0 VC=0 T1=15NS V1=0
V2A 103 0 PWL PERIOD=30NS T0=0 VO=0 T1=15NS V1=0
V3A 104 0 PWL PERIOD=30NS T0=0 VO=0 T1=15NS V1=0
V4A 119 0 PWL PERIOD=30NS T0=0 VO=0 T1=15NS V1=0
V5A 120 0 PWL PERIOD=30NS T0=0 VO=0 T1=15NS V1=0
V6A 134 0 PWL PERIOD=15NS T0=0 VO=0 T1=6.5NS V1=0 T2=7NS V2=5 \ T3=14.5NS V3=5
IC 131=5 127=0 115=0 116=0 134=5 114=5 125=0 126=0 190=5
;----------------------------------------
RELAX2 OPTIONS STOP=60NS WINDOWSIZE=60NS DODC=1 DOTRAN=1
;----------------------------------------
PLOT 102 134 115 125 127
APPENDIX 5

RELAX Input deck for the 32*7 static ROM
; RELAX INPUT deck for simulation of a STATIC CMOS ROM
; Prepared by P.V.R. Raja for his M.A.Sc thesis
GLOBAL 0
;-----------------------------------------------
; Description of the ROM circuit
;-----------------------------------------------
VDD 100 0 DC V=5
RA1 102 105 R R=0.300K
RA2 103 106 R R=0.300K
RA3 104 107 R R=0.300K
CA1 105 0 C C=0.23267PF
CA2 106 0 C C=0.23267PF
CA3 107 0 C C=0.23267PF
X11 100 105 108 INV1
X12 100 106 109 INV1
X13 100 107 110 INV1
RA4 108 111 R R=0.450K
RA5 109 112 R R=0.450K
RA6 110 113 R R=0.450K
CA4 111 0 C C=0.243964PF
CA5 112 0 C C=0.243964PF
CA6 113 0 C C=0.243964PF
X14 100 111 112 113 114 115 ANAND3
RA7 114 116 R R=0.720K
CA7 116 0 C C=0.1270672PF
RA8 115 117 R R=0.720K
CA8 117 0 C C=0.176448PF
RA9 118 120 R R=0.450K
RB1 119 121 R R=0.450K
CA9 120 0 C C=0.03894PF
CB1 121 0 C C=0.03894PF
X15 100 120 122 INV1.
X16 100 121 123 INV1
RB1 122 124 R R=0.450K
RB2 123 125 R R=0.450K
CB1 124 0 C C=0.048PF
CB2 125 0 C C=0.048PF
X17 100 124 125 126 127 ANAND2
RB3 126 128 R R=0.720K
CB3 128 0 C C=0.040616PF
RB4 127 129 R R=0.720K
CB4 129 0 C C=0.047402PF
X18 117 130 NCELL
RBB 130 131 R R=0.003K
CBB 131 0 C C=0.017933PF
X19 131 129 138 COLNTRAN
RC1 138 140 R R=0.003K
CC1 140 0 C C=0.132924PF
CB9 132 0 C C=0.132924PP
R19 133 132 R R=0.003K
X20 100 134 128 133 COLPTRAN
RC4 135 134 R R=0.003K
CC4 134 0 C C=0.017933PF
X21 100 116 135 PCELL

;CONSTRUCTED COMPLEX MODEL.
MODEL NMOD NMOS VTO=0.7 KF=50U GAMMA=1.1 PHI=0.6 LAMBDA=0.01 \ CGSO=3.0E-10 CGDO=3.0E-10 CGBO=5.0E-10 TOX=500E-10 

;MODEL NPDMOD D IS=3E-9 CJO=0.73E-10 M=0.33
MODEL NADMOD D IS=3E-9 CJO=0.77E-4 M=0.5

DEFINE NQSHARE ( D G S B )
PARAMETERS W=10U L=3U AS=1.0 AD=1.0 PS=1.0 PD=1.0
  DEVICE D G S B NMOD W=W L=L
  DPS B S NPDPMOD AREA=PS
  DPD B D NPDPMOD AREA=PD
  DPS B S NADMOD AREA=AS
  DPD B D NADMOD AREA=AD
END NQSHARE

;CONSTRUCTED COMPLEX MODEL.
MODEL PMOD PMOS VTO=-0.8 KP=16E-6 GAMMA=0.6 PHI=0.6 LAMBDA=0.03 \ 
  CDSO=2.5E-10 CGDO=2.5E-10 CGBQ=5.0E-10 TOX=500E-10

MODEL PFDMOD D IS=1E-9 CJO=2.2E-10 M=0.33
MODEL PADMOD D IS=3E-9 CJO=1.4E-4 M=0.5

DEFINE PQSHARE ( D G S B )
PARAMETERS W=10U L=3U AS=1.0 AD=1.0 PS=1.0 PD=1.0
  DEVICE D G S B PMOD W=W L=L
  DPS S B PPDPMOD AREA=PS
  DPD D B PPDPMOD AREA=PS
  DPS S B PADMOD AREA=AS
  DPD D B PADMOD AREA=AD
END PQSHARE

;Definition of a single inverter

DEFINE INV1 ( 1 2 3 )
M1 3 2 1 1 PQSHARE L=3.0U W=5.4U AS=64.8P PS=34.8U AD=42.12P PD=26.4U
M2 3 2 0 0 NQSHARE L=3.0U W=3U AS=57.6P PS=34.8U AD=36.6P PD=26.4U
END INV1

;Definition of a 2-input NAND-Inverter gate

DEFINE ANAND2 ( 1 2 3 4 6 )
M1 4 2 1 1 PQSHARE L=3U W=5.4U AS=64.6P PS=34.8U AD=12.96P PD=15.6P
M2 4 3 1 1 PQSHARE L=3U W=5.4U AS=42.12P PS=26.4U AD=55.08P PD=31.2U
M3 4 3 5 0 NQSHARE L=3U W=5.4U AS=25.92P PS=21.6U AD=42.12P PD=26.4U
M4 5 2 0 0 NQSHARE L=3U W=5.4U AS=64.8P PS=34.0U AD=25.92P PD=21.6U
RA1 4 40 R R=0.300K
CA1 40 0 C C=0.02214PF
M5 6 40 1 1 PQSHARE L=3U W=16.2U AS=194.46P PS=56.4U AD=64.8P PD=48.0U
M6 6 40 0 0 NQSHARE L=3U W=5.4U AS=42.12P PS=26.4U AD=64.8P PD=34.8U
END ANAND2

;Definition of a 3-input NAND-Inverter gate

DEFINE ANAND3 ( 1 2 4 6 7 10 )
M1 7 2 1 1 PQSHARE L=3.0U W=5.4U AS=64.8P PS=34.8U AD=12.96P PD=15.6U
M2 7 4 1 1 PQSHARE L=3.0U W=5.4U AS=55.08P PS=30.0U AD=12.96P PD=15.6U
M3 7 6 1 1 PQSHARE L=3.0U W=5.4U AS=55.08P PS=30.0U AD=42.12P PD=26.4U
M4 7 6 8 0 NQSHARE L=3.0U W=5.4U AD=42.12P PD=26.4U AS=29.16P PS=21.6U
DEFINE COLTRAN ( 2 3 4 )
M1 4 3 2 0 NQSHARE L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
END COLTRAN

;*-----------------------------------------------
;Definition of column decoding p-channel gate
;*-----------------------------------------------
DEFINE COLDTRAN ( 1 2 3 4 )
M1 4 3 2 1 PQSHARE L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
END COLDTRAN

;*-----------------------------------------------
;Definition of an n-channel memory gate
;*-----------------------------------------------
DEFINE NCELL ( 3 2 )
M1 2 3 0 0 NQSHARE L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
END NCELL

