A Novel Current-Sensing Completion-Detection Circuit Adapted to the Micropipeline Methodology

by

Michael J. Gamble

A thesis
presented to the University of Manitoba
in partial fulfilment of the
requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba
Canada

© 1994, Michael J. Gamble

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

THE AUTHOR HAS GRANTED AN IRREVOCABLE NON-EXCLUSIVE LICENCE ALLOWING THE NATIONAL LIBRARY OF CANADA TO REPRODUCE, LOAN, DISTRIBUTE OR SELL COPIES OF HIS/HER THESIS BY ANY MEANS AND IN ANY FORM OR FORMAT, MAKING THIS THESIS AVAILABLE TO INTERESTED PERSONS.

L'AUTEUR A ACCORDE UNE LICENCE IRREVOCABLE ET NON EXCLUSIVE PERMETTANT A LA BIBLIOTHEQUE NATIONALE DU CANADA DE REPRODUIRE, PRETER, DISTRIBUER OU VENDRE DES COPIES DE SA THESE DE QUELQUE MANIERE ET SOUS QUELQUE FORME QUE CE SOIT POUR METTRE DES EXEMPLAIRES DE CETTE THESE A LA DISPOSITION DES PERSONNE INTERESSEES.

THE AUTHOR RETAINS OWNERSHIP OF THE COPYRIGHT IN HIS/HER THESIS. NEITHER THE THESIS NOR SUBSTANTIAL EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT HIS/HER PERMISSION.

L'AUTEUR CONSERVE LA PROPRIETE DU DROIT D'AUTEUR QUI PROTEGE SA THESE. NI LA THESE NI DES EXTRAITS SUBSTANTIELS DE CELLE-CI NE DOIVENT ETRE IMPRIMES OU AUTREMENT REPRODUITS SANS SON AUTORISATION.

ISBN 0-315-99108-9

Canada

Name _____

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

ELECTRONICS ENGINEERING

SUBJECT TERM

0544

SUBJECT CODE

U·M·I

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS

Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION

General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523
Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS

Language
- General 0679
- Ancient 0289
- Linguistics 0290
- Modern 0291

Literature
- General 0401
- Classical 0294
- Comparative 0295
- Medieval 0297
- Modern 0298
- African 0316
- American 0591
- Asian 0305
- Canadian (English) 0352
- Canadian (French) 0355
- English 0593
- Germanic 0311
- Latin American 0312
- Middle Eastern 0315
- Romance 0313
- Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY

Philosophy 0422
Religion
- General 0318
- Biblical Studies 0321
- Clergy 0319
- History of 0320
- Philosophy of 0322

Theology 0469

SOCIAL SCIENCES

American Studies 0323
Anthropology
- Archaeology 0324
- Cultural 0326
- Physical 0327

Business Administration
- General 0310
- Accounting 0272
- Banking 0770
- Management 0454
- Marketing 0338

Canadian Studies 0385
Economics
- General 0501
- Agricultural 0503
- Commerce-Business 0505
- Finance 0508
- History 0509
- Labor 0510
- Theory 0511

Folklore 0358
Geography 0366
Gerontology 0351
History
- General 0578
- Ancient 0579
- Medieval 0581
- Modern 0582
- Black 0328
- African 0331
- Asia, Australia and Oceania 0332
- Canadian 0334
- European 0335
- Latin American 0336
- Middle Eastern 0333
- United States 0337

History of Science 0585
Law 0398
Political Science
- General 0615
- International Law and Relations 0616
- Public Administration 0617

Recreation 0814
Social Work 0452
Sociology
- General 0626
- Criminology and Penology 0627
- Demography 0938
- Ethnic and Racial Studies 0631
- Individual and Family Studies 0628
- Industrial and Labor Relations 0629
- Public and Social Welfare 0630
- Social Structure and Development 0700
- Theory and Methods 0344

Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES

Agriculture
- General 0473
- Agronomy 0285
- Animal Culture and Nutrition 0475
- Animal Pathology 0476
- Food Science and Technology 0359
- Forestry and Wildlife 0478
- Plant Culture 0479
- Plant Pathology 0480
- Plant Physiology 0817
- Range Management 0777
- Wood Technology 0746

Biology
- General 0306
- Anatomy 0287
- Biostatistics 0308
- Botany 0309
- Cell 0379
- Ecology 0329
- Entomology 0353
- Genetics 0369
- Limnology 0793
- Microbiology 0410
- Molecular 0307
- Neuroscience 0317
- Oceanography 0416
- Physiology 0433
- Radiation 0821
- Veterinary Science 0778
- Zoology 0472

Biophysics
- General 0786
- Medical 0760

EARTH SCIENCES

Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES

Environmental Sciences 0768
Health Sciences
- General 0566
- Audiology 0300
- Chemotherapy 0992
- Dentistry 0567
- Education 0350
- Hospital Management 0769
- Human Development 0758
- Immunology 0982
- Medicine and Surgery 0564
- Mental Health 0347
- Nursing 0569
- Nutrition 0570
- Obstetrics and Gynecology 0380
- Occupational Health and Therapy 0354
- Ophthalmology 0381
- Pathology 0571
- Pharmacology 0419
- Pharmacy 0572
- Physical Therapy 0382
- Public Health 0573
- Radiology 0574
- Recreation 0575
- Speech Pathology 0460
- Toxicology 0383

Home Economics 0386

PHYSICAL SCIENCES

Pure Sciences

Chemistry
- General 0485
- Agricultural 0749
- Analytical 0486
- Biochemistry 0487
- Inorganic 0488
- Nuclear 0738
- Organic 0490
- Pharmaceutical 0491
- Physical 0494
- Polymer 0495
- Radiation 0754

Mathematics 0405
Physics
- General 0605
- Acoustics 0986
- Astronomy and Astrophysics 0606
- Atmospheric Science 0608
- Atomic 0748
- Electronics and Electricity 0607
- Elementary Particles and High Energy 0798
- Fluid and Plasma 0759
- Molecular 0609
- Nuclear 0610
- Optics 0752
- Radiation 0756
- Solid State 0611

Statistics 0463

Applied Sciences

Applied Mechanics 0346
Computer Science 0984
Engineering
- General 0537
- Aerospace 0538
- Agricultural 0539
- Automotive 0540
- Biomedical 0541
- Chemical 0542
- Civil 0543
- Electronics and Electrical 0544
- Heat and Thermodynamics 0348
- Hydraulic 0545
- Industrial 0546
- Marine 0547
- Materials Science 0794
- Mechanical 0548
- Metallurgy 0743
- Mining 0551
- Nuclear 0552
- Packaging 0549
- Petroleum 0765
- Sanitary and Municipal 0554
- System Science 0790

Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY

General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

Nom _____

Dissertation Abstracts International est organisé en catégories de sujets. Veuillez s.v.p. choisir le sujet qui décrit le mieux votre thèse et inscrivez le code numérique approprié dans l'espace réservé ci-dessous.

_____ ☐☐☐ U·M·I

SUJET — CODE DE SUJET

Catégories par sujets

HUMANITÉS ET SCIENCES SOCIALES

COMMUNICATIONS ET LES ARTS
Architecture 0729
Beaux-arts 0357
Bibliothéconomie 0399
Cinéma 0900
Communication verbale 0459
Communications 0708
Danse 0378
Histoire de l'art 0377
Journalisme 0391
Musique 0413
Sciences de l'information 0723
Théâtre 0465

ÉDUCATION
Généralités 515
Administration 0514
Art 0273
Collèges communautaires 0275
Commerce 0688
Économie domestique 0278
Éducation permanente 0516
Éducation préscolaire 0518
Éducation sanitaire 0680
Enseignement agricole 0517
Enseignement bilingue et multiculturel 0282
Enseignement industriel 0521
Enseignement primaire. 0524
Enseignement professionnel 0747
Enseignement religieux 0527
Enseignement secondaire 0533
Enseignement spécial 0529
Enseignement supérieur 0745
Évaluation 0288
Finances 0277
Formation des enseignants 0530
Histoire de l'éducation 0520
Langues et littérature 0279
Lecture 0535
Mathématiques 0280
Musique 0522
Orientation et consultation 0519
Philosophie de l'éducation 0998
Physique 0523
Programmes d'études et enseignement 0727
Psychologie 0525
Sciences 0714
Sciences sociales 0534
Sociologie de l'éducation 0340
Technologie 0710

LANGUE, LITTÉRATURE ET LINGUISTIQUE
Langues
- Généralités 0679
- Anciennes 0289
- Linguistique 0290
- Modernes 0291

Littérature
- Généralités 0401
- Anciennes 0294
- Comparée 0295
- Médiévale 0297
- Moderne 0298
- Africaine 0316
- Américaine 0591
- Anglaise 0593
- Asiatique 0305
- Canadienne (Anglaise) 0352
- Canadienne (Française) 0355
- Germanique 0311
- Latino-américaine 0312
- Moyen-orientale 0315
- Romane 0313
- Slave et est-européenne 0314

PHILOSOPHIE, RELIGION ET THÉOLOGIE
Philosophie 0422
Religion
- Généralités 0318
- Clergé 0319
- Études bibliques 0321
- Histoire des religions 0320
- Philosophie de la religion 0322

Théologie 0469

SCIENCES SOCIALES
Anthropologie
- Archéologie 0324
- Culturelle 0326
- Physique 0327

Droit 0398
Économie
- Généralités 0501
- Commerce-Affaires 0505
- Économie agricole 0503
- Économie du travail 0510
- Finances 0508
- Histoire 0509
- Théorie 0511

Études américaines 0323
Études canadiennes 0385
Études féministes 0453
Folklore 0358
Géographie 0366
Gérontologie 0351
Gestion des affaires
- Généralités 0310
- Administration 0454
- Banques 0770
- Comptabilité 0272
- Marketing 0338

Histoire
- Histoire générale 0578
- Ancienne 0579
- Médiévale 0581
- Moderne 0582
- Histoire des noirs 0328
- Africaine 0331
- Canadienne 0334
- États-Unis 0337
- Européenne 0335
- Moyen-orientale 0333
- Latino-américaine 0336
- Asie, Australie et Océanie 0332

Histoire des sciences 0585
Loisirs 0814
Planification urbaine et régionale 0999
Science politique
- Généralités 0615
- Administration publique 0617
- Droit et relations internationales 0616

Sociologie
- Généralités 0626
- Aide et bien-être social 0630
- Criminologie et établissements pénitentiaires 0627
- Démographie 0938
- Études de l' individu et de la famille 0628
- Études des relations interethniques et des relations raciales 0631
- Structure et développement social 0700
- Théorie et méthodes. 0344
- Travail et relations industrielles 0629

Transports 0709
Travail social 0452

SCIENCES ET INGÉNIERIE

SCIENCES BIOLOGIQUES
Agriculture
- Généralités 0473
- Agronomie. 0285
- Alimentation et technologie alimentaire 0359
- Culture 0479
- Élevage et alimentation 0475
- Exploitation des pâturages 0777
- Pathologie animale 0476
- Pathologie végétale 0480
- Physiologie végétale 0817
- Sylviculture et faune 0478
- Technologie du bois 0746

Biologie
- Généralités 0306
- Anatomie 0287
- Biologie (Statistiques) 0308
- Biologie moléculaire 0307
- Botanique 0309
- Cellule 0379
- Écologie 0329
- Entomologie 0353
- Génétique 0369
- Limnologie 0793
- Microbiologie 0410
- Neurologie 0317
- Océanographie 0416
- Physiologie 0433
- Radiation 0821
- Science vétérinaire 0778
- Zoologie 0472

Biophysique
- Généralités 0786
- Medicale 0760

SCIENCES DE LA TERRE
Biogéochimie 0425
Géochimie 0996
Géodésie 0370
Géographie physique 0368
Géologie 0372
Géophysique 0373
Hydrologie 0388
Minéralogie 0411
Océanographie physique 0415
Paléobotanique 0345
Paléoécologie 0426
Paléontologie 0418
Paléozoologie 0985
Palynologie 0427

SCIENCES DE LA SANTÉ ET DE L'ENVIRONNEMENT
Économie domestique 0386
Sciences de l'environnement 0768
Sciences de la santé
- Généralités 0566
- Administration des hôpitaux 0769
- Alimentation et nutrition 0570
- Audiologie 0300
- Chimiothérapie 0992
- Dentisterie 0567
- Développement humain 0758
- Enseignement 0350
- Immunologie 0982
- Loisirs 0575
- Médecine du travail et thérapie 0354
- Médecine et chirurgie 0564
- Obstétrique et gynécologie 0380
- Ophtalmologie 0381
- Orthophonie 0460
- Pathologie 0571
- Pharmacie 0572
- Pharmacologie 0419
- Physiothérapie 0382
- Radiologie 0574
- Santé mentale 0347
- Santé publique 0573
- Soins infirmiers 0569
- Toxicologie 0383

SCIENCES PHYSIQUES
Sciences Pures
Chimie
- Généralités 0485
- Biochimie 487
- Chimie agricole 0749
- Chimie analytique 0486
- Chimie minérale 0488
- Chimie nucléaire 0738
- Chimie organique 0490
- Chimie pharmaceutique 0491
- Physique 0494
- PolymÈres 0495
- Radiation 0754

Mathématiques 0405
Physique
- Généralités 0605
- Acoustique 0986
- Astronomie et astrophysique 0606
- Électronique et électricité 0607
- Fluides et plasma 0759
- Météorologie 0608
- Optique 0752
- Particules (Physique nucléaire) 0798
- Physique atomique 0748
- Physique de l'état solide 0611
- Physique moléculaire 0609
- Physique nucléaire 0610
- Radiation 0756

Statistiques 0463

Sciences Appliqués Et Technologie
Informatique 0984
Ingénierie
- Généralités 0537
- Agricole 0539
- Automobile 0540
- Biomédicale 0541
- Chaleur et thermodynamique 0348
- Conditionnement (Emballage) 0549
- Génie aérospatial 0538
- Génie chimique 0542
- Génie civil 0543
- Génie électronique et électrique 0544
- Génie industriel 0546
- Génie mécanique 0548
- Génie nucléaire 0552
- Ingénierie des systèmes 0790
- Mécanique navale 0547
- Métallurgie 0743
- Science des matériaux 0794
- Technique du pétrole 0765
- Technique minière 0551
- Techniques sanitaires et municipales 0554
- Technologie hydraulique 0545

Mécanique appliquée 0346
Géotechnologie 0428
Matières plastiques (Technologie) 0795
Recherche opérationnelle 0796
Textiles et tissus (Technologie) 0794

PSYCHOLOGIE
Généralités 0621
Personnalité 0625
Psychobiologie 0349
Psychologie clinique 0622
Psychologie du comportement 0384
Psychologie du développement 0620
Psychologie expérimentale 0623
Psychologie industrielle 0624
Psychologie physiologique 0989
Psychologie sociale 0451
Psychométrie 0632

A NOVEL CURRENT-SENSING COMPLETION-DETECTION CIRCUIT ADAPTED TO THE MICROPIPELINE METHODOLOGY

BY

MICHAEL J. GAMBLE

A Thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

© 1994

Permission has been granted to the LIBRARY OF THE UNIVERSITY OF MANITOBA to lend or sell copies of this thesis, to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film, and UNIVERSITY MICROFILMS to publish an abstract of this thesis.

The author reserves other publications rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's permission.

I hereby declare that I am the sole author of this thesis.

I authorize the University of Manitoba to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Michael J. Gamble

I furthermore authorize the University of Manitoba to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Michael J. Gamble

Abstract

A micropipeline circuit utilizes delay elements in the request path to effect a bounded data path in a processing circuit. The stated implementation is not a true delay-insensitive circuit because it places constraints on the request signal that are related to the data path only through estimation. This condition hinders the potential speed of the circuit, creates a structure that is difficult to test, and introduces an element that is difficult to optimize through synthesis (the delay circuit). This thesis promotes the use of a novel circuit design that uses current-sensing completion-detection to eliminate the need for the delay element and, consequently, the said problems. The proposed circuit utilizes current-mode circuitry in a CMOS current comparator which proves to be a more efficient implementation than previous designs. The circuit employed in a data processor results in a power-efficient and potentially fast approach to asynchronous data processing. A four-bit Booth-encoded multiplier has been designed using a 1.2 micron CMOS process and sent for fabrication.

Acknowledgements

Firstly, I would like to thank Bob McLeod for key recommendations during the development of this thesis. His enthusiastic support of my endeavours had a positive influence on the outcome of this project and I look forward to working with him on future ones.

Equipment loans, software, and technical support were provided through the Canadian Microelectronics Corporation (CMC) and fabrication of the IC was subsidised by Northern Telecom Electronics Incorporated with the financial assistance of the National Sciences and Engineering Research Council (NSERC) of Canada. The assistance is gratefully appreciated.

I also would like to thank Adrian Earle and Peter Stokes for their help in the design and submittal of the IC and Jeff Fitchett and Budi Rahardjo for the stimulating conversations about asynchronous systems.

Finally, I would like to dedicate these works to my best friend and prime motivator, Lanette Kuchenski.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Digital system designers are continually looking to improve the processing speed of their hardware; whether it be with new fabrication processes or with innovative methods of design. Progress in this area is driven by the need to solve computationally intensive problems such as real-time simulation and image processing. It is well known that semiconductor manufacturers are reaching their limits in terms of producing smaller device geometries. This fact leaves integrated circuit (IC) designers to invent new circuit techniques and architectures that will enhance computation speed.

This thesis presents a novel circuit technique for asynchronous digital design. The circuit is based on a completion-detection circuit adapted to the micropipeline design methodology by [Suth89]. A custom 1.2 micron CMOS (complementary metal-oxide-semiconductor) four-bit Booth-encoded multiplier circuit has been designed and is currently undergoing fabrication. It comprises 2102 transistors and has a core area of 1.52mm^2. A custom leaf-cell library was devel-

oped to enable the placement and routing of the design. The resulting circuit proves to be a feasible asynchronous design.

