DESIGN AND TEST OF A MULTIPLE-VALUED LOGIC
CMOS STANDARD CELL LIBRARY

A Thesis Submitted to the College of
Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in the Department of Electrical Engineering

University of Saskatchewan
Saskatoon

By

Vincenzo M. Fortugno

Spring 1996

© Copyright Vincenzo M. Fortugno, 1996. All rights reserved.
To my parents,
my brothers and sisters,
and especially to Corinne.
Copyright

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or in part should be addressed to:

Head of the Department of Electrical Engineering
University of Saskatchewan
57 Campus Drive
Saskatoon, Saskatchewan
S7N 5A9
DESIGN AND TEST OF A MULTIPLE-VALUED LOGIC
CMOS STANDARD CELL LIBRARY

Candidate: Vincenzo M. Fortugno
Supervisors: Dr. R. J. Bolton and Dr. M. H. Abd-El-Barr
M.Sc. Thesis Presented to the College of Graduate Studies and Research
December 1995

Abstract

Multiple-valued logic (MVL) circuits can be designed and implemented, utilizing 4 levels of logic, in current mode CMOS technology to compete with and improve upon conventional binary logic in certain target areas. The chosen technology is important to the success of MVL in augmenting and contributing to the semiconductor logic framework. CMOS is the technology selected for the implementation of the MVL standard cell library in this thesis. It is essential that these circuits can coexist with and perform the functions of existing binary logic circuits, the majority of which are fabricated in CMOS. The design and simulation and/or test of a number of multiple-valued logic standard cells are performed to verify the thesis statement.

The MVL designs are made up of six basic elements; the switch, constant, current mirror, threshold, summing node and voltage reference circuits. Built upon these basic components are the MVL operators including the min, max, tsum, literal, complement of literal, cycle, level restorer and t-gate circuits. In addition, the binary-to-quaternary encoder and quaternary-to-binary decoder circuits are designed, further demonstrating the compatibility between MVL and binary logic. Methods are provided for combining the operators and basic components to synthesize functions via existing
binary tools and representations; the Karnaugh map and the sum of products form. Each of the circuits are presented with detailed design specifications. All circuits are simulated with HSPICE and the results verifying operation are graphed. Standard cell library VLSI layout diagrams are presented with accompanying descriptions for a subset of the operators and basic elements. Test results confirming the basic functionality of the fabricated standard cell circuits are also presented.

The complete design of two implementations of a quaternary full adder are developed and simulated to reveal optional design methodologies available in MVL, and to confirm the validity of the application of MVL design for realizing functions.
Acknowledgements

The author wishes to express his appreciation, for the advice and assistance offered throughout the course of this research and in the preparation of this thesis, to his supervisors Dr. R. J. Bolton and Dr. M. H. Abd-El-Barr.

The author would also like to acknowledge the Canadian Microelectronics Corporation for providing the required facilities to undertake this research work and Northern Telecom Limited for fabricating the Multiple-Valued Logic standard cell Integrated Circuits.
Table of Contents

Copyright ................................................. i
Abstract ................................................. ii
Acknowledgements .................................... iv
Table of Contents ...................................... v
List of Figures .......................................... ix
List of Tables ........................................... xii
List of Abbreviations ................................... xiii

1. Introduction ........................................... 1
   1.1 Research Motivation .................................. 1
      1.1.1 Conventional Logic Problem .................. 1
      1.1.2 Benefits of Multiple-Valued Logic .......... 2
   1.2 Definition .......................................... 2
   1.3 Implementation ..................................... 3
   1.4 Philosophical Foundation for Multiple-Valued Logic .... 3
   1.5 Transformation of Multiple-Valued Logic ........ 4
   1.6 Very Large Scale Integration ..................... 5
      1.6.1 Transistor Technology ......................... 6
      1.6.2 Transistors ..................................... 6
      1.6.3 Voltage Mode CMOS Logic ..................... 7
      1.6.4 Current Mode CMOS Logic .................... 7
      1.6.5 Standard Cell Design .......................... 8
   1.7 Outline ............................................. 9

2. Multiple-Valued Logic Algebra ....................... 10
   2.1 Logic Design with Multiple-Valued Logic .......... 10
   2.2 Algebraic Operators ................................ 12
   2.3 Multiple-Valued Logic Equation Synthesis ....... 14

3. Design Considerations ................................ 16
3.1 Multiple-Valued Logic Building Blocks ................................................................. 16
  3.1.1 Logical Sum ........................................................................................................... 16
  3.1.2 Reference Voltages ......................................................................................... 17
  3.1.3 Constant Current ............................................................................................ 19
  3.1.4 The Current Mirror ......................................................................................... 19
  3.1.5 Threshold Devices ......................................................................................... 22
  3.1.6 Voltage Controlled Switch ............................................................................. 24
  3.2 Design Synthesis ................................................................................................ 26

4. Detailed Design ....................................................................................................... 27
  4.1 Multiple-Valued Logic Current Levels ................................................................. 27
  4.2 Reference Voltages ............................................................................................ 28
  4.3 Basic Standard Cell Circuits ............................................................................. 29
    4.3.1 Literal .............................................................................................................. 29
    4.3.2 Complement of Literal ................................................................................. 30
    4.3.3 Cycle .............................................................................................................. 31
    4.3.4 Level Restorer ............................................................................................... 32
    4.3.5 Binary-to-Quaternary Encoder ...................................................................... 32
    4.3.6 Quaternary-to-Binary Decoder ....................................................................... 33
    4.3.7 Tsum ............................................................................................................... 35
    4.3.8 Min .................................................................................................................. 36
    4.3.9 Max ................................................................................................................ 37

5. Circuit Simulations ................................................................................................ 38
  5.1 Software ................................................................................................................ 38
    5.1.1 Simulator - HSPICE ...................................................................................... 38
    5.1.2 Layout Tool - ELECTRIC .............................................................................. 39
    5.1.3 Hardware Description Language - VHDL ....................................................... 39
    5.1.4 Silicon Compiler - QUISC .............................................................................. 39
  5.2 HSPICE Component Simulations ....................................................................... 40
    5.2.1 Reference Voltages ....................................................................................... 40
    5.2.2 N-type and P-type Mirrors ........................................................................... 40
  5.3 HSPICE Multiple-Valued Logic Circuit Simulations ........................................... 44
List of Figures

| Figure 1.1 | Simplified conceptual form of a VLSI digital circuit layout | 7 |
| Figure 1.2 | Views illustrating the MOS transistor geometry | 8 |
| Figure 2.1 | K-map for the 4-valued 2-variable function \( \text{max}_4(x_1, x_2) \) | 11 |
| Figure 2.2 | Example K-map for the \( \text{max}_4(x_1, x_2) \) function with adjacent squares grouped | 15 |
| Figure 3.1 | Summing node with \( m \) inputs, \( x_1 \ldots x_m \), and one output, \( y \) | 17 |
| Figure 3.2 | Reference voltage circuit | 18 |
| Figure 3.3 | MVL constant circuit. (a) Symbol. (b) P-type constant. (c) N-type constant | 20 |
| Figure 3.4 | NMOS current mirror schematic and symbolic layout | 21 |
| Figure 3.5 | (a) NMOS current mirror symbol and corresponding circuit. (b) PMOS current mirror symbol and corresponding circuit | 23 |
| Figure 3.6 | Threshold circuit. (a) Symbolic representation. (b) N-type threshold circuit. (c) P-type threshold circuit | 25 |
| Figure 3.7 | Switch circuit. (a) Symbol. (b) N-type switch circuit. (c) P-type switch circuit | 25 |
| Figure 3.8 | Top level design blocks for a MVL standard cell circuit | 26 |
| Figure 4.1 | P and N reference sources | 28 |
| Figure 4.2 | Literal circuit, \( kX(a,b) \) | 30 |
| Figure 4.3 | Complement of literal circuit, \( k\overline{X(a,b)} \) | 30 |
| Figure 4.4 | Cycle circuit, \( x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldots \rightarrow x \rightarrow x \rightarrow \ldot...) | 31 |
| Figure 4.5 | Level restorer circuit | 32 |
| Figure 4.6 | Binary-to-quaternary encoder circuit | 33 |
| Figure 4.7 | Quaternary-to-binary decoder circuit | 34 |
| Figure 4.8 | Tsum circuit | 35 |
| Figure 4.9 | Min circuit | 36 |
| Figure 4.10 | Multiple-input max circuit | 37 |
| Figure 5.1 | Simulation of a minimum sized n-type current mirror | 41 |
| Figure 5.2 | Output transfer characteristics for the minimum sized p-mirror | 42 |
| Figure 5.3 | Simulation of a double width p-type current mirror | 43 |
| Figure 5.4 | Binary-to-quaternary encoder simulation | 45 |
Figure 5.5  Binary-to-quaternary encoder generating a step output logic sequence. ................................................................. 46
Figure 5.6  Literal operator simulation. ......................................................... 47
Figure 5.7  Simulation response of the literal operator for an input which skips through intermediate logic levels. ......................... 48
Figure 5.8  Simulation results for the complement of literal operator. ............ 49
Figure 5.9  Simulation of the cycle operator. ................................................. 50
Figure 5.10 Tsum operator simulation.......................................................... 51
Figure 5.11 Min operator simulation............................................................ 52
Figure 5.12 Level restorer simulation.......................................................... 53
Figure 5.13 Simulation output of the quaternary-to-binary circuit.................... 54
Figure 6.1  Voltage reference source values for biasing of the n-type and p-type transistors......................................................... 57
Figure 6.2  Test results for the n-type current mirror..................................... 59
Figure 6.3  Test results for the p-type current mirror..................................... 59
Figure 6.4  Binary-to-quaternary test results............................................... 60
Figure 6.5  Literal operator test results....................................................... 61
Figure 6.6  Test results of the current level, restorer circuit........................... 62
Figure 6.7  Quaternary-to-binary circuit test results...................................... 63
Figure 7.1  Layout organization of a leaf cell............................................... 67
Figure 7.2  Example placement of standard cells........................................ 68
Figure 7.3  Legend distinguishing the various layers in the VLSI layouts........... 69
Figure 7.4  Layout of the binary CMOS inverter from the nc3lib standard cell library................................................................. 70
Figure 7.5  Layout of the binary-to-quaternary encoder.................................. 71
Figure 7.6  Layout of the quaternary-to-binary decoder.................................. 73
Figure 7.7  Layout of the n-reference voltage source..................................... 75
Figure 7.8  Layout of the p-reference voltage source..................................... 77
Figure 7.9  Layout of a single output, one-to-one, n-transistor current mirror..... 79
Figure 7.10 Layout of a single output, one-to-one, p-transistor current mirror..... 81
Figure 7.11 Layout of a three output, one-to-one, n-transistor current mirror..... 83
Figure 7.12 Layout of the quaternary level restorer circuit............................. 85
Figure 7.13 Layout of the literal operator (I)............................................... 87
Figure 7.14  Layout of the literal operator (II) ................................................................. 89
Figure 7.15  Layout of the complement of literal operator (I) ........................................ 91
Figure 7.16  Layout of the cycle operator ........................................................................ 93
Figure 7.17  Layout of sk_001, the first group of MVL library cells combined into a single circuit using VHDL and QUISC .................................................. 95
Figure 7.18  Layout of sk_002, the second group of MVL standard cells combined into a single circuit using VHDL and QUISC ........................................... 96
Figure 7.19  Layout of SKVF0, the fabricated circuit containing sk_001 connected to the pad frame ......................................................................................... 97
Figure 7.20  Layout of SKVF1, the fabricated circuit containing sk_002 connected to the pad frame ......................................................................................... 98
Figure 8.1  3-digit 4-valued parallel adder ....................................................................... 101
Figure 8.2  Circuit scheme to produce a sum function ..................................................... 102
Figure 8.3  K-map of S, the sum portion of a full adder ...................................................... 103
Figure 8.4  T-Gate circuit ................................................................................................. 105
Figure 8.5  Half Adder (SCA) comprised of standard cells, no carry ............................ 106
Figure 8.6  Full adder configuration using the summing node half adder ...................... 107
Figure 8.7  Half Adder (SNA) including both the Sum and Carry outputs ..................... 108
Figure 8.8  Full adder (SCA) simulation, with no carry output ....................................... 109
Figure 8.9  Full adder (SNA) simulation, with carry ....................................................... 110
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1.1</td>
<td>Negation table for the 4-valued variable $x$</td>
<td>5</td>
</tr>
<tr>
<td>Table 2.1</td>
<td>Truth table for the 4-valued 2-variable function $\text{max}_4$</td>
<td>11</td>
</tr>
<tr>
<td>Table 2.2</td>
<td>A subset of MVL algebraic operators (logic formulas)</td>
<td>12</td>
</tr>
<tr>
<td>Table 2.3</td>
<td>Functionally complete sets of MVL operators</td>
<td>14</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Current-to-binary-to-binary truth table.</td>
<td>35</td>
</tr>
<tr>
<td>Table 6.1</td>
<td>Key process parameters for the CMOS3 DLM p-devices</td>
<td>58</td>
</tr>
<tr>
<td>Table 6.2</td>
<td>Key process parameters for the CMOS3 DLM n-devices</td>
<td>58</td>
</tr>
<tr>
<td>Table 7.1</td>
<td>Description of the binary-to-quaternary encoder</td>
<td>72</td>
</tr>
<tr>
<td>Table 7.2</td>
<td>Description of the quaternary-to-binary decoder</td>
<td>74</td>
</tr>
<tr>
<td>Table 7.3</td>
<td>Description of the n-reference voltage source</td>
<td>76</td>
</tr>
<tr>
<td>Table 7.4</td>
<td>Description of the p-reference voltage source</td>
<td>78</td>
</tr>
<tr>
<td>Table 7.5</td>
<td>Description of the n-transistor current mirror</td>
<td>80</td>
</tr>
<tr>
<td>Table 7.6</td>
<td>Description of the p-transistor current mirror</td>
<td>82</td>
</tr>
<tr>
<td>Table 7.7</td>
<td>Description of the n-transistor three output current mirror</td>
<td>84</td>
</tr>
<tr>
<td>Table 7.8</td>
<td>Description of the quaternary level restorer</td>
<td>86</td>
</tr>
<tr>
<td>Table 7.9</td>
<td>Description of the literal circuit, layout I</td>
<td>88</td>
</tr>
<tr>
<td>Table 7.10</td>
<td>Description of the literal circuit, layout II</td>
<td>90</td>
</tr>
<tr>
<td>Table 7.11</td>
<td>Description of the complement of literal circuit</td>
<td>92</td>
</tr>
<tr>
<td>Table 7.12</td>
<td>Description of the cycle circuit</td>
<td>94</td>
</tr>
<tr>
<td>Table 8.1</td>
<td>Partial truth table for a 4-valued full adder</td>
<td>101</td>
</tr>
</tbody>
</table>
List of Abbreviations

ALU  Arithmetic Logic Unit
ASIC  Application Specific Integrated Circuit
BiCMOS  Bipolar Complementary Metal Oxide Semiconductor
CAD  Computer-Aided Design
CCD  Charge-Coupled Device
CIF  Caltech Intermediate Form
CMC  Canadian Microelectronics Corporation
CMCL  Current-Mode CMOS Logic
CMOS  Complementary Metal Oxide Semiconductor
DLM  Double Layer Metal
DUT  Design Under Test
ESD  ElectroStatic Discharge
IC  Integrated Circuit
MOS  Metal Oxide Semiconductor
MVL  Multiple-Valued Logic
POS  Product of Sums
QUISC  Queen’s University Interactive Silicon Compiler
SCA  Standard Cell Adder
SNA  Summing Node Adder
SOP  Sum of Products
TTL  Transistor-Transistor Logic
VHDL  Very High Speed Integrated Circuit Hardware Description Language
VLSI  Very Large Scale Integration
VMCL  Voltage-Mode CMOS Logic
1. Introduction

1.1 Research Motivation

One of the objectives of this thesis is to provide insight into the implementation of a multiple-valued logic (MVL) standard cell library which will aid integrated circuit designers in the design of novel subsystems to meet some of the challenges facing electronics engineers today. This thesis provides a concise introduction to MVL and its considerations for design and development. It is hoped that the material presented will act as a catalyst to not only broaden the awareness of the concepts of MVL, but also to inspire an interest for the widespread implementation of MVL using very large scale integration (VLSI). Presently there are related areas yet to be researched, such as design refinements for circuit speed improvement and discovering applications for MVL circuits. In time, with more attention given to the study, the greater the probability of obtaining faster more efficient MVL subsystems.