;*-----------------------------------------------
;Definition of p-channel memory gate
;*-----------------------------------------------
DEFINE PCELL ( 1 3 2 )
M1 2 3 1 1 PQSHARE L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
END PCELL

;*-----------------------------------------------
V1A 102 0 PWL PERIOD=20NS T0=0 V0=0 T1=20NS V1=0
V2A 103 0 PWL PERIOD=20NS T0=0 V0=0 T1=20NS V1=0
V3A 104 0 PWL PERIOD=20NS T0=0 V0=0 T1=20NS V1=0
V4A 118 0 PWL PERIOD=20NS T0=0 V0=0 T1=20NS V1=0
V5A 119 0 PWL PERIOD=20NS T0=0 V0=0 T1=20NS V1=0
IC 132=0 140=5 114=5 116=5 126=5 128=5 115=0 117=0 127=0 129=0

;*-----------------------------------------------
RELAX2 OPTIONS STOP=60NS WINDOWSIZE=60NS DODC=1 DOTRAN=1
;*-----------------------------------------------

PLOT 102 115 127 132 140
APPENDIX 6

RELAX Input deck for the fault-tolerant systolic cell
; TEST OF A SYSTOLIC CELL
; A RELAX input deck for the simulation of a systolic cell
; The systolic cell consists of a Dynamic CMOS ROM, an EXCLUSIVE-
; 2 input OR gate, an OR2 gate, and a Latch.
; Written by P.V.R. Raja for his M.A.Sc Thesis

GLOBAL 0

; Description of the circuit

VDD 100 0 DC V=5

RA1 102 105 R R=0.150K
RA2 103 106 R R=0.150K
RA3 104 107 R R=0.150K
CA1 105 0 C C=0.17826PF
CA2 106 0 C C=0.17826PF
CA3 107 0 C C=0.17826PF
X11 100 105 108 INV1
X12 100 106 109 INV1
X13 100 107 110 INV1

RA4 108 111 R R=0.330K
RA5 109 112 R R=0.330K
RA6 110 113 R R=0.330K
RA7 154 114 R R=0.180K
CA4 111 0 C C=0.2101PF
CA5 112 0 C C=0.2101PF
CA6 113 0 C C=0.2101PF
CA7 114 0 C C=0.1248PF

X14 100 114 111 112 113 190 115 DYAND3

RA8 115 116 R R=0.720K
CA8 116 0 C C=0.110406PF
;*------------------------
X15 116 118 ROMCELL
;* COLUMN DECODER circuit description
;*------------------------
RB0 119 121 R R=0.150K
RB1 120 122 R R=0.150K
CB0 121 0 C C=0.092164PF
CB1 122 0 C C=0.092164PF
;*------------------------
X16 100 121 131 INV1
X17 100 122 132 INV1
;*------------------------
RB2 131 123 R R=0.350K
RB3 132 124 R R=0.350K
CB2 123 0 C C=0.060121PF
CB3 124 0 C C=0.060121PF
;*------------------------
X18 100 123 124 125 AND2
;*------------------------
RB4 125 126 R R=0.720K
CB4 126 0 C C=0.02PF
X19 127 126 128 COLTRANS
RB5 118 128 R R=0.003K
CB5 128 0 C C=0.016036PF
RB6 129 127 R R=0.004K
CB6 127 0 C C=0.0711984PF
RB7 157 135 R R=0.720K
CB7 135 0 C C=0.04PF
X20 100 135 129 PTRANS
X21 100 127 150 151 EXOR
X22 100 151 152 153 OR
X23 100 154 155 INV1
X24 100 156 157 INV1
X25 100 153 154 155 156 157 158 159 303 304 LATCH

; -----------------------------
; CONSTRUCTED COMPLEX MODEL.
MODEL NMOS VTO=0.7 KP=50U GAMMA=1.1 PHI=0.6 LAMBDA=0.01 \ 
CGSO=3.0E-10 CGDO=3.0E-10 CGBO=5.0E-10 TOX=500E-10
; -----------------------------
MODEL NPD MOD D IS=3E-9 CJ0=0.73E-10 M=0.33
MODEL NAD MOD D IS=3E-9 CJ0=0.77E-4 M=0.5

DEFINE NGSHARE ( D G S B )
PARAMETERS W=10U L=3U AS=1.0 AD=1.0 PS=1.0 PD=1.0
    DEVICE D G S B NMOD W=W L=L
    DPS B S NPD MOD AREA=PS
    DPD B D NPD MOD AREA=PD
    DPS B S NAD MOD AREA=AS
    DPD B D NAD MOD AREA=AD
END NGSHARE

; -----------------------------
; CONSTRUCTED COMPLEX MODEL.
MODEL PMOS VTO= -0.8 KP=16E-6 GAMMA=0.6 PHI=0.6 LAMBDA=0.03 \ 
CGSO=2.5E-10 CGDO=2.5E-10 CGBO=5.0E-10 TOX=500E-10
; -----------------------------
MODEL PPD MOD D IS=1E-9 CJ0=2.2E-10 M=0.33
MODEL PAD MOD D IS=3E-9 CJ0=1.4E-4 M=0.5

DEFINE PSHARE ( D G S B )
PARAMETERS W=10U L=3U AS=1.0 AD=1.0 PS=1.0 PD=1.0
    DEVICE D G S B PMOD W=W L=L
    DPS S B PPD MOD AREA=PS
    DPD D B PPD MOD AREA=PS
    DPS S B PAD MOD AREA=AS
    DPD D B PAD MOD AREA=AD
END PSHARE

; -----------------------------
; DEinition of Dynamic ROM *
; -----------------------------

; Definition of Single Inverter
DEFINE INV1 (.1 2 3 )
M1 3 2 1 1 PSHARE L=3.0U W=5.4U AS=64.8P PS=34.8U AD=42.12P PD=26.4U
M2 3 2 0 0 NSHARE L=3.0U W=3U AS=57.6P PS=34.8U AD=36.6P PD=26.4U
END INV1

; Definition of Dynamic CMOS AND3 gate for ROW decoding
X21 100 127 150 151 EXOR
X22 100 151 152 153 OR
X23 100 154 155 INV1
X24 100 156 157 INV1
X25 100 153 154 155 156 157 158 159 303 304 LATCH

;CONSTRUCTED COMPLEX MODEL.
MODEL NMOS VTO=0.7 KP=50U GAMMA=1.1 PHI=0.6 LAMBDA=0.01 \ CGSO=3.0E-10 CGDO=3.0E-10 CGBO=5.0E-10 TOX=500E-10

MODEL NPMDM D IS=3E-9 CJO=0.73E-10 M=0.33
MODEL NAADOM D IS=3E-9 CJO=0.77E-4 M=0.5

DEFINE NQSHARE ( D G S B )
PARAMETERS W=10U L=3U AS=1.0 AD=1.0 PS=1.0 PD=1.0

DEVICE D G S B NMOD W=W L=L
DPS B S NPMDM AREA=PS
DPD B D NPMDM AREA=PD
DPS B S NAADOM AREA=AS
DPD B D NAADOM AREA=AD

END NQSHARE

;CONSTRUCTED COMPLEX MODEL.
MODEL PMOD PMOS VTO=0.8 KP=16E-6 GAMMA=0.6 PHI=0.6 LAMBDA=0.03 \ CGSO=2.5E-10 CGDO=2.5E-10 CGBO=5.0E-10 TOX=500E-10