Digital design could be categorized into two very different design methodologies: synchronous and asynchronous. Synchronous design assumes binary data and discrete time. Discrete time is accomplished by applying a clock signal to trigger the capture of data in sequential and/or parallel storage cells. Asynchronous design does not use the assumption of discrete time. It uses handshaking signals to control the capture of data. The removal of this assumption, theoretically, should provide a system with greater potential [Hauc94].

1.1 The Water-Bucket Queue

Have you ever observed or perhaps participated in a chain of firefighters passing buckets of water to dowse a raging fire? This system is, in fact, an asynchronous queue. More specifically, it may be designated as a *self-timed* system since it operates flawlessly without concern for the time delays in the course of its spatially localized actions. Each firefighter in the chain follows the same rules for transferring the buckets (see Figure 1.1).

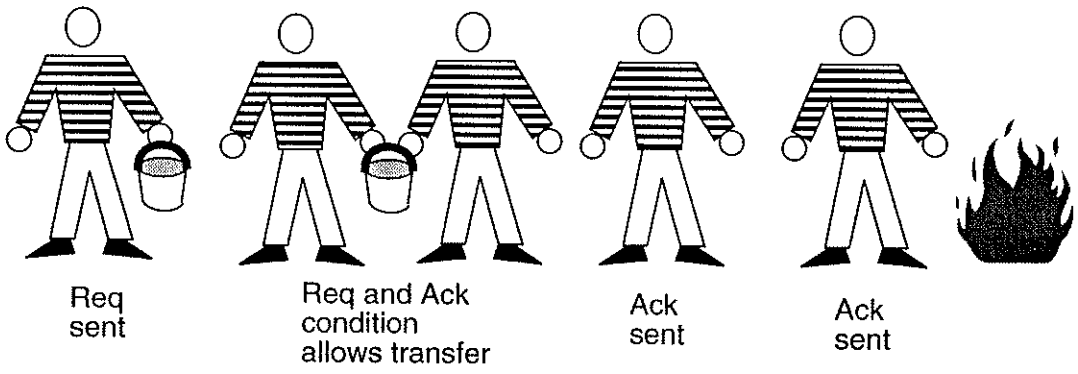


Figure 1.1. A Chain of Firefighters

When a firefighter does not have a bucket, an outstretched hand to the previous firefighter signifies an *acknowledge* state. When the firefighter is in possession of a bucket, a hand holding the bucket is extended towards the subsequent firefighter. This represents a *request* state. A relevant observation is that the data (the bucket) and the request are said to be *bundled*. Bundled data will be discussed further in Section 2.1. When two sequential firefighters are in opposite states (acknowledge and request) a bucket can be transferred between them. Of course, there are two temporary states that occur when a firefighter is in neither the request nor the acknowledge state. A firefighter is in a temporary state when in possession of a bucket and in the process of swinging it to a position where the next firefighter can grab it (request state), and when the firefighter is not in possession of a bucket and is in the process of reaching back to the previous firefighter where an oncoming bucket can be grabbed (acknowledge state). It is these two temporary states that create travel-time delay or *latency* in the advancing buckets.

1.2 Assets of Asynchronous Systems

The analogy of the water-bucket queue is provided to point out a few important and beneficial properties of asynchronous systems:

- The asynchronous queue is *elastic* because the number of buckets in it can vary. This property allows the queue to decrease the latency in the transfer of a bucket from the beginning to the end when the queue is partially full. This condition helps increase the *average throughput* in a system where the queue is partially full at all times. This is possible because the firefighters that do not possess buckets will likely be in the acknowledge state (a *stable*

state) which requires only one temporary state (*an unstable state*) to effect a transfer. [SSM94] reports that a latch-based pipeline attains a maximum throughput when it is half full.

- The elasticity of the queue also allows it to be more efficient because less power is consumed when the queue is not full. Instead of having the firefighters swing their arms to the beat of a drum (i.e., a synchronous system), they work on response to a request which allows them to rest if no buckets are available.
- If one firefighter is weaker (i.e. slower) than the rest, the *throughput* of that queue is based on that particular firefighter's speed if the queue is full. This would also be true for a synchronous system since one must clock a pipeline at the speed of the slowest element. However, unlike a synchronous system, the throughput of concurrent processes in the asynchronous system are not affected by one slow process since faster processes may operate as quickly as they can.
- If the firefighter chain is working under conditions where there is poor footing and high temperatures, the speed of bucket transfer will drop. But, if the chain is working under more suitable conditions, the queue will likely speed up. The *adaptability* of the asynchronous queue to its physical state and environment removes the concern that synchronous designers have: worst-case conditions must always be considered when choosing a clock speed.
- Another advantage to this system is its compatibility with the inputs offered from the naturally asynchronous world. Synchronous systems do not have this *robust mutual exclusion* property and are therefore subject to *metastability* problems at an asynchronous/synchronous interface [ChMo73].

These are just a few properties of asynchronous systems based on the observations of the water-bucket queue. The list above is sufficient to induce the systems community to invest in researching the problems associated with asynchronous design. Other papers discuss these advantages and disadvantages in more detail [Raha93 and Hauc94]. The scope of this document is restricted to the discussion of micropipelines and the correction of their associated problems with the proposed invention. Feedback in asynchronous systems such as finite-state machines (FSMs) and more complex structures are not discussed.

1.3 Thesis Overview

This document is organized into the following chapters:

- **Chapter Two** discusses micropipelines and some aspects of micropipelines which demand improvement. Completion-detection methods are introduced as another asynchronous design solution with results from recent studies. The idea for completion-detection circuitry integrated with a micropipeline is shown to be an amiable solution. A list of desired properties of a novel asynchronous circuit are presented as goals for this thesis.
- **Chapter Three** details the development of the invention. A timing diagram is used to describe the circuit working in a data processing system which is the main aid for understanding the operation of the circuit.
- **Chapter Four** details the development of the leaf-cell library and shows the results of simulations for each leaf cell.
- **Chapter Five** discusses the implementation of a prototype circuit that is used to verify the circuit theory. The circuit implemented is a four-bit Booth-

encoded multiplier using the leaf-cell library. A rudimentary simulation verifies the operation of the control circuit.

- **Chapter Six** discusses the proposed test procedures which will be used to verify the fabricated IC.
- **Chapter Seven** gives a summary of what this research has accomplished and offers conclusions. Applications for this circuit are discussed. Future work and goals are also indicated.
- **Appendix A** contains the schematic diagrams and layouts of the leaf cells used to implement the circuit.
- **Appendix B** contains the schematic diagrams of the four-bit Booth-encoded multiplier circuit and its layout.

CHAPTER 2

Asynchronous Design

2.1 Micropipelines

Micropipelines is a term used to denote an asynchronous control-circuit implementation for data transfer. In [Suth89], micropipelines were formally introduced and since that time this digital systems paradigm has been used to implement various projects [Brun91, KSRA91, LiGo92, PDF+92, TrDu92, GRM94, and Roin94]. The intended uses for this structure are integrated pipelined data-processing circuits; hence the word micropipelines.

The two-cycle signalling convention described by [Seit80] is utilized by the micropipeline methodology to produce an elegant and efficient circuit for asynchronous data transfer. The timing diagram in Figure 2.1(a) depicts how the signal levels interact. The relation between the signals is clearly shown to be based on voltage-level transitions or *events*. This convention is also known as non-return-to-zero (NRZ) because the signal levels do not have to return to a low voltage level to enable a second latching of data. The event can be either a rising or

falling transition and it is this property of the two-cycle convention that makes it faster and more power efficient than four-cycle signalling or return-to-zero (RZ) signalling (Figure 2.1(b)). However, the choice of protocals has been vehemently disputed by members of the asynchronous research community.

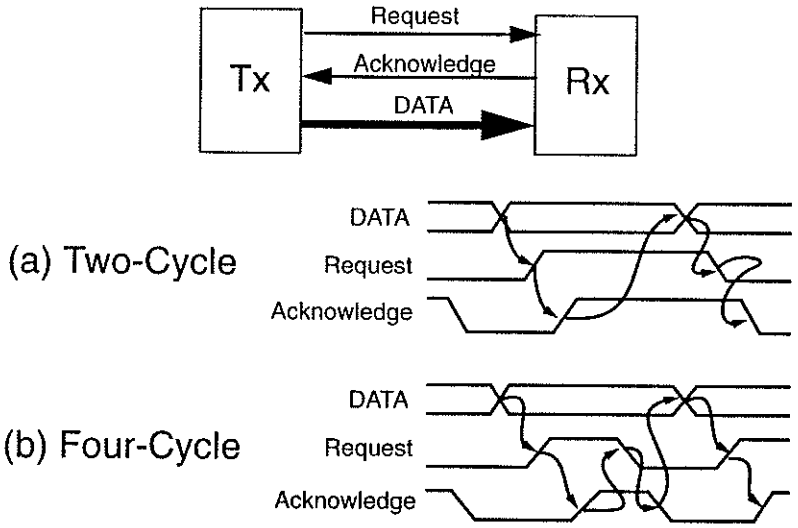


Figure 2.1: (a) The Two-Cycle Signalling Convention used in Micropipelines and (b) The Four-Cycle Signalling Convention (After [Siet80])

This simple handshaking protocol is implemented with one simple logic primitive: the *Muller C-element* (see Table 2.1 for the truth table). A C-element is a memory circuit that produces a logical-one at its output if all inputs are logical-one and a logical-zero at its output if all inputs are logical-zero; otherwise, the output will not change. This device is also known as a *rendezvous* or *join*.

Table 2.1: The Muller C-Element Truth Table

X	Y	Z(n+1)
0	0	0
0	1	Zn
1	0	Zn
1	1	1

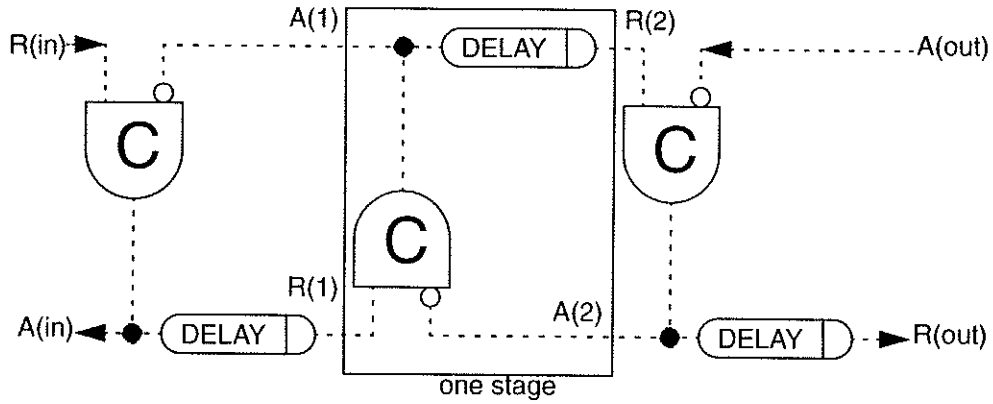


Figure 2.2: Control Circuit for the Micropipeline (After [Suth89])

The schematic diagram in Figure 2.2 depicts the control circuit for the micropipeline. The C-elements control the handshaking signals to effect the two-cycle convention. The delay element is inserted in the request-signal path to allow time for the data to settle at the inputs of the next stage. The delay element is a necessity in processing circuits since the delay of the logic gates in the processing circuit block must be accommodated by the delay in the request signal; otherwise the next stage in the pipeline might capture erroneous data with the premature arrival of the request event. If there is no processing logic between registers (i.e., a FIFO), the delay element can be ommited since the delay through the Muller C-elements and the associated wiring should be enough to allow the data levels to settle at the inputs to the next register. If a processing circuit block in between two registers has a logic-gate depth of four gates then the combined delay of the request event propagating through the delay element and the Muller C-element (which controls the capture of data in the subsequent register) must be greater than the delay of four gates. This implementation further categorizes micropipelines as two-cycle *bundled-data* circuits since the data

and the request signal must be bounded together to provide integrity of the data transfer.

The micropipeline also uses a different memory element for storing data. Unlike the traditional D-type flip-flop used in most circuits, micropipelines employ *event-controlled storage elements* [Suth89]. A schematic diagram of the event-controlled storage cell is depicted in Figure 2.3.

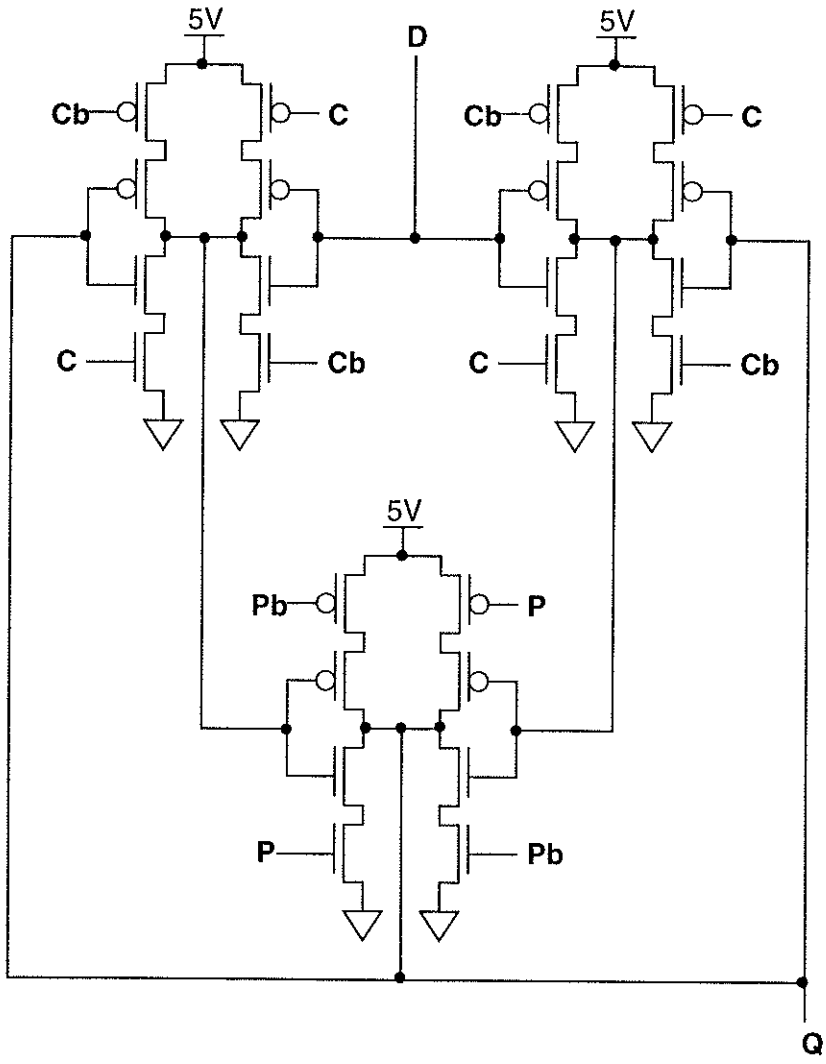


Figure 2.3: Schematic Diagram of an Event-Controlled Storage Cell (After [Suth89])

The *C* (capture) and *P* (pass) inputs and their complements (*Cb* and *Pb*) control the latching of data. When the event-controlled storage cell is *open* the data can pass from input to output like a latch. This property allows data to flow directly from input to output of a pipeline when the pipeline is *empty*.

2.1.1 Micropipeline Shortcomings

It is surmised that the main disadvantage with micropipelines is that they are not truly self-timed or delay-insensitive circuits. The delay element is required to enable the data to be bundled with the request signal. The control circuit is delay-insensitive but the bounded data path imposes the need for constraints on the request path. These constraints are compromised by the addition of the delay element which imposes a number of problems.

There are two ways to implement a delay element. The first is very demanding in terms of design time. The pretentious designer could observe each stage in the micropipeline and estimate the depth of logic to indicate the length of delay needed. The delay could then be implemented with a string of inverters; the length of which should emulate the amount of delay required. This is not a reasonable method due to the monolithic IC designs that can be realized with micropipelines. The synthesis of the delay element is a more acceptable approach where the maximum delay of a processing circuit could be estimated with simulation and instantiated with the appropriate number of inverters. The large number of processing logic blocks in a monolithic IC makes the synthesis of these delay elements computationally intensive.

This latter method was used in [Røin94] by combining HSPICE with a custom timing tool to estimate the time delay in the processing logic. This method unfortunately failed (five of twenty samples worked); even when a 50% safety margin was used. The conclusion from this study is that synthesis of the delay element requires more research into the different parasitics that exist in a CMOS circuit and the associated wiring. Varying temperature and manufacturing processes may affect the delay circuit and processing block differently which could result in underestimating the delay element size.

A more significant drawback when using delay elements is that they are gauged to emulate delays greater than the delays of the processing logic block. This defeats the purpose of using asynchronous circuits which should operate as fast as their physical properties and temperature conditions allow if they are truly delay-insensitive circuits. Micropipelines have a worst-case performance rather than the preferred average-case performance.

Another area of concern for micropipelines is testability. The performance of the micropipeline is determined by the control circuit. This makes it difficult to test the data path and to debug. Is the fault caused by a delay-type fault or perhaps even a design error? When micropipelines are used in complex systems this problem increases exponentially. More research in this area is required for micropipelines to be commercially accepted.

2.2 Completion-Detection Circuits

Another method of insuring data integrity at the inputs to the next register in the pipeline is by using completion detection in each processing logic

block. Rather than using a delay element in the control circuit, the speed of the processing logic block dictates the time delay in the request signal. This is accomplished by using different techniques to trigger a signal after the completion of the data processing.

2.2.1 Dual-Rail Encoding

Most completion-detection circuits have been based on dual-rail circuit implementations using either four-phase or two-phase signalling with the majority using four-phase signalling. A common approach is to use DCVSL (dynamic cascade voltage switch logic) [MBM89]. This method uses the DCVSL logic family to implement a four-phase signalling circuit that provides high speed and low power (Figure 2.4). Problems associated with DCVSL are charge sharing between the NMOS and PMOS FETs and the data must be latched after each gate otherwise it must be refreshed.