1.1.1 Conventional Logic Problem

Physical limitations are hindering the processes for improving the speed, power and functionality of conventional binary digital circuits. Advances in circuit fabrication techniques have helped to increase circuit speeds, decrease power consumption per unit structure and increase the processing power and density of integrated circuits through refinements in lithographic techniques and miniaturization or scaling. The scaling down of device dimensions, however, can only continue to a certain point, beyond which integrated circuits become unreliable, succumbing to the effects of radiation or interference [1], electrostatic shock, and phenomena such as electromigration. Electromigration results in defects within a conduction path caused by the physical flow of metal ions from direct current as influenced by high current densities, the surrounding temperature and the underlying crystal structure [2]. The advances which have allowed for the packing of all the separate functions, of a computer for example, onto a single silicon wafer results in quicker responses and smaller sizes; however this occurs at the expense of greater interconnection area, larger pinouts and higher operating
temperatures. These drawbacks will reach a point where they will become unworkable. Solutions to these problems are required and multiple-valued logic circuit design has the potential to address these limitations.

1.1.2 Benefits of Multiple-Valued Logic

The major advantage of multiple-valued logic circuits, namely, increased information flow, produces important additional benefits. These benefits relate directly to circuit fabrication. As a result of the multi-level encoding and manipulation of data, area is reduced, circuit interconnections (number of pins and wires) are reduced, speed can be increased or at least maintained, and designs can be readily incorporated into existing binary systems [3].

1.2 Definition

Multiple-Valued Logic, abbreviated MVL, is a generalized form of binary logic that can be used to address the limitation imposed by the two discrete voltage levels available in contemporary digital logic circuit design. For a given digital output signal the amount of information which can be conveyed need not be restricted to two values. If the output of a digital signal is limited to 0 and 5 volts then this can at most represent two possible states. However, if the output of the same digital signal can be 0, 2, 3 or 5 volts, then it is possible to represent four states (alternatively the output may consist of a series of current levels). By using four states one can replace two binary logic paths in a circuit with a single MVL path through the same circuit. Theoretically, a MVL signal can provide an enormous amount of information by utilizing hundreds of discrete states, being constrained only by limitations in technology and complexity of design (the design complexity is directly proportional to the number of logic levels). The manipulation of this information through the use of an associated algebra constitutes MVL circuit design.
1.3 Implementation

The implementation of MVL in integrated circuit applications requires that readily familiar techniques for design and development are maintained. If the MVL techniques are awkward or require special consideration for implementation, they would hinder advances in rapidly evolving VLSI design. Undoubtedly, design with MVL requires an advanced form of conceptualization, but without an alteration in the design process; in fact, the ability to readily merge MVL into current binary design methods is one of its strengths.

"If the order of the multiple valued logic is finite \( n \geq 2 \), then the \( n \) logical states

\[
0 = e_0 < e_1 < e_2 < \cdots < e_{n-1} = 1
\]

are coded as binary values 0 or 1 on \((n-1)\) wires...Thus, all \( n \)-stable devices within the computer can operate with binary symbols, preserving reliability features of binary systems and allowing such devices to be incorporated easily into these systems" [4].

The above statement is relevant to the method for encoding and decoding multiple-valued signals. The ability to encode and decode these signals enables a specialized MVL circuit to coexist with or within a binary system.

1.4 Philosophical Foundation for Multiple-Valued Logic

MVL is based on the algebraic concepts required for implementing an indeterminate method of reasoning. Since every proposition cannot be categorized into the constrictive true and false states, it must be accepted that intermediate states do exist. An example of the classification can involve the following two groupings, the first titled probabilistic modalities and the second alethic modalities [5].
<table>
<thead>
<tr>
<th>Probabilistic Modalities</th>
<th>Alethic Modalities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certainly True</td>
<td>Necessarily True</td>
</tr>
<tr>
<td>Probably True</td>
<td>Contingently True</td>
</tr>
<tr>
<td>Indifferent</td>
<td>Contingently False</td>
</tr>
<tr>
<td>Probably False</td>
<td></td>
</tr>
<tr>
<td>Certainly False</td>
<td>Necessarily False</td>
</tr>
</tbody>
</table>

The term alethic describes a portion of Aristotle’s philosophical work related to the area of (ethical) logic. Modality is defined as “the classification of propositions on the basis of whether they assert or deny the possibility, impossibility, contingency or necessity of their content” [6]. The concept (of propositional modality) can be clarified with the example involving two “worlds”, the Actual world (A) and a Particular alternate world (P). For every possible proposition one of four possible states may prevail [5].

<table>
<thead>
<tr>
<th>STATE</th>
<th>DESCRIPTION</th>
<th>NARRATIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>True in A and in P</td>
<td>Necessarily true</td>
</tr>
<tr>
<td>1</td>
<td>True in A but not in P</td>
<td>Actually but not necessarily true</td>
</tr>
<tr>
<td>2</td>
<td>False in A but true in P</td>
<td>Actually but not necessarily false</td>
</tr>
<tr>
<td>3</td>
<td>False in both A and P</td>
<td>Necessarily False</td>
</tr>
</tbody>
</table>

A numerical equivalence becomes evident through this philosophical (discrete) separation of states. If each state in the alethic modalities is assigned a value, with 0 being associated with *Necessarily True* and 3 being associated with *Necessarily False* then an ordered representation results, which can be manipulated algebraically.

### 1.5 Transformation of Multiple-Valued Logic

The transformation from English (propositions) to mathematics results in an MVL form that can be adapted for the development of intricate logic circuits. In the following truth table, Table 1.1, the numerical negations in turn have a valid...
propositional meaning. The encoding of the four logic levels is represented by the digits 0, 1, 2 and 3 in sequence. The Boolean negation operator (¬), also denoted not, of the multiple-valued variable x can be defined [7] according to the function

\[ \neg x = (r - 1) - x \]

where \( r \) is the number of logical levels, in this example \( r = 4 \). Therefore \( \neg 3 \) equals 0, or propositionally, not Necessarily False equals Necessarily True.

<table>
<thead>
<tr>
<th>( x )</th>
<th>( \neg x )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1.1 Negation table for the 4-valued variable x.

Although the negation function has been alternatively defined, resulting in a different truth table, as discussed in [5], the method described above is adopted simply because it provides for practical application, since in this form it is an extension of binary negation \( (r = 2) \). It can be argued that binary logic is in fact the base case for MVL. This argument is valid because for every binary logic function there exists a more general multiple-valued function. For practical purposes, only a select number of multiple-valued functions are required. A subset of these functions will be discussed in Chapter 2.

1.6 Very Large Scale Integration

Current binary logic integrated circuits may consist of millions of transistors on a single silicon wafer. MVL, therefore, must also be implemented in large scale integration in order to be considered for deployment with or in place of binary logic.
Figure 1.1  Simplified conceptual form of a VLSI digital circuit layout.

1.6.3 Voltage Mode CMOS Logic

This use or mode of design, in which the transistor’s switching characteristics are exploited, is often termed voltage mode (VMCL). Although current flow occurs (the MOS transistors are in fact voltage controlled current sources) electric potential is the dominant control mechanism throughout. Essentially the switches pass 0’s and 1’s. The 0 is a low (0 volts in TTL) and the 1 is a high (5 volts in TTL). The TTL level of 5 volts can be misleading in VLSI design, due to the advances in scaling; a high voltage may be as low or lower than 3 volts.

1.6.4 Current Mode CMOS Logic

The multiple-valued signals in this design, in contrast to the voltage signals in binary design, are current based and thus a current mode CMOS logic (CMCL) process is necessary; however, voltage mode processes are also utilized, interestingly enough at an advantage. The benefit is a further integration of both forms of logic. CMCL involves the utilization of both n-type and p-type transistors to produce and control a defined set of current levels. The value of the current, not the voltage, passing through a node becomes more important. Thus, the sizing of the transistors to produce the desired current levels takes on a greater significance.
Figure 1.2 Views illustrating the MOS transistor geometry.

Figure 1.2 illustrates the device parameters for an NMOS (n-type) transistor. The understanding of the physical device characteristics is important for producing a solid VLSI design. Although a circuit designer has no control over the material parameters, there are few limitations for specifying the geometric dimensions. The width and length of each transistor is controlled by the designer with two minor limits. First, the transistor device dimensions can only be as small as the defined process. For a 3µm process the minimum width and length allowed is 3µm. Second, the resolution in specifying the device dimensions is limited. For example, in this design implementation, the dimensions are adjusted to the nearest half micron. Thus by biasing and varying the geometry of the transistors, and carefully controlling their interconnections, the basic circuit elements for CMCL [10] can be achieved. These elements, along with the developed operators, constitute the MVL standard cell library.

1.6.5 Standard Cell Design

The method of standard cell design in standard and custom VLSI is essential. This modular design process allows for practical design refinement from initial conception, circuit synthesis, primary circuit simulation, circuit layout, circuit extraction, secondary circuit simulation (simulation including technology specific parameters) to circuit fabrication and final circuit testing (physical electrical measurements). By designing a set of MVL circuits which can be connected to produce a required set of functions and having these circuits available in a common collection or library, the
objective to facilitate design in MVL is met. This thesis outlines the considerations and processes involved in creating this library and verifying the circuit operations, thereby substantiating the practicality and usefulness of MVL design.

1.7 Outline

Chapter 1 provides an introduction to MVL, describing the motivation and benefits as well as providing a well rounded definition and understanding for MVL. In addition, considerations for implementing MVL in VLSI and standard cell design are discussed. Chapter 2 presents the logic operators for designing MVL subsystems (comparable to the algebraic concepts for implementing a binary arithmetic logic unit (ALU) for example, or rather the tools necessary to design an ALU). It provides a view of how the individual logic operators are used to represent functions. Chapter 3 discusses the transistor level circuit descriptions for the basic units which comprise the MVL standard cells. The discussion includes how key transistors are sized. Chapter 4 includes a discussion of, and notes for, the design and layout of the standard cells; the actual transistor sizes are provided. The simulation of the cells with predicted behaviour are presented in Chapter 5. The actual test results are provided in Chapter 6 with discussion and comparison of the expected and obtained results. Chapter 7 provides the schematics (VLSI layouts) for the standard cells and some comments on the topic. Two complete MVL designs for a radix 4 full adder are presented in Chapter 8. The final chapter, Chapter 9, completes the thesis with the conclusion containing a summary and comments on future work in MVL.
2. Multiple-Valued Logic Algebra

2.1 Logic Design with Multiple-Valued Logic

The individual logic levels alone cannot produce the envisioned range of realizable circuits. Rather it is necessary to combine and compare the levels to produce intermediary outputs according to a certain set of rules or criteria. These rules can be described by a collection of MVL functions. These base functions, called operators, are quite similar to the basic logic gates associated with Boolean algebra. The familiar AND$_2$ and OR$_2$ gates, for example, used in binary systems have sister AND$_4$ and OR$_4$ gates in 4-valued systems, called the minimum (min) and maximum (max) functions respectively [23]. In fact, the scope of tools available for designing in binary logic can be utilized for $r$-valued logic. These include, among other things, truth tables, Karnaugh maps (K-map), two level gating reduction and simplification techniques. The standard product of sums (POS) and standard sum of products (SOP) forms [21] for representing a user defined function is utilized for $r$-valued functions as well [23]. There are a large number of references which deal with the topic of Boolean algebra and its associated forms of representation and methods of simplification; [21] provides one of many good sources.

For purposes of explanation and clarity the max$_4$ function will be referenced throughout this chapter. The numeric subscript 4 attached to the function name max denotes the number of logic levels to which the function is associated, thus max$_r$ refers to the max function dealing with $r$-valued logic. The truth table of the max$_4$ function is provided in Table 2.1, which defines the maximum of the two variables $x_1$ and $x_2$.

A pictorial or graphical representation of the max$_4$ function illustrated in the K-map of Figure 2.1 helps to visualize the behaviour of the function and generate a simplified algebraic expression. The simplified expression translates into a reduction in the number of gates required to implement the function. Each square of the K-map can be defined by a term consisting of literals. The literal is discussed in the next section which deals with the MVL algebraic operators.
Table 2.1  Truth table for the 4-valued 2-variable function max_4.

<table>
<thead>
<tr>
<th>x_1</th>
<th>x_2</th>
<th>max_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Figure 2.1  K-map for the 4-valued 2-variable function max_4(x_1,x_2).

A complete work outlining the transformation from the K-map to a circuit realizable SOP form is detailed in the work of Su and Cheung [7] and the following sections are based in part on the concepts presented in that work. The K-map of Figure 2.1 can be translated directly into a SOP or POS form. The transformation relies upon the min, max and other related operators.
2.2 Algebraic Operators

In decimal arithmetic it would be difficult to form equations without the use of the basic operations of multiplication, division, subtraction and addition. The same difficulty holds true for MVL algebra, but with the major difference being the set of required operators. The more essential MVL operators [7, 10, 23] are summarized in Table 2.2, and the examples provided are for a 4-valued logic. The operators listed allow for the manipulation of the logic levels based on derived functions for realizing a system design. The symbols + and - represent mathematical addition and subtraction respectively in Table 2.2. The \( r \)-valued logic, in the relations provided in the table, is associated with the set \( L = \{0,1,2,...,r - 1\} \).

Table 2.2 A subset of MVL algebraic operators (logic formulas).

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Symbolic Form</th>
<th>Algebraic Form</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>( min ) (AND)</td>
<td>{ ( \cdot )}</td>
<td>( a \cdot b )</td>
<td>( min(a,b) )</td>
</tr>
<tr>
<td>( max ) (OR)</td>
<td>{ ( \varnothing )}</td>
<td>( a \varnothing b )</td>
<td>( max(a,b) )</td>
</tr>
<tr>
<td>complement (NOT)</td>
<td>( \overline{a} )</td>
<td>( (r - 1) - a )</td>
<td>( \overline{2} = (4 - 1) - 2 = 1 )</td>
</tr>
<tr>
<td>( tsum )</td>
<td>{ ( \oplus )}</td>
<td>( a \oplus b )</td>
<td>( min(a + b, r - 1) )</td>
</tr>
<tr>
<td>literal</td>
<td>( kX(a,b), a \leq b )</td>
<td>( k ) if ( a \leq X \leq b )</td>
<td>( k ) if ( a \leq X \leq b )</td>
</tr>
<tr>
<td>complement of literal</td>
<td>( k\overline{X(a,b)}, a \leq b )</td>
<td>( k ) if ( X &lt; a ) or ( X &gt; b )</td>
<td>( k ) if ( X &lt; a ) or ( X &gt; b )</td>
</tr>
</tbody>
</table>
The OR, and AND, functions alluded to previously are the max and min functions respectively. The min function, as its name implies, selects the smallest of its n inputs, where \( n \geq 2 \). The NOT, function, also referred to as the negation or complement, simply inverts its input according to the rule listed. The uncomplemented literal, literal for short, performs a translation on an \( r \)-valued variable which produces a resulting 2-valued variable [4]. The complement of literal is simply the complement or negation of the literal. Both forms of the literal are utilized because for a given expression the use of the complement of literal may produce fewer terms and vice-versa.

Additional operators also exist which relate directly to the summation of logic levels for specific mathematical type functions. The sum function which is the algebraic sum of its inputs, is the central operator. In CMCL the sum function is obtained by simply connecting all the inputs to a single node; the sum of the inputs equals the sum of the output(s) by Kirchoff’s Current Law. The truncated sum (tsum) operator restricts the output current of a summing node from exceeding the maximum logic level for the given \( r \)-valued logic. If the addition of two inputs for a 4-valued CMCL system is logic level 5, then the sum of the inputs is irrelevant, because the maximum logic level is 3. Therefore, the tsum operation can be used to truncate the sum of a set of inputs. Other similar functions exist, such as the cycle, and can be found in [23]. The selection of operators and resulting implementation depends on the technology chosen and the function being considered [23].

One example of a function which is not well suited to implementation in CMCL is the max operator. A circuit realizing the max operator is presented in Chapter 4, but it suffers from a limitation in its maximum attainable output current. The circuit may be redesigned, but this is not necessary since other operators may be used in its place. The type and number of operators required to realize a function in MVL is not restricted. A number of functionally complete sets of MVL operators, a group of operators belonging to a set \( S \) which can synthesize any function without the introduction of any operator(s) foreign to \( S \), can be defined [10, 30]. It is important to realize that the same function can be defined by a number of different sets, however one set may provide a more optimal design over another. Therefore, it is beneficial to have available two or more complete
sets, in order to obtain minimal sized designs for each function to be considered. A set \( S \) may provide optimal configurations for functions \( d \) and \( e \), but functions \( f, g \) and \( h \) may be more efficiently defined using the set \( Q \). Table 2.3 provides a number of functionally complete sets of operators [7, 10, 23, 30]. The sets 1 through 7 are described in references [10, 30] and are summarized in the table. Sets 4 through 8 in Table 2.3 are readily implemented in CMCL, where the \( tsum \) circuit replaces the \( max \) circuit. The \( T \)-gate, although it is functionally complete [23], is better suited to more specialized application. The \( T \)-gate is further discussed in Chapter 8.