MODEL PPMDM D IS=1E-9 CJO=2.2E-10 M=0.33
MODEL PAPMDM D IS=3E-9 CJO=1.4E-4 M=0.5

DEFINE PQSHARE ( D G S B )
PARAMETERS W=10U L=3U AS=1.0 AD=1.0 PS=1.0 PD=1.0

DEVICE D G S B PMOD W=W L=L
DPS S B PPMDM AREA=PS
DPD D B PPMDM AREA=PS
DPS S B PAPMMD AREA=AS
DPD D B PAPMMD AREA=AD

END PQSHARE

*DEFINITION of a Dynamic ROM *

:Definition of Single Inverter
DEFINE INV1 ( 1 2 3 )
M1 3 2 1 1 PQSHARE L=3.0U W=5.4U AS=64.8P PS=34.8U AD=42.12P PD=26.4U
M2 3 2 0 0 NQSHARE L=3.0U W=3U AS=57.6P PS=34.8U AD=36.6P PD=26.4U

END INV1

*Definition of a Dynamic CMOS AND3 gate for ROW decoding
DEFINE DYAND3 (1 2 4 6 8 30 10)
M1 3 2 1 1 PQSHARE L=3U W=5.4U AS=46.8P PS=34.5U AD=42.12P PD=26.4U
M2 3 4 5 0 PQSHARE L=3U W=5.4U AS=42.12P PD=26.4U AS=8.1P PS=13.8U
M3 5 6 7 0 PQSHARE L=3U W=5.4U AD=8.1P PD=13.8U AS=8.1P PS=13.8U
M4 7 8 9 0 PQSHARE L=3U W=5.4U AS=8.1P PS=13.8U AD=8.1P PD=13.8U
M5 9 2 0 0 PQSHARE L=3U W=5.4U AD=8.1P PD=13.8U AS=64.8P PS=34.8U
RA6 3 30 R R=0.250K
CA6 30 0 C C=0.023PF
M6 10 30 1 1 PQSHARE L=3U W=16.2U AS=194.46P PS=56.4U AD=126.36P PD=48.0U
M7 10 30 0 0 PQSHARE L=3U W=5.4U AS=42.12P PS=26.4U AD=64.8P PD=34.8U
END DYAND3

;Definition of a STATIC CMOS AND2 gate for column decoding

DEFINE AND2 (1 2 3 6)
M1 4 2 1 1 PQSHARE L=3U W=5.4U AS=64.6P PS=34.8U AD=8.1P PD=13.8U
M2 4 3 1 1 PQSHARE L=3U W=5.4U AS=42.12P PS=26.4U AD=55.08P PD=31.2U
M3 4 3 5 0 PQSHARE L=3U W=5.4U AS=55.08P PS=31.2U AD=42.12P PD=26.4U
M4 5 2 0 0 PQSHARE L=3U W=5.4U AD=42.12P PS=34.0U AD=8.1P PD=13.8U
RA7 4 40 R R=0.210K
CA7 40 0 C C=0.0214PF
M5 6 40 1 1 PQSHARE L=3U W=16.2U AS=194.46P PS=56.4U AD=64.8P PD=48.0U
M6 6 40 0 0 PQSHARE L=3U W=5.4U AS=42.12P PS=26.4U AD=64.8P PD=34.8U
END AND2

;------------------------------------------------------

DEFINE ROMCELL (3 2)
M1 2 3 0 0 PQSHARE L=3U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
END ROMCELL

;------------------------------------------------------

DEFINE COLTRANS (2 3 4)
M1 2 3 4 0 PQSHARE L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
END COLTRANS

;------------------------------------------------------

DEFINE PTRANS (1 2 3)
M1 2 3 4 1 PQSHARE L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
END PTRANS

;------------------------------------------------------

;******************************************************************

;Definition of an EXCLUSIVE 2-input OR circuit

;******************************************************************

DEFINE EXOR (1 20 40 8)
RA1 20 2 R R=0.15K
CA1 20 0 C C=0.02PF
RA2 40 4 R R=0.15K
CA2 40 0 C C=0.02PF
M1 3 2 1 1 PQSHARE L=3.0U W=5.4U AS=64.8P PS=34.8U AD=42.12P PD=26.4U
M2 5 4 1 1 PQSHARE L=3.0U W=5.4U AS=64.8P PS=34.8U AD=42.12P PD=26.4U
M3 7 4 3 1 PQSHARE L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
M4 7 5 2 1 PQSHARE L=3.0U W=5.4U AS=42.12P PS=26.4U AD=42.12P PD=26.4U
M5 8 7 1 1 PQSHARE L=3.0U W=5.4U AS=64.8P PS=34.8U AD=42.12P PD=26.4U
M6 8 7 0 0 PQSHARE L=3.0U W=3U AS=64.8P PS=34.8U AD=36.36P PD=26.4U
M7 7 4 2 0 PQSHARE L=3.0U W=3U AS=36.36P PS=26.4U AD=36.36P PD=26.4U
M8 7 5 3 0 PQSHARE L=3.0U W=3U AS=36.36P PS=26.4U AD=36.36P PD=26.4U
M9 3 2 0 0 PQSHARE L=3.0U W=3U AS=64.8P PS=34.8U AD=36.36P PD=26.4U
M10 5 4 0 0 NQSHARE L=3.0U W=3U AS=64.8P PS=34.8U AD=36.36P PD=26.4U
END EXOR

;***************************************************************
;DEFINITION OF A 2-INPUT OR CIRCUIT *
;***************************************************************
DEFINE OR ( 1 20 40 6 )
RA1 20 2 R R=0.3K
CA1 2 0 C C=0.02PF
RA2 40 4 R R=0.3K
CA2 4 0 C C=0.02PF
M1 3 2 1 1 PQSHARE L=3.0U W=5.4U AS=64.8P PS=34.8U AD=29.26P PD=21.6U
M2 5 4 3 1 PQSHARE L=3.0U W=5.4U AS=25.92P PS=20.4U AD=42.12P PD=26.4U
M3 5 2 0 0 NQSHARE L=3.0U W=3.0U AS=57.6P PS=39.6U AD=21.7P PD=11.4U
M4 5 4 0 0 NQSHARE L=3.0U W=3.0U AS=36.36P PS=26.4U AD=21.7P PD=11.4U
RA3 5 50 R R=0.3K
CA3 50 0 C C=0.02PF
M5 6 50 1 1 PQSHARE L=3.0U W=5.4U AS=64.8P PS=34.8U AD=42.12P PD=26.4U
M6 6 50 0 0 NQSHARE L=3.0U W=3.0U AS=57.6P PS=38.6U AD=36.36P PD=26.4U
END OR

;***************************************************************
;DEFINITION OF A LATCH *
;***************************************************************
DEFINE LATCH ( 1 40 60 20 130 90 8 11 3 10 )
RA1 40 4 R R=0.013K
CA1 4 0 C C=0.16PF
RA2 20 2 R R=0.013K
CA2 2 0 C C=0.013PF
RA3 60 6 R R=0.18K
CA3 6 0 C C=0.013PF
RA5 90 9 R R=0.18K
CA5 9 0 C C=0.02PF
RA6 130 13 R R=0.18K
CA6 13 0 C C=0.02PF
M1 3 2 1 1 PQSHARE L=3.0U W=5.4U AS=77.76P PS=45.0U AD=16.2P PD=24.6U
M2 8 4 3 1 PQSHARE L=3.0U W=5.4U AS=16.2P PS=24.6U AD=55.08P PD=36.6U
M5 8 4 5 0 NQSHARE L=3.0U W=3.0U AS=8.1P PS=13.8U AD=42.12P PD=26.4U
M6 5 6 0 0 NQSHARE L=3.0U W=3.0U AS=64.8P PS=34.8U AD=8.1P PD=13.8U
RA7 8 80 R R=0.3K
CA7 80 0 C C=0.02PF
M9 10 9 1 1 PQSHARE L=3.0U W=5.4U AS=55.08P PS=36.6U AD=16.2P PD=24.6U
M10 11 80 10 1 PQSHARE L=3.0U W=5.4U AS=16.2P PS=24.6U AD=55.08P PD=36.6U
M11 11 80 14 0 NQSHARE L=3.0U W=3.0U AS=42.12P PS=26.4U AD=8.1P PD=13.8U
M12 14 13 0 0 NQSHARE L=3.0U W=3.0U AS=29.16P PS=21.6U AD=8.1P PD=13.8U
END LATCH