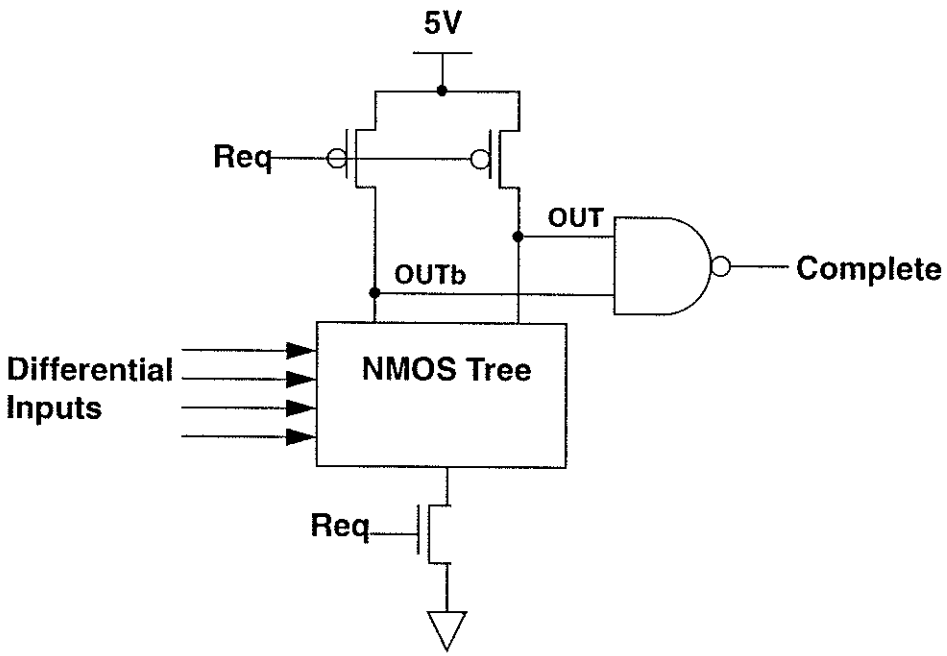


Figure 2.4: Schematic Diagram of a DCVSL Circuit for Completion Detection (After [MBM89])

An improvement over DCVSL is claimed by [Lu93] with the use of ECDL (enable/disable CMOS differential logic). It is a static implementation so the problems with dynamic logic are eliminated but at the expense of additional transistors.

One serious concern with these types of implementations is that they must be hazard-free designs. The complexity of these circuits will thus increase with larger function blocks and their synthesis and logic decomposition becomes a difficult problem. CAD tools that are currently available are mainly directed at the placement and routing of standard leaf-cell designs. Custom layout will, therefore, have to be used in some circumstances which is not practical for monolithic ASICs (application specific integrated circuits). Design time must be reduced to the level attained by synchronous circuit design to promote an investment in designing asynchronous circuits.

As with all asynchronous circuits, the problem of silicon overhead is also a concern. The additional wires needed in dual-rail schemes and the extra transistors required for the control circuitry hinders the use of these implementations because of the use of extra silicon real estate. However, in VLSI (very large scale integration) systems the clocking circuitry must be routed with care and can often consume large percentages of silicon area of an IC (sometimes as high as 25% to 30% of the die for large designs). Often, the routing of the clock and the buffering of the clock signal requires manual placement. The placement and routing of an asynchronous circuit is thus easier and reduces design time without depleting too much extra silicon area than a synchronous system. Nevertheless, the reduction of asynchronous control circuit overhead is a concern for

designers of asynchronous circuits albeit less a concern than performance and power efficiency.

2.2.2 CSCD (Current-Sensing Completion Detection)

Another approach to providing completion-detection of data processing is to monitor the current in a CMOS processing logic block. CMOS circuits draw current from the power and ground rails when a transition occurs at a CMOS circuit output (see Figure 2.5).

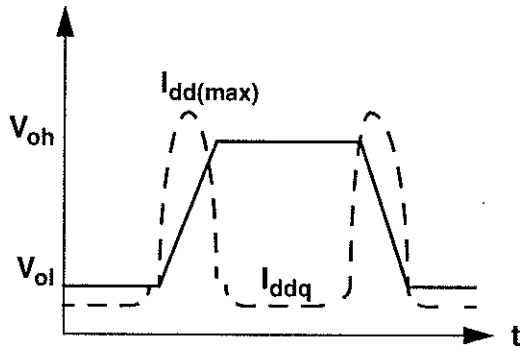


Figure 2.5: Voltage/Current Output Characteristics of a CMOS Circuit

The main advantage that current sensing has over other completion-detection techniques is that the processing logic block does not have to be designed as a hazard-free circuit. Logic block outputs may toggle prematurely at the inputs to the next register in the pipeline which suggests *static*, *dynamic*, or *delay hazards* [Jaco89]. An example of a static hazard is shown in Figure 2.6.

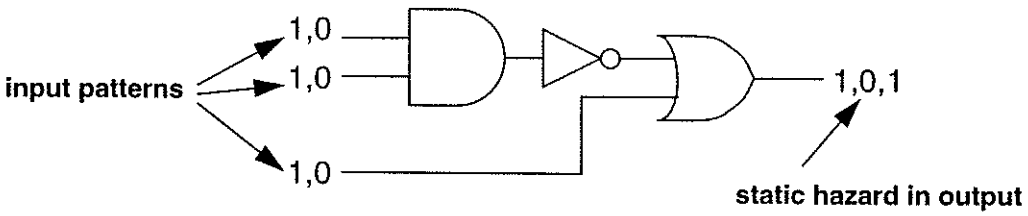


Figure 2.6: Example of a Circuit with a Static Hazard

Previous schemes would relay the request signal when the hazard appeared at the output of the logic block and cause the latching of erroneous data. In current-sensing completion detection, any hazard would be masked by the continuous flow of current from the supply rails. When the current draw from the supply rails ceases, the appropriate detection circuitry will trigger the completion signal. This permits the use of any logic decomposition of a function.

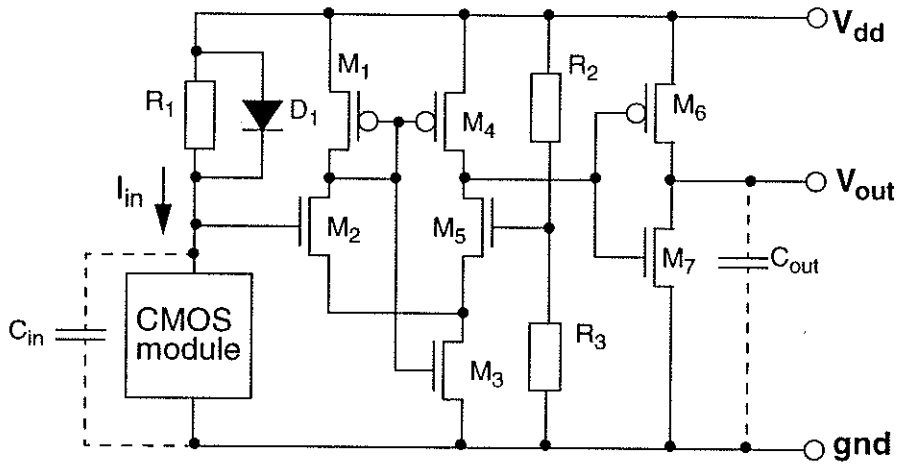


Figure 2.7: A Current-Sensing Completion-Detection Circuit (After [IST90])

An investigation by [IST90] resulted in the circuit implementation depicted in Figure 2.7. The circuit works by applying the pseudo *Vdd* voltage level at the top of the CMOS module to a differential amplifier whose reference voltage level is based on the voltage divider (*R2* and *R3*). The voltage level at the pseudo *Vdd* rail drops from a quiescent level of approximately 4.3 Volts when an output transition occurs within the CMOS module. The current drawn into the CMOS module (*Iin*) is transformed into a proportional voltage level which is the pseudo *Vdd* voltage level. When the pseudo *Vdd* level drops below the threshold level the output voltage level *Vout* goes high. When the pseudo *Vdd* level flips back above

the threshold level, *Vout* goes low. The CMOS module utilized in their simulations is a string of five inverters.

The performance of the circuit was limited by several factors. The diode resistor in parallel at the power rail of the circuit is the main detriment. It's parasitic influence slowed the circuit for the CMOS process that was simulated. The inverter train was impeded by the diode-resistor circuit, changing the latency from 11ns to 16ns according to the simulation data. The reaction of the differential amp presented the completion signal to the next stage approximately 6ns after the current in the CMOS module subsided to a quiescent current level (*Iddq*) and 10ns after the inverter train had finished processing if it was without the diode-resistor circuit appended to its power rail.

One large concern for this type of circuit is the presumption that a continuous and strong current draw will occur at the power rail during the transitioning of outputs in the logic gates of the CMOS module. There is always a current draw from the power and ground rails during an output transition of a CMOS logic gate because, for an instant of time, the NMOS and PMOS FETs are both turned *on* in varying degrees according to the voltage levels on the gate inputs at that particular instant of time. This causes an instantaneous short between the power and ground rails. This power surge is not, however, as significant as the current caused by the voltage level of the output capacitance discharging through either the power or ground rail.

A four-input NOR-gate was simulated to examine these effects in a 1.2 micron

CMOS process (Figure 2.8). The results showed that with increasing output capacitance on the output of the NOR-gate the current surge caused by the short between power and ground actually decreased. This was observed by examining the current flow from the power rail when the output was conditioned to go from five volts to zero volts with the following input combination sequence:

ABCD (0000 => 0001)

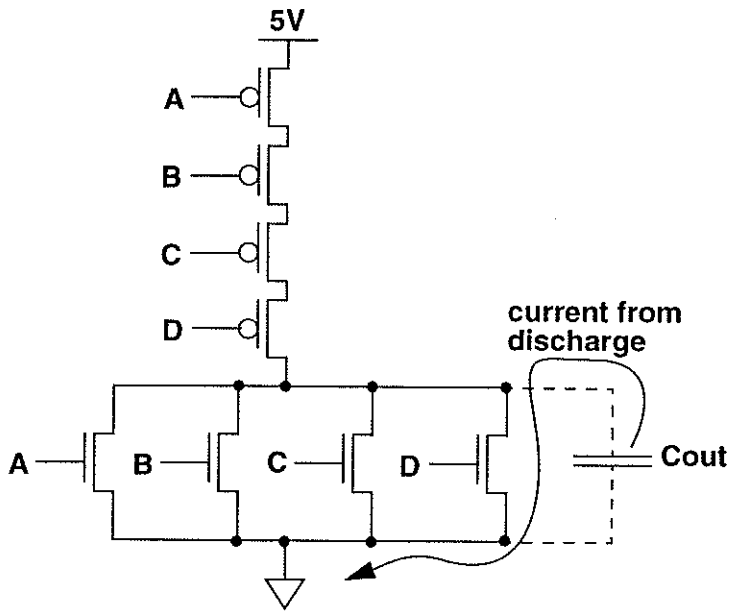


Figure 2.8: Current Flow in a Switching Four-Input NOR-Gate

The effect of increasing output capacitance reduced the current draw through the power rail by overpowering the instantaneous current surge from the ground rail with an abundance of charge from the output capacitance. The key to this result is the difference between the rise-times and fall-time of the inputs and output. The output capacitance causes the fall-time to be slower than the rise-time of input *D*. This enables the PMOSFET driven by *D* to turn *off* before the

capacitance has been fully discharged which results in only a small source-to-drain voltage (V_{DS}) across the PMOSFET during the transition on the gate of the PMOSFET. The fact that the V_{DS} value will be less than the gate-to-source voltage (V_{GS}) for a time will cause the PMOSFET to operate linearly during that transition. This lowers the maximum current drive in that PMOSFET. The string of four PMOSFETs in series also contributes somewhat to the reduction of current flow because of the additional resistance. This effect was not observed by [IST90] because of their use of an inverter circuit which draws a significant current through both supply rails with minor loading on the output (i.e., a fanout of one inverter gate).

This experiment verifies the concern with the method of current sensors on the power rail which may be alleviated by providing current sensors on the ground rail too. This solution, however, presents additional circuitry which may make the control circuit overhead unwieldy and, worse still, reduces the performance of the circuit further. Another CMOS logic family that discharges the output capacitance through both the power and ground rails is thought to be a better solution if the CMOS family does not present too much additional circuitry. This idea is used in the final implementation and is elaborated upon in Section 3.1.

Another problem associated with the diode-resistor circuit is that it presents a quiescent current draw across the resistor of approximately 140μA (0.7V/5kΩ). This increases the quiescent power draw for each CMOS module in the circuit which is again unwanted because one reason for using asynchronous circuits is their low power operation as explained in Section 1.2. Also, the resistors used to

implement the voltage reference level also draw a significant quiescent current since *R2* and *R3* add up to 5kΩ (according to the values given) which draws 1mA (5V/5kΩ). The diode-resistor is also difficult to place and route with existing CAD tools since it will have to be a large enough device to dissipate the large power draw from a CMOS module that could be as large as a hundred logic gates.

The differential amplifier is a robust circuit in terms of being adaptable to process changes but it is a large circuit with some transistors needing to be implemented with widths as wide as 200 microns (*M2* and *M5*). This is again difficult to place and route with existing CAD tools and hints at a manual implementation requirement. Furthermore, the resistors themselves are difficult to implement in common CMOS processes with high degrees of tolerance. Yet another difficulty is the susceptibility of the differential amplifier to noise. Surges on the power rail and ground rail due to effects such as ground bounce or low-level electro-magnetic interference (EMI) could prematurely trigger a completion signal because of the need for a threshold voltage very close to the voltage diode drop (approximately 0.8 Volts).

Overall, the analysis of the prior circuit shows that there is potential for the circuit but other configurations would be necessary. Another attempt to implement a CSCD circuit by [DDH91] uses the same basis as [IST90] but with minor variations. Using various configurations of BiCMOS (bipolar-CMOS hybrid technology) and CMOS circuits they experimented with an adder logic block as the CMOS module. One very important observation made from their study is that

the state similarity of sequential inputs to the logic block (hamming distance) reduced the amount of current drawn by the CMOS circuitry which was beneficial in increasing the speed of the circuit. The cessation of transient current-flow happened quicker when the hamming distance between consecutive bit patterns was small which produced a decreased latency in the completion signal. If complete state similarity occurred (i.e., zero hamming distance) a minimum delay-generating (MDG) circuit was required to ensure the toggling of the completion signal (see Figure 2.9). In some cases the toggling was not detected and the MDG provided a backup in case of that predicament.

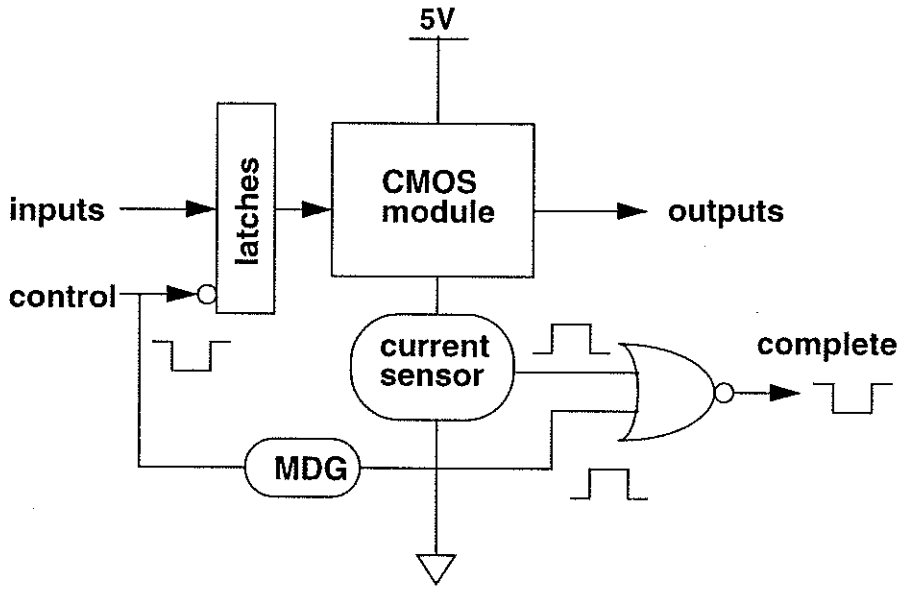


Figure 2.9: CSCD Block Diagram (After [DDH91])

Again, the use of only one current sensor is utilized despite the danger of reduced current flow in one supply rail or another according to our studies. However, claims are made by [DDH91] that the CMOS-variant simulations only required one current sensor. The basic circuit used for the current sensor is depicted in Figure 2.10.

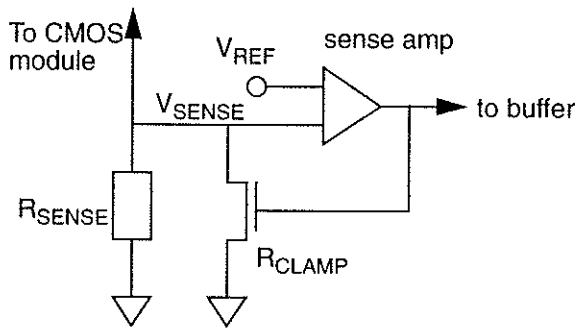


Figure 2.10: Current Sensor (After [DDH91])

Rsense is chosen to maintain a 100mV quiescent voltage level for the CMOS module pseudo-ground. A diode is not used in this implementation because the NMOSFET labelled *Rclamp* is used to ensure that *Vsense* does not go above 800mV. The transistor representing *Rclamp* will have to be very large physically because of the large transient currents that could develop from a large CMOS module. The parasitic effects of the large drain capacitance will hamper the speed of the circuit and the speed of its switching is questionable unless the sense amp has a large drive which requires large transistors in that configuration. This all contributes to higher power requirements for the IC. This is a rational alternative but it still introduces a parasitic element *Rsense* which reduced the speed of the CMOS module by a reported 3%. The use of resistors is again deemed to be an implementation problem due to their size, low tolerance to process variations, and power consumption. In addition, various resistance values will have to be synthesized depending on the number of gates in the CMOS module. The problems associated with differential amplifiers again

appears. The researchers claim that these difficulties will be overcome with further research into different configurations. It is the aim of this thesis to follow that premise.