### Table 2.3 Functionally complete sets of MVL operators.

<table>
<thead>
<tr>
<th>Set Number</th>
<th>Operators Belonging to the Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cycle, min, max [33]</td>
</tr>
<tr>
<td>2</td>
<td>window literal, min, max [34]</td>
</tr>
<tr>
<td>3</td>
<td>cycle, min, max, unary inverter [36]</td>
</tr>
<tr>
<td>4</td>
<td>window literal, min, tsum [3, 37]</td>
</tr>
<tr>
<td>5</td>
<td>literal, complement of literal, cycle, complement of cycle, min, tsum [10]</td>
</tr>
<tr>
<td>6</td>
<td>literal, complement of literal, min, tsum [7, 10]</td>
</tr>
<tr>
<td>7</td>
<td>literal, min, tsum [10]</td>
</tr>
<tr>
<td>8</td>
<td>T-gate [23]</td>
</tr>
</tbody>
</table>

### 2.3 Multiple-Valued Logic Equation Synthesis

Each square in the K-map corresponds to an expression involving a constant with a \textit{literal} term for each variable. The value in the top right square at the intersection of \( x_1 = 3 \) and \( x_2 = 0 \) is 3 in Figure 2.2. A product term which describes this element is \( 3 \cdot x_1(3,3) \cdot x_2(0,0) \). The expression will take on one of two values when evaluated. If the value of \( x_1 = 3 \) and the value of \( x_2 = 0 \) the expression will become \( 3 \cdot 3 \cdot 3 = 3 \). Conversely, if either \( x_1 \) or \( x_2 \) takes on a value other than 3 and 0 respectively, the product term would evaluate to 0, since 0 AND anything else equals 0. Rather than incorporate a product term for every coordinate in the K-map, groupings of coordinates can be made
to simplify the resulting SOP equation. Figure 2.2 shows 6 combined sets for expressing the \textit{max} function resulting in Equation (2.1). The groupings of adjacent squares in the figure provide one possible configuration for producing a reduced SOP expression.

$$max(x_1, x_2) = 1 \cdot x_1(1,1) \cdot x_2(0,1) + 1 \cdot x_1(0,1) \cdot x_2(1,1) + 2 \cdot x_1(2,2) \cdot x_2(0,2)$$
$$+ 2 \cdot x_1(0,2) \cdot x_2(2,2) + 3 \cdot x_1(3,3) \cdot x_2(0,3) + 3 \cdot x_1(0,3) \cdot x_2(3,3)$$  \hspace{1cm} (2.1)

Equation (2.1) can be further reduced by eliminating the \textit{literals} which evaluate to a constant in both the 5th and 6th product terms. For example, the term \(x_2(0,3)\), also written \(1 \cdot x_2(0,3)\), reduces to 1 since the expression \(0 \leq x_2 \leq 3\) is always true.

$$max(x_1, x_2) = 1 \cdot x_1(1,1) \cdot x_2(0,1) + 1 \cdot x_1(0,1) \cdot x_2(1,1) + 2 \cdot x_1(2,2) \cdot x_2(0,2)$$
$$+ 2 \cdot x_1(0,2) \cdot x_2(2,2) + 3 \cdot x_1(3,3) + 3 \cdot x_2(3,3)$$  \hspace{1cm} (2.2)

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{k_map.png}
\caption{Example K-map for the \textit{max}\(_4\)\((x_1, x_2)\) function with adjacent squares grouped.}
\end{figure}

Computer simplification techniques are required for MVL functions of greater than 2 or 3 variables as the K-map method becomes tedious. Once a reduced function expression is obtained, the implementation of the function becomes solely dependent upon the availability of each of the operators in working circuit form.
3. Design Considerations

3.1 Multiple-Valued Logic Building Blocks

The formation of the MVL circuits relies on a number of basic elements. Two of the more significant units include current sources, NMOS and PMOS current mirrors, and voltage controlled switches, referred to as n-type and p-type switches. In order for the current sources to provide the correct output levels, voltage reference sources, essentially CMOS voltage dividers, are required to bias the transistors which supply the current. The detection of these current levels is important; this is accomplished through the use of current sensing circuits, known as n-type and p-type threshold devices, which are also biased by the same reference voltages. A method for generating constant current levels, referred to as n-type and p-type constants, is essential. Once generated, these current levels can be manipulated, for example, through the use of summation nodes. A detailed description of each of these six basic MVL elements follows.

3.1.1 Logical Sum

The logical sum of a set of MVL inputs is the simplest circuit element to implement because it requires only an ohmic connection of inputs to a single node. The summing node, obeying the property defined by Kirchoff's current law, can provide for both a summation and a difference of currents, depending upon the direction of the inputs [10]. Figure 3.1 provides the symbol used to represent the logical sum in part (a) and its corresponding circuit diagram in part (b). The direction for the input current $x_m$ is shown opposite to that of the other inputs; this indicates that the sum can be a difference of inputs. The cycle is one operator which takes advantage of the summing node utilized to provide a difference of inputs.
3.1.2 Reference Voltages

The reference voltages are required to bias transistors in saturation to achieve specified current levels. Two separate voltages are required depending upon which type of transistor is generating the current. The need for two voltages is due primarily to the difference in the majority carrier mobility of p-type and n-type transistors. The required reference voltage is calculated using the equation for the drain-to-source current ($I_{ds}$) for an enhancement MOS transistor in saturation [2].

$$I_{ds} = \frac{\mu \varepsilon}{2t_{ox}} \left( \frac{W}{L} \right) \left( V_{gs} - V_t \right)^2$$  \hspace{1cm} (3.3)

The symbol $\mu$ is the drift mobility of the majority carrier, $\varepsilon$ is the permittivity of the gate insulator, $t_{ox}$ is the thickness of the gate insulator and $V_t$ is the device threshold voltage. The symbols $W$ and $L$ are the channel width and channel length of the transistor respectively. $V_{gs}$ is the gate-to-source voltage, the voltage of interest, which sets the value for the reference voltage. Equation (3.3) is rearranged to solve for the gate-to-source voltage.
\[ V_{gs} = \sqrt{I_{ds} \frac{2t_{ox}}{\mu \varepsilon} \frac{L}{W}} + V_i \] (3.4)

The device dimensions are chosen for a minimum sized transistor and the current \( I_{ds} \) is set to \( I_o \), logic level one. The reference voltage circuit itself consists of two transistors connected in series between power and ground. Each transistor, one PMOS and one NMOS, is connected as a resistance. The circuit illustrated in Figure 3.2 is essentially a voltage divider. The reference voltage circuit provides a biasing voltage, \( V_{ref} \), which determines the value of current through a transistor.

![Figure 3.2](image)

**Figure 3.2** Reference voltage circuit.

The size of the transistors \( T_1 \) and \( T_2 \) can be determined based on the effective resistance, \( R_{eff} \), of each transistor. \( R_{eff} \) is directly proportional to the mobility of the charge carriers and the ratio of the length to width of the conduction path, \( R_{eff} \propto \mu L/W \). However, the effective resistance is nonlinear and is dependent upon the drain to source
current and the drain to source voltage. The size of $T_2$ is chosen to be minimum sized, thus the size of $T_1$ need only be determined. An expression relating the length to the width of $T_1$, the PMOS transistor (denoted by the $p$ subscript on $L_p, W_p$ and $\mu_p$), is derived from Equation (3.3).

$$L_p = \frac{\mu_p e}{2t_{ox}} \left( \frac{1}{I_{ds}} \right) (V_g - V_s - V_t)^2 W_p$$  \hspace{1cm} (3.5)

The value of the gate voltage, $V_g$, and the source voltage, $V_s$, for $T_1$ in Equation (3.5) is determined by KVL. The current, $I_{ds}$, through $T_1$ is equal to the current through $T_2$ by KCL. The value of current through $T_2$ is determined by setting $W_l/L_n = 1$, since $T_2$ is minimum sized, and $V_{gs}$ equal to $V_{ref}$ in equation (3.3).

### 3.1.3 Constant Current

According to the algebraic MVL expressions, constants are required in order to implement certain functions. A constant can be achieved by biasing an n-type or p-type transistor with the correct reference voltage. The reference voltage $V_{N-ref}$ is used to designate the reference voltage for an n-type transistor and $V_{P-ref}$ that of a p-type transistor. Figure 3.3 illustrates the constant, a circuit used to produce a constant level of current. The current, $K$, acts as a sourcing current in (b) and a sinking current in (c).

### 3.1.4 The Current Mirror

The current sources, called current mirrors, consist of “mirrored” transistors which provide stable and consistent current levels. Current mirrors are also used to regenerate and scale signals. The actual current levels used to represent the multiple-valued signals depend upon the number of signals desired, the limitations of the chosen technology (effects of noise and other phenomena such as latchup and electromigration) and the important goal of area minimization. It is important to mention here that area minimization is meaningful because it is directly proportional to the yield, $Y$, which is one of the advantages of MVL. The number of good ICs per wafer increases with higher
density. Two models are provided in [2], Seed's model for large chips and Murphy's model for small chips.

\[ Y = e^{-\sqrt{AD}} \]

\[ Y = \left( \frac{1 - e^{-AD}}{AD} \right)^2 \]

The variable \( A \) represents the chip area and \( D \) is defined as the defect density.

\[ V_{P\text{-ref}} = K \]

\[ V_{N\text{-ref}} = K \]

\[ V_{DD} \]

\[ V_{SS} \]

**Figure 3.3** MVL constant circuit. (a) Symbol. (b) P-type constant. (c) N-type constant.
The current levels need to be small enough to utilize minimum sized transistors, but large enough to be readily discernible without error; this is not only to ensure that static values are valid, but also, as with many other circuits, to guard against the dynamic or transient states of the system which may produce temporary erroneous results. These transient errors are, in part, dependent upon the speed variations which can exist between certain interdependent devices within the circuit. This issue will be discussed when the test results are presented.

The successful operation of the current mirror relies on the fact that the output current is isolated from the input current. This isolation can be achieved by having an identical controlling voltage (gate to source voltage which is determined by the input current) producing the output current. The circuit diagram of Figure 3.4(a) can clarify the operation of the current mirror; the label \( x \) designates the input current and the label \( y \) designates the output current [3, 10, 22]. The power rail shown in (b) would provide the source for the circuit which would produce the current \( x \), the input to the n-mirror. For a n-mirror there is no direct connection to power; similarly, there would be no direct connection to ground for a p-mirror.

![Diagram](image)

(a) n-mirror circuit schematic  
(b) n-mirror symbolic layout (VLSI)

**Figure 3.4** NMOS current mirror schematic and symbolic layout.
The n-mirror current source in Figure 3.4 is the central component in the CMCL MVL VLSI designs being presented. The current mirror illustrated is a two transistor, unit scaled device. The drain (D), source (S) and gate (G) are labeled for the first transistor (T1) in the circuit schematic Figure 3.4(a) and in the corresponding VLSI layout Figure 3.4(b). A similar structure, called a p-mirror, can be formed with p-type transistors to provide current flow in the opposite direction. If the width to length ratio of the gate region on T2 equals that of T1 then the scaling factor, a, is 1. If the ratio for T2 divided by the ratio for T1 equals two then the output current y would be twice the value of x. The number of output transistors is not restricted. The relationship between any output and the input is defined as,

\[ y_i = a_i x \]

Figure 3.5 provides the symbols and representations for the two types of mirrors. The operation of the mirror relies on setting the controlling gate voltage on the output transistor(s) to match that of the gate voltage on the input transistor, as is evident in the figure. The input current is converted into a voltage which in turn generates an output current equal in magnitude to the input current.

3.1.5 Threshold Devices

The ability to recognize or detect a certain logic level is necessary in order to determine the steps required to achieve a desired output given the input(s). In binary logic, the inputs turn on or off a series of switches relaying the results to the output in a type of domino effect. Each preceding switch affects each subsequent switch as in a configuration of dominos. In the CMCL implementation of MVL, this direct control of the switching, becomes indirect. First, a current cannot directly drive a CMOS switch, and second, a simple conversion to the on and off states no longer exists. The p-type and n-type threshold devices allow a mapping from current to voltage to exist. A threshold device will output a binary logic high if the input current, x, is greater than or equal to the threshold value, k. Otherwise the output, V_y, will produce a binary logic low. The threshold current can be defined as the value of current required to produce the saturated current flow through the threshold transistor.
Figure 3.5  (a) NMOS current mirror symbol and corresponding circuit. (b) PMOS current mirror symbol and corresponding circuit.
Figure 3.6 provides the circuit diagrams for the threshold devices used to detect if the magnitude of the input $x$ matches and/or exceeds $k$. The threshold level is determined by the size of reference voltage, $V_{N-ref}$ and the transistor size. In the subsequent discussion the following variables are equivalent, $V_{gs} = V_{N-ref}$, $V_{ds} = V_y$ and $I_{ds} = x$. If the input current for the n-type threshold device in (b) is less than the threshold current $k$, then the transistor will operate in the linear region, $0 < V_y < (V_{N-ref} - V_t)$. For a transistor operating in the linear region $V_{ds}$ is proportional to $I_{ds}$, thus, $V_y$ is proportional to $x$. $V_y$ increases with further increases in $x$ until saturation occurs. After saturation is reached, $V_y > (V_{N-ref} - V_t)$, $x$ is limited to $k$. Recall that the I-V characteristics are horizontal within the saturation region for a constant $V_{gs}$. Therefore, $V_y$ continues to increase while $x$ remains constant; $V_y$ becomes binary high less a small voltage drop across the device providing the input current. The operation of the threshold device rests on the fact that $V_{N-ref}$ is less than $V_{DD}/2$ otherwise $V_y$ could become binary high before $x$ reaches a magnitude of $k$. A similar process can be described for the p-type threshold device in (c).

### 3.1.6 Voltage Controlled Switch

A standard CMOS switch is utilized to permit or inhibit the flow of current. The p-type and n-type switches provided in Figure 3.7 regulate the flow of current, irrespective of magnitude (up to the level of saturation current for the transistor). Part (a) depicts the circuit symbol for the switch. The circuit diagrams in (b) and (c) represent the n-type and p-type switches respectively. The switch operation is straightforward, given the n-type switch in Figure 3.7(b) if $V_{in}$ is binary high then $y = x$, otherwise $y = 0$. 
Figure 3.6  Threshold circuit. (a) Symbolic representation. (b) N-type threshold circuit. (c) P-type threshold circuit.

Figure 3.7  Switch circuit. (a) Symbol. (b) N-type switch circuit. (c) P-type switch circuit.
3.2 Design Synthesis

In addition to the basic components provided, some conventional binary logic gates are incorporated into the standard cell designs to form the logic function control circuits. A pictorial view of the synthesis of the circuits illustrates the signal flow and interconnection of the blocks.

Figure 3.8 Top level design blocks for a MVL standard cell circuit.

Figure 3.8 illustrates a template for the development of the standard cells. Each standard cell can adopt the basic template tailoring the individual blocks accordingly. The control logic block, different for each cell, necessitates the majority of the modifications. The control circuit consists of conventional binary logic gates which allow for the control of the switches on the current source. The current source in most of the standard cells is actually made up of two individual current sources, supplying two different levels of current. The switch block is a set of two switches; each switch is turned on and off in combination (on-on, on-off, off-on, off-off) to produce the proper output current level.

The next chapter provides the detailed circuit schematics comprising the standard cell library.
4. Detailed Design

The detailed circuit schematics are presented in this chapter with notes on relevant details to the design choices and methods utilized. The circuit operations are discussed qualitatively. Each of the circuits were prepared for fabrication with the dimensions provided. Some of the circuit designs were preliminary, but appropriate for testing the MVL circuit operations.

A brief description of the notations and conventions used will be helpful. The circuit schematics provide the detailed transistor interconnections, and incorporate logic symbols/labels where appropriate. Voltages always begin with the letter V, currents always begin with the letter I, with the exception that x usually designates the input current and y the output current to the MVL circuits. Transistor sizes are often provided in either lambda (λ) units or microns (μm). Lambda units, referred to as design scale microns, are drawing units which when multiplied by the corresponding scale factor provide the actual size in microns. The width (W) to length (L) ratio is the convention used (W:L) in the schematics, and all units are assumed to be in microns unless explicitly provided in lambda units. The W to L ratio of 6:3, for example, represents a transistor with a width of 6 µm and a length of 3 µm.