;Input Waveforms
;**************************
V1A 102 0 PWL PERIOD=30NS T0=0 V0=0 T1=30NS V1=0
V2A 103 0 PWL PERIOD=30NS T0=0 V0=0 T1=30NS V1=0
V3A 104 0 PWL PERIOD=30NS T0=0 V0=0 T1=30NS V1=0
V4A 119 0 PWL PERIOD=30NS T0=0 V0=0 T1=30NS V1=0
V5A 120 0 PWL PERIOD=30NS T0=0 V0=0 T1=30NS V1=0
V7A 150 0 PWL PERIOD=30NS T0=0 V0=5 T1=30NS V1=5
V8A 152 0 PWL PERIOD=30NS T0=0 V0=0 T1=30NS V1=0
V9A 154 0 PWL PERIOD=30NS T0=0 V0=0 T1=11.5NS V1=0 T2=12NS V2=5 \T3=29.5NS V3=5
V10A 156 0 PWL PERIOD=30NS T0=0 V0=5 T1=10.5NS V1=5 T2=11NS V2=0 \T3=29.5NS V3=0
IC 131=5 127=0 115=0 116=0 114=5 125=0 126=0 190=5
IC 151=0 152=0 158=5 159=0 155=0 157=0 303=0 304=0

RELAX2 OPTIONS STOP=50NS WINDOWSIZE=50NS DODC=1 DOTRAN=1

PLOT 102 127 151 153 154 156 158 159
APPENDIX 7

FORTRAN program for generation of various layout options for a ROM
APPENDIX 7

Part I: A program to find an optimal layout configuration of a ROM

Bayoumi has given an algorithm for finding out various layout options of a ROM for RNS architectures [7]. This algorithm requires a modulus and the space between two adjacent bits. Based on these two inputs, the algorithm calculates memory array capacity, area of the ROM, length of the ROM, width of the ROM, time delay of the ROM, number of words, word length of the ROM and the area*time product. The algorithm generates several layout options with respect to these parameters. A designer can select one of these options based on his requirements or the optimal option. A FORTRAN program, Block.for, has been written based on this algorithm. This program generates various ROM layout options for a given modulus. The spacing between two adjacent bits is assumed as 7 microns based on the CMOS layout rules.

Part II: Use of program Block.for to generate layout options of a ROM

Using the program Block.for, layout parameters for a ROM based on a modulus 7 are generated. The various layout parameters generated are given in the file Model.Out. The area*time efficiency occurs in option 4 where it has the lowest value. The option suggests the organization of a ROM should have 49 locations with a word length of 3 bits. It also suggests that the ROM should have 12 horizontal bits and 11 vertical bits, and the layout must be square shaped. The program Block.for and the results are given in the following pages.
The quality of this microfiche is heavily dependent upon the quality of the thesis submitted for microfilming.

Please refer to the National Library of Canada target (sheet 1, frame 2) entitled:

NOTICE
Program to find out the various layout options for a ROM in the
in the block level
Written by: P.U.R. RAJA for his Master's thesis
Department of Electrical Engineering
University of Windsor

program ATOPT
This program gives the ROM designer various layout options
for area*time delay, area and time delay minimization. This
program incorporates the Bayoumi's model (M.A. Bayoumi, Ph.D thesis
University of Windsor, 1985).

c modulus of the ROM
    TYPEA, 'Give the base modulus:'
    READA, mod
    mod=7
    level=1
    n=1
    alpha- the spacing between two adjacent cells
    alpha=7e-6
    call atmodel(mod, n, alpha)
    stop
end subroutine atmodel(mod, n, alpha)
integer width, lb(100), wb, hab
real modi
OPEN(UNIT=1, FILE='MODEL.OUT', TYPE='NEW')
print*, 'The results file can be found in the file Model.out'
write(1,*) 'Various layout options of the ROM for modulus: ', mod
write(1,*) '*******************************
modi=mod-1.0
width - the word length
width=modi
width=int(width)+1
The above line makes width an integer
write(1,*) 'Word length: ', width
acp- the capacity of the ROM array
acp=width*(modi**2)
write(1,*) 'Array capacity: ', acp
Algorithms described in the report
hab=0.0
i=1
100
alpha=1*width
hb-number of bits in the vertical axis
wbnumber of bits in the horizontal axis
wb=alpha**n
if(wb.gt.modi**2) then
    type= 'Termination'
close(UNIT=1)
return
else
    hb(i)=int(acp/wb)+1
endif
    if(hb(1).eq.hab) then
      i=i+1
    elseif(hb(1).lt.width) then
      n=n+1
    else
      beta=integer((acp/wb)**(1/n))+1
      yal=y(alpha)
      ybet=y(beta)
    end if
  c  h - height of the ROM
  c  w - width of the ROM
  c  s - "
  c  (labs(int(yal)))+1.0)\times hb(1)
  c  w = (alpha1/(1-alpha-1))+(2*alpha-1)/(alpha-1))\times wb
  c  ft = n*alpha*alpha*alpha
  c  s = n*beta*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha*alpha
  c  a-area; t-time delay; at-area*time delay
  c  st=n*beta*alpha*alpha
  c  if(ft.gt.st) then
  c    t=ft
  c  else
  c    t=st
  c  endif
  c  endif
  c  hab=hb(1)
  c  a=w*h
  c  at=w*t
  c  n=n+1 !not executed- because n=1    
  i=i+1
  write(1,'******','option@),(i-1)', '************
  write(1,'N=',n
  write(1,'\times wb=', wb, 'HB=', hab
  write(1,'\times \alpha ', 'height=', h
  write(1,'\times \alpha ', 'time=', t
  write(1,'\alpha ', 'area=', at
  goto 100
  end
  c  function to generate logarithm of x for base 2
  function y(x)
    y=alog(x)/alog(2.0)
  return
  end
Various layout options of the ROM for modulus:

<table>
<thead>
<tr>
<th>Option</th>
<th>Word Length</th>
<th>Array Capacity</th>
<th>Width</th>
<th>Height</th>
<th>Area</th>
<th>Time</th>
<th>Area * Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>147,0000</td>
<td>23.2502</td>
<td>53.08240</td>
<td>1234.167</td>
<td>3.50E+004</td>
<td>0.4319585</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
<td>17.7604</td>
<td>30.29531</td>
<td>538.0461</td>
<td>1.75E+004</td>
<td>9.41E-006</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td></td>
<td>19.68756</td>
<td>24.63684</td>
<td>485.0393</td>
<td>1.19E+004</td>
<td>5.77E-006</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td></td>
<td>19.93397</td>
<td>20.85426</td>
<td>415.7081</td>
<td>9.10E+003</td>
<td>3.78E-005</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td></td>
<td>22.72970</td>
<td>18.14822</td>
<td>412.5035</td>
<td>1.05E+004</td>
<td>4.33E-005</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td></td>
<td>25.59874</td>
<td>19.68756</td>
<td>503.9768</td>
<td>1.26E+004</td>
<td>6.35E-005</td>
</tr>
<tr>
<td>7</td>
<td>21</td>
<td></td>
<td>28.50765</td>
<td>18.93883</td>
<td>539.9016</td>
<td>1.47E+004</td>
<td>7.93E-005</td>
</tr>
<tr>
<td>8</td>
<td>24</td>
<td></td>
<td>31.45000</td>
<td>20.31391</td>
<td>575.8280</td>
<td>1.69E+04</td>
<td>9.15E-004</td>
</tr>
<tr>
<td>N</td>
<td>Wb</td>
<td>Hb</td>
<td>Width</td>
<td>height</td>
<td>Area</td>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
<td>--------</td>
<td>--------</td>
<td>-------</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>24</td>
<td>7</td>
<td>29.30830</td>
<td>18.27783</td>
<td>535.6920</td>
<td>1.6800000E-04</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>27</td>
<td>6</td>
<td>32.27238</td>
<td>17.76004</td>
<td>573.1589</td>
<td>1.8900000E-04</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td>5</td>
<td>35.24397</td>
<td>17.50003</td>
<td>616.7707</td>
<td>2.1000000E-04</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>33</td>
<td>0</td>
<td>35.24397</td>
<td>17.50003</td>
<td>616.7707</td>
<td>2.1000000E-04</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>39</td>
<td>4</td>
<td>44.18587</td>
<td>20.88892</td>
<td>922.9951</td>
<td>2.7300000E-04</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>42</td>
<td>0</td>
<td>44.18587</td>
<td>20.88892</td>
<td>922.9951</td>
<td>2.7300000E-04</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>48</td>
<td>4</td>
<td>53.15018</td>
<td>20.88892</td>
<td>1110.250</td>
<td>3.3600000E-04</td>
<td></td>
</tr>
</tbody>
</table>