2.3 A List of Solution Requirements

In retrospect of the above analysis, we have identified some weaknesses with micropipelines and other asynchronous circuit methodologies. The initiation of this study was built on the realization that micropipelines were hampered by the need for delay elements in the path of the request signal. Knowing that Iddq detection methods are used in built-in self-testing (BIST) circuits [MaNi91], it was conjectured that a similar current-monitoring circuit could be used to implement the completion-detection. To our knowledge only two groups have studied this problem [IST90 and DDH91] with simulations of various experimental circuits. With that background, a new circuit must be realized and implemented in silicon to achieve a feasible solution. A list of desirables for a new circuit is given below:

1. The circuit must be compatible with micropipelines. It will follow the same handshaking protocol (the two-phase signalling convention).
2. The circuit will use current-sensing completion detection on one supply rail to minimize control circuit size.
3. The control circuit must be as fine-grained as the rest of the standard leaf-cells so that common CAD tool placement and routing (P&R) can take place without concern for manual placement.
4. The control circuitry will not expropriate more than 30% of the core silicon area.

5. Resistors will not be utilized.
6. The circuit will be implemented in a 1.2 micron CMOS process.

With this modest list, a suitable circuit was developed and implemented. The requirements were met and a feasible implementation is now available.

CHAPTER 3

The Neo-Micropipeline

In previous implementations of a CSCD circuit [IST90 and DDH91], the current at a pseudo supply rail was transformed into a proportional voltage and input to a differential amplifier. An alternative and more prudent approach is to utilize current-mode circuitry [TLH90]. The circuit design presented in this chapter is built on the idea that current-mode circuitry is potentially a more effective and efficient way of implementing the current sensor. The development of the rest of the circuit fans out from this initial commitment. Subsequently, this chapter details the invention [GaMc94] with a bottom-up description.

3.1 The Current Sensor

Knowing the limitations of differential amplifier-based CSCD circuits in this application, we examine the current-mode current comparator as a means to eliminate some of these problems. The circuit developed by [FrCu83] proves to be a practical building block for our purposes. Figure 3.1 depicts the modified comparator circuit used in the current sensor.

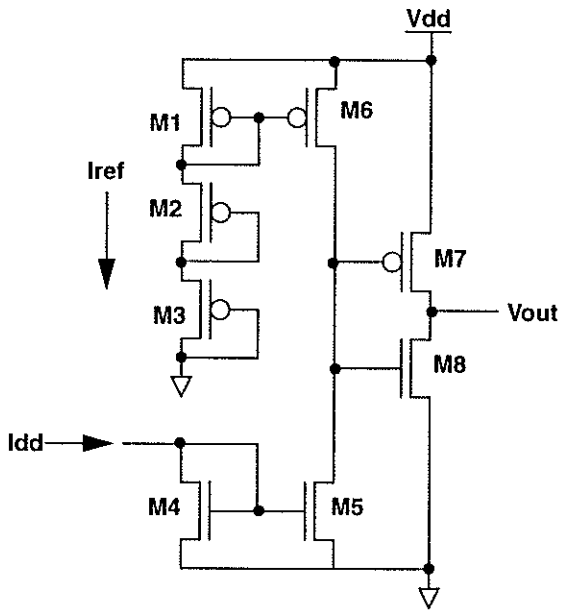


Figure 3.1: A CMOS Current-Mode Current Comparator

The input to the comparator is effectively a diode-connected NMOSFET which provides a low input resistance (*Zin = 1/gm*). The PMOSFETs *M1*, *M2*, and *M3* set up a quiescent reference current (*Iref*) of 12μA. This value is chosen to provide a reference level that is safely above the quiescent current level of the CMOS processing logic block where noise spikes will not significantly affect the current level. The PMOSFET *M6* mirrors the current in the reference leg to provide the pullup drive on the inverter created by FETs *M7* and *M8*. However, PMOSFET *M6* is usually increased in width to have a 40μA pullup drive to offset the quiescent current in the logic block (see Section 5.3). The quiescent current increases linearly with the number of gates in the logic block. The input current (*Idd*) provided by a high impedance output (explained hereinafter) is mirrored by NMOSFET *M5* which provides the pulldown drive on the inverter input. The circuit

outputs a logic-one when *Idd* is less than *Iref* and logical-zero when *Idd* is greater than *Iref*. The size of the transistors in the comparator do not have to be larger than those used in the leaf cells because the circuit has a low input impedance and inherent current limiting. Power consumption of the control circuit is therefore significantly less than previous differential amplifier-based current sensors.

The input of the comparator is coupled to the outputs of a current mirror on each CMOS gate in the processing logic block. Rather than using a large current mirror for each CMOS module, each gate is appended with a Widlar-type current mirror at its power rail to provide a grid of high impedance *Idd* sources which can be input to the comparator. This results in a finer granularity of the circuit which allows P&R with existing CAD tools.

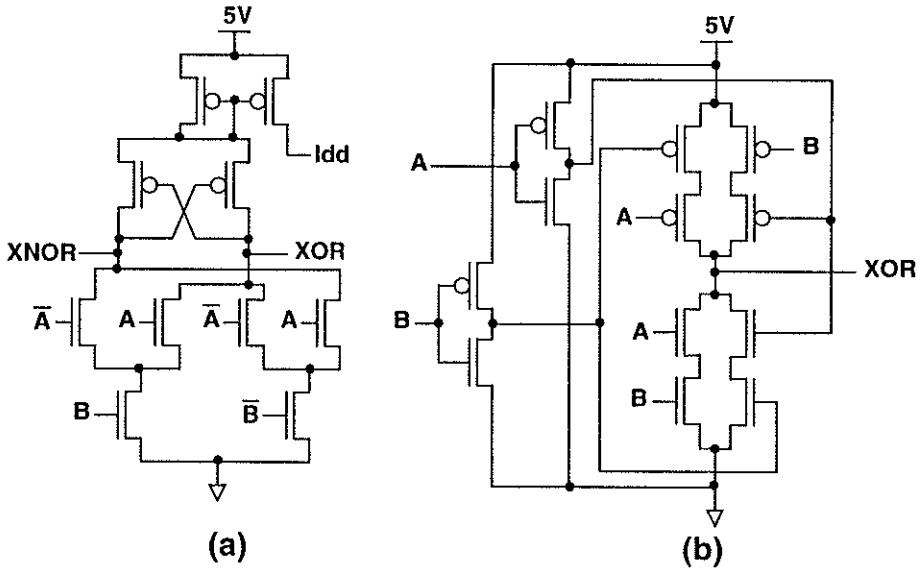


Figure 3.2: (a) CVSL Logic Family Implementation of an XOR/XNOR-Gate with Current Mirror and (b) a Standard CMOS Family Implementation of an XOR-Gate.

Recall from Section 2.2.2 that a CMOS gate must draw significant current from the power rail to allow the current sensor to detect it. To provide a CMOS family that draws significant transient current from the power rail during a high-to-low transition of the output, we use the cascode voltage switch logic (CVSL) family [HeGr84]. It uses a pair of PMOSFETs as pullups and a differential input NMOS-tree which is the design precursor to the DCVSL family. This CMOS family typically uses two more transistors than primitive NAND/NOR CMOS gate families. Appending the current mirror to the power rail requires an additional two FETs for each gate (FETs required $\leq 2N+4$). AND/NAND and OR/NOR-gates require four more additional FETs than their standard counterparts but the XOR/XNOR-gate implementation is actually less complex than a standard active XOR-gate implementation. The CVSL XOR/XNOR-gate implementation with current mirror appended comprises ten transistors while the standard active XOR gate comprises twelve transistors (see Figure 3.2). The CVSL gate also provides both true and complemented forms of the gate function which is advantageous when reducing gate delays and decomposing functions into logic gates. This makes CVSL comparatively better than standard CMOS logic when implementing arithmetic circuits. The main disadvantage with CVSL is the requirement of dual-rail logic inputs which increases the amount of wiring in the circuit.

Another aspect of the new logic gate implementation is that it has a reduced voltage swing on its outputs. Because of the current mirror, the maximum voltage at the pseudo *Vdd* rail is now five Volts minus the threshold voltage of the PMOSFETs (5 - 0.8 = 4.2Volts). This has the effect of reducing the power

requirements for each logic gate.

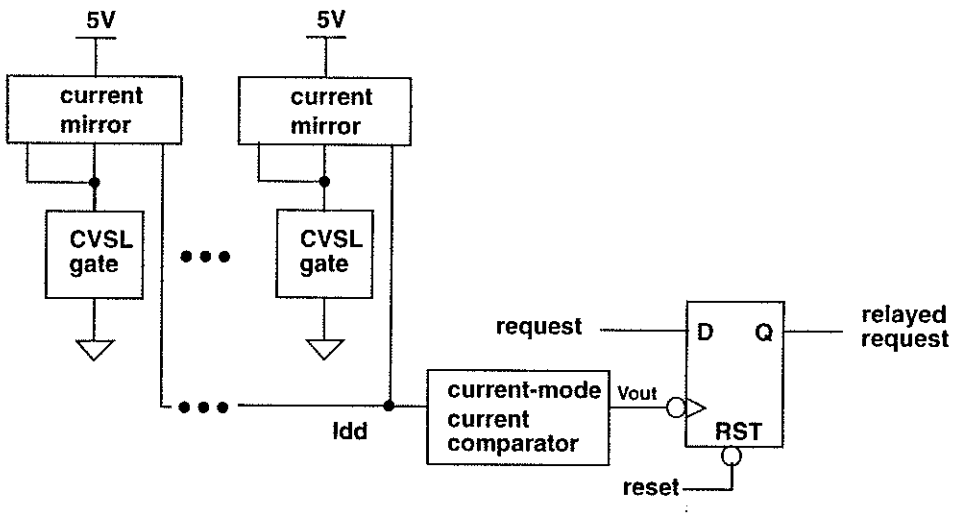


Figure 3.3: Block Diagram of the Current Sensor

A block diagram of the current-sensor is depicted in Figure 3.3. The output of the comparator is used to latch the request signal using a D-type flip-flop (DFF). The request signal is thus relayed to the next stage of the micropipeline which is input to a C-element. The truth table for the DFF is shown in Table 3.1.

Table 3.1: Truth Table for a D-type Flip-Flop

CLK	RST	D	Q(t)
X	0	X	0
X	1	X	Q(t-1)
↓	1	0	0
↓	1	1	1

It was mentioned that a complete state similarity in the CMOS logic block would not increase the quiescent current significantly. This circumstance would cause

the circuit to deadlock since the comparator output will not toggle and the request signal would not be relayed. To overcome this problem, the C-elements that control the handshaking signals are also appended with current mirrors. The C-element must be implemented with a static form like the cross-coupled inverter implementation depicted in Figure 3.4. This circuit will draw current from the power rail during any output transition which is, in almost all cases, an event signalling a request to the next stage.

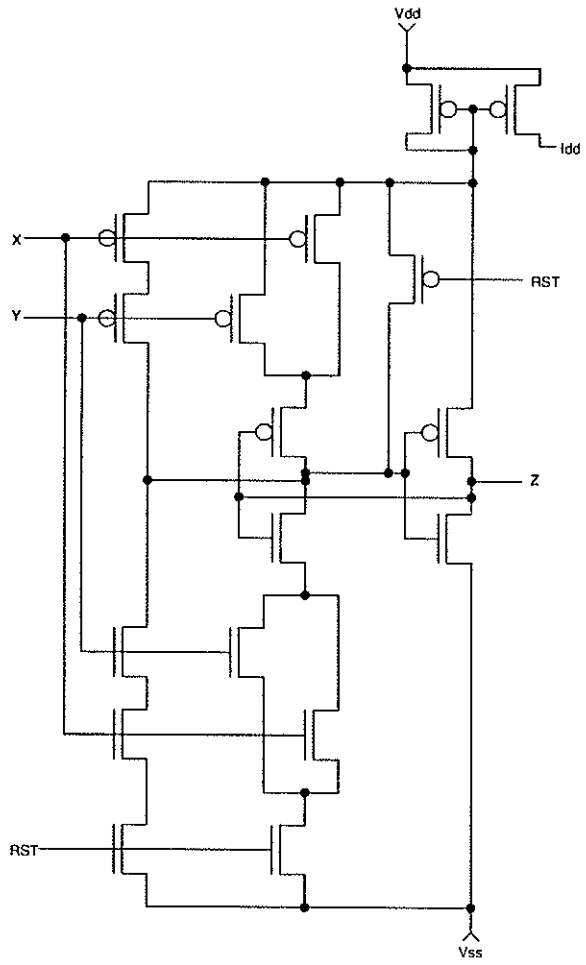


Figure 3.4: A Static Implementation of a Muller C-Element

Both the true and complementary outputs of the Muller C-element are utilized to provide the *Pass* and *Capture* signals plus their complements for triggering the event-controlled storage cells (see Figure 2.3). To *clear* the pipeline, reset signals must be sent to the DFFs and the Muller C-Elements. This leaves the micropipeline in an empty state. The micropipeline must be reset after power-up to initiate the micropipeline in the empty state.

3.2 The Timing Diagram[1]

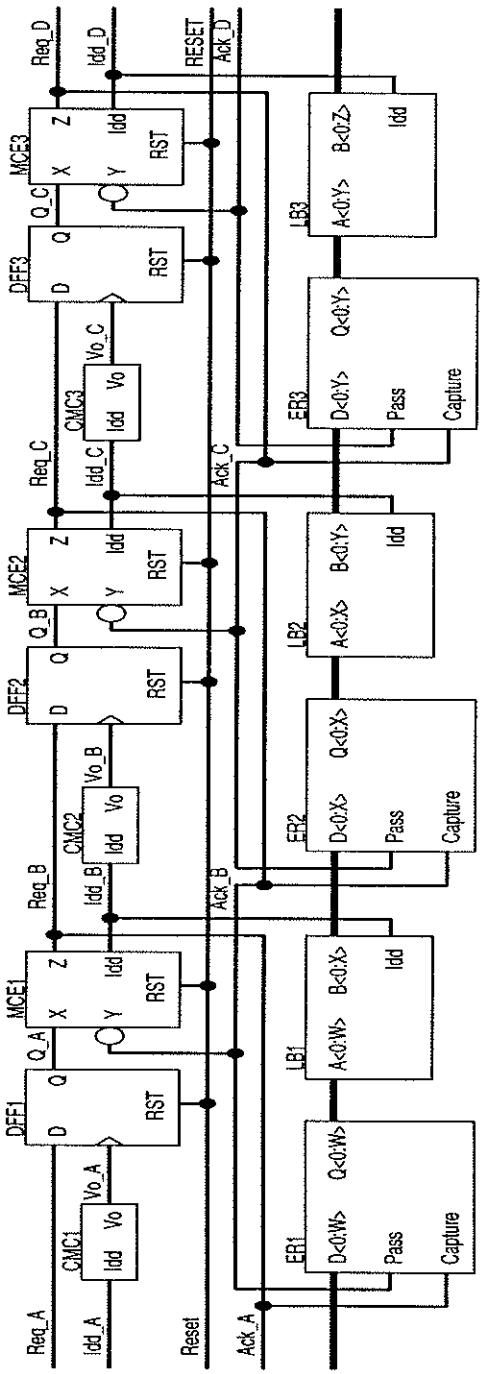


Figure 3.5: A Block Diagram of the Neo-Micropipeline (After [GaMc94])

1. Excerpts in this section are taken from the UK provisional patent entitled "An Asynchronous Circuit for Data Processing" (Serial Number 9406122.3 [GaMc94])

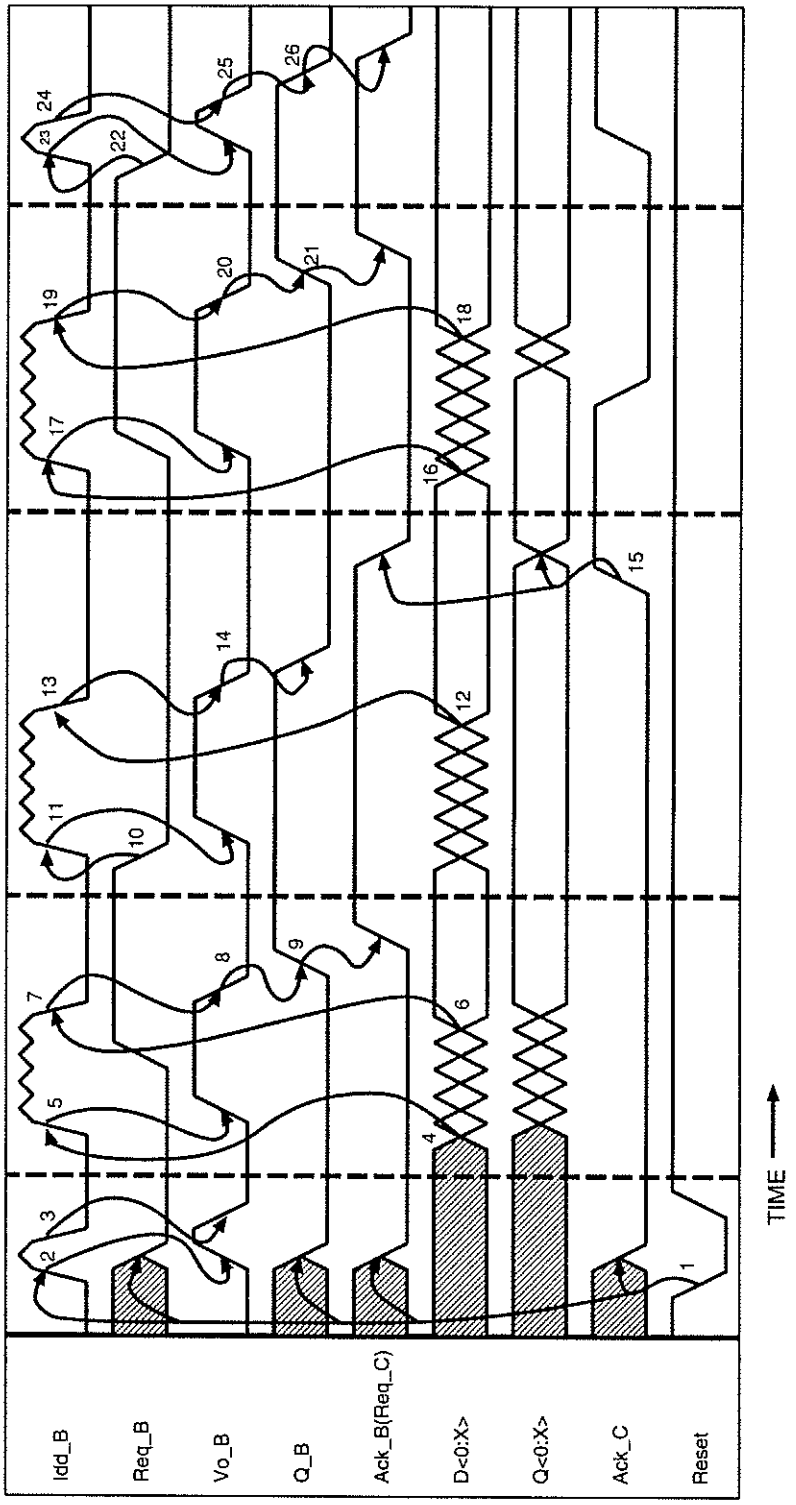


Figure 3.6: Timing Diagram for the Neo-Micropipeline (After [GaMc94])

The complete functionality of the circuit is described below. A full understanding of the circuit and its operation should result from the study of this section. Timing diagram references are based on the block diagram of the neo-micropipeline that is depicted in Figure 3.5. There are three stages of micropipeline in the circuit depicted in Figure 3.5. The timing diagram in Figure 3.6 illustrates the signal levels of the second stage in the micropipeline during several operation cycles.