4.1 Multiple-Valued Logic Current Levels

The value of the MVL signals do not need to be fixed, but rather can be set proportionately. The symbol $I_o$ is used to represent the first MVL logic level, that above the base logic level of 0. Each subsequent logic level is a factor of $I_o$. Thus the 4-valued logic levels are 0, $I_o$, $2I_o$ and $3I_o$. The value of $I_o$ in the design is chosen at 20 μA. It is worth mentioning that in relation to the actual fabrication process, the value of $I_o$ is not important given that all dimension and process parameter variances affect each transistor equally; however, if the variation is too great from the original specifications, problems may arise whereby circuits are no longer operable. For example, delays may be more pronounced, currents may become too large or too small and other unpredictable behaviour may result. Alternatively, in relation to circuit operation, the value of $I_o$ is
important, because it must be small enough to minimize logic stepping delays - this matter will be discussed further in Chapter 6. In addition, the current values need to be large enough to adequately drive standard binary CMOS gates.

4.2 Reference Voltages

The setting of $I_0$ determines the size of transistors required to generate the bias voltages $V_{P-ref}$ and $V_{N-ref}$. The magnitude of $V_{P-ref}$ required to produce a current of $I_0$ is 2.25 volts. The value of $V_{N-ref}$ necessary to produce the same current in the minimum sized n-type transistors is 1.73 volts. The reference voltages are determined through calculations as discussed in Section 3.1.2.

![Figure 4.1 P and N reference sources.](image)

The circuits in Figure 4.1, which include the transistor sizes, are used to produce the gate biasing voltages. The transistor sizes are calculated using Equations (3.3) and (3.5). These circuits are omitted from the standard cell schematics, and the labels exist in their place.
4.3 Basic Standard Cell Circuits

Each of the MVL circuits presented were adapted from [10], unless otherwise noted, and were redesigned to accomplish the research objectives. The *quaternary-to-binary decoder* and *binary-to-quaternary encoder* circuits are original designs.

4.3.1 Literal

The *literal* maps an \( r \)-valued variable into a two valued variable. The circuit shown in Figure 4.2 is one implementation of the literal \( kX(1,1) \). The constant \( k \) for an \( r \)-valued logic is more often selected as \( r - 1 \); for this case, the literal is referred to as a *window literal* [10]. The size of \( T_3 \) determines the value of \( k \). Currently \( T_3 \) is minimum sized (\( W:L = 3:3 \)), therefore \( k \) is set to 1, which corresponds to a current of \( I_o \) and a logic level of 1. In this configuration \( y \) alternates between the two values 0 and \( I_o \), depending upon \( x \). If the width of \( T_3 \) is tripled (\( W:L = 9:3 \)), then the constant \( k \) is 3 and \( y \) alternates between 0 and 3\( I_o \). The transistors \( T_1 \) and \( T_2 \), threshold transistors, can be sized accordingly to obtain the full range of literals from \( kX(0,0) \) to \( kX(3,3) \), with a few exceptions. \( T_1 \) activates at a current level of \( 4/3 I_o \) and \( T_2 \) at a current level of \( 3/4 I_o \) as shown in the preliminary design of the literal \( kX(a,b) \) provided in Figure 4.2. The variable \( k \), equal to 1, corresponds directly to the second of the two possible output logic levels (0 and 1). Work done to optimize the literal performance, described by \( X(a,b) \), in the presence of imperfect logic values, sets the threshold current levels for \( T_2 \) at \( (a - 0.5)I_o \) and that for \( T_1 \) at \( (b + 0.5)I_o \) [10].

The operation of the literal circuit relies on the detection of the input current level. The currents \( I_1 \) and \( I_2 \) are initially mirrors to the input current \( x \). Transistor \( T_1 \) detects whether the value of \( I_1 \) is greater than \( b \), and \( T_2 \) detects whether \( I_2 \) is less than \( a \). If either condition is met, the switch, transistor \( T_4 \), is off and the output current is 0, otherwise, \( T_4 \) is on, resulting in an output current of \( I_o \). The p-mirror connected to \( T_4 \) changes the output from a sinking to a sourcing current.

The literal is important for realizing the terms of a function, as discussed in Chapter 2. A typical logic expression realizing a function may consist of the *min*, *max*,
or _tsum_ of a number of product terms. The product terms may include the _literal_, _complement of literal_, or _cycle_ operators.

![Figure 4.2 Literal circuit, kX(a,b).](image)

### 4.3.2 Complement of Literal

The _complement of literal_ function is used to realize _literals_ that cannot be obtained by the _literal_ circuit, namely, X(0,0), X(0,1) and X(0,2). These three _literals_ can be obtained using the _complement of literal_ functions X(1,3), X(2,3), and X(3,3) respectively [10]. The operation of the circuit in Figure 4.3 parallels that of the _literal_ circuit.

![Figure 4.3 Complement of literal circuit, kX(a,b).](image)
4.3.3 Cycle

The forward cycle operator, denoted $x \xrightarrow{m} y$, transposes the input by adding $m$, where $m$ is limited to the set $\{1, 2, \ldots, r - 1\}$. If $m = 2$, then the input logic sequence $\{0,1,2,3\}$ would produce the output sequence $\{2,3,0,1\}$. Based on modulo arithmetic when $m$ plus the input exceeds the highest available logic level, $r$ is subtracted to obtain the cycle output for a $r$-valued logic.

The circuit of Figure 4.4 is the forward cycle operator $x \xrightarrow{3}$ or equivalently $x \xleftarrow{1}$ the reverse ($m$ is subtracted from the input) cycle operator. The circuit operation is determined by the size of transistors $T_1$, $T_2$ and $T_3$. $T_1$ is a threshold which detects any logic level greater than zero, $T_2$ sources $I_0$ and $T_3$ is sized to sink $I_0$. The two switches between $T_2$ and $T_3$ are controlled by the same connection point so that if one is on the other is off. If $x$ is 0, then the sourcing current is allowed to flow and the sinking current is disabled. If $x$ is greater than 0, then the sourcing current is disabled and the sinking current is enabled. The effect is to subtract $I_0$ from the value of the input. The two inverters in series, shown in Figure 4.4, condition the signal for driving the n-type switch and p-type switch connected to the output of the second inverter.

![Figure 4.4](image)

**Figure 4.4** Cycle circuit, $x \xrightarrow{m} y$. 
4.3.4 Level Restorer

The level restorer is used to restore a current (logic) level to its proper size (value). The inclusion of the level restorer depends upon which operation is being performed. The literal circuit is more restoring than the cycle circuit, that is, there is less chance of error in the output logic level of the literal. The literal provides an output from one source as opposed to three sources connected to a summing node in the cycle. Since the level restorer adds extra delay to the overall circuit it should only be used at suspect nodes.

The level restorer circuit in Figure 4.5 utilizes the threshold transistors $T_1$, $T_2$ and $T_3$ to detect the presence of the three current levels $0.5I_o$, $1.5I_o$ and $2.5I_o$. If the current level is detected, then the corresponding switch is turned on, otherwise, it is off. Transistors $T_a$, $T_b$ and $T_c$ each provides a current size of $I_o$, therefore, if all three switches ($S_a$, $S_b$ and $S_c$) were on, $x$ greater than or equal to $2.5I_o$, the output current would equal $3I_o$, equivalently, the output logic level would equal 3.

![Figure 4.5 Level restorer circuit.](image)

4.3.5 Binary-to-Quaternary Encoder

A method of interfacing from a binary system to a MVL system [3] is accomplished through a binary to $r$-ary encoder. The binary-to-quaternary encoder converts two binary (voltage) input signals into a single 4-valued logic (current) level.
The *quaternary encoder* design is similar to that presented in [19] if transistor T₂ in the top row of Figure 4.6(b) is doubled in width to provide 2I₀ current flow and the two right-most transistors, T₃ and S₃, are removed. The encoder maps a base 2 numeral into a base 4 numeral. The mapping is outlined in the truth table of Figure 4.6(a) where, for example, the binary input 10₂ (Vᵢₜ₀ = binary high, Vᵢₜ₁ = binary low) is converted to 2₄ (2I₀). If Vᵢₜ₀ is binary high, then S₂ and S₃ are turned on in unison allowing T₂ and T₃ to supply current to the output, otherwise no current flows from T₂ or T₃. Similarly if Vᵢₜ₀ is binary high, S₁ turns on which allows T₁ to supply current to the output, otherwise current flow is cut off from T₁.

\[
\begin{array}{ccc}
V_{i\text{t}0} & V_{i\text{t}1} & y \\
0 & 0 & 0 \\
0 & 1 & I_0 \\
1 & 0 & 2I_0 \\
1 & 1 & 3I_0 \\
\end{array}
\]

(a)  

Figure 4.6  Binary-to-quaternary encoder circuit.

### 4.3.6 Quaternary-to-Binary Decoder

One method of interfacing a MVL system to a binary system is through an r-ary to binary decoder. The *quaternary-to-binary decoder* decodes a single 4-valued logic (current) level into two separate binary (voltage) outputs. Figure 4.7 provides the *quaternary-to-binary decoder* circuit.

The operation of the decoder, as with the *level restorer*, relies on the detection of three separate levels of current. The threshold transistors T₁, T₂ and T₃ detect the
Figure 4.7 Quaternary-to-binary decoder circuit.

presence of the three current levels $0.5I_o$, $1.5I_o$ and $2.5I_o$. If $x$ supplies enough current to activate a threshold device, then the corresponding voltage at node a, b, c or some combination, will become binary high, thus the task becomes one of converting three binary signals into the required two binary outputs. It is important to note that at this point, the delay experienced by the activation of the threshold devices becomes consequential. To clarify the discussion Table 4.1 is referenced.

By observation, it can be seen that the voltage at node b, $V_b$, in the table is identical to the expected output high bit, $V_{HI}$. The one to one mapping of $V_b$ to $V_{HI}$ is revealed by the branch containing two binary inverters connected to node b. Also, when $V_b$ is low $V_a$ matches the expected output for $V_{LO}$. Similarly, when $V_b$ is high $V_c$ matches the expected output for $V_{LO}$. The matched patterns are highlighted by the solid ellipses and rectangles. Therefore, $V_a$ and $V_c$ are ORed together for $x \leq I_o$. $V_c$ and 0 are ORed together for $x \geq 2I_o$. The OR of $V_c$ with 0 is accomplished through the two pass transistors connected to node b, a p-type pass transistor $T_p$ and a n-type pass transistor $T_n$. If $V_b$ is low then $T_p$ is on, $T_n$ is off and $V_a$ is passed through. If $V_b$ is high then $T_n$ (connected to ground) is on, $T_p$ is off and 0 is passed through.
Table 4.1  Current-to-binary-to-binary truth table.

<table>
<thead>
<tr>
<th>x</th>
<th>Va</th>
<th>Vb</th>
<th>Vc</th>
<th>VH</th>
<th>VLO</th>
</tr>
</thead>
<tbody>
<tr>
<td>0I₀</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1I₀</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2I₀</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3I₀</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

4.3.7 Tsum

The truncated sum operator, tsum, performs the addition of logic levels. If the result of the addition exceeds the highest logic level available in the MVL system, then truncation to the highest logic level is performed. The tsum selects the minimum of the addition of the inputs or the logic level $r - 1$. For example, with $r = 4$ the tsum of $x_1 = 1$ and $x_2 = 1$ is $tsum(1,1) = min(1+1, 3) = 2$. The tsum of $x_1 = 1$ and $x_2 = 2$ and $x_3 = 1$ is $tsum(1,2,1) = min(1+2+1,3) = 3$. The circuit schematic of Figure 4.8 details the design of the tsum operator [10].

![Tsum circuit](image_url)

Figure 4.8  Tsum circuit.

The summation of the currents $x_1 + x_2 + ... + x_m = x$ flows into a n-mirror with two outputs. The logic level $r - 1 = 3$ is supplied to node a. At node a, three currents
exist with the following KCL relation, $I_2 = I_1 - 3I_0$. If $I_1$, which tries to mirror $x$, exceeds $3I_0$, then $I_2$ provides the additional current. At node b, once again, three currents exist in the following relation, $I_4 = x - I_3$. $I_3$ attempts to mirror $I_2$ and $y$ is the mirror of $I_4$, thus $y = x - I_2$. Given these expressions, the following two scenarios are possible, $x \leq 3I_0$ and $x > 3I_0$. For $x \leq 3I_0$, $I_2$ would need to be negative or zero, since $I_2$ cannot be negative it must be zero, therefore $y = x$. For $x > 3I_0$, $I_2 = x - 3I_0$ since $I_1$ mirrors $x$. Substituting this expression for $I_2$ in the expression for $y$, $y = x - (x - 3I_0)$ and simplifying yields the relation $y = 3I_0$. In simplified terms, node a determines if $x$ exceeds $3I_0$ and passes this information onto node b which outputs the correct logic level.

### 4.3.8 Min

The min operator outputs the minimum of two input logic levels. Close observation of the min circuit [10] provided in Figure 4.9, compared with the tsum circuit illustrated in Figure 4.8, reveals that they are essentially the same circuits. The tsum utilizes the min circuit to determine the minimum of the summed inputs and $r - 1$. $T_1$ in Figure 4.8 is removed and the constant current $3I_0$ is replaced by the variable current $x_2$ in Figure 4.9.

![Figure 4.9 Min circuit.](image-url)
The operation of the \textit{min} circuit parallels that of the \textit{tsum} circuit with $x$ replaced by $x_1$ and $3I_0$ replaced by $x_2$.

### 4.3.9 Max

The \textit{max} circuit of Figure 4.10, adapted from [18], outputs the maximum value of its multiple inputs.

![Multiple-input max circuit](image)

**Figure 4.10** Multiple-input max circuit.

The authors, Baturone et al. [18], describe this \textit{max} circuit as an interconnected set of improved Wilson current mirrors. Each mirror uses the single diode-connected transistor, $T_4$, as a common source resistance. Based on simulations, with the transistor sizes as provided in Figure 4.10, a maximum input current of 40 $\mu$A is possible, but at least 60 $\mu$A is required. Voltage drops across the transistors constrain the maximum attainable input current. The circuit thus exhibits a limitation which requires redesign in order to utilize the circuit as a MVL operator.
5. Circuit Simulations

Simulation of integrated circuits is essential in order to validate circuit operation and reveal potential flaws in the design. Simulation also helps in the refinement of circuit design; it takes only minutes to change a transistor size and re-simulate, although an actual simulation run may take many minutes to complete. Software simulators developed for modeling electronic circuits have proven to be correct, providing accurate results for predicting circuit behaviour.

5.1 Software

All of the software used in the creation of the MVL standard cells runs under the Sun Operating System on Sun workstations. The software used includes: HSPICE, ELECTRIC, VHDL and QUISC. Software tailored for applications in VLSI design is an important tool required by integrated circuit (IC) designers. Computer aided design (CAD) is essential to effect the rapid development of VLSI systems, and working knowledge of the individual CAD tools certainly requires priority.

5.1.1 Simulator - HSPICE

The simulation software used to confirm circuit functionality is HSPICE by Meta-Software, Inc. HSPICE is a powerful electrical circuit simulator which makes use of standardized models for transistors or other devices incorporated into the circuits under analysis [24]. Key transistor fabrication specific parameters, like the oxide thickness for example, are user configurable. The model level used in the circuit simulations is the LEVEL 3 IDS Empirical Model. This model takes into consideration the mobility modulation, saturation field factor, maximum drift velocity of the carriers, as well as others. Level 3 simulation is sufficient for the purposes of this thesis, not too advanced (Level 8) and yet not too basic (Level 1). The purpose of the simulations is to obtain results that will accurately predict the behaviour of the resulting physical circuit and the Level 3 model meets this requirement.
Two separate simulations are carried out for each circuit prepared for fabrication. The initial simulation is performed using a user prepared netlist (circuit description). The custom layout takes many hours, so it is advantageous to know that a circuit will perform as expected before laying it out. The second simulation is performed using a netlist generated by the layout tool, in which the parameters are extracted from the actual layout. This simulation helps ensure that the layout is correct.

5.1.2 Layout Tool - ELECTRIC

ELECTRIC is a layout editor that allows a VLSI designer to place and size transistors and run metal and polysilicon wires with geometric precision. ELECTRIC is a highly specialized graphics drawing package which integrates a number of important tools including an editor, a database of libraries, a design rule checker, a netlist generator, a compaction tool, a conversion tool, and a simple simulator, as well as others. The final Caltech Intermediate Form (CIF) code describing the integrated circuit(s) is generated by ELECTRIC. It is the data required for submission to effect the fabrication of the chips containing the designs.