**********option# 9 **********
N = 1
Wb = 27 Hb = 6
Width = 32.27238 height = 17.76004
Area = 573.1589 Time = 1.8900000E-04
Area*Time = 0.1083270

**********option# 10 **********
N = 1
Wb = 30 Hb = 5
Width = 35.24397 height = 17.50003
Area = 616.7707 Time = 2.1000000E-04
Area*Time = 0.1295218

**********option# 12 **********
N = 1
Wb = 33 Hb = 0
Width = 35.24397 height = 17.50003
Area = 616.7707 Time = 2.1000000E-04
Area*Time = 0.1295218

**********option# 13 **********
N = 1
Wb = 39 Hb = 4
Width = 44.18587 height = 20.88892
Area = 922.9951 Time = 2.7300000E-04
Area*Time = 0.2519777

**********option# 15 **********
N = 1
Wb = 42 Hb = 0
Width = 44.18587 height = 20.88892
Area = 922.9951 Time = 2.7300000E-04
Area*Time = 0.2519777

**********option# 16 **********
N = 1
Wb = 48 Hb = 4
Width = 53.15018 height = 20.88892
Area = 1110.250 Time = 3.3600000E-04
Area*Time = 0.3730439

**********option# 17 **********
N = 1
Wb = 48 Hb = 4
Width = 53.15018 height = 20.88892
Area = 1110.250 Time = 3.3600000E-04
Area*Time = 0.3730439

**********option# 18 **********
N = 1
Wb = 48 Hb = 4
Width = 53.15018 height = 20.88892
Area = 1110.250 Time = 3.3600000E-04
Area*Time = 0.3730439

**********option# 19 **********
N = 1
Wb = 48 Hb = 4
Width = 53.15018 height = 20.88892
Area = 1110.250 Time = 3.3600000E-04
Area*Time = 0.3730439

**********option# 20 **********
N = 1
Wb = 48 Hb = 4
Width = 53.15018 height = 20.88892
Area = 1110.250 Time = 3.3600000E-04
Area*Time = 0.3730439
APPENDIX 8

FORTRAN program for optimization of CMOS circuits
APPENDIX 8

Part I: An optimization program for CMOS circuits

In the transistor level optimization, width and length of individual transistor are considered. This level of optimization is particularly suited to MOS circuits, in which the delay time of the whole circuit is based on individual transistors. By changing the width and length, we can change the delay time of a MOS transistor. This is the basic idea behind several optimization algorithms. One such algorithm is given by S.M. Kang [39]. This algorithm generates area*time calculations of standard cell type CMOS circuits based on numerous parameters. The most important parameters are the width and length of each transistor. The model in this algorithm is given below:

Nomenclature

- $Ca(p), Ca(n)$: Capacitance per unit area in p- and n-diffusion (pF/μm²)
- $CL$: Load capacitance (pF)
- $Cox$: Gate capacitance per unit area (pF/μm²)
- $Cp(p), Cp(n)$: Capacitance per unit perimeter in p and n-diffusion area (pF/μm)
- $Dd$: Diffusion width in drain region (Um)
- $fo$: Number of fan-outs
- $Kox$: Relative permittivity of silicon di-oxide (3.9)
- $Lp, Ln$: Gate channel length in p-channel and n-channel transistor (μm)
tox                          Gate oxide thickness (Å)
Vth(p),Vth(n)               Threshold voltage of p- and n-channel transistors
Wp, Wn                       Gate channel width in p and n-channel transistors (µm)
ε0                          Permittivity of free space
µp, µn                       Mobility of holes and electrons.
vn                           The n-channel transistor propagation delay
vp                           The p-channel transistor propagation delay
Cw                           The wiring capacitance
Vin                          Input to the circuit in volts
Vout                         Output of the circuit in volts
Vdd                          The supply rail voltage
Vss                          The ground voltage
Vg                           Gate to Source voltage
Vds                          Drain to Source voltage
Ci                           Internal wire capacitance
Co                           Output node capacitance

The functional relationship between propagation delays and important circuit parameters can be described by the following equations.

\[ T_n = f(W_n, C_o + C_w + C_i; V_i(t)) \]
\[ T_p = g(W_p, C_o + C_w + C_i; V_i(t)) \]

Where \( V_i(t) \) denotes the input waveform and \( C_o, C_w, \) and \( C_i \) are the implicit functions of \( W_n \) and \( W_p \).

Let us derive an analytical expression for \( T_n \) based on the Inverter shown in Fig. 8.1 assuming that the input voltage \( V_i(t) \) has an ideal pulse waveform.
Fig. 8.1 A CMOS Inverter and its input/output waveform pair
The state equations:

\[ \text{NFET in saturation (Vg-Vout \leq Vth(n))} \]

CL \[ \frac{dVout}{dout} = -Bn (Vg-Vth(n))^2 \] \[ \longrightarrow (2.a) \]

\[ \text{NFET in linear region (Vg-Vout \geq Vth(n))} \]

CL \[ \frac{dVout}{dt} = -Bn(2(Vg-Vth(n))Vout - Vout^2) \] \[ \longrightarrow (2.b) \]

where

\[ Bn = \frac{\mu_n \lambda_n C_{ox}}{2L_n} \]

\[ C_{ox} = \frac{K_{ox} C_o}{t_{ox}} \]

The time interval for the NFET to remain in saturation denoted by \( ts \) can be obtained from (2.a)

\[ \int_{Vdd}^{Vth(n)} \frac{dVout}{Vdd - Vth(n)^2} dt = -Bn \]

\[ \int_{Vdd}^{Vout} \frac{dVout}{Vdd - Vth(n)} = Bn(2(Vdd-Vth(n))Vout - Vout^2) \]

\[ \longrightarrow (3.0) \]

\[ \int_{Vdd}^{Vout} \frac{dVout}{Vdd - Vth(n)} = 0 \]

Solving (3) for \( ts \) yields

\[ ts = \frac{Vth(n)CL}{Bn(Vdd-Vth(n))^2} \]

\[ \longrightarrow (4.0) \]

The time interval for the NFET to stay in the linear region until it reaches one half of \( Vdd \) is obtained from (2.b)

\[ \frac{1}{2} Vdd \]

\[ \int_{Vdd}^{Vth(n)} \frac{dVout}{2(Vdd-Vth(n))Vout - Vout^2} \]

\[ \frac{1}{2} Vdd \]

\[ \int_{Vdd}^{Vout} \frac{dVout}{Vdd - Vth(n)} = -Bn \int_{Vdd}^{Vout} dt \] \[ \longrightarrow (5) \]

\[ \int_{Vdd}^{Vout} \frac{dVout}{Vdd - Vth(n)} = ts \]

and, therefore, the propagation delay in the n-channel transistor \( T_n \) has the form
\[ T_n = \frac{CL}{\text{Un} \times \text{Wn} \times \text{Kox} \times \text{Co}} \times \text{Ln} \times \text{tox} \quad \text{(6)} \]

where

\[ A = \left[ \frac{V_{\text{th}(n)}}{2} + \frac{1}{\ln\left(\frac{1.5 \times V_{\text{dd}} - 2V_{\text{th}(n)}}{0.5 \times V_{\text{dd}} - V_{\text{th}(n)}}\right)} \right] \]

Similarly, \( T_p \) can be expressed by (6) with subscript \( n \) replaced by \( p \).