First, examine the circuit in Figure 3.5 to identify the various building blocks that define the neo-micropipeline. Each stage of the pipeline has a main circuit assembly and a control circuit assembly. The main circuit assembly comprises an event-controlled register and a CMOS processing logic block. The control circuit assembly comprises a current-mode comparator, a DFF, and a C-element. Thus, each stage of the circuit in Figure 3.5 comprises one current-mode comparator (labelled *CMC 1* through *3* for its corresponding stage number), one DFF (labelled *DFF 1* through *3* for its corresponding stage number), one Muller C-element (labelled *MCE 1* through *3* for its corresponding stage number), one event-controlled register (labelled *ER 1* through *3* for its corresponding stage number), and one CMOS processing logic block (labelled *LB 1* through *3* for its corresponding stage number).

Referring to Figure 3.6, *Idd_B* is the current level in the net that is sourced by the output *Idd* of *MCE1* and the output *Idd* of *LB1*. A simplification of what current level could be present on this net is featured by depicting an ideal current flow in *Idd_B*. Two basic states exist in the timing diagram for the current in

Idd_B. One state occurs when there are no transitions in *MCE1* or *LB1* showing zero current. The other state is when transitions are occurring in *MCE1* and/or *LB1* that cause a source of current on *Idd_B* which fluctuates slightly but maintains a strong current flow. Correspondingly, the output of *LB1* (*D<0:X>*) has an output vector that relates to *Idd_B* in an idealistic manner. When *Idd_B* sources a strong current flow, *CMC2* reacts by driving *Vo_B* to a logical-one state. After the current flow in *Idd_B* subsides to zero, *CMC2* reacts by driving *Vo_B* to a logical-zero state. The rising and falling of *Vo_B* is used to clock *DFF2*. By clocking *DFF2*, the request signal (labelled *Req_B*) from *MCE1* can be relayed to the *X*-input of *MCE2*. This operation effectively replaces the need of a delay element in the path of the request signal which is the main object of the invention. It effectively speeds up the circuit when state similarity between subsequent data inputs to the pipeline causes an early cessation of current.

There are several numerated points in the timing diagram (labelled ***1*** through ***26***). These points are used to describe the operation of the invention in a sequential manner.

1. The integrated circuit in Figure 3.5 has been powered up with five Volts on the Vdd rail and zero Volts on the Vss rail. The *Reset* signal has been driven low to reset all the DFFs and C-elements, thus *Req_B*, *Q_B*, *Ack_B*(*Req_C*), and *Ack_C* have a low transition.
2. *MCE1* draws a current because it has a low transition which appears as a rise in the current level on *Idd_B*. Initially, *MCE1*, or any C-element in the data processor, is at an unknown state at power-up. In this case, *Req_B* was initially at a high level, therefore it sourced a current as it transitioned

to a low voltage level. This caused *Vo_B* to go high. Almost immediately, the current level on *Idd_B* drops to zero because the transition on *MCE1* causes a current draw for a small period of time.

3. This drop in current level causes *Vo_B* to go low. *DFF2* is clocked because of this rise and fall on *Vo_B* but it does nothing to change the output state of *DFF2* because the output was previously reset to a logical-zero state and *Req_B* was reset to a logical-zero state. Thus ends the initial cycle in the operation of the circuit.
4. Transitions begin to appear on *D<0:X>* because a data change has been applied to the beginning of the pipeline and all the event-controlled registers pass data through from input to output (i.e. they are transparent). The request signal to the second stage of the data processor is yet to appear because it is lagging behind the data signal propagation. These transitions in *LB1* are reflected in the rise in current on *Idd_B*. Notice also that the transitions also appear on *Q<0:X>* although slightly delayed because *ER2* is passing data from input to output as previously mentioned.
5. The rise in current on *Idd_B* causes *Vo_B* to go high. At a point during the output transitions in *LB1*, *Req_B* goes high.
6. The transitions in *LB1* stop and the cessation of current draw into *LB1* is reflected on *Idd_B*.
7. The drop in current on *Idd_B* causes *CMC2* to drive *Vo_B* low.
8. The low transition on *Vo_B* causes *DFF2* to pass the request signal which drives *Q_B* high.
9. The transition on *Q_B* causes *MCE2* to drive *Ack_B* high which effects a request to stage three and an acknowledge to stage one; thus ending a sec-

ond cycle in the operation of the circuit.

10. *MCE1* drives *Req_B* low. At the same time, the output transitions in *LB1* start because a new set of data has been passed into it from *ER1*. Notice that the output of *ER2* (labelled *Q<0:X>*) is not changing because the acknowledge signal from the third stage (*Ack_C*) in the pipeline has not set *ER2* to be transparent. The current on *Idd_B* increases to a significant level because of the initial transition on *Req_B* and the continuing flow of current in *LB1*.
11. The rise in current on *Idd_B* causes *Vo_B* to go high.
12. The transitions in *LB1* stop and the cessation of current draw into *LB1* is reflected on *Idd_B*.
13. The drop in current on *Idd_B* causes *CMC2* to drive *Vo_B* low.
14. The low transition on *Vo_B* causes *DFF2* to pass the request signal which drives *Q_B* low. The request signal has now been relayed to *MCE2* but the output on *MCE2* does not change because the acknowledge signal event has not been received from the third stage in the circuit. This likely means that stage two has not completed its processing (*LB2* is drawing transient current).
15. The acknowledge signal has arrived sending *Ack_C* high. This signal passes the data through *ER2* (causing transitions on *Q<0:X>*) and triggers *MCE2* to send *Ack_B* low which effects a request to stage three and an acknowledge to stage one; thus ending a third cycle in the operation of the circuit.
16. A new set of data has been passed into *LB1* and again both the request signal from *MCE1* and the acknowledge signal from *MCE3* have not arrived yet. However, the output transitions in *LB1* draw current which is reflected

in *Idd_B*.

17. The rise in current on *Idd_B* causes *Vo_B* to go high. Notice that *ER2* does not pass the data on to *LB2* (observe *Q<0:X>*) until *Ack_C* goes low.
18. The transitions in *LB1* stop and the cessation of current draw into *LB1* is reflected on *Idd_B*.
19. The drop in current on *Idd_B* causes *CMC2* to drive *Vo_B* low.
20. The low transition on *Vo_B* causes *DFF2* to pass the request signal which drives *Q_B* high.
21. The transition on *Q_B* causes *MCE2* to drive *Ack_B* high which effects a request to stage three and an acknowledge to stage one; thus ending a fourth cycle in the operation of the circuit.
22. *MCE1* drives *Req_B* low. The current in *Idd_B* reflects the draw of current by *MCE1*.
23. The rise in current on *Idd_B* causes *Vo_B* to go high. An important aspect in this particular operation cycle of the data processor is that the data captured by *ER1* is the same data set as the previous one (i.e., *D<0:X>* does not change). *LB1*, therefore, does not draw any transient current.
24. The current in *Idd_B* immediately subsides because no current is drawn by *LB1* and the output transition in *MCE1* causes a current draw for a relatively short time. The drop in current on *Idd_B* causes *CMC2* to drive *Vo_B* low.
25. The low transition on *Vo_B* causes *DFF2* to pass the request signal which drives *Q_B* low.
26. The transition on *Q_B* causes *MCE2* to drive *Ack_B* low which effects a request to stage three and an acknowledge to stage one; thus ending a fifth

cycle in the operation of the circuit. This last cycle example clearly shows that a data processing circuit that employs the invention will have an increase in data throughput average if the state similarity between consecutive data input vectors is close enough to effect a decrease in current draw. In the case of similar consecutive data sets, a current may be drawn but this also depends on the structure of the processing logic block.

By focusing on the control signals *Req_B* and *Ack_B* in Figure 3.6, the two-phase bundled data convention is clearly seen to be present and effective in the present invention.

3.3 Repercussions on Circuit Testing

Circuit testing is an area that has generated much concern for asynchronous systems. It is one of the main factors that has caused many researchers in the digital design community to criticize the use of asynchronous circuits. As explained in section 2.1.1, micropipelines are not easily testable. The introduction of the CSCD circuit helps eliminate some of these problems.

If a short or open circuit fault exists in a processing logic block, the *Idd* net for that processing logic block may consistently draw a significant current (see Figure 3.7). With knowledge of the operation of the neo-micropipeline, it is easy to deduce that the pipelined circuit will not pass data through that stage because of the excess current draw. The circuit will, therefore, deadlock. Another example of how a fault could cause the failure of the pipelined circuit is apparent if the fault causes a logic gate in the processing logic block to remain in a stuck-at

condition which may cause the circuit to prematurely send the request signal and capture erroneous data at the next register. A fault in the circuit that employs the invention could be said to be more easily exercised and detected because of the catastrophic effects a single fault may cause. Furthermore, the test vector set for the circuit that uses the invention may be less in size than a set of test vectors needed to determine a fault if the invention is not used in the same circuit. The fault coverage of a test vector set could be said to be higher than the same test vector set used to test a circuit that does not employ the invention. This results in a simplification of the test procedure required to determine the integrity of the circuit-under-test.

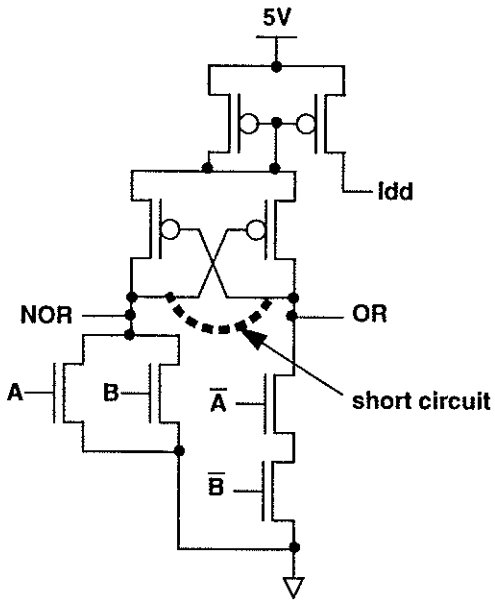


Figure 3.7: A Short Circuit in a CVSL OR/NOR-Gate Causing Excessive Current Draw

CHAPTER 4

The Leaf Cells

Implementation of the neo-micropipeline in silicon was resolved with the design of a custom set of leaf cells. The logic primitives implemented as leaf cells were chosen to enable the design of any data processing circuit that would be suitable for pipelined operation. A four-bit Booth-encoded multiplier circuit was chosen as the prototype for the neo-micropipeline. More detail on the multiplier circuit is described in Chapter Five. Appendix B provides a compilation of all the schematics and layouts for each leaf cell.

4.1 The Leaf-Cell Set

The list of leaf cells required for implementing prototypes of the neo-micropipeline are:

- andNand2: a CVSL two-input AND/NAND-gate
- andNand3: a CVSL three-input AND/NAND-gate
- andNand4: a CVSL four-input AND/NAND-gate
- cmcc: a current-mode current comparator

- eventCell: an event-controlled storage cell
- mce: a Muller C-element with reset
- orNor2: a CVSL two-input OR/NOR-gate
- orNor3: a CVSL three-input OR/NOR-gate
- orNor4: a CVSL four-input OR/NOR-gate
- xorXnor2: a CVSL two-input XOR/XNOR-gate

These leaf-cells were developed using a 1.2-micron double-metal n-well CMOS process available through the Canadian Microelectronics Corporation [CMC90a]. The height of the leaf-cells was chosen to be 75 design-scale microns (DSM[1]). This height allowed the new cell set to be compatible with the standard cell library offered through the Canadian Microelectronics Corporation [CMC90b]. It is compatible because the supply rails match up during a placement and routing of the cells when both libraries are utilized. Also, the I/O of the new cell set were designed to be compatible with the [CMC90b] library. This permitted the use of a couple of cells that were previously available in this technology and hence did not need to be customized. The DFF and inverter leaf cells in the [CMC90b] library were required for the implementation. The DFF is used in the control circuit and the inverter is used to invert the outputs from the event-controlled storage cells. The complements from the event-controlled registers are needed for the inputs to the CVSL gates.

Two other leaf cells that could be of use would be a two-to-one multiplexer (MUX) and a three-to-one MUX. These devices fit the CVSL scheme nicely since

1. one DSM equals 0.8 microns

their implementation would result in a single gate delay for both.

4.2 The Layout of the Leaf Cells

The initial layout was based on the 75 design-scale micron (DSM) height requirement. All mask levels were manually placed but were not optimized by compaction tools or by back-annotating information from simulations. The layout of each cell was conducted with the goal of simply achieving an error-free design. Despite this, the leaf cells were relatively compact. Table 4.1 shows a comparison between a subset of standard cells used in industry [CMC90b] and the corresponding prototype leaf cells. The relationship between the number of inputs and the size of the prototype leaf cell shows that more inputs increases the percentage in size difference between the standard cells and the prototype cells. This is brought about by the dual-input requirement for the CVSL gates. However, because CVSL offers both true and complemented outputs it does not need an inverter to provide a complemented version of its output. If this is required in the decomposition of a function, the CVSL-gate becomes more space-efficient. In the case of the XOR/XNOR CVSL-gate, we see that it is quite comparable to the standard XOR-gate when an XNOR function is required. As previously mentioned in Section 3.1, this makes the CVSL-gates more effective when implementing arithmetic functions. Optimization of these leaf cells for size would improve these comparisons.

4.2.1 The Current Mirror

Appending the current mirror to the circuit was accomplished by placing two wide transistors on the top of the leaf cell. A schematic diagram of the two-

input CVSL AND/NAND-gate and it layout is depicted in Figure 4.1. Before the PMOSFETs are placed, the NMOSFETs were placed to determine the width of the cell. Once the width of the cell is determined, the PMOSFETs providing the current mirror were spaced so that they would assume the total width of the cell at the top. The NMOSFETs were placed as close to the ground rail as possible in an arrangement where the polysilicon gates run vertically parallel to each other. The width of the NMOSFETs are made as small as possible (3.12μm) unless they are in series which must be accommodated with a larger width depending on how many NMOSFETs are in series. All transistor lengths were made to be 1.2 microns which is the smallest geometry allowed for the technology used. The width of the two pull-up PMOSFETs are made to be four microns. The extra width is needed to increase the pull-up power over the NMOSFETs because of the higher mobility of carriers in the NMOSFETs.