5.1.3 Hardware Description Language - VHDL

VLSI Hardware Description Language, VHDL, is a programming language similar to BASIC and PASCAL, but much more tightly controlled, which is used to describe the interconnection of a circuit based on the available inputs and outputs of a number of leaf (standard) cells.

5.1.4 Silicon Compiler - QUISC

The Queen's University Integrated Silicon Compiler QUISC, takes VHDL code along with a standard cell library of integrated circuits created using ELECTRIC as its input and produces an interconnected integrated circuit. Only the pads and pad connections need to be physically added to complete the circuit to be submitted for fabrication.
5.2 HSPICE Component Simulations

Each of the cells were simulated initially without loading and then with loading to see how performance was affected. The simulations are presented with loading included. The type of load was chosen to represent a typical input impedance of another cell. Depending upon which direction the output current flows, an n-mirror or a p-mirror was used to model a typical load in the following graphs. The simulations presented are based on the geometric dimensions of the actual layouts, that is, the netlist is extracted from the layout editor. The input waveforms were set to have a switching time of 5 ns, the time it takes for the signal to rise or fall to the next specified logic level. No correction was made for a signal which may pass through more than one logic level before reaching its destination. The graphs are plotted with the data generated by HSPICE. One HSPICE listing for the literal is provided in Appendix II.

5.2.1 Reference Voltages

The simulation of the reference voltages provided values of 2.3254 volts and 1.681 volts for the \( V_{p\text{-ref}} \) and \( V_{n\text{-ref}} \) sources respectively. The simulation values are included in Figure 6.1 in Chapter 6. The variance in the p-reference voltage of +0.0754 volts, up from 2.25 volts, is due to the fact that the p-transistor size changed from W:L of 6.3:3 to 6.6:3 in order to address limitations imposed by precision in the specification of dimensions in design scale microns for fabrication. Initially at 6.3 \( \mu \text{m} \) (10.5\( \lambda \)), resolution errors cannot be avoided unless the dimension is rounded up or down to the nearest whole number, 11\( \lambda \) was selected resulting in the W = 6.6 \( \mu \text{m} \). Similarly for the n-reference voltage the smaller variance of -0.049 volts, down from 1.73 volts, comes from the change in L of the p-transistor from 5.2 \( \mu \text{m} \) to 5.4 \( \mu \text{m} \) as a result of rounding up from 8.7\( \lambda \) to 9\( \lambda \).

5.2.2 N-type and P-type Mirrors

The n-type and p-type current mirrors enable the individual MVL cells to be used interchangeably, since each complements the other with respect to current direction.
5.2.2.1 N-Mirror

The graph of Figure 5.1 reveals the behaviour of an n-mirror for a linear voltage input from 0 volts at 0 ns to 5 volts at 400 ns. The input voltage controlled a p-transistor connected as a current source. In this configuration, the voltage was applied to the gate of a p-transistor with its source connected to V_{DD} and drain connected to the input node of the n-mirror. A p-transistor was connected to the output of the n-mirror to act as a load. The output current, y, is plotted as a function of the input current generated by the voltage controlled current source.

N-MIRROR SIMULATION

![Graph showing the relationship between input current and output current.]

**Figure 5.1** Simulation of a minimum sized n-type current mirror.

The graph of Figure 5.1 indicates that the n-mirror output transistor begins to saturate at approximately 110 µA. The output mirrors the input successfully for the MVL designs. An improvement in saturation level can be obtained by doubling the width of both n-transistors at a small cost in area. The upper limit for I_{ds} increases since I_{ds} is directly proportional to the width, W, as is evident in Equation (3.3).
5.2.2.2 P-Mirror

The minimum sized p-mirror, as with the n-mirror, mirrors its input well up to approximately 65 µA. The p-transistors begin to saturate beyond this current level. The p-mirror characteristics are illustrated in Figure 5.3. The output current, $y$, forms a straight line, indicative of its one to one relationship with the input current. Once again, a voltage controlled current source was used to generate the input current to the p-mirror. An ideal input current source could not be used directly in the simulation because it could not sufficiently raise the voltage on the drain of the input transistor which effectively disables the mirror transistor, since the gate of the mirror transistor is connected to the drain of the input transistor. This configuration can be seen in Figure 3.4 (a) where the input transistor is $T_1$ and the mirror transistor is $T_2$.

![P-MIRROR SIMULATION](image)

**Figure 5.2** Output transfer characteristics for the minimum sized p-mirror.

In the simulation of the p-mirror, the voltage was applied to the gate of an n-transistor with its source connected to $V_{ss}$ and its drain connected to the input of the p-
mirror. A load, consisting of an n-transistor configured as a resistance, was also attached to output of the p-mirror.

The output current levels for the literal, complement of literal, tswn and min MVL cells require p-mirrors with increased widths to reflect the proper logic levels. Again, as with the n-mirror, an increase in the width allows for an increase in $I_{ds}$. By doubling the width of both p-transistors the saturation current is effectively doubled. The graph of Figure 5.3 shows the linear characteristics of a 6:3 p-mirror. The p-transistors begin to saturate at approximately 110 $\mu$A, close to the saturation level for the 3:3 n-mirror.

P-MIRROR SIMULATION

![Graph showing linear characteristics of a 6:3 p-mirror.]

Figure 5.3 Simulation of a double width p-type current mirror.
5.3 HSPICE Multiple-Valued Logic Circuit Simulations

5.3.1 Binary-to-Quaternary Encoder

The binary-to-quaternary encoder simulations are provided in Figure 5.4 and Figure 5.5. The top part of each graph provides the two binary voltage inputs. The bottom part of each graph contains the output current generated. Figure 5.4, containing the input binary sequence \{00,01,11,10,00,01,11,10\}, causes the output quaternary sequence \{0,1,3,2,0,1,3,2\}. The top part of the graph provides the two binary input voltages, \(V_{in0}\) and \(V_{in1}\), the bottom part the output current \(y\). The graph reveals an output current spike at 300 ns in the downward transition from logic 3 to logic 2, which would indicate the circuit's tendency to overcompensate due to the rapid switching time. Figure 5.5 is provided to illustrate the generation of the linear quaternary output sequence \{0,1,2,3\} for the binary input sequence \{00,01,10,11\}.

The delay in the circuit simulations can help to determine the maximum operating speed for an MVL subsystem to provide proper performance without error. Both of the graphs reveal negligible delays, in the order of a few nanoseconds, compared to the much larger delays experienced by the MVL operator circuits. It is safe to ignore the delay in conversion from a binary to a quaternary system and more important to concentrate on the speed of the intermediary MVL circuits.

Close observation of the output reveals that the first logic level is closer to 18.5 \(\mu A\) rather than 20 \(\mu A\). Levels 2 and 3 are out by the same proportion, since they are factors of \(I_0\). This is a direct result of the mentioned dimension readjustments of the p-type transistor used in the n-type and p-type reference voltage circuits.
Figure 5.4  Binary-to-quaternary encoder simulation.
Figure 5.5  Binary-to-quaternary encoder generating a step output logic sequence.
5.3.2 Literal Operator

The simulation output for the literal 1X(1,1) is shown in Figure 5.6. The output $y$ is at logic 1 only when the input $x$ is at logic 1. The graph provides the expected (correct) response, but with a marked delay. HSPICE allows for accurate determination of time between events through a .MEASURE statement added to the analysis portion of a netlist. The measured rise time and fall time of the current pulse, taken at 10% and 90% of its maximum value, are 6.2 ns and 12.4 ns respectively. Perhaps more noticeable, are the switching delay times from input to output of the literal. The rising delay measured between the input rising through 10 µA and the output rising through 10 µA is 34 ns. A similar measurement, for the falling delay measured between the input rising through 30 µA and the output falling through 10 µA is much smaller at 16 ns. The threshold device with the smaller W (3 µm), used to detect logic level 1, takes more time to switch the control logic than the threshold transistor with the larger W (4 µm).

![Literal simulation graph]

**Figure 5.6** Literal operator simulation.
Figure 5.7  Simulation response of the literal operator for an input which skips through intermediate logic levels.

Figure 5.7 provides a better view of the unequal delay experienced by the threshold transistors and the delay due to the input transition from one logic level through another to its final destination. The current spike is a result of the logic transition through logic level 1 from logic level 2 to logic level 0 occurring at 300 ns.

5.3.3 Complement of Literal Operator

The simulation of the complement of literal $1\bar{x}(1,1)$ is provided in Figure 5.8. The output $y$ is logic 1 except when the input $x$ is logic 1. The largest delay experienced is the switching transition occurring at 200 ns. The measured delay from the input rising through 10 µA to the output falling through 10 µA is 34 ns.
5.3.4 Cycle Operator

The simulation of the cycle operator for $x \overset{1}{\leftarrow}$ is shown in the graph of Figure 5.9. The output $y$ is always one logic level lower than the input $x$. The graph indicates that the circuit operates correctly with two notable observations. The delay is negligible except in the input transition from logic 0 to logic 1. The measured delay, at $x$ rising through 10 µA and $y$ falling through 10 µA, is 40 ns. The lack of symmetry in the delays exhibited is not critical since the circuit cannot be operated at speeds faster than its slowest operating point. The output current levels, although within restorable limits, are out by ±5 µA depending upon the logic level. The discrepancy can be accounted for, because the input logic current levels are the ideal values, while the cycle circuit is designed to accept input current levels at a few microampere less than the desired values because of the adjusted reference voltages.
5.3.5 Tsum Operator

Both the \textit{tsum} and the \textit{min} operator experience the smallest delay times. Recall that the \textit{tsum} and the \textit{min} are essentially the same circuit. The \textit{tsum} circuit does not use any conventional logic and does not make use of any threshold transistors, both of which adversely affect delay. The delay is primarily due to the time required for the threshold transistors to reach saturation. Only a single input to the \textit{tsum}, designated by $x$, is provided. The \textit{tsum} can handle multiple inputs by simply making a metal connection of all the input signals to the same node. In Figure 5.10, only the sum of the inputs, $x$, is provided in the simulation. The sum $y$ is truncated to logic 3 when the input $x$ exceeds the current limit which represents logic 3.
It is evident from the graph that the output matches the sum of the inputs until the sum exceeds the maximum allowable logic level. As the sum approaches 90 µA the output remains at 60 µA, logic 3. The simulation presented is based on the tsum circuit diagram of Chapter 4. Alternate simulation runs, with the p-transistors sized at 3:3 as opposed to 6:3, provide the same response, with the exception that the output is limited to 50 µA and not the desired 60 µA.

5.3.6 Simulation of the Min Operator

The simulation of the min operator of Figure 5.11 validates the circuit operation. The output y follows the minimum of the two inputs $x_1$ and $x_2$. The minimum of the two inputs $x_1$ and $x_2$ is passed through to the output with little delay. The input to output delay is in the order of 10 ns.
5.3.7 Level Restorer Simulation

Due to the lack of self-restoring ability of the MVL operators, the regeneration of the signals becomes necessary. In the graph of Figure 5.12 the output $y$ restores the degraded input signal $x$ to the proper logic levels. It is unfortunate that the level restorer exhibits the largest delay time, as can be seen by the simulation in Figure 5.12. The larger delay is experienced in the input transition from logic 0 to logic 1. The measured delay is in the order of 41 ns. For the transition from logic 0 to logic 3, not provided in the graph, the delay is closer to 44 ns. Therefore, to assure valid circuit operation, the maximum switching speed would be in the order of 22 MHz, based on the delays determined by the simulation results. The current mirrors used in the MVL designs would require a level restorer to be used after every 4 to 5 stages of successive mirrors. If two sets of mirrors are isolated by digital logic gates, such as in the literal, this would count as one stage. If the more complicated cascode current mirrors were used in the
MVL designs then level restoration would only be required after 10 to 12 successive stages. A trade off in area occurs because the cascode current mirrors utilize larger transistor widths and each mirror requires two additional transistors.

**LEVEL RESTORER SIMULATION**

![Graph of level restorer simulation](image)

**Figure 5.12** Level restorer simulation.

### 5.3.8 Simulation of the Quaternary-to-Binary Decoder

Figure 5.13 details the simulation results for the conversion of a quaternary MVL signal into two binary signals. The *decoder* utilizes threshold devices similar to the level restorer circuit, but does not experience the same delay. Unlike the *level restorer*, the *decoder* does not have to convert its control signals back into current levels. The largest delay experienced, measured from the input rising through 10 µA and the output low bit \( V_{LO} \) rising through 2.5 volts, is 32 ns. It is clear from the graph that the MVL input sequence \( \{0,1,2,3\} \) produces the correct binary output sequence \( \{00,01,10,11\} \).
Figure 5.13  Simulation output of the quaternary-to-binary circuit.
6. Circuit Tests

6.1 Introductory Remarks on Testing

VLSI circuitry requires testing, among other more obvious reasons, because of potential problems which can occur in the fabrication process. Open and short circuits in the interconnection layers are some of the more dominant causes of circuit failure [2]. A greater effort should be placed on producing fault-tolerant designs and designs which incorporate self-test circuitry to facilitate and guard against potential pitfalls. To leave the testing to the end and concentrate solely on circuit function is easier in the short term, but makes testing more difficult in the long term. This is especially true as the number of inputs and outputs of binary circuits continue to increase as well as the number of possible values at the inputs and outputs for MVL circuits. The design time and cost increase with any increase in circuit testability. Another factor which tends to discourage designing with testing in mind is the increase in circuit complexity, which becomes unavoidable. In the MVL standard cell designs for this thesis, the concentration is aimed towards examining the basic functionality rather than testability of each of the cells; this is due, in part, to time constraints.

One challenge facing design in VLSI current mode circuitry, particularly the standard cells, is in its testing. The MVL operators work with very small currents, while the majority of high speed devices are centered around voltages in the millivolt to volt range. There are, as a result, more readily available (affordable) equipment designed to measure and supply these voltages rather than current. An ammeter can measure small currents but cannot provide the same valuable information obtained by observing waveforms on an oscilloscope. Fortunately, the use of a Semiconductor Parameter Analyzer is made available [25]. Unfortunately, problems with this older model of analyzer exist. The analyzer’s limited ability for supplying a variety of input waveforms and its low speed of operation, make it appropriate to test the functionality (for a given
input level does the circuit provide the correct output level?) of a VLSI circuit, but not the equally important high speed operation and timing of a circuit.

6.2 Test Equipment

The HP4145A Semiconductor Parameter Analyzer is a specialized, but limited, oscilloscope with a contained environment for performing complete IC tests with no external connections. The HP4145A can provide input signals to the device under test (DUT) and perform measurements on selected outputs. It can also perform limited time domain measurements whereby, the applied current and/or voltage signals are held constant and the response of the selected output(s) is plotted against time [25]. The Analyzer was used to plot the transfer functions of the fabricated circuits. The transfer functions enable the functionality of the circuits to be verified.

Standard lab equipment was necessary to make the circuit analysis possible, including a signal generator, oscilloscope and DC power supply. A special breadboard was designed in order to handle the 68 pin IC.

6.3 Fabricated Circuits

The standard cells to be tested were contained in two 24 pad frame circuits. The two circuits sent for fabrication arrived back in two separate sets of five 68 pin IC packages. The two circuits were labeled SKVF0 and SKVF1 for identification and each contained a subset of the designed MVL operators and supporting circuitry. No ESD (electrostatic discharge) protection was included in the design so careful handling procedures were necessary. A layout of each of the circuits used to obtain the test results is provided in Chapter 7.
6.4 Test Results

6.4.1 Reference Voltage Circuits

The value of the reference voltages shown in Figure 6.1 exceeded the desired specifications. The expected and simulated values are plotted for comparison.

Figure 6.1 Voltage reference source values for biasing of the n-type and p-type transistors.