For \( m \)-input NAND and NOR gates, the propagation delays in the p-channel and n-channel can be approximated as follows:

\[
\begin{align*}
T_p \ (\text{NAND} \ m) &= T_p \\ T_p \ (\text{NOR} \ m) &= mT_p \\ T_n \ (\text{NAND} \ m) &= mT_n \\ T_n \ (\text{NOR} \ m) &= T_n
\end{align*}
\]

\text{(7)}

For an Inverter, the propagation delays can be approximated as follows:

\[
\begin{align*}
T_p \ (\text{INV}) &= T_p \\ T_n \ (\text{INV}) &= T_n
\end{align*}
\]

Based on the above approximation, the mean delay in an average poly cell, which uses all the gates above can be expressed as by:

\[
\bar{T} = \frac{1}{N} \sum_{j=1}^{N=5} \frac{1}{2} (T_p(j) + T_n(j))
\]

Where \( T_n(j) \) and \( T_p(j) \) denote the propagation delays in the \( j \)th polycell.

\( \bar{T} \) can be generalized by the following expression:
\[ \bar{T} = \left( \frac{Kn}{Un} + \frac{Kp}{Wp} \right) \cdot \frac{CL}{Wn} \rightarrow (8) \]

Where

\[ Kz = \frac{1.6 \cdot Lz \cdot tox \cdot Vth(z)}{Kox \cdot Eo} \quad \frac{1}{\ln\left( \frac{1.5 \cdot Vdd - 2 \cdot Vth(z)}{(Vdd - Vth(z))^2} \right)} + \frac{0.5 \cdot Vdd}{2(Vdd - Vth(z))} \]

for \( z = n \) or \( p \)

The load capacitance \( CL \) can be expressed by:

\[ CL = Co + Cw + Ci \]

\[ = \left( \frac{Ca(p) \cdot Wp + Ca(n) \cdot Wn \cdot Dd}{Cw(Wp + Wn) \cdot Fo(Q)} + Cw(Wp + Wn) \cdot Fo(Wp + Wn) \cdot Cox \right) \rightarrow (9) \]

The effective chip area for an average polycell can be represented by

\[ A = Wp + Wn + S \rightarrow (10) \]

where \( S \) denotes the spacing between \( p \)-channel and \( n \)-channel thinoxide regions specified by the design rules.

From equations (8) to (10), the product of propagation delay and chip area, henceforth called the objective function, can be written as:

\[ H(Wp, Wn) = A \cdot \bar{T} = (Wp + Wn + S) \left( \frac{Kn}{Un} + \frac{Kp}{Wp} \right) \left( \frac{Xp \cdot Wp + Xn \cdot Wn + Xo}{Wp} \right) \rightarrow (11) \]

\[ Xz = Ca(z) \cdot Dd + 2 \cdot Cp(z) + Fo \cdot Cox \cdot L(z) \text{ for } z = p \text{ or } n \]

\[ Xo = 2 \cdot Dd \left( Cp(p) + Cp(n) \right) + Cw \]

A FORTRAN program (Optimize.for) which uses the aforementioned algorithm has been developed. This program
calculates and finds out the optimal area*time value of a specified CMOS circuit's p-MOS and n-MOS transistors' widths and lengths. A CMOS circuit can be specified by the following inputs:

1) The number of p-channel transistors in the circuit
2) The number of n-channel transistors in the circuit
3) The width of the n-channel transistors in the circuit

The number of transistors of any type (p- or n-channel) can be calculated as follows. For example, let us calculate the number of p-channel transistors and the number of n-channel transistors in a 3-input AND gate. This gate consists of a 3-input NAND gate and an Inverter. In the 3-input NAND gate, there are 3 p-channel transistor connected in a parallel form, and 3 n-channel transistors connected as a series. The algorithm requires us to take the number of p-channel transistors in the circuit as 1, and the number of n-channel transistors in the circuit as 3. Hence, we can conclude that there are 2 p-channel transistors and 4 n-channel transistors, including the transistors in the Inverter. This type of simplicity provides feasibility to optimize larger circuits. The program Optimize.for is presented in the following pages.

Part II: Optimization of a 2-input EX-OR and OR2 gate

Using program optimize.for, a circuit consisting of a series connected EX-OR2 and OR2 gates is optimized. The file opt.out shows the results given by the program. The optimum widths and lengths for the area*time efficiency are: 1) p-channel transistor width is 15 microns, length is 3 microns, and 2) n-channel transistor width is 3 microns and the length is 3 microns. These
vaules have been verified using extensive SPICE simulations. The results are given in the following pages.
************

OPTIMIZATION PROGRAM FOR CMOS CELL LAYOUTS
WRITTEN BY P.V.R. RAJA FOR HIS MASTER'S THESIS

Objective of this program:
-------------------------------
The main goal is to write the simplest program for the
optimization of CMOS standard cell type circuits.

The main feature of this program is the incorporated model
for area/time efficiency.

How to work with the program:
-----------------------------
This program works based on two inputs. The inputs are:
1. the varying P-channel transistor width, and 2. the fixed
width of N-channel transistor. The program starts with
asking the number of P-channel transistors and the number
of N-channel transistors. The number of transistors can be
calculated in the following ways: 1. When there is a set of
parallel transistors in the circuit, the whole set is
considered a single transistor. On the other hand, if the
set consists of N number of transistors in series, then
the answer for this question should be N.

Example:
-------
In a 2-input AND gate, there are 2 P-channel transistors
(one in the NAND portion and the other in the Inverter)
and there are 3 N-channel transistors.

The program finds out the optimal P-channel width of for
initial N-transistor width. Using the values given by this program
in SPICE simulation is a way to verify the accuracy of this
program.

Using the program, circuits of any size can be
optimized very easily.

*************************************************************************

PROGRAM ATOPT
DIMENSION A(50)
REAL LM,LP,KM,KP
OPEN(UNIT=5,TYPE='W',FILE='OPT.OUT')
TYPEA,'NUMBER OF P-TRANSISTORS IN THE CIRCUIT:'
READA,CONTP
TYPEA,'NUMBER OF N-TRANSISTORS IN THE CIRCUIT:'
CONTA

C Just an approximation
C Example: for NAND 0 of P trans=1, 0 of N trans=n
READA,CONSTM
TYPEA,'Give the width for N channel transistor:'
READA,NN
WRITE(5,'(4,1X,A,F12.4)') 'NDEF' , 'PDEF' , 'ADEF' , 'NDEF',


c increment the width by 1 micron
   DEL=1.E-6
   WP=WP+DEL

   c set the initial width of the P trans equal to 6 microns
   WPPI=6.E-6
   WP=WPPI

   c ARETIM is the area*time parameter
   CALL MODEL(CONSTP,CONSTN,WP,KN,T,ARETIM)
   A(1)=ARETIM

   c do simple one dimensional search by increasing the width of P
   c P width should be increased to increase the speed
   30  I=I+1
       K=I-1
       WP=WP+(I-1)*1E-6
       CALL MODEL(CONSTP,CONSTN,WP,KN,T,ARETIM)
       WRITE(5,*) WP,ARETIM,T
       A(I)=ARETIM

   c if there is no convergence in 50 iterations, exit.
   IF(I.GT.50) THEN
       GO TO 50
   ELSE IF(A(I).GT.A(K)) THEN
       GO TO 40
   ELSE
       GO TO 30
   ENDIF

   c print out the value of the P Trans. width at optimum
   40  TYPE*, 'THE OPTIMUM AREA*TIME OCCURS AT:
       PRINT*, (WP-DEL)
       WRITE(5,*) ' The optimum Area*Time occurs at WP=', (WP-DEL)
       GO TO 100