Table 4.1: Size Comparisons for Two Leaf-Cell Sets

Logic Primitive	Standard Cell Area (μm^2)	Prototype Cell Area (μm^2)	% Difference	% Difference if inverter is added to Standard Cell
two-input NAND-gate[a]	1267.2	2006.4	58% greater	8% greater
a DFF with reset and set[b]	4108.8	4372.8	6% greater	not applicable
a four-input NOR-gate[c]	1766.4	3168.0	79% greater	43% greater
a two-input XOR-gate[d]	1876.8	2515.2	34% greater	equal

a. a two-input CVSL AND/NAND-gate for the prototype
b. an event-controlled storage cell for the prototype
c. a four-input CVSL OR/NOR-gate for the prototype
d. a two-input CVSL XOR/XNOR-gate for the prototype

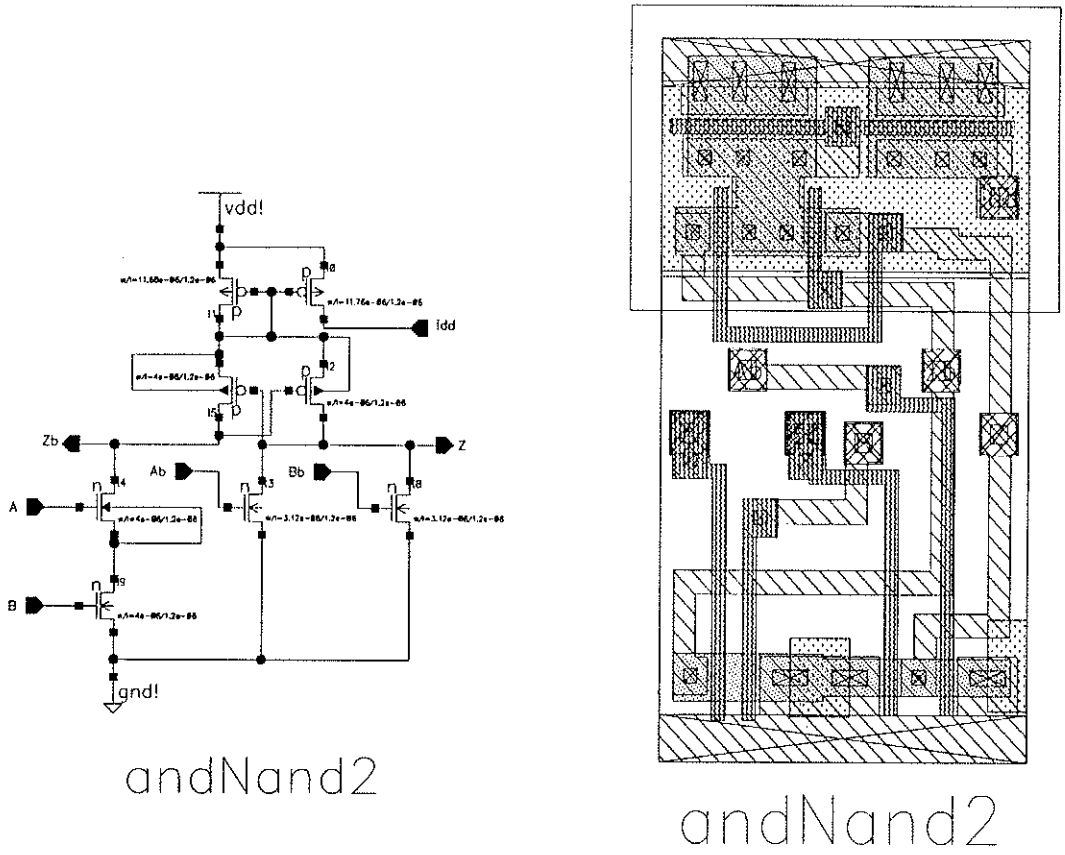


Figure 4.1: Schematic Diagram and Layout of a Two-Input CVSL AND/NAND-Gate

4.2.2 The Input/Output Terminals

The complementary outputs and the complementary inputs were placed so that two CVSL gates could be abutted with stub routing between them. The number of I/O is substantially increased with this dual-rail technology but the stub routing eliminates the overhead for one set of complementary inputs (if the placement and routing takes advantage of this capability). The two-input AND/NAND-gate has seven I/O (including the *Idd* port) while a standard CMOS AND-gate primitive has three I/O. Routing wire and silicon overhead is thus

increased when using dual-rail logic over the use of single-rail logic.

4.3 Simulation Results

The speed of each cell could be improved but *HSPICE* simulations verified their functionality. Despite the fact that these customized cells were not optimized, they were relatively compact and their performance in the *HSPICE* simulations showed reasonable levels of speed. Table 4.2 shows the propagation delays during various transitions in the CVSL-gate implementations, the Muller C-element, and the event-controlled storage cell. These simulations were conducted without the use of a capacitive load on the output. Also, the values supplied are the worst-case output propagation delays which are determined by one output pattern that causes the longest delay for that particular cell. The results show that the high-to-low propagation delays are comparable to other CMOS logic families but the low-to-high propagation delay needs to be improved. It was considered that increasing the size of the two PMOSFET pull-ups in each CVSL-gate would significantly improve performance of the rising transition. Doubling the width of the two PMOSFETs was simulated but this did not improve the rise time. The main detriment to the gates is the current mirror on the power rail. Despite this drawback, the circuit will optimize its speed because of its self-timed property which is one of the main reasons for using this scheme.

Table 4.2: Propagation Delays from Simulation

Logic Primitive	Z↑(ns)	Z↓(ns)	Zb↑(ns)	Zb↓(ns)
two-input AND/NAND	1.3	0.1	1.4	0.2
three-input AND/NAND	1.5	0.1	1.8	0.2
four-input AND/NAND	1.7	0.1	2.3	0.4
two-input OR/NOR	1.3	0.2	1.2	0.1
three-input OR/NOR	1.7	0.2	1.5	0.1
four-input OR/NOR	2.1	0.1	1.5	0.1
two-input XOR/XNOR	1.5	0.2	1.5	0.2
Muller C-element	1.1	1.6	1.7	0.8
event-controlled cell[a]	1.6(1.1)	1.4(0.6)	na	na

a. The first value given is the delay from flowthrough operation where the data can pass freely from input to output. The values in brackets is the delay when a pass signal controls the opening of the cell which allows data flowthrough.

CHAPTER 5

The Circuit Prototype

To verify the scheme in silicon, a circuit prototype was chosen and implemented with the leaf cells. The chosen test circuit is a four-bit Booth-encoded multiplier. Schematic capture was used as the design method. Complete schematics of the circuit are provided in Appendix B. A layout showing the two metal layers of the IC is depicted in Figure 5.1. The 1.2 micron CMOS process used to design the leaf cells resulted in an IC with a core area of $1.52mm^2$.

5.1 The IC Layout

The placement and routing was accomplished automatically through the use of *Cadence Cell Ensemble*. This satisfies one of the goals of the thesis which was to be able to implement a design without any manual placement. Custom layout design requires a skilled designer and the manual placement is time consuming. Once the schematic is completed, it is only a matter of seconds for the automatic P&R algorithm to do its job.

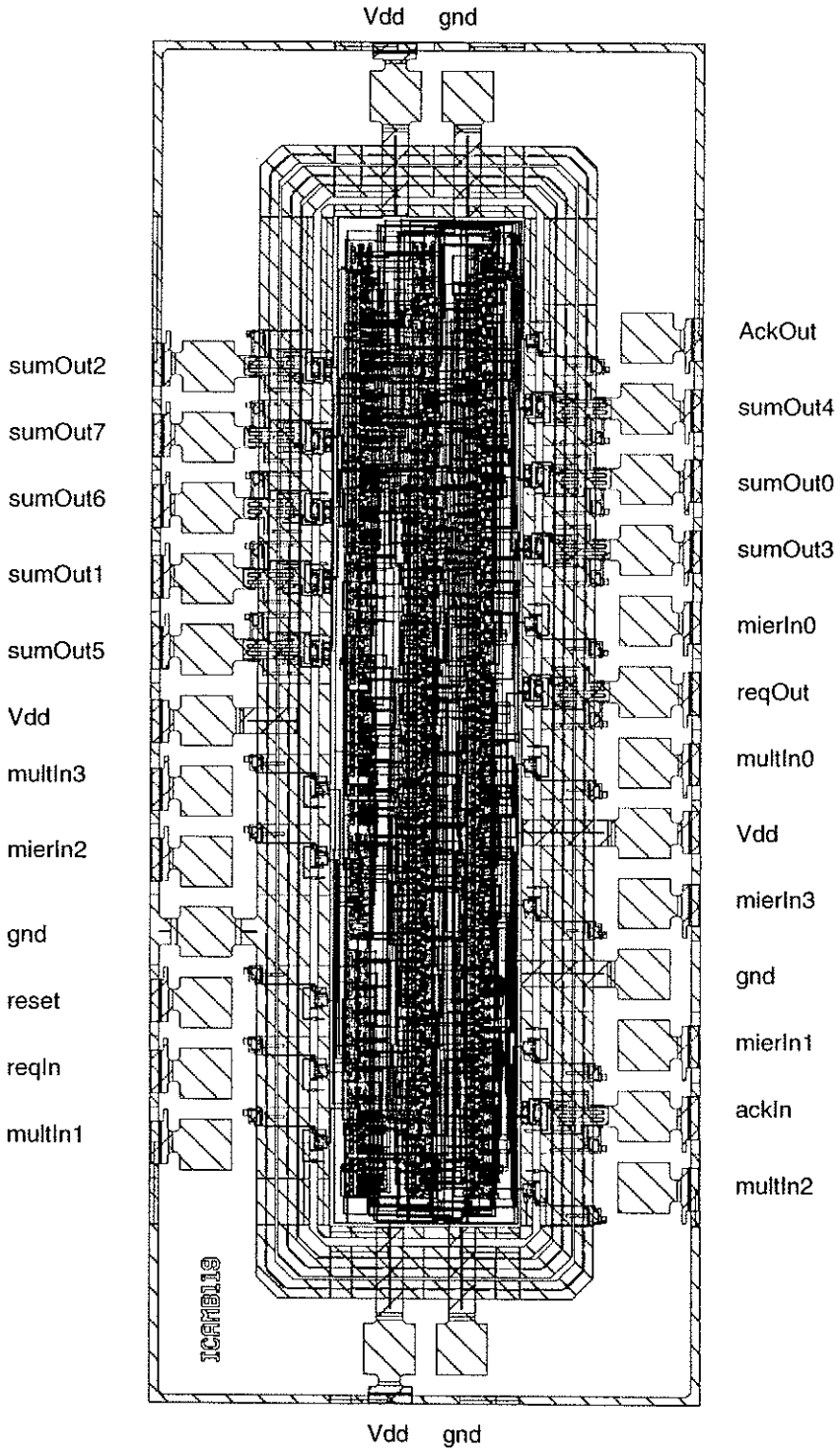


Figure 5.1: Layout of the Four-Bit Booth-Encoded Multiplier (metal layers one and two only)

5.2 Booth-Encoded Multiplication

Multiplication in circuits is best accomplished using parallel multiplication for fast operation. Array multiplication accomplishes this by summing the partial products from successive stages of parallel adders. An improvement over array multiplication is the computation of the partial products by observing three bits of the multiplicand at a time. This is called Radix-4 multiplication or Booth-encoded multiplication. The three bits of the multiplicand are re-coded to determine whether a 1*, -1*, 2*, or -2* is added to that rank of the multiplicand. The multiplier works on 2's complement integers.

The design used is based on the schematics provided in [WeEs93]. The use of Booth-encoding is not as efficient in a four-bit multiplier as one that computes on larger operands such as a sixteen-bit multiplier. The four-bit multiplier is reduced from four partial product stages to three while a sixteen-bit multiplier is reduced from sixteen partial product stages to eight. A floorplan of the four-bit multiplier is shown in Figure 5.2.

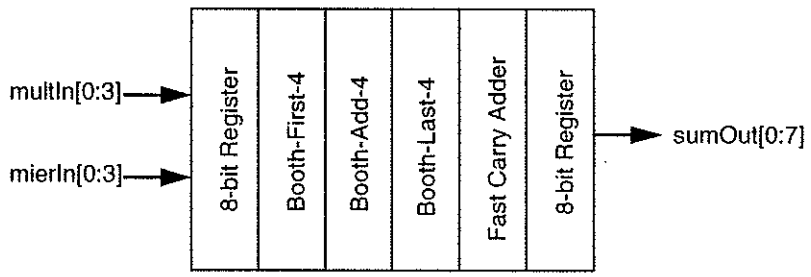


Figure 5.2: Floorplan of the Prototype Multiplier Circuit

The design in [WeEs93] used pass-logic design to implement the multiplier circuit. Those gates were converted to the CVSL-gate family. The two registers each comprise of eight event-controlled storage cells which are controlled by the application of the pass and capture signals. Since the majority of logic is devoted to the processing logic block, the requirement for the asynchronous control circuitry to be less than 30% of the silicon area used is satisfied. Even when the current mirrors are computed as part of the control circuit area (approximately 15% of the area of each gate), the processing circuitry still dominates.

To verify the functionality of the multiplier design, a *Xilinx* [Xili94] FPGA (XC4003) was configured with the circuit and tested. As a result of that study, a micropipeline circuit was emulated in the *Xilinx* FPGA architecture and showed considerable speed [GRM94].

The resulting layout required over 180 CVSL gates in the processing logic block between the registers. Originally, it was intended that a large processing logic block would severely test the capabilities of the control circuit. This was overwhelmingly satisfied by the multiplication circuit. With such a large number of gates driving the *Idd* net, the functionality of the comparator circuit will be effectively tested since simulations were not conducted with large processing logic blocks. Unfortunately, after we had committed the first prototype, more strident simulations showed that adjustments were needed in the widths of certain MOSFETs to provide a faster implementation. The next section will detail this revelation.

5.3 Simulations of the Control Circuit

The control circuitry was evaluated with a rudimentary simulation. A two-stage neo-micropipeline is instantiated with a schematic diagram (see Figure 5.4) and simulated with *HSPICE*. The logic block between the two registers is an eight-to-one multiplexor (MUX). We use a level-three *HSPICE* simulation with extracted capacitances and FET netlists from the leaf-cell layouts. The parasitics of the wires connecting the gates are not included in the simulation. However, gate and drain capacitances are typically more significant than wire-interconnect capacitances.

To make the schematic diagram more simple (i.e. less wires) a different notation was used and must be explained. The dashed lines represent the two wires of a differential signal. If a straight line is tapped off this differential line it may represent a true or complement signal wire depending on whether the line is terminated with a bubble (inversion) symbol. The current-mirror symbol that appears on a gate in the logic block signifies a coupling of the *Idd* output to the input of the comparator which is also marked with the current-mirror symbol and the label *CMC*. The C-elements are also marked with the current-mirror symbol but only the C-element in the first stage is coupled with the comparator. The C-element from the second stage may be connected to the *Idd* net of that stage otherwise it is tied high (which it is in this simulation).

The execution of this simulation revealed a number of shortcomings in the initial leaf-cell library. First, the PMOSFET *I26* (see Figure A.4) required a width increase from 4.4 to 16 microns because the quiescent current in the logic block

increases with the number of gates. The quiescent current is created because the PMOSFET in each gate that sources *Idd* is in triode operation and will source a small leakage current even when the gate voltage is at the threshold level. Fortunately, this effect is linearly proportional to the number of gates in the logic block. Therefore, a small set of comparators with different widths for PMOSFET *I26* would be needed to optimize logic blocks with different numbers of gates.

Second, it is not necessary for the PMOSFET that sources *Idd* to be equal in width to the PMOSFET that provides the channel for the power supply. When examining these simulations it was discovered that provided the C-element sources a significant current to the *Idd* net, only a small current was required to maintain the output of the comparator high. This allows us to use the smallest width of PMOSFET (3.12 microns) for sourcing the Idd net of any gate. The exception is the C-element which retained the original size of its PMOSFETs because it must provide that initial surge of current to the comparator when a request is forwarded.

These optimizations were made to the netlist and the simulation executed. The throughput speed of this circuit with a specific input sequence is at a minimum rate because the MUX output will toggle with each new data-word input. The measurements from simulation reveals a throughput cycle time of 26 nanoseconds (or 38.5 MB/s). If the data word remained the same for each cycle the circuit attained a cycle time of 14 nanoseconds (or 71.4 Mbyte/s). If these two values are the extremes of throughput operation the average speed could have a cycle time of 20 nanoseconds (or 50 Mbyte/s). This performance is comparable

to modern processor speeds which use smaller CMOS geometries and commercially-developed standard cells. Simulation data is provided below in Figure 5.3. Referring to Figure 5.4, trace *63* is *D0*, trace *60* is *Qouti*, trace *88* is *Qout*, trace *43* is *AckIn*, trace *7* is *Vout*, and trace *53* is *ReqOut*.