Deviation in fabrication parameter values such as the doping concentrations, carrier mobility, insulator thickness and error in dimension tolerances accounts for the differences. Table 6.1 and Table 6.2 summarize the typical parameters used for the simulations, and those of the actual fabricated circuits for the CMOS3 DLM process, along with their percent difference. The percent difference is calculated by dividing the absolute value of the difference between the typical and actual values by the average of the two. The typical parameter values were obtained from guidelines provided by the CMC [9, 26] which would be representative of values from previous fabrication runs.
Table 6.1  Key process parameters for the CMOS3 DLM p-devices.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TYPICAL Value</th>
<th>ACTUAL Value</th>
<th>DIFFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_T$</td>
<td>-0.800 V</td>
<td>-0.806 V</td>
<td>0.75 %</td>
</tr>
<tr>
<td>$D_{t1}$</td>
<td>$5.00 \times 10^{15}$ /cm$^3$</td>
<td>$3.02 \times 10^{15}$ /cm$^3$</td>
<td>49.38 %</td>
</tr>
<tr>
<td>$T_{ox}$</td>
<td>50 nm</td>
<td>47.2 nm</td>
<td>5.76 %</td>
</tr>
<tr>
<td>$\mu$</td>
<td>250 cm$^2$/Vs</td>
<td>166 cm$^2$/Vs</td>
<td>40.38 %</td>
</tr>
</tbody>
</table>

Table 6.2  Key process parameters for the CMOS3 DLM n-devices.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TYPICAL Value</th>
<th>ACTUAL Value</th>
<th>DIFFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_T$</td>
<td>0.700 V</td>
<td>0.699 V</td>
<td>0.14 %</td>
</tr>
<tr>
<td>$D_{t1}$</td>
<td>$17 \times 10^{15}$ /cm$^3$</td>
<td>$16.5 \times 10^{15}$ /cm$^3$</td>
<td>2.99 %</td>
</tr>
<tr>
<td>$T_{ox}$</td>
<td>50 nm</td>
<td>47.2 nm</td>
<td>5.76 %</td>
</tr>
<tr>
<td>$\mu$</td>
<td>775 cm$^2$/Vs</td>
<td>565 cm$^2$/Vs</td>
<td>31.34 %</td>
</tr>
</tbody>
</table>

The new parameter, $D_{t1}$, represents the doping concentration. The oxide thickness is the same for both device types. The p-devices experience the larger percent difference in the threshold voltage and mobility, both of which are affected by the doping concentration.

6.4.2  Mirror Operation

The n-mirror output current is plotted as a function of input voltage, as illustrated in Figure 6.2. The n-mirror output exhibits characteristics indicative of the transfer characteristics of a n-transistor. The n-mirror, consisting of two n-transistors with their sources tied to ground, could not source the current supplied by the HP4145A, it required a voltage input, for the same reason as outlined in the section on simulation of the n-mirror. The output (drain) current begins to flow around $V_T$ as expected and increases proportional to the square of the input voltage. The input voltage is essentially the gate voltage on the mirror transistor.
**N-MIRROR CIRCUIT TEST**

![Graph showing output current vs. input voltage for the n-type current mirror.](image)

**Figure 6.2** Test results for the n-type current mirror.

**P-MIRROR CIRCUIT TEST**

![Graph showing output current vs. input current for the p-type current mirror.](image)

**Figure 6.3** Test results for the p-type current mirror.
The p-mirror output current is plotted as a function of input current as provided in Figure 6.3. In the case of the p-mirror, the output current is almost identical to the input current, demonstrated by the straight line graph. Based on the results it is reasonable to state that the transistors are exact mirrors of one another, and that the current source is stable.

6.4.3 Binary-to-Quaternary Encoder Circuit

The encoder circuit requires five connection points, \( V_{DD}, V_{SS}, V_{in0}, V_{in1} \) and \( y \), to perform the proper measurements. The inputs \( V_{in0} \) and \( V_{in1} \) were applied to the DUT and held constant, while the logic output level was measured. This was performed four times to include all combinations of inputs. The results are plotted in Figure 6.4 with the output current level as a function of input voltage combination.

![Binary-to-Quaternary Encoder Circuit Test](image)

**Figure 6.4** Binary-to-quaternary test results.

The current levels, which do not match the desired levels, are larger than expected. The difference is explained by the larger p-reference voltage biasing the current generating transistors in the encoder circuit.
6.4.4 Literal Circuit

The output current is plotted as a function of input current for the literal in Figure 6.5. The current pulse representing logic 1 occurs when the input is at logic 1. Two outputs are plotted y.1 and y.2 to show the effect of selecting an alternate logic transition value. The circuit producing y.1 was designed to recognize logic 1 as any current value between $0.75I_o$ and $1.5I_o$. The second circuit generating y.2 recognized logic 1 as any current value between $0.75I_o$ and $1.75I_o$. Both y.1 and y.2 should go to logic 1 at the same time but y.1 should switch to logic 0 before y.2 since the input reaches $1.5I_o$ before $1.75I_o$. The tolerances define the width of the pulse or the coverage of a logic value.

LITERAL CIRCUIT TEST

![Graph showing output current as a function of input current with two curves labeled y.1 and y.2]

Figure 6.5 Literal operator test results.

It is interesting to note the difference between y.1 and y.2 for the literal circuit, since both of these circuits exist on the same substrate and are exact duplicates, except for the aspect ratio of a single transistor. The value of y.1 and y.2 should be identical everywhere outside the input current range between 30 μA and 40 μA. From the graph it is evident that y.2 reaches 25 μA before y.1. The discrepancy could be attributed to the inherent imperfections during ion implantation or deposition during fabrication.
6.4.5 Level Restoring Circuit

The level restorer input/output transfer characteristics are provided in Figure 6.6. The restorer utilizes three n-type threshold devices, designed to detect the MVL current levels at 0.5$I_0$, 1.5$I_0$, and 2.5$I_0$. However, because the transistors were laid out with aspect ratios of 0.5, 1.6, and 2.6 respectively and the process parameters used in the design differed from those of the fabricated circuit, as noted in Table 6.1 and Table 6.2, the switching points have deviated markedly. Since the 3rd logic level for a fully restored signal was at a current value of ~65 µA, the level restorer, which switched at ~70 µA, could not perform its full function with the other circuits in this fabrication run. The same holds true for the quaternary-to-binary decoder discussed in the next section.

**LEVEL RESTORER CIRCUIT TEST**

![Graph showing output current vs. input current](image)

**Figure 6.6** Test results of the current level, restorer circuit.

6.4.6 Quaternary-to-Binary Decoder

The decoder converts a multiple-valued signal into a 2 bit binary word which acts as an interface to the binary world. The transfer characteristics are provided in Figure 6.7. The transfer functions of both binary output voltages, $V_{HB}$ and $V_{LB}$, are provided as a function of input current. $V_{LB}$ (low output bit) switched states at 8 µA, 36 µA and 69

---

*Note: The graph and figure references are placeholders and should be replaced with actual images or figures.*
μA, but was designed to switch at 10 μA, 30 μA and 50 μA. \( V_{HB} \) (high output bit) activated at 39 μA and was designed to activate at 30 μA. The discrepancies, similar to those presented for the level restorer circuit, are caused by the same phenomenon.

![Quaternary-to-Binary Decoder Circuit Test](image)

**Figure 6.7** Quaternary-to-binary circuit test results.

### 6.5 Observations

Each of the tested circuits functioned correctly. The output characteristics of the standard cell circuits are not ideal but match closely with the characteristics predicted through simulation. Process parameter variations and the rounding off of transistor dimensions contributed to the deviations experienced, especially for those standard cells which relied heavily on the threshold devices, such as the level restorer. The inappropriate sizing of a pass transistor can limit the maximum current levels obtained due to the effect of a transistor reaching maximum saturation.
7. Standard Cell Library

7.1 Definition and Description

Design experience with VLSI layouts provides an appreciation for the concept of a standard cell library. The amount of time it takes to hand place each individual transistor and route and place the p-wells, n-wells, vias, contact cuts, metal1 metal2 and polysilicon wires for an individual cell is on the order of many hours. Assuming that an average cell may contain 20 transistors and a serious VLSI circuit contains on the order of 500 000 transistors, it soon becomes evident that a manually crafted layout would be next to impossible to complete. Given the proper supporting tools, once the library of cells is complete, the design time to circuit completion, can be reduced to a few days.

A standard cell library can be defined as a collection of a class of objects, each with its own special characteristics, that, when inter-linked, can function as a completely new entity. There are two distinct levels with which a standard cell library can be viewed. The first view is that of an information resource, revolving around computers, based on high powered software for database object and visual manipulation. In this respect, a layout is reduced to nothing more than a simple binary file. The second, more important view, is that of electronic circuitry, rooted in the understanding of the electrical behaviour of individual subsystems for the development, refinement and production (i.e. synthesis) of a larger, more complete, physically operational system. Each standard cell becomes a reusable circuit component, not unlike the traditional passive electronic components of resistors capacitors and inductors.

The hardware and software views are clearly distinguishable, yet each cannot progress without the other. Often, when a model is used to represent a real system, compromises and/or approximations are introduced. In this case, the software models the hardware at every level. In order to fully utilize the potential of a standard cell library consisting of physically formed circuits, automatic placement and routing of
interconnections for the cells is essential. This automation of circuit synthesis defines the rules to which a library must conform. The standardization of cells affords the software the ability to interpret a designer's view of a system in order to generate the hardware.

7.1.1 Software

The software, as described in Chapter 6, includes an integration among three separate systems. ELECTRIC, VHDL and QUISC were used together to create and utilize a standard cell library. The process began and ended with ELECTRIC, where each cell was initially custom designed. Next VHDL was used to describe, on a qualitative level, how and what cells should be connected to create the application specific integrated circuit (ASIC). QUISC then used this VHDL code to generate a database of the ASIC, describing the physical placement and interconnections. Finally, the database was compiled within ELECTRIC to generate a full layout. The final layout is completely accessible for modification by hand, if required, because of certain limitations which can be experienced by QUISC. The two ICs containing the MVL standard cells generated by QUISC required physical adjustment because of cell overlap conflicts. Each cell had its inputs and outputs connected to an input/output pad [11], and were not interconnected amongst one another in order to facilitate testing. Due to the unusual connections (or lack thereof) of the cells, the QUISC compiler was unable to account for the extra space required between cells. Under normal use, QUISC automatically provides correct spacing for fully interconnected cells.

7.1.2 Standard Cell Library Guidelines

A standard cell library, as indicated by its name, must maintain a certain set of standards in order for it to be useable and portable. At the transistor level, there are design rules which must be followed to limit problems; if a metal or polysilicon wire is too thin, breaks during the deposition can occur; the placement of a via too close to a polysilicon wire can result in a short circuit; other rules also exist to help reduce phenomena, such as stray currents due to parasitic diode formations. At the cell level, similar rules exist in the development of a library to guard against the same type of
problems, especially to avoid short circuits and interconnection conflicts. Some of the more important guide-lines for the development of a standard cell library are listed, and are relevant to the QUISC program [29], but may equally be applied to other so-called silicon compilers.

- A cell requires its ports to be defined as either an input, output, bi-directional, power or ground.
- Access to ports from both the top and bottom is necessary.
- A minimum separation is required between ports accessible on the same edge of a cell.
- Power and ground ports must be routed on the first metal layer and be situated at the top and bottom of the cell respectively.
- All cells should have the same height.
- Cells need to be oriented horizontally to facilitate cell coupling (joining).
- P-wells need to be placed an equal distance from the cell bottom and should also be of the same height. (assumes a P-well CMOS process)

A cell is a rectangular object with a top, bottom, left and right side. Electrically, it is a circuit, such as the cycle operator, with the circuit-forming interconnected transistors sandwiched between one track of wire supplying power and another supplying ground. A good organizational view of a leaf cell, a leaf cell being the smallest distinguishable component in a library (i.e., a library can contain cells made up of two or more leaf cells), is illustrated in Figure 7.1. The dotted outline of the cell represents a fictitious boundary which QUISC creates internally to determine the spacing of cells in the placement phase. This boundary is not always adequate and care is required when placing ports near the cell boundaries because adjacent cells may be overlapped, potentially leading to connection conflicts. The height, \( H \), represents the true height of the cell. It is not clear how the placement phase of the compiler will merge cells together, but ports \( d \) and \( e \) may be too far to the right with the potential for causing a short circuit with its neighbouring cell.
Figure 7.1 Layout organization of a leaf cell.

Cells are horizontally aligned and divided into rows as required, for meeting area and interconnection restrictions. Figure 7.2 provides a view of a possible QUISC layout with two rows of cells. No interconnections are shown, but typically wires running horizontally would be on the first metal layer and those running vertically would be on the second metal layer. Notice that the cells, although the same height, are not restricted in their length. The compiler places the cells in horizontal rows lining up the power and ground rails and merging the cells.
7.2 Standard Cell Layouts

Each circuit was designed to conform to the existing standard cell library of binary logic gates. This was done to facilitate integration of the two separate levels of design tools into one library. The limitation placed on the layout of the MVL gates and circuits is that of height, which causes a reduction in area efficiency. This is also the case in any standard cell library since the height of each cell is dictated by the height of the most complex cell. The height of the MVL cells is 126X, designed to match the existing cells of the nc3lib cell library [26]. The following figures constitute the MVL library which was created to verify and test the operation of the 4-valued logic operators. The first figure, Figure 7.3, provides a legend which can be used to discern the various material layers in the VLSI layouts. The second figure, Figure 7.4, shows a CMOS binary inverter, taken from nc3lib for comparison with the MVL cells. Each of the MVL standard cells has an accompanying data sheet providing relevant characteristics and details.
Figure 7.3  Legend distinguishing the various layers in the VLSI layouts.
Figure 7.4 Layout of the binary CMOS inverter from the nc3lib standard cell library.
Figure 7.5  Layout of the binary-to-quaternary encoder.
Table 7.1 Description of the binary-to-quaternary encoder.

## Binary-to-Quaternary Encoder

### Layout Description

Encode two binary voltage inputs to a single quaternary current value

<table>
<thead>
<tr>
<th>Cell Dimensions</th>
<th>INPUTS</th>
<th>signal type</th>
<th>OUTPUTS</th>
<th>signal type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>126λ</td>
<td>In1</td>
<td>voltage</td>
<td>y</td>
</tr>
<tr>
<td>Width</td>
<td>259λ</td>
<td>In0</td>
<td>voltage</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>vdd, gnd</td>
<td>voltage</td>
<td></td>
</tr>
</tbody>
</table>

Technology: CMOS3-DLM

### Characteristics

**Electrical Parameters**

- Static Power Dissipation: n/a
- Time Delay: < 5 ns

### Operator Function (Truth Table)

<table>
<thead>
<tr>
<th>In0</th>
<th>In1</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>
Figure 7.6

Layout of the quaternary-to-binary decoder.
Table 7.2 Description of the quaternary-to-binary decoder.

<table>
<thead>
<tr>
<th>Quaternary-to-Binary Decoder</th>
</tr>
</thead>
</table>

### Layout Description
Decode a quaternary current value into two binary voltages

<table>
<thead>
<tr>
<th>Cell Dimensions</th>
<th>INPUTS signal type</th>
<th>OUTPUTS signal type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>126λ</td>
<td>Out0 voltage</td>
</tr>
<tr>
<td>Width</td>
<td>434λ</td>
<td>Out1 voltage</td>
</tr>
</tbody>
</table>

Technology: CMOS3-DLM

### Characteristics

**Electrical Parameters**

- Static Power Dissipation: n/a
- Time Delay: < 33 ns

### Operator Function (Truth Table)

<table>
<thead>
<tr>
<th>x</th>
<th>Out0</th>
<th>Out1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Figure 7.7 Layout of the n-reference voltage source.
Table 7.3  Description of the n-reference voltage source.

<table>
<thead>
<tr>
<th>N-reference Voltage Source</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Layout Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage reference for n-type transistors</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cell Dimensions</th>
<th>INPUTS signal type</th>
<th>OUTPUTS signal type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>126(\lambda)</td>
<td>vdd, gnd voltage</td>
</tr>
<tr>
<td>Width</td>
<td>52(\lambda)</td>
<td></td>
</tr>
</tbody>
</table>

Technology: CMOS3-DLM

<table>
<thead>
<tr>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical Parameters</td>
</tr>
<tr>
<td>Static Power Dissipation: 110 (\mu)W</td>
</tr>
<tr>
<td>Time Delay: n/a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operator Function (Truth Table)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value Description</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>designed</td>
</tr>
<tr>
<td>simulated</td>
</tr>
<tr>
<td>measured</td>
</tr>
</tbody>
</table>
Figure 7.8  Layout of the p-reference voltage source.
Table 7.4  Description of the p-reference voltage source.

<table>
<thead>
<tr>
<th>P-reference Voltage Source</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Layout Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage reference source for p-type transistors</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cell Dimensions</th>
<th>INPUTS</th>
<th>OUTPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>126λ</td>
<td>VDD, GND</td>
</tr>
<tr>
<td>Width</td>
<td>52λ</td>
<td>V_P_ref</td>
</tr>
</tbody>
</table>

 Technology: CMOS3-DLM  

<table>
<thead>
<tr>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical Parameters</td>
</tr>
</tbody>
</table>

- Static Power Dissipation: 250 μW
- Time Delay: n/a

<table>
<thead>
<tr>
<th>Operator Function (Truth Table)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value Description</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>designed</td>
</tr>
<tr>
<td>simulated</td>
</tr>
<tr>
<td>measured</td>
</tr>
</tbody>
</table>
Figure 7.9  Layout of a single output, one-to-one, n-transistor current mirror.
Table 7.5 Description of the n-transistor current mirror.