   50  TYPE*, 'NO COVERAGE IN 50 ITERATIONS'
       CLOSE(UNIT=5)

   100  TYPE*, 'The output file is OPT.OUT'
       STOP

C******************************************************************************
C SUBROUTINE MODEL(CONSTP,CONSTN,WP,KN,T,ARETIM)
C REAL LN,LP,KN,WP
C c VDD - the power supply in Volts
VDD=5.0

   c CAP,CAN: capacitance per unit area in p and n diffusion(pF/(micron)**2)
   CAP=1.0E-4
   CAN=1.0E-4

   c CPW,CPP: capacitance per unit perimeter in p and n diffusion(pF/micron)
   CPW=9.0E-10
   CPP=8.0E-10

   c CW - the wiring capacitance
   CW=25.0E-10

   c TOX - Thickness of the gate oxide( Å )
   TOX=5000E-8

   c AKOX - relative permittivity of the silicon di oxide
   AKOX=3.9

   c VTHN,VT- threshold voltage of N and P transistors respectively (volts)
   VTHN=0.7
VTHP = -0.8

C DD - diffusion width in the drain area (microns)
    DD = 7E-6

C FO the number of fanouts
    FO = 0

C S - the distance between the P and N diffusion (microns)
    S = 1E6-6

C LP, LN - lengths of P and N channel transistors (microns)
    LP = 3.0E-6
    LN = 3.0E-6

C AMUP, AMUN - mobility of minority carriers in p and n channel transistors
    AMUP = 0.025
    AMUN = 0.060

C EPSIL - permittivity of free space
    EPSIL = 8.854E-14

C CALCULATION begins

AKZP4 = ALOG((1.5*VDD-2.0*VTHP)/(0.5*VDD))
AKZP3 = 1/(2*(VDD-VTHP))
AKZP2 = VTHP/(2*(VDD-VTHP))
AKZP1 = CONSTP*LP/TOX/(AKOX*EPSIL)
KP = AKZP1*(AKZP2+AKZP3+AKZP4)

COX = AKOX*EPSIL/TOX

CL = (CAP*WP+CAP*WN)/DD+(2.0*(WP+DD)*CPP)+(2*N+DD)*CPN+
1.0*(FO*(WP+LN+LN*COX)
AKZN4 = ALOG((1.5*VDD-2.0*VTHN)/(0.5*VDD))
AKZN3 = 1/(2*(VDD-VTHN))
AKZN2 = VTHN/(2*(VDD-VTHN))
AKZN1 = CONSTN*LN/TOX/(AKOX*EPSIL)
KN = AKZN1*AKZN2*AKZN3*AKZN4

C Time delay calculation

T = ((KN1/(AMUN*WN)) + (KP*1/(AMUP*WP)))*CL
ALPHP = (CAP*DD)(2*CPP)(FO*COX*LP)
ALPHN = (CAP*DD)(2*CPN)(FO*COX*LN)
ALPHO = 2*DD*CPP*CPN*CW

C Print line

PRINT, KH, KP, AMUN, AMUP, KN, WP

C Area*time delay computation

ARETIM = (KP*WP-S)*(KN1/(AMUN*WN))*KP/(AMUP*WP)

1 = (ALPHP*WP+ALPHN*WN+ALPHO)
RETURN
END
<table>
<thead>
<tr>
<th>WP</th>
<th>ARETIM</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7.000000E-06</td>
<td>2.1317383E-04</td>
<td>8.198993</td>
<td></td>
</tr>
<tr>
<td>8.000000E-06</td>
<td>2.0435252E-04</td>
<td>7.568612</td>
<td></td>
</tr>
<tr>
<td>9.000000E-06</td>
<td>1.9819282E-04</td>
<td>7.078315</td>
<td></td>
</tr>
<tr>
<td>1.000000E-05</td>
<td>1.9389627E-04</td>
<td>6.686079</td>
<td></td>
</tr>
<tr>
<td>1.100000E-05</td>
<td>1.9095476E-04</td>
<td>6.365159</td>
<td></td>
</tr>
<tr>
<td>1.200000E-05</td>
<td>1.8902947E-04</td>
<td>6.097725</td>
<td></td>
</tr>
<tr>
<td>1.300000E-05</td>
<td>1.8788596E-04</td>
<td>5.871436</td>
<td></td>
</tr>
<tr>
<td>1.400000E-05</td>
<td>1.8715667E-04</td>
<td>5.677474</td>
<td></td>
</tr>
<tr>
<td>1.500000E-05</td>
<td>1.8731873E-04</td>
<td>5.509375</td>
<td></td>
</tr>
<tr>
<td>1.600000E-05</td>
<td>1.8768008E-04</td>
<td>5.362288</td>
<td></td>
</tr>
</tbody>
</table>

The optimum Area-Time occurs at WP = 1.500000E-05
$W_n = 3 \mu m$

Fig. 8.2 Area*Time optimization curve
APPENDIX 9

Testing of fabricated relevant designs
In this section we discuss the testing of fabricated designs which are relevant to the thesis. These designs have been fabricated by the Canadian Microelectronics Corporation. The fabricated designs are: 1) the 24*5 static ROM discussed in Section 3.4, 2) the 3-to-7 dynamic CMOS decoder discussed in Section 3.5, and 3) the 24*5 dynamic ROM discussed in section 3.5.1. The completed designs for fabrication are shown in Figures 9.1, 9.2, and 9.3 respectively. The multiproject bonding diagrams of these designs obtained from the Canadian microelectronics corporation are shown in Figures 9.4, 9.5, and 9.6.

Testing of these designs were carried out using the computer aided IC tester. The test station consists of several items of equipment including an IBM personal computer and a Hewlet-Packard 8212A data analyzer. The type of testing carried out is functional testing. In this type of testing, test vectors are fed to the station through the computer. The test software generates input signals to the IC chip under test according to the test vectors. The data analyzer displays the output of the IC chip in the form of waveforms.

The test results for the 24*5 static ROM are given in Table 9.1. Test vectors for input address are given in this table. The input address represents two numbers. A multiplier look-up table is programmed in the ROM using the function (A*B)MOD31. The least three significant bits represent the number Add2 and the most two significant digits represent the number Add1. Hence, the
number of bits in the input address is 5. The output data can be obtained from the function \((\text{Add1} \times \text{Add2}) \mod 31\). The expected data and the obtained data are given in Table 9.1; the expected output waveform is given in Fig. 9.7. The output data displayed by the 8212A data analyzer is shown in the Fig. 9.8. We can note that the expected waveform and the obtained waveform are the same.

Test results for the 3-to-7 Domino CMOS logic decoder are shown in Table 9.2. The input vector (the input to the chip) has three bits. The input addresses (input test vectors), the expected data, and the obtained data are given in the Table. The expected output waveforms are shown in Fig. 9.9 and the obtained waveforms are shown in Fig. 9.10. We can note that both the expected and obtained waveforms and data are the same. The clock applied for operation of this dynamic decoder is not shown in this figure.