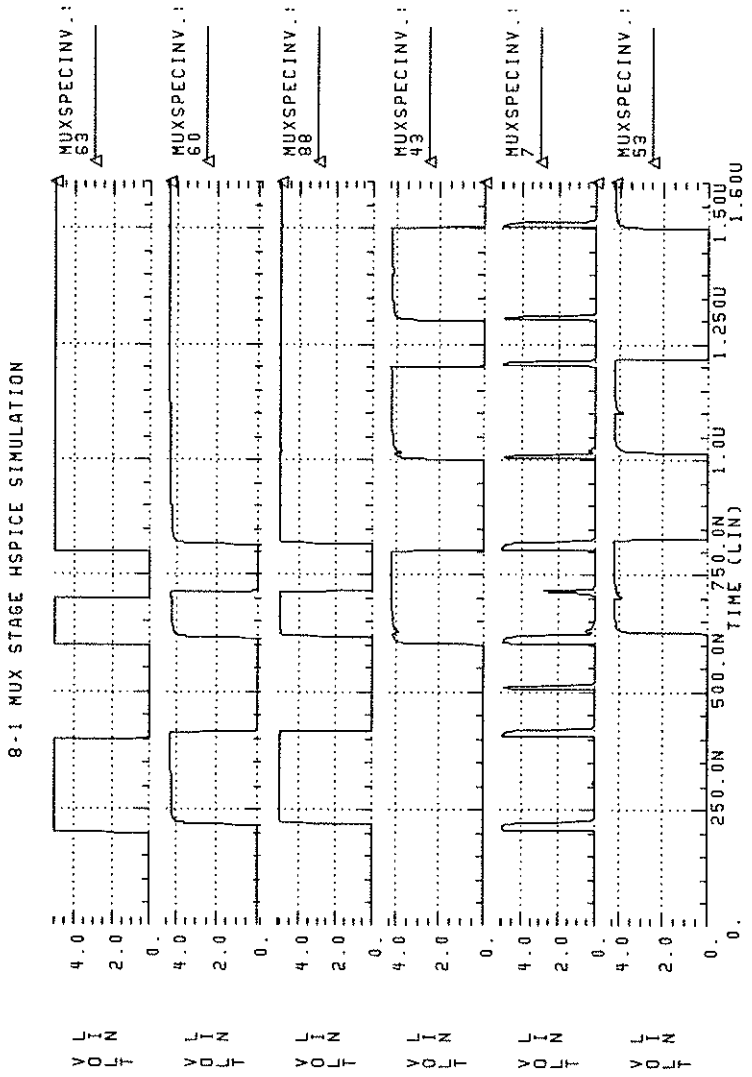


Figure 5.3: Simulation Data for the 8-to-1 MUX Stage

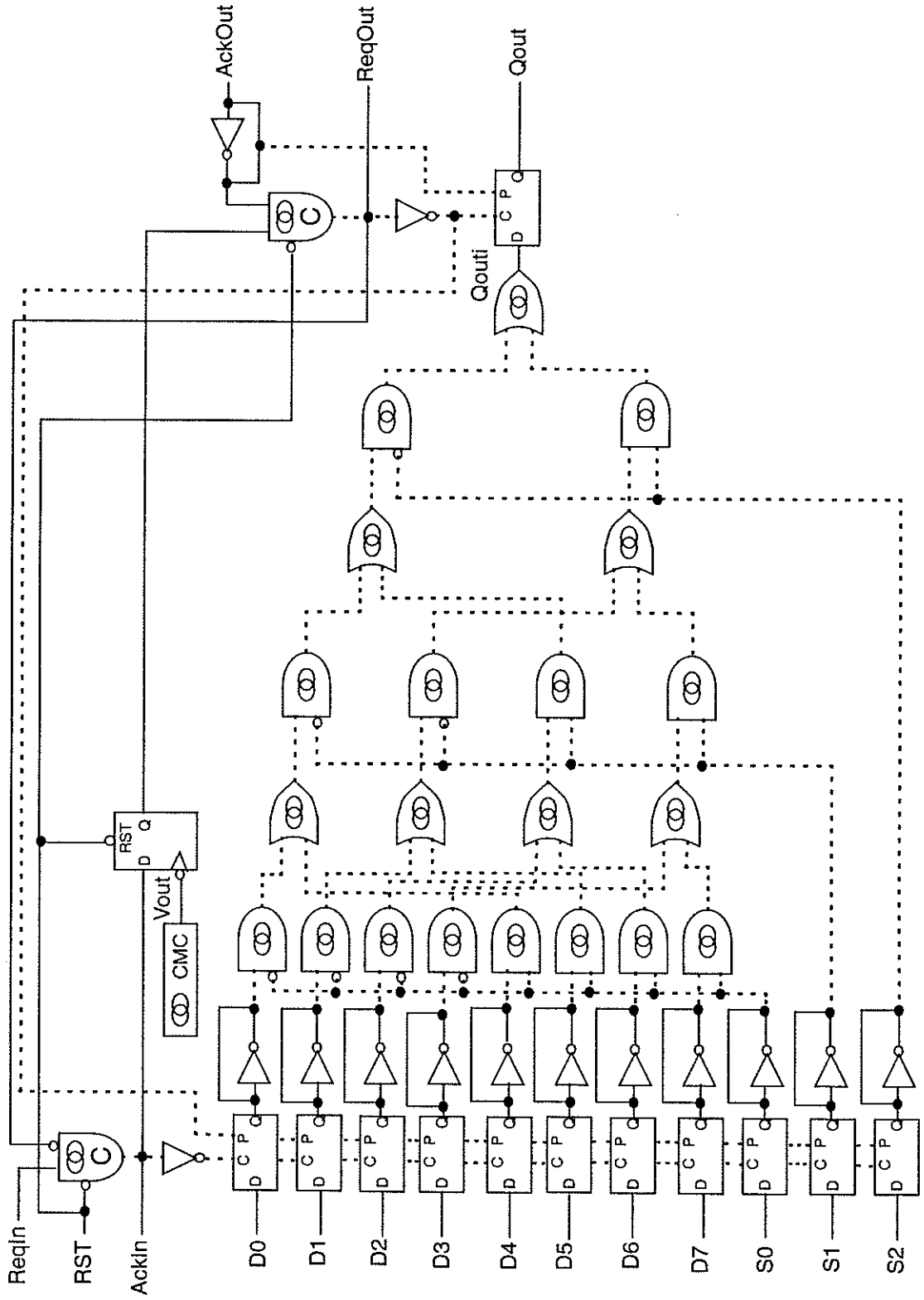


Figure 5.4: Schematic Diagram of an 8-to-1 Multiplexor Stage

CHAPTER 6

The Test Procedure

Several characteristics of this circuit design must be considered to confirm its validity. Power draw, size, and speed are the main concerns for modern IC design. The size of the proposed circuit has been evaluated as being fit for acceptance and the power of this circuit is effectively lower than synchronous designs because of the inherent low-power aspect of asynchronous design. The speed of the circuit shows potential in the limited simulations that were performed but it does not provide concrete evidence. A test of the multiplier IC will quantitatively offer conclusions in this regard. In retrospect, the speed of this first prototype will not be altogether noteworthy due to the lack of transistor width optimization in the comparator leaf cell.

6.1 The Test Circuit Setup

The two tests that will determine the performance of the multiplier circuit will check for latency and throughput of the IC. Figure 6.1 depicts the test setup to enable the testing of the IC for these two performance measures.

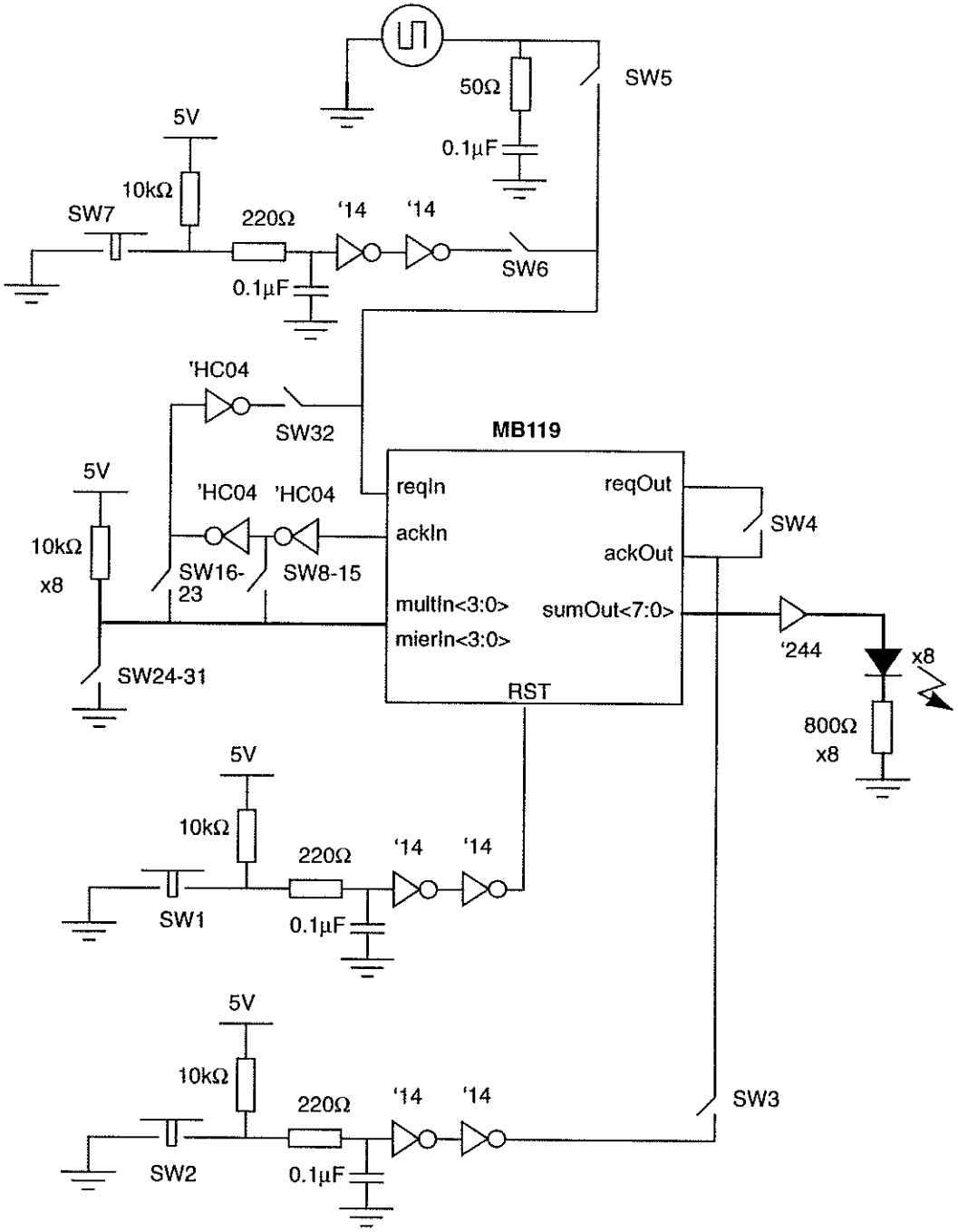


Figure 6.1: The Test Setup

Each I/O is capable of being probed by an oscilloscope or logic analyser because of the provision of probe points on the test head. The input to the *RST* pin is debounced as is one of the options for the *reqIn* pin and the *ackOut* pin. The other options for the *reqIn* pin is the ability to provide an input from a function generator (50Ω output) or the inverted output from the *ackIn* pin. The *ackOut* pin can also be fed by the *reqOut* pin. The *sumOut* pins drive LEDs. The inputs to the *mierIn* and *multIn* pins can be configured to provide alternating inputs every cycle with each bit being in-phase or out-of-phase with the *ackIn* pin output. Also, they can be manually set.

A number of switches provide configuration and control of the test procedure. They are:

- **SW1:** This switch allows the manual toggling of the *RST* pin.
- **SW2:** This switch allows the manual toggling of the *ackOut* pin.
- **SW3:** When this switch is *ON* it enables the use of *SW2*.
- **SW4:** When this switch is *ON* the *ackOut* pin is fed by the *reqOut* pin.
- **SW5:** When this switch is *ON* the function generator input is applied to the *reqIn* pin.
- **SW6:** When this switch is *ON* it enables the use of *SW7*.
- **SW7:** This switch allows the manual toggling of the *reqIn* pin.
- **SW8-15:** Each of these switches enables its respective input pin to be out-of-phase with the output of *ackIn*.
- **SW16-23:** Each of these switches enables its respective input pin to be in-phase with the output of *ackIn*.
- **SW24-31:** Each of these switches enables its respective input pin to be tog-

gled manually.

- **SW7:** When this switch is *ON* it enables the *reqIn* pin to be out-of-phase with the *ackIn* pin output.

6.2 The Latency Test

This test will measure the total delay for the circuit to react to a request event or an acknowledge event. The *reqIn* pin is manually toggled and the transition is used to trigger one channel of a storage oscilloscope. The second channel of the storage oscilloscope monitors the *reqOut* pin. The measured delay between these two events will determine the forward latency of the circuit.

The reverse direction of the control signals can also be tested with this method. After the request is sent, the *ackOut* pin can be manually toggled and this event will be used to trigger the first channel of the storage oscilloscope. The second channel will be used to monitor *ackIn*. This latency measurement should be less than the forward latency since this operation only depends on the operation of the Muller C-elements and not the current-mode circuit (i.e., the request signal is bounded but the acknowledge signal is not).

When measuring the forward latency, various combinations on the input pins to the multiplier circuit will vary the delay. If a hamming distance of zero occurs between successive bit patterns, then the forward latency should theoretically be 40ns (according to simulations) plus the delay of the I/O pads. The forward latency of the circuit with transitioning inputs is effectively measured by using the *ackOut* pin to cause the circuit to bring an acknowledge signal through the

circuit to the first stage where the Muller C-element can toggle which will cause the capture of data in the first register and simultaneously send a request signal to the next register.

6.3 The Throughput Test

The throughput test will determine the rate of data throughput in the circuit. Again, this measurement will depend on the sequential states of the multiplier inputs. Various combinations can be created using switches *SW16-31*. The *reqIn* pin will be fed with the function generator input and the output will be monitored on the *reqOut* pin. The input frequency is increased until the two signals (*reqIn* and *reqOut*) are no longer synchronized which will determine the maximum throughput level. As the IC heats up due to its high speed operation, this measurement value will likely drop. This test was successfully used to test the throughput of a micropipeline emulation in a *Xilinx* FPGA in [GRM94].

6.4 Functional Testing

Exhaustive testing of all the possible input/output combinations could be performed with the proposed test setup but this would be a tedious job to manually toggle those switches. There are 256 possible input patterns for the multiplier. A test head interfaced with a *VXIbus* I/O pattern card will be used to provide a functional test which will include all of the possible input vectors [CMC93].

CHAPTER 7

Conclusions

This study examined the possibility of improving the micropipeline methodology. The delay element in the request path of the micropipeline provided a bundled-data condition which reduces the efficiency of the micropipeline and complicates the synthesis of a data processing circuit. This research has produced a novel CSCD circuit that is to be incorporated with a micropipeline. The CSCD circuit eliminates the need for delay elements and their associated problems. Also, this circuit proves to be a more feasible implementation than previous completion-detection methods. The key to the improvement is the use of current-mode circuitry in the comparator circuit. The neo-micropipeline has the following characteristics:

- It follows the two-cycle signalling convention.
- The control circuit is relatively small compared to previous completion-detection methods.
- The control circuit has lower power requirements than previous completion-detection methods.

- The CVSL gates that are used in the processing logic block have a reduced power requirement because of the reduced voltage swing on the outputs (0 to 4.2Volts).
- The CVSL gates are more useful for decomposing logic functions because they have differential outputs. Moreover, the logic block will require fewer CVSL gates for implementing a function than standard logic gates.
- The CVSL gates are most effective when used for implementing arithmetic functions because of the reduced size of XOR-gate implementations.
- The CVSL gate is slower than standard logic primitives because of the parasitic effects on a rising transition of an output. However, this effect is countered by the inherent ability of the circuit to optimize its speed. The speed optimization occurs because the circuit is delay-insensitive which allows its speed to be dependant only on the physical characteristics of the circuit and the operating conditions.
- The average delay in the forward latency of the data processing circuit is reduced due to the state similarity characteristics of sequential inputs. If the hamming distance between two sequential inputs is small or zero, the current draw in the logic block will taper off more quickly than for other sequential patterns. The circuit can react to this early completion and trigger the request signal through to the next register.
- The circuit can be automatically placed-and-routed with common CAD tools which also simplifies the design entry.
- The logic block does not require hazard-free synthesis. This further simplifies the design-entry stage since any logic decomposition of a function will be suitable.

- There will be a reduction in the number of test patterns required to test a data processing circuit incorporating the proposed circuit because of the increased fault coverage a pattern may have.

7.1 Applications

With the characteristics listed prior, the proposed circuit is considered a feasible approach to designing asynchronous circuits. Simulations show that further optimization of the leaf cells will result in speeds that are comparable to modern devices. Combined with its low power requirements, this circuit will be suitable for various applications.

A supercomputer requires fast message passing to effectively use its parallel processing abilities. Asynchronous communication between processing elements has been highly regarded for this function [TrDu92 and Fitc93]. Also, because of the problems with clock skew in large systems, the asynchronous approach to interconnecting processors provides a simpler design requirement than for synchronous systems. However, most supercomputers use locally synchronous processors.

The proposed circuit in this thesis is aimed at processing elements. Therefore, in a supercomputing environment, the ideal system will be globally and locally asynchronous. Interconnections and FIFOs would maintain the classical micropipeline approach and processing logic would utilize the neo-micropipeline. The main difficulty for such an implementation is that the entire system would have

to be custom designed. FPGAs could be used for implementing the micropipelined message passing but the processors would have to be custom built.

In a wafer-scale environment, the circuit would be highly feasible. It could be used to implement single-instruction multiple-data (SIMD) structures. Also, a mesh architecture of cellular automata could be used to implement some high speed algorithms [SBPS94]. The asynchronous circuit would be able to *relax* to a minimum solution rather than working in lockstep fashion. The characteristic of the neo-micropipeline allows each processing element to be globally independent of the speed of other processes which would optimize speed. The temporal and spatial locality of such algorithms are well-suited to an asynchronous architecture.

Low power requirements is one of the main problems when designing portable applications. Today's high-speed reduced-instruction-set IC (RISC) processors are not readily adaptable to portable applications because of their power consumption. The clocking network in a large IC will draw a significant amount of transient current because it much charge and discharge the clock grid (a large capacitance load) even when the circuit is not actively processing data. The proposed circuit could be used to develop a low-power RISC processor for this application. Overall, this circuit could be used for any design that requires low power and complex digital signal processing. It is speculated that the use of this circuit in industry would highlight asynchronous design as an acceptable form of digital design.

7.2 Future Work and Goals

The testing of the fabricated IC is the next step in verifying the functionality of the proposed circuit. In any event, more stringent simulations will be conducted on the layout of the IC with *HSPICE.*

The change to the PMOSFET width in the comparator will be implemented in the next revision of the comparator leaf cell. This change (4.4 microns to 16 microns) should significantly speed up the circuit. All leaf cells will be optimized for size and speed if possible.

Future IC prototypes will be more complex than the four-bit Booth-encoded multiplier. Several applications have been considered. One application is the development of a neural network processor that utilizes stochastic arithmetic. The problems with feedback and arbitration in micropipelines is another interesting topic and will need to be examined in a more complex design.

Schematic design entry needs to be improved by removing the need to provide the complements of every signal in the processing logic block. The redundant wires and labels in the schematic makes for an unwieldy form of design entry. A hardware description language could simplify the design entry. Use of VHDL (very high speed IC hardware description language) will be investigated as a solution.

Bibliography

[Brun91] E. Brunvand, "A Cell Set for Self-Timed Design Using Actel FPGAs", Technical Report (UUCS-91-013), Department of Computer Science, University of Utah, August 1991.