### N-Transistor Current Mirror

#### Layout Description

Regenerate an input current with a scale of 1:1, single input - single output

<table>
<thead>
<tr>
<th>Cell Dimensions</th>
<th>INPUTS</th>
<th>signal type</th>
<th>OUTPUTS</th>
<th>signal type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>126λ</td>
<td>x</td>
<td>y</td>
<td>current</td>
</tr>
<tr>
<td>Width</td>
<td>80λ</td>
<td>vdd, gnd</td>
<td></td>
<td>voltage</td>
</tr>
</tbody>
</table>

Technology: CMOS3-DLM

#### Characteristics

**Electrical Parameters**

- Static Power Dissipation: n/a
- Time Delay: < 5 ns

#### Operator Function (Truth Table)

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Figure 7.10  Layout of a single output, one-to-one, p-transistor current mirror.
Table 7.6 Description of the p-transistor current mirror.

<table>
<thead>
<tr>
<th>Layout Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regenerate an input current with a scale of 1:1, single input - single output</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cell Dimensions</th>
<th>INPUTS</th>
<th>signal type</th>
<th>OUTPUTS</th>
<th>signal type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>126λ</td>
<td>x</td>
<td>y</td>
<td>current</td>
</tr>
<tr>
<td>Width</td>
<td>76λ</td>
<td>vdd, gnd</td>
<td>voltage</td>
<td>current</td>
</tr>
</tbody>
</table>

Technology: CMOS3-DLM

<table>
<thead>
<tr>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical Parameters</td>
</tr>
<tr>
<td>Static Power Dissipation: n/a</td>
</tr>
<tr>
<td>Time Delay: &lt; 5 ns</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operator Function (Truth Table)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>
Figure 7.11  Layout of a three output, one-to-one, n-transistor current mirror.
Table 7.7 Description of the n-transistor three output current mirror.

<table>
<thead>
<tr>
<th>Cell Dimensions</th>
<th>INPUTS signal type</th>
<th>OUTPUTS signal type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>126Å</td>
<td>y1 current</td>
</tr>
<tr>
<td>Width</td>
<td>143Å</td>
<td>y2 current</td>
</tr>
<tr>
<td></td>
<td></td>
<td>y3 current</td>
</tr>
<tr>
<td>Technology: CMOS3-DLM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Characteristics

Electrical Parameters

Static Power Dissipation: n/a

Time Delay: < 5 ns

Operator Function (Truth Table)

<table>
<thead>
<tr>
<th>x</th>
<th>y1</th>
<th>y2</th>
<th>y3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Figure 7.12 Layout of the quaternary level restorer circuit.
Table 7.8  Description of the quaternary level restorer.

Quaternary Level Restorer

<table>
<thead>
<tr>
<th>Layout Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore the input current to its correct 4-valued logic level</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cell Dimensions</th>
<th>INPUTS signal type</th>
<th>OUTPUTS signal type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>x current</td>
<td>y current</td>
</tr>
<tr>
<td>Width</td>
<td>vdd, gnd voltage</td>
<td></td>
</tr>
<tr>
<td>Technology: CMOS3-DLM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical Parameters</td>
</tr>
</tbody>
</table>

| Static Power Dissipation: n/a |
| Time Delay: < 44 ns |

<table>
<thead>
<tr>
<th>Operator Function (Truth Table)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>~1</td>
</tr>
<tr>
<td>~2</td>
</tr>
<tr>
<td>~3</td>
</tr>
</tbody>
</table>
Figure 7.13  Layout of the literal operator (I).
Table 7.9  Description of the literal circuit, layout I.

<table>
<thead>
<tr>
<th>Literal (I)</th>
</tr>
</thead>
</table>

**Layout Description**

Generate a logic 1 if the input current is a logic 1: \( \{1X(1,1)\} \)

<table>
<thead>
<tr>
<th>Cell Dimensions</th>
<th>INPUTS</th>
<th>signal type</th>
<th>OUTPUTS</th>
<th>signal type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height 126λ</td>
<td>x</td>
<td>current</td>
<td>y</td>
<td>current</td>
</tr>
<tr>
<td>Width 308λ</td>
<td>vdd, gnd</td>
<td>voltage</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Technology: CMOS3-DLM

**Characteristics**

**Electrical Parameters**

Static Power Dissipation: n/a

Time Delay: < 35 ns

**Operator Function (Truth Table)**

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>
Figure 7.14  Layout of the literal operator (II).
Table 7.10 Description of the literal circuit, layout II.

<table>
<thead>
<tr>
<th>Literal (II)</th>
</tr>
</thead>
</table>

### Layout Description

Generate a logic 1 if the input is a logic 1: \(1X(1,1)\) (threshold adjusted)

<table>
<thead>
<tr>
<th>Cell Dimensions</th>
<th>INPUTS</th>
<th>signal type</th>
<th>OUTPUTS</th>
<th>signal type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height 126(\lambda)</td>
<td>x</td>
<td>current</td>
<td>y</td>
<td>current</td>
</tr>
<tr>
<td>Width 308(\lambda)</td>
<td>vdd, gnd</td>
<td>voltage</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Technology: CMOS3-DLM

### Characteristics

Electrical Parameters

- Static Power Dissipation: n/a
- Time Delay: < 35 ns

### Operator Function (Truth Table)

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>
Figure 7.15  Layout of the complement of literal operator (I).
Table 7.11  Description of the complement of literal circuit.

<table>
<thead>
<tr>
<th>Complement of Literal</th>
</tr>
</thead>
</table>

## Layout Description

Generate a logic 1 for every input logic level except logic 1: \( \{ \overline{1X(1,1)} \} \)

<table>
<thead>
<tr>
<th>Cell Dimensions</th>
<th>INPUTS</th>
<th>signal type</th>
<th>OUTPUTS</th>
<th>signal type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>126(\lambda)</td>
<td>x</td>
<td>current</td>
<td>y</td>
</tr>
<tr>
<td>Width</td>
<td>310(\lambda)</td>
<td>vdd, gnd</td>
<td>voltage</td>
<td>-</td>
</tr>
</tbody>
</table>

Technology: CMOS3-DLM

## Characteristics

### Electrical Parameters

- Static Power Dissipation: n/a
- Time Delay: < 35 ns

## Operator Function (Truth Table)

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
Figure 7.16  Layout of the cycle operator.
Table 7.12  Description of the cycle circuit.

<table>
<thead>
<tr>
<th>Cycle</th>
</tr>
</thead>
</table>

### Layout Description

Step the output logic level back one from the input logic level: \{ x^{−1} \}

<table>
<thead>
<tr>
<th>Cell Dimensions</th>
<th>INPUTS signal type</th>
<th>OUTPUTS signal type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>126λ</td>
<td>x current</td>
</tr>
<tr>
<td>Width</td>
<td>324λ</td>
<td>vdd, gnd voltage</td>
</tr>
</tbody>
</table>

Technology: CMOS3-DLM

### Characteristics

**Electrical Parameters**

- Static Power Dissipation: n/a
- Time Delay: < 40 ns

### Operator Function (Truth Table)

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
Figure 7.17  Layout of sk_001, the first group of MVL library cells combined into a single circuit using VHDL and QUISC.
Figure 7.18 Layout of sk_002, the second group of MVL standard cells combined into a single circuit using VHDL and QUISC.
Figure 7.19 Layout of SKVF0, the fabricated circuit containing sk_001 connected to the pad frame.
Figure 7.20  Layout of SKVF1, the fabricated circuit containing sk_002 connected to the pad frame.
8. Four-Valued Full Adder

In theory, MVL circuit design concepts are intriguing; however, unless they are tailored to perform a specific function their full value cannot be appreciated. One specific function is addition, which is a function basic to many larger systems, such as an ALU. Thus, the design of a circuit to perform the fundamental operation of addition is beneficial for realizing the potential of MVL. Two designs for the implementation of a 4-valued full adder are presented in this chapter to further demonstrate the practical application of MVL. Two approaches can be taken to implement an adder in MVL. The first approach is to base the design on the standard MVL operators by direct implementation of the defining truth table (and corresponding K-map) with the introduction of a new operator, the T-gate, to allow for a reduction in the overall design size. The second approach is to take advantage of the current summing node and tailor a circuit around this MVL element. Other similar adders have been designed such as the CMCL quaternary threshold logic full adder by Current [19]. The design presented in this thesis is a complete implementation; it will realize the entire truth table defining the addition of three radix 4 inputs. The input carry is not limited to logic 0 and logic 1 as in the standard implementation of a full adder, but can also assume the values of logic 2 and logic 3. The concept of a binary full adder comprised of two interconnected half adders is utilized. This concept is well defined and is quite readily adapted to MVL.

8.1 Full Adder Concept

The addition of two 4-valued digits to produce a sum (S) and a corresponding carry (C) can be referred to, in circuit form, as a half adder [21]. The addition of two digits, \(2_4 + 2_4 = 10_4\), for example, produces the sum \(S = 0\) and the carry \(C = 1\). The subscript 4 denotes the base of the numbers and is omitted from the addition examples provided and from the subsequent discussion.
In order to add two n-digit 4-valued numbers, represented by the symbols A and B (A = A_nA_{n-1}...A_0 and B = B_nB_{n-1}...B_0), the inclusion of the carry from the lower order digit to the higher order digit in the addition is necessary. The addition of the numbers A = 13 and B = 01 produces a carry C_1 from the addition of A_0 = 3 and B_0 = 1. By including the carry in the addition of A_1 = 1 and B_1 = 0, the correct value of 2 is obtained. This is illustrated in the following example where the sum S = A + B.

\[
\begin{array}{cccc}
1 & 1 & 3 & 0 \\
+1 & +3 & +3 & +3 \\
S = 2 & S = 0 & S = 2 & S = 3 \\
C = 0 & C = 1 & C = 1 & C = 0
\end{array}
\]

The adder circuit required to perform the addition of three separate digits, the augend, the addend and the carry is called a full adder [32]. A number of examples are provided below. The carry is normally limited to a maximum of 1, however the full adder presented was designed to add three radix 4 inputs, that is, the third input was not confined to being a carry input. The removal of the restriction on the carry input was done to increase the functionality and potential use of the circuit.

\[
\begin{array}{cccc}
 & C_1 & & 1 \\
A = A_1A_0 & 1 & 3 \\
B = B_1B_0 & +01 \\
S = S_1S_0 & 20
\end{array}
\]

\[
\begin{array}{cccc}
1 & 2 & 3 & 0 \\
1 & 1 & 3 & 0 \\
+1 & +3 & +3 & +3 \\
S = 3 & S = 2 & S = 1 & S = 3 \\
C = 0 & C = 1 & C = 2 & C = 0
\end{array}
\]
The number of possible additions for \( n \), \( r \)-valued digits is obtained by the relation \( r^n \). In the implementation of the 4-valued full adder, \( 4^3 = 64 \) separate additions (entries in an exhaustive truth table) are possible. A partial truth table is provided in Table 8.1 showing 8 of the 64 entries.

<table>
<thead>
<tr>
<th>( A_i )</th>
<th>( B_i )</th>
<th>( C_i )</th>
<th>( S_i )</th>
<th>( C_{i+1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

A flow diagram of a 3-digit parallel adder, which consists of three 4-valued full adders cascaded together, is provided in Figure 8.1. The diagram, detailing the interconnection, also illustrates how the symbols in the truth table, \( A_i \), \( B_i \), \( C_i \), \( S_i \) and \( C_{i+1} \), are related.
Concentration on \( S_i \), the sum function, is further developed in the next section to illustrate the function simplification and design method for the standard cell based design, full adder design \#1. However, both \( S_i \) and \( C_{i+1} \) are generated in the summing node design, full adder design \#2.

### 8.2 Function Minimization

It is evident from the partial truth table, Table 8.1, that minimization is required in order to develop a functional circuit to realize the sum of the full adder function. Reorganization of the truth table into a set of K-maps helps to visualize the function behaviour and provide clues for implementation. The set of K-maps demonstrating the function \( S_i \) is provided in Figure 8.3. In studying the map, it appears there are no squares which can be combined to reduce the number of terms. Each square requires a defining product term consisting of literals, except the 0s of the function. Thus, 12 terms are required to define map \( a \), the map corresponding to \( C_i = 0 \). The remaining three maps \( b, c \) and \( d \) can be accounted for by noting that each map can be obtained by adding the value of \( C_i \) of the map of interest to each square in map \( a \). For instance, the square corresponding to \((A_i = 1, B_i = 0)\) in map \( c \) contains the value 3, the value of \( C_i \) for map \( c \) is 2, the value for \((A_i = 1, B_i = 0)\) in map \( a \) is 1, therefore \( 1 + 2 = 3 \). The diagram of Figure 8.2 shows this configuration, where the block \( F(A_i, B_i) \) defines the function to realize map \( a \) which is in fact a half adder.

![Circuit scheme to produce a sum function.](image)

**Figure 8.2** Circuit scheme to produce a sum function.
Figure 8.3  K-map of $S_i$, the sum portion of a full adder.
The function $F(A_i, B_i)$ can be realized with a logic circuit utilizing three cycle operators and a $T$-gate [23], instead of the less efficient circuit comprised of twelve sets of literal terms (min of two literals) and a tsum gate. An examination of map $a$ in Figure 8.3 reveals the following set of propositions:

- If $B_i = 0$ then $S_i = A_i$
- If $B_i = 1$ then $S_i = A_i^{1/2}$
- If $B_i = 2$ then $S_i = A_i^{2/3}$
- If $B_i = 3$ then $S_i = A_i^{3/4}$

The implementation of this set of rules requires three separate cycle functions and a $T$-gate. The $T$-gate is essentially a 4-valued multiplexer not unlike its digital counterpart.

**8.2.1 T-Gate Operation**

The 4-valued $T$-gate selects one of its four possible inputs and passes only that input through. The decision is based on a separate 4-valued control input. The circuit diagram of Figure 8.4 provides the detailed interconnection of the $T$-gate. The threshold transistors $T_1$, $T_2$ and $T_3$ are sized to detect logic 1, 2 and 3 respectively. Each of the four inputs, $x_1$, $x_2$, $x_3$ and $x_4$, are connected to a series of pass transistors. The pass transistors are interconnected in such a way as to restrict flow from the other three inputs. The detection of each logic level is not independent, that is, if the input, $x$, is at logic 3 then logic 1 and logic 2 are also detected and their corresponding p-type transistors are switched on; however, their corresponding n-type transistors are turned off, effectively disabling their matching inputs from reaching the output. Therefore, if logic 3 is detected, then the voltages at node $c$, $b$, and $a$ are all high; the effect is that only input $x_4$ is passed through to the output.
8.3 Circuit Realizations of the Half Adder

The two circuits to be presented are referred to as half adder design #1, the Standard Cell Adder (SCA), and half adder design #2, the Summing Node Adder (SNA).

8.3.1 Design #1 - Standard Cell Half Adder

The SCA design is provided in Figure 8.5 and is a direct implementation of the propositions describing map $a$ in Figure 8.3, $S_i = A_i + B_i$. The n-mirror connected to the input $A_i$ replicates the current so that each of the cycle operators can function with its own copy of the input. The select input on the $T$-gate is controlled by the input $B_i$. The p-mirrors are used to ensure current flows from the $T$-gate output. The full adder is obtained by connecting the SCA in the configuration shown in Figure 8.2.
8.3.2 Design #2 - Summing Node Adder

The SNA circuit diagram is shown in Figure 8.7. The operation relies on passing the direct addition of $A_i$ and $B_i$, via a summing node, through to the output $S$. If the sum exceeds logic level 3, then the pass transistor $T_2$ is turned off inhibiting the flow and $T_3$ is turned on activating the carry output $C_i$. A p-type threshold device, transistor $T_1$, is sized to detect a logic 4. The binary NOR gates are used to determine the output level when the natural sum (sum via the summing node) exceeds logic 3. The inputs to the binary gates are generated via the threshold devices; $T_6$ and $T_7$ detect logic 2 and 3 for input $A_i$, $T_9$ and $T_8$ detect logic 2 and 3 for input $B_i$.