Testing procedure for the 24×5 dynamic ROM is the same as that for the 24×5 ROM discussed above since the stored look-up table for both the memories are the same. The input addresses, the expected data, and the obtained data are given in Table 9.3. The expected waveforms and the obtained waveforms are shown in Fig. 9.11, and Fig. 9.12 respectively. We can note that both the waveforms and the data are the same.
Fig. 9.1 Test chip for the 24x5 static ROM
Fig. 9.2 Test chip for the 3-to-7 Domino CMOS decoder
Fig. 9.3 Test chip for the 24 x 5 dynamic ROM
**TEST RESULTS OF A 24*5 STATIC ROM**

Test chip: Fig. 9.1

CMC multi-project bonding diagram: Fig. 9.4

Expected output: Fig. 9.7

Tested results: Figure 9.8 (Output of the HP 8182A data analyzer)

Table 9.1 Test results of the 24*5 static ROM

<table>
<thead>
<tr>
<th>Input address Test vectors</th>
<th>Expected Data (Add1*Add2) mod 31</th>
<th>Measured Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>A4 A3 A2 A1 A0</td>
<td>D4 D3 D2 D1 D0</td>
<td>D4 D3 D2 D1 D0</td>
</tr>
<tr>
<td>1) 0 0 0 0 0</td>
<td>0 0 0 0 0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>2) 0 0 0 0 1</td>
<td>0 0 0 0 0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>3) 0 0 0 1 0</td>
<td>0 0 0 0 0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>4) 0 0 0 1 1</td>
<td>0 0 0 0 0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>5) 0 0 1 0 0</td>
<td>0 0 0 0 0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>6) 0 0 1 0 1</td>
<td>0 0 0 0 0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>7) 0 1 0 0 0</td>
<td>0 0 0 0 0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>8) 0 1 0 0 1</td>
<td>0 0 0 0 1</td>
<td>0 0 0 0 1</td>
</tr>
<tr>
<td>9) 0 1 0 1 0</td>
<td>0 0 0 1 0</td>
<td>0 0 0 1 0</td>
</tr>
<tr>
<td>10) 0 1 0 1 1</td>
<td>0 0 0 1 1</td>
<td>0 0 0 1 1</td>
</tr>
<tr>
<td>11) 0 1 1 0 0</td>
<td>0 0 1 0 0</td>
<td>0 0 1 0 0</td>
</tr>
<tr>
<td>12) 0 1 1 0 1</td>
<td>0 0 1 0 1</td>
<td>0 0 1 0 1</td>
</tr>
<tr>
<td>13) 1 0 0 0 0</td>
<td>0 0 0 0 0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>14) 1 0 0 0 1</td>
<td>0 0 0 1 1</td>
<td>0 0 0 1 1</td>
</tr>
<tr>
<td>15) 1 0 0 1 0</td>
<td>0 0 1 0 0</td>
<td>0 0 1 0 0</td>
</tr>
<tr>
<td>16) 1 0 0 1 1</td>
<td>0 0 1 1 0</td>
<td>0 0 1 1 0</td>
</tr>
<tr>
<td>17) 1 0 1 0 0</td>
<td>0 1 0 0 0</td>
<td>0 1 0 0 0</td>
</tr>
</tbody>
</table>
Fig. 9.4 Bonding diagram of the fabricated 24*5 static ROM test chip
<table>
<thead>
<tr>
<th></th>
<th>18) 1 0 1 0 1</th>
<th>19) 1 1 0 0 0</th>
<th>20) 1 1 0 0 1</th>
<th>21) 1 1 0 1 0</th>
<th>22) 1 1 0 1 1</th>
<th>23) 1 1 1 0 0</th>
<th>24) 1 1 1 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1 0 1 0</td>
<td>0 0 0 0 0</td>
<td>0 0 0 1 1</td>
<td>0 0 1 1 0</td>
<td>0 1 0 0 1</td>
<td>0 1 1 0 0</td>
<td>0 1 1 1 1</td>
</tr>
</tbody>
</table>
TEST RESULTS OF A 3-TO-7 DYNAMIC CMOS LOGIC DECODER

Test chip: Fig. 9.2

CMC multiproject bonding diagram: Fig. 9.5

Expected output: Fig. 9.9

TEST results: Figure 9.10 (Output of the 8182A Data analyzer)

Table 9.2 Test results of the 3-to-7 dynamic CMOS decoder

<table>
<thead>
<tr>
<th>Input address test vectors</th>
<th>Expected Data</th>
<th>Measured Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2 A1 A0</td>
<td>D6 D5 D4 D3 D2 D1 D0</td>
<td>D6 D5 D4 D3 D2 D1 D0</td>
</tr>
<tr>
<td>1)</td>
<td>0 0 0 0 0 0 0 1</td>
<td>0 0 0 0 0 0 1</td>
</tr>
<tr>
<td>2)</td>
<td>0 0 1 0 0 0 1 0</td>
<td>0 0 0 0 1 0 0</td>
</tr>
<tr>
<td>3)</td>
<td>0 1 0 0 0 0 1 0</td>
<td>0 0 0 1 0 0 0</td>
</tr>
<tr>
<td>4)</td>
<td>0 1 1 0 0 0 1 0</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>5)</td>
<td>1 0 0 0 0 1 0 0</td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>6)</td>
<td>1 0 1 0 0 0 0 0</td>
<td>0 1 0 0 0 0 0</td>
</tr>
<tr>
<td>7)</td>
<td>1 1 0 1 0 0 0 0</td>
<td>1 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
Fig. 9.5 Bonding diagram of the fabricated 3-to-7 dynamic CMOS decoder test chip
TEST RESULTS OF A 24*5 DYNAMIC ROM

Test chip: Fig. 9.3

CMC multi-project bonding diagram: Fig. 9.6

Expected output: Fig. 9.11

Test results: Figure 9.12 (Output of the 8182A data analyzer)

Table 9.3 Test results of the 24*5 dynamic ROM

<table>
<thead>
<tr>
<th>Input address test vectors</th>
<th>Expected Data (Add1*Add2) mod 31</th>
<th>Measured Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad1</td>
<td>Ad2</td>
<td>D4</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>----</td>
</tr>
<tr>
<td>1)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>13)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>14)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>15)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>16)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>17)</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

209
<table>
<thead>
<tr>
<th></th>
<th>18) 1 0 1 0 1</th>
<th>19) 1 1 0 0 0</th>
<th>20) 1 1 0 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1 0 1 0</td>
<td>0 0 0 0 0</td>
<td>0 0 0 1 1</td>
</tr>
<tr>
<td>21) 1 1 0 1 0</td>
<td>0 0 1 1 0</td>
<td>0 0 1 1 0</td>
<td></td>
</tr>
<tr>
<td>22) 1 1 0 1 1</td>
<td>0 1 0 0 1</td>
<td>0 1 0 0 1</td>
<td></td>
</tr>
<tr>
<td>23) 1 1 1 0 0</td>
<td>0 1 1 0 0</td>
<td>0 1 1 0 0</td>
<td></td>
</tr>
<tr>
<td>24) 1 1 1 0 1</td>
<td>0 1 1 1 1</td>
<td>0 1 1 1 1</td>
<td></td>
</tr>
</tbody>
</table>
Fig. 9.6 Bonding diagram of the fabricated 24×5 dynamic ROM test chip
Fig. 9.7 Expected test results of the 24x5 static ROM
PHOTOGRAPHS
Fig. 9.8 Obtained test results of the 24*5 static ROM
Fig. 9.9 Expected test results of the 3-to-7 dynamic CMOS decoder
Fig. 9.11 Expected test results of the 24*5 dynamic CMOS ROM
VITA AUCTORIS

P. V. R. Raja was born in Paruvachi, Tamil Nadu, India in 1962. In 1979 he received his Pre-University certificate from the R. K. Vivekananda College, Madras, India, and in 1984 Bachelor of Engineering degree in Electronics and Instrumentation Engineering with distinction from the Annamalai University, Annamalai Nagar, India. He joined the Madras Institute of Technology, Madras, India in June 1984. He joined the University of Windsor in 1985 and is presently a candidate for the Master of Applied Science degree in Electrical Engineering.