[ChMo73] T. J. Chaney and C. E. Molnar, "Anomalous Behaviour of Synchronizers and Arbiters", *IEEE Transactions on Computers*, vol. C-22, pp.421-422, April 1973.

[CMC90a] Canadian Microelectronic Corporation, "Design Rules and Process Parameters for the Northern Telecom CMOS4S Process", Technical Report (IC90-01), February 1990.

[CMC90b] Canadian Microelectronics Corporation, "The CMOS4S Standard Cell Library", Technical Report (ICI-021R0), March 1990.

[CMC93] Canadian Microelectronics Corporation, "User Manual: CMC VXI Test System", Technical Report (ICI-057), November 1993.

[DDH91] M. E. Dean, D. L. Dill, and M. Horowitz, "Self-Timed Logic Using Current-Sensing Completion Detection (CSCD)" *Proceedings of the 1991 IEEE International Conference on Computer Design*, IEEE Computer Society Press, pp.187-191, 1991.

[Fitc93] J. W. Fitchett, *A Locally Synchronous Globally Asynchronous Vertex-8 Processing Element for Image Reconstruction on a Mesh*, MSc Thesis, Department of Electrical and Computer Engineering, University of Manitoba, 1993.

[FrCu83] D. A. Freitas and K. W. Current, "CMOS Current Comparator Circuit", *Electronics Letters*, 19(17):695-697, August 1983.

[GaMc94] M. J. Gamble and R. D. McLeod, *An Asynchronous Circuit for Data Processing*, UK Provisional Patent, serial number (9406122.3), March 1994.

[GRM94] M. Gamble, B. Rahardjo, and R. D. McLeod, "Reconfigurable FPGA Micropipelines", *Proceedings of the 1994 Canadian Workshop on Field-Programmable Devices*, pp.3.1.1-3.1.7, June 1994.

[Hauc94] S. Hauck, *Asynchronous Design Methodologies: An Overview*, Department of Computer Science and Engineering, University of Washington, 1994.

[HeGr84] L. G. Heller and W. R. Griffin, "Cascode Voltage Switch Logic: A Differential CMOS Logic Family", *1984 IEEE ISSCC*, February 1984.

[IST90] O. A. Izosimov, I. I. Shagurin, and V. V. Tsylyov, "Physical Approach to CMOS Module Timing", *Electronics Letters*, 26(22):1835-1836, October 1990.

[Jaco89] G. M. Jacobs, *Self-Timed Integrated Circuits for Digital Signal Processing*, PhD Thesis, Electrical and Computer Science, University of California at Berkeley, 1989.

[KSRA91] S. Karthik, I. de Souza, J. T. Rahmeh, and J. A. Abraham, "Interlock Schemes for Micropipelines: Application to a Self-Timed Rebound Sorter", *Proceedings of the 1991 IEEE International Conference on Computer Design*, IEEE Computer Society Press, pp.393-396, 1991.

[LiGo92] A. Liebchen and G. Gopalakrishnan, "Dynamic Reordering of High Latency Transactions Using a Modified Micropipeline", *Proceedings of the 1992 IEEE International Conference on Computer Design*, IEEE Computer Society Press, pp.336-340, 1992.

[Lu93] S.-L. Lu, "Design of Hardware Efficient Self-Timed Circuits", *Electronics Letters*, 29(1):6-7, January 1993.

[MaNi91] W. P. Maly and P. J. Nigh, *Built-In Current Testing of Integrated Circuits*, US Patent, serial number (5025344), June 1991.

[MBM89] T. H.-Y. Meng, R. W. Brodersen, D. G. Messerschmitt, "Automatic Synthesis of Asynchronous Circuits from High-Level Specifications", *IEEE Transactions on Computer-Aided Design*, 8(11):1185-1205, November 1989.

[PDF+92] N. C. Paver, P. Day, S. B. Furber, J. D. Garside, and J. V. Woods, "Register Locking in an Asynchronous Microprocessor", *Proceedings of the 1992 IEEE International Conference on Computer Design*, IEEE Computer Society Press, pp.351-355, 1992.

[Raha93] B. Rahardjo, "Asynchronous Hardware Design: An Overview", Technical Report (UMECE TR-92-215), Department of Computer and Electrical Engineering, University of Manitoba, February 1993.

[Røin94] P. T. Røine, "Building Fast Bundled Data Circuits with a Specialized Standard Cell Library", *International Symposium on Advanced Research in Asynchronous Circuits and Systems*, November 1994.

[SBPS94] N. R. S. Simons, G. E. Bridges, B. W. Podaima, and A. Sebak, "Cellular Automata as an Environment for Simulating Electromagnetic Phenomena", *IEEE Microwave and Guided Wave Letters*, 4(7):1-3, July 1994.

[Seit80] C. L. Seitz, "System Timing" in *Introduction to VLSI Systems*, C. A. Mead and L. A. Conway (Eds.), Addison-Wesley, 1980.

[SSM94] R. F. Sproull, I. E. Sutherland, C. E. Molnar, "The Counterflow Pipeline Processor Architecture", *IEEE Design & Test of Computers*, 11(3):48-59, Fall 1994.

[Suth89] I. E. Sutherland, "Micropipelines", *Communications of the ACM*, 32(6):720-738, June 1989.

[TLH90] C. Toumazou, F. J. Lidgey, and D. G. Haigh, *Analogue IC Design: The Current-Mode Approach*, Peter Peregrinus Limited, 1990.

[TrDu92] R. Traylor and D. Dunning, "Routing Chip Set for Intel Paragon Parallel Supercomputer", *Hot Chips IV*, Stanford University, August 1992.

[WeEs93] N. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, Addison-Wesley, 1993.

[Xili94] Xilinx Incorporated, *The Programmable Logic Data Book*, 1994.

Appendix A

The Leaf-Cell Library

Schematic diagrams and layouts of each leaf cell are provided in this appendix. Information on the schematic diagram comprises the netlist of the cell and the width/length of the individual transistors. Each leaf cell is 75 DSM in height and the I/O placement is matched to the rules in [CMC90b].

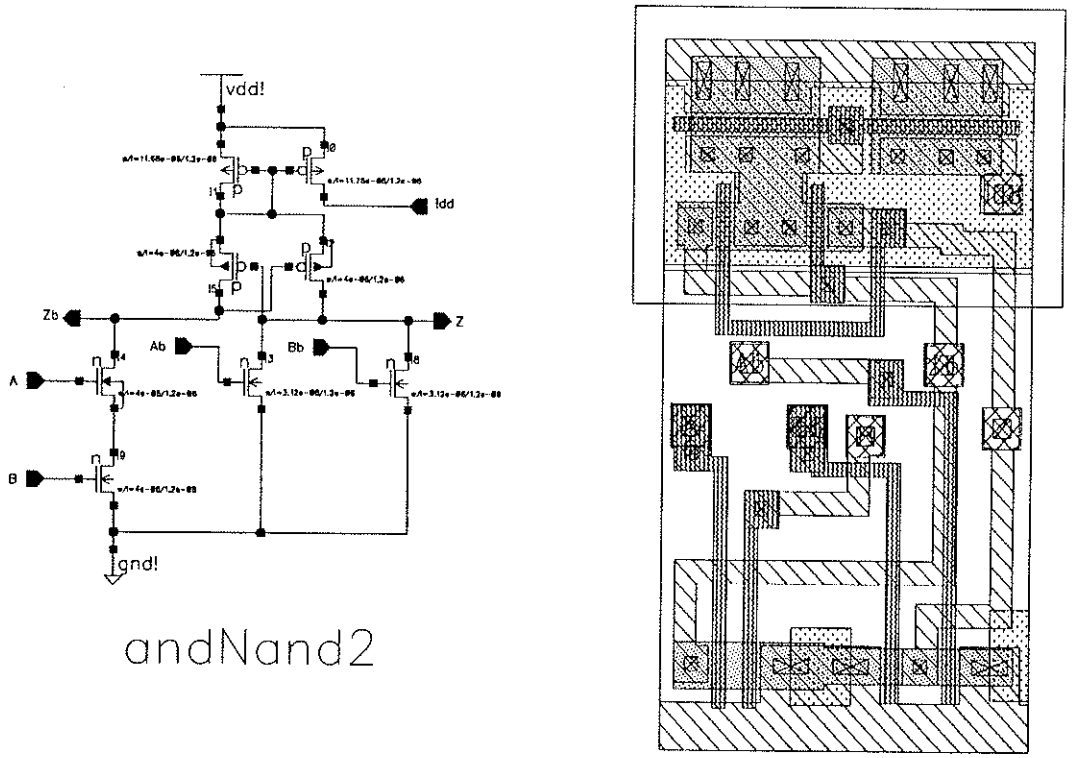


Figure A.1: Schematic and Layout of a Two-Input CVSL AND/NAND Gate

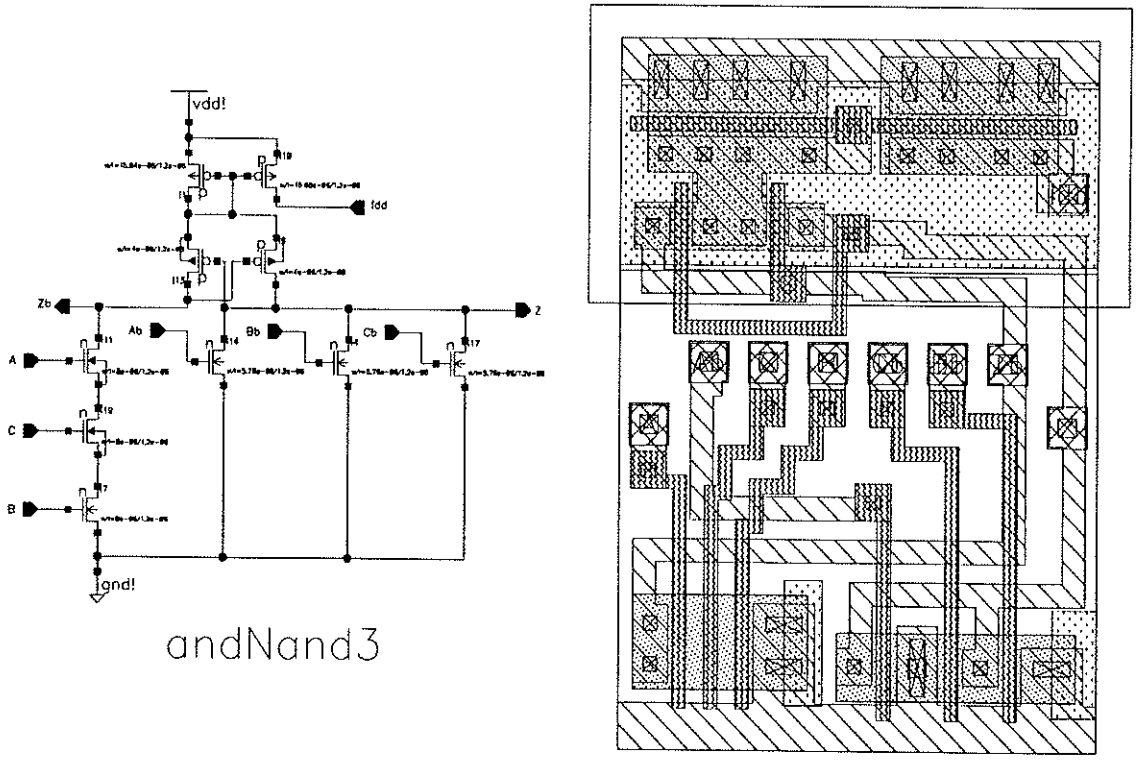


Figure A.2: Schematic and Layout of a Three-Input CVSL AND/NAND Gate

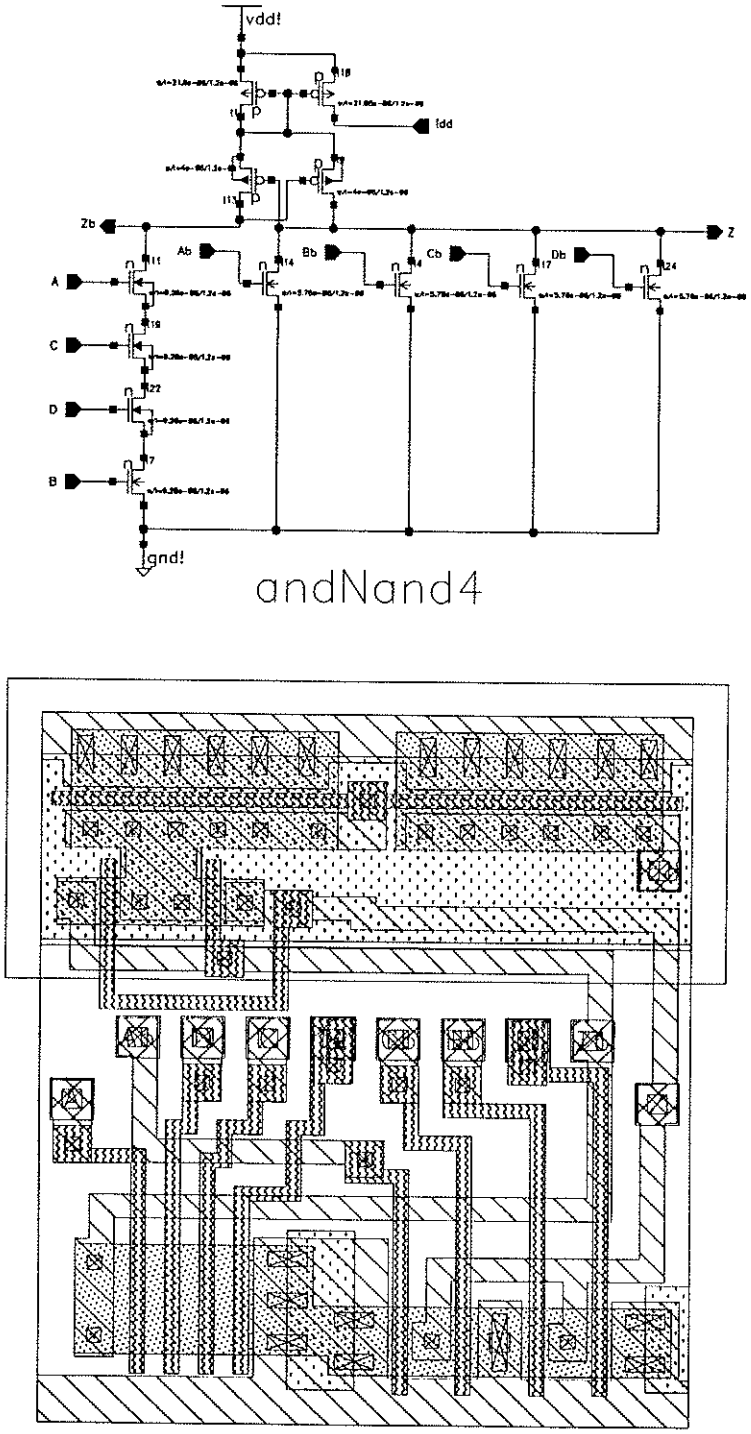


Figure A.3: Schematic and Layout of a Four-Input CVSL AND/NAND Gate

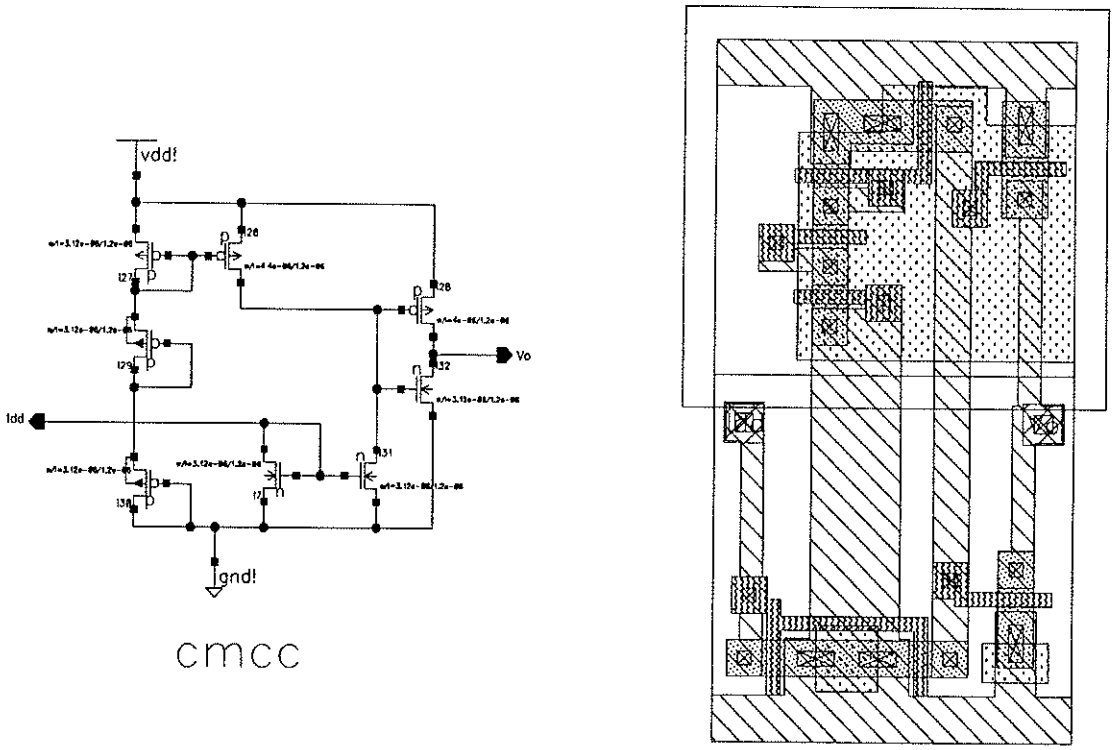


Figure A.4: Schematic and Layout of a Current-Mode Current Comparator