The configuration of NOR gates controlling $T_3$ determines if either $A_i$ or $B_i$ is at logic 3 and the other input is at least at logic 2. If the condition is met, then $T_3$ is turned on and $I_o$ is added to the output. The single NOR gate detects the condition when both
inputs are at logic 3 and turn on $T_4$, adding an additional $I_o$ to the output. Thus, the sum is logic 2 when both $A_i$ and $B_i$ are at logic 3; 0 passing through $T_2$, $I_o$ passing through $T_3$ and $I_o$ passing through $T_4$. The n-type and p-type mirrors are used to replicate and change the current directions to achieve the sum and carry outputs. The full adder is obtained by combining the SNA in the configuration depicted in Figure 8.6. The carry outputs are simply joined together forming a summing node. The carry output will produce the correct value for every possible input combination. Simulation results for both full adders are provided in the next section.

![Figure 8.6](image)

**Figure 8.6** Full adder configuration using the summing node half adder.
Figure 8.7  Half Adder (SNA) including both the Sum and Carry outputs.
8.4 Simulations of the Full Adder Designs

The SCA full adder circuit was first simulated with a MVL simulator before performing the HSPICE circuit level simulation. The logic simulator confirmed the adder's functionality. The electrical simulation verified the circuit operation. The graph of Figure 8.8 shows how the SCA full adder provided the correct addition of the two inputs $A_i$ and $B_i$ plus the carry $C_i$. The delay is primarily a result of the delay experienced by the cycle operator.

The SNA full adder circuit, because it contains a mixture of logic elements and the more basic MVL building blocks (switches, thresholds, and mirrors), was not simulated with a logic simulator; however, the more important circuit level simulation was performed. Both outputs, the sum $S_i$ and carry $C_{i+1}$, are graphed along with the inputs $A_i$, $B_i$ and $C_i$ in Figure 8.9. The simulation provides good results with the sum and carry exhibiting marginal delays.

**Figure 8.8** Full adder (SCA) simulation, with no carry output.
Figure 8.9 Full adder (SNA) simulation, with carry.

Both graphs are equivalent, that is, the inputs for both simulations are the same in order to provide a comparison between both adder designs. The SNA output signal S provides a more precise response with a smaller delay compared to the SCA output signal. At 100 ns the delay in the SCA output S is almost double that of the SNA. Also S does not fully return to 0 between 600 ns and 700 ns in the SCA, this is not the case in the SNA. However, both full adders work within tolerated limits and each can be utilized within a larger MVL design.

8.4.1 Comparing Multiple-Valued and Binary Logic

The full adder designs, if optimized, can help illustrate the potential reduction in area that is obtainable through the increased information content offered by MVL. A comparison can be made between a binary full adder and a quaternary full adder. The
standard definition of a full adder limits the carry to the values 0 and 1 [19]. Therefore, an optimized quaternary full adder design for producing a sum and carry for the addition of two 4-valued variables consists of 20 transistors. A combinational binary full adder (equivalently a transmission gate full adder [2]) cascaded with a second binary full adder to produce the same sum and carry consists of 48 transistors in total. This clearly illustrates a 50% reduction in the number of transistors required to perform quaternary addition. The area reduction achieved depends upon the average transistor sizes used in each design, in this case, the average sizes are equal. Thus, the reduction in the number of transistors corresponds to a 50% reduction in area. The comparison is valid for the inclusion of the individual adders within a larger system, the MVL adder within a MVL subsystem and the binary adder within a binary system. That is, the reduction in area is lost if the quaternary full adder is used directly within a binary system since it takes almost 30 transistors in total to perform a binary-to-quaternary and quaternary-to-binary conversion. A more concrete example is the MVL based CMOS design of a 32 x 32 bit signed digit multiplier designed and fabricated by Kameyama et al [16]. The multiplier, while maintaining comparable speed, reduced both the area and power dissipation to almost half that of an equivalent CMOS binary multiplier. This novel design further demonstrates the "effectiveness" of implementing MVL in VLSI.
9. Conclusion

This thesis examined the relation between VLSI and MVL through the design and test of a standard cell library. The MVL circuits gathered from a number of sources, including original design, were first simulated, laid out and refined. The circuits were then validated through re-simulation, fabrication and finally testing. A complete view of MVL with insight and recommendations throughout each stage in the research was provided to further assist in the development of novel MVL systems, demonstrated through the introduction of two new designs for a 4-valued full adder. The designs verified that MVL design is not purely abstract and does have practical application.

9.1 Summary

The fabricated circuits are CMOS realizable, based on the employment of four discrete levels of current to represent the individual logic levels. Each MVL operator and supporting circuitry was treated as an individual cell. The layout of each cell conformed to the nc3lib CMOS3 DLM binary logic standard cell library, with some of the operators incorporating binary logic gates (inverters, nand and nor gates) into their design. The cells laid out for fabrication included the binary-to-quaternary encoder, literal, complement of literal, cycle, level restorer, n-mirror, p-mirror and the quaternary-to-binary decoder. The cells, investigated through simulation, included the fabricated cells as well as the tsum, min, and max operators. The fabricated circuits functioned as predicted through simulation, with some deviation in results.

Given the operability of the quaternary logic, some factors of concern are evident. MVL circuits suffer from a inter-logic level delay which binary circuits are immune to. A binary circuit switches between two states so detection is straight forward (not an issue). In MVL, threshold detection is unavoidable and becomes a critical issue. Detecting a logic level can introduce unequal added delay as a result of the incorrect detection of an intermediary logic level as a signal jumps from logic 0 to logic 3, for example. The effect is that it limits the maximum speed with which a quaternary operator can work. The
behaviour of the threshold devices themselves contributes to this delay. The MVL operators are not self-restoring, thereby requiring signal restoration at successive stages [17,19]. Static power dissipation exists, which could prohibit the use of current mode MVL from certain applications. Although static power is a concern, overall power of an MVL system may still be less than an equivalent binary system due to the increase in information content. One other potential problem can occur with the sizing of pass transistors (switches) and transistors comprising the current mirrors. For example, if a transistor saturates at a level below the maximum logic level attempting to be conveyed, then it will lead to output truncation which can be misconstrued as signal deterioration. It is important to recognize and size these transistors to pass all logic levels without truncation. In fact, in certain instances, the \( t_{sum} \) function can be implemented with a simple pass transistor sized accordingly to saturate at the maximum logic level.

The design of the MVL operators in CMOS technology introduces another problem with respect to design tolerances. The need for two separate reference voltages to bias the n-devices and the p-devices results in a few of the circuits, for example the level restorer operating partially out of range, due to constraints on layout transistor sizing precision and the use of typical values of the process parameters as opposed to actual values. Therefore, for practical design, it is important to have full prior knowledge of the stringent requirements of the fabrication facility and allow for variations during processing which are beyond a designer's control. Design in higher radix logic could prove to be too complex, given the difficulties which are experienced with the quaternary cells.

MVL design utilizing the basic operators and/or the basic components including switches, summing nodes, constants, current mirrors and threshold devices has practical application. The circuit designs for the two quaternary full adders along with their electrical simulations provides two views for approaching design with MVL, standard cell and transistor level based design. The simulation results verified the operation of each adder. Both adder circuits are useable in practical MVL design, with the summing node based design providing better performance. An argument for the utilization of MVL in
Certain applications was made through the comparison of the optimized quaternary full adder and the signed digit multiplier with their binary equivalents.

Logic, the process of breaking down seemingly complex matters into their basic elements, is not limited to two separate states. The implementation of a number of 4-valued operators in CMOS technology, current mode systems incorporating voltage mode control, challenges the traditional binary view of logic. Each of the subsets of MVL operators fabricated proved valid and provided reasonable comparability to their traditional counterparts. The integration of the MVL cells with the binary standard cell library has positive implications with respect to incorporating MVL into binary systems.

9.2 Future Areas of Study

Only a subset of one complete set of MVL operators was laid out in the interest of preliminary testing and verification. First, it would be beneficial to adjust the existing cells to account for differences between the typical and actual fabrication parameters, and second, to complete the set to form a full quaternary standard cell library, maintaining compatibility with the existing binary logic standard cells.

The hybrid technology, BiCMOS, can offer significant advantages in speed improvement [10] at a potential increase in power consumption. Further examination is necessary, but the creation of a 4-valued logic standard cell library for a BiCMOS process is advantageous for the advancement of MVL.

The re-emergence of MVL has seen only a handful of practical circuits designed and reported on. The full scale design of an ASIC utilizing the MVL algebra along with the associated logic reduction and combination techniques would allow the MVL standard cell library to be fully utilized. An investigation must be part of the design and involve guidelines and recommendations as to where the MVL circuits could best be utilized to tackle the problems of pin limitations, area constraints and interconnection nightmares.
References


Appendix I

MOS Device Equation

A review of the current-voltage relationship for an enhancement FET (Field Effect Transistor) in saturation is provided along with some typical values of the device parameters.

\[ I_{ds} = \frac{\mu \varepsilon}{2t_{ox}} \left( \frac{W}{L} \right) (V_{gs} - V_t)^2 \left( 1 + \lambda V_{ds} \right) \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Typical Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mu )</td>
<td>majority carrier drift mobility</td>
<td>775 cm(^2)/vs</td>
</tr>
<tr>
<td>( \varepsilon )</td>
<td>permittivity of the gate insulator</td>
<td>0.354 pF/cm</td>
</tr>
<tr>
<td>( t_{ox} )</td>
<td>thickness of the gate insulator</td>
<td>50 nm</td>
</tr>
<tr>
<td>( V_t )</td>
<td>device threshold voltage</td>
<td>0.7 volts</td>
</tr>
<tr>
<td>( W )</td>
<td>transistor channel width</td>
<td>3 ( \mu )m</td>
</tr>
<tr>
<td>( L )</td>
<td>transistor channel length</td>
<td>3 ( \mu )m</td>
</tr>
</tbody>
</table>

![Side View](image1)

![Top View](image2)

118
Appendix II

HSPICE Listing

This HSPICE listing is provided to provide an example of the netlist input data used to generate the simulation graphs presented.

************* Hspice Listing for the Literal (I) Circuit Simulation *************
**********************************************************************
*** literal (x(1,1), output = 0 or Io) ***
*** facet literal1 from library mvllib ***
*** version 1
*** extracted by electric vlsi design system, version 4.05
*** uc spice ***, min_resist 50.000000, min_capac 0.040000ff

* northern telecom 3 micron cmos process
* parameters expressed in terms of real microns

.options post nomod nopage

.options defl=3um defw=3um
+ limpts=20000 itl3=10 itl4=30 itl5=40000
+ lvltim=2 method=gear gmin=1.e-10
+ absto1=10pa vnto1=10uv

.model n nmos ( level=3 vto=0.7 kp=40e-6 gamma=1.1 phi=0.6
+ lambda=1.0e-2 pb=0.7 cgso=3.e-10 cgdo=3.e-10 cgbo=5.0e-10
+ rsh=25 cj=4.4e-4 mj=0.5 cjsw=4e-10 mjsw=0.3 js=1.0e-5
+ tox=5.0e-8 nsub=1.7e+16 tpg=1 xj=6.0e-7 ld=3.5e-7 uo=775
+ rd=40 rs=40 nss=0 nfs=0 vmax=1.0e5 )

.model p pmos ( level=3 vto=-0.8 kp=12.e-6 gamma=0.6 phi=0.6
+ lambda=3.0e-2 pb=0.6 cgso=2.5e-10 cgdo=2.5e-10 cgbo=5.0e-10
+ rsh=80 cj=1.5e-4 mj=0.6 cjsw=4.e-10 mjsw=0.6 js=1.0e-5
+ tox=5.0e-8 nsub=5.0e+15 tpg=1 xj=5.e-7 ld=2.50e-7 uo=250
+ rd=100 rs=100 nss=0 nfs=0 vmax=0.7e5 )

* mosfet element - statement variables
* l=channel length,w=channel width
* ps=perimeter of the source junction,pd=perimeter of the drain junction
* as=source diffusion area, ad=drain diffusion area

119
.subckt literal 8 3 2
** pin 8: y
** pin 3: vdd
** pin 2: x
*** component n-transistor:
m1 4 6 0 0 n l=3.00u w=3.00u as=28.44p ad=14.52p ps=20.22u pd=10.40u
*** component n-transistor:
m2 6 9 0 0 n l=3.00u w=3.00u as=28.44p ad=18.18p ps=20.22u pd=13.20u
*** component p-transistor:
m3 6 9 3 3 p l=3.00u w=6.00u as=33.86p ad=23.40p ps=19.12u pd=13.80u
*** component p-transistor:
m4 12 6 3 3 p l=3.00u w=6.00u as=33.86p ad=30.60p ps=19.12u pd=16.20u
*** component p-transistor:
m5 3 2 9 3 p l=3.00u w=6.00u as=23.40p ad=13.80u ps=19.12u
*** component p-transistor:
m6 7 2 3 3 p l=3.00u w=6.00u as=33.86p ad=23.40p ps=19.12u pd=13.80u
*** component p-transistor:
m7 3 2 2 3 p l=3.00u w=6.00u as=11.70p ad=33.86p ps=6.90u pd=19.12u
*** component p-transistor:
m8 4 7 12 3 p l=3.00u w=6.00u as=30.60p ad=46.80p ps=16.20u pd=27.60u
*** component p-transistor:
m9 3 1 1 3 p l=3.00u w=6.00u as=15.60p ad=33.86p ps=9.20u pd=19.12u
*** component p-transistor:
m10 8 1 3 3 p l=3.00u w=6.00u as=33.86p ad=46.80p ps=19.12u pd=27.60u
*** component n-transistor:
m11 0 7 4 0 n l=3.00u w=3.00u as=14.52p ad=28.44p ps=10.40u pd=20.22u
*** component n-transistor:
m12 11 10 0 0 n l=3.00u w=3.00u as=12.60p ad=16.20u ps=20.22u pd=11.40u
*** component n-transistor:
m13 1 4 11 0 n l=3.00u w=6.00u as=12.60p ad=54.00p ps=11.40u pd=30.00u
*** component n-transistor:
m14 7 10 0 0 n l=3.00u w=4.20u as=28.44p ad=19.62p ps=20.22u pd=13.20u
*** component n-transistor:
m15 0 10 9 0 n l=4.20u w=3.00u as=18.18p ad=28.44p ps=13.20u pd=20.22u
*** component p-transistor:
m16 3 10 10 3 p l=5.40u w=3.00u as=18.63p ad=33.86p ps=13.50u pd=19.12u
*** component n-transistor:
m17 10 10 0 0 n l=3.00u w=3.00u as=28.44p ad=6.06p ps=20.22u pd=4.40u
** extracted parasitic elements:
c1 10 0 42.49f
c2 29 0 21.04f
c3 3 8 0 4.69f
c4 7 0 24.22f
c5 6 0 18.52f
c6 4 0 19.79f
.subckt n_mirror 4 2 1
  ** pin 4: vdd
  ** pin 2: x
  ** pin 1: y
  *** component n-transistor:
  ml 0 2 0 n 1=3.00u w=3.00u as=12.72p ad=23.58p ps= 9.20u pd=16.80u
  *** component n-transistor:
  m2 1 2 0 0 n 1=3.00u w=3.00u as=23.58p ad=38.16p ps=16.80u pd=27.60u
  ** extracted parasitic elements:
  c1 4 0   8.07f
  c2 2 0   20.24f
  c3 1 0   6.52f
.ends n_mirror

*=============================================================================*
* voltage and/or current sources defined
*
* single step input, 5ns transition
ia 1 0 pwl 0ns 0ua, 200ns 0ua, 205ns 20ua, 400ns 20ua, 405ns 40ua,
+       600ns 40ua, 605ns 60ua, 800ns 60ua

vdd 3 0 dc 5 $ vdd

*=============================================================================*
* meters, subckt and component connections
*
vom 8 9 dc 0  $ current meter - for output current

x1 8 3 1   literal1
x2 3 9 11  n_mirror   $ n_mirror as load
x3 11 3 12  literal1   $ connect to the input of a second literal

*=============================================================================*
* plot or graph and measurement statements
*
.meASURE tran rtimedlay trig i(ia) val=10ua rise=1
+         targ i(vom) val=10ua rise=1
.meASURE tran ftimedlay trig i(ia) val=30ua rise=1
+         targ i(vom) val=10ua fall=1
.meASURE tran risetime  trig i(vom) val=1.78ua rise=1
targ i(vom) val=16.02ua rise=1
.measure tran falltime trig i(vom) val=16.02ua fall=1
+ targ i(vom) val=1.78ua fall=1
.print tran 'x'=i(ia) 'y'=i(vom) (0,80u)

*---------------------------------------------------------------
* transient analysis time
* 
.trans lns 800ns

.width out=80
.end

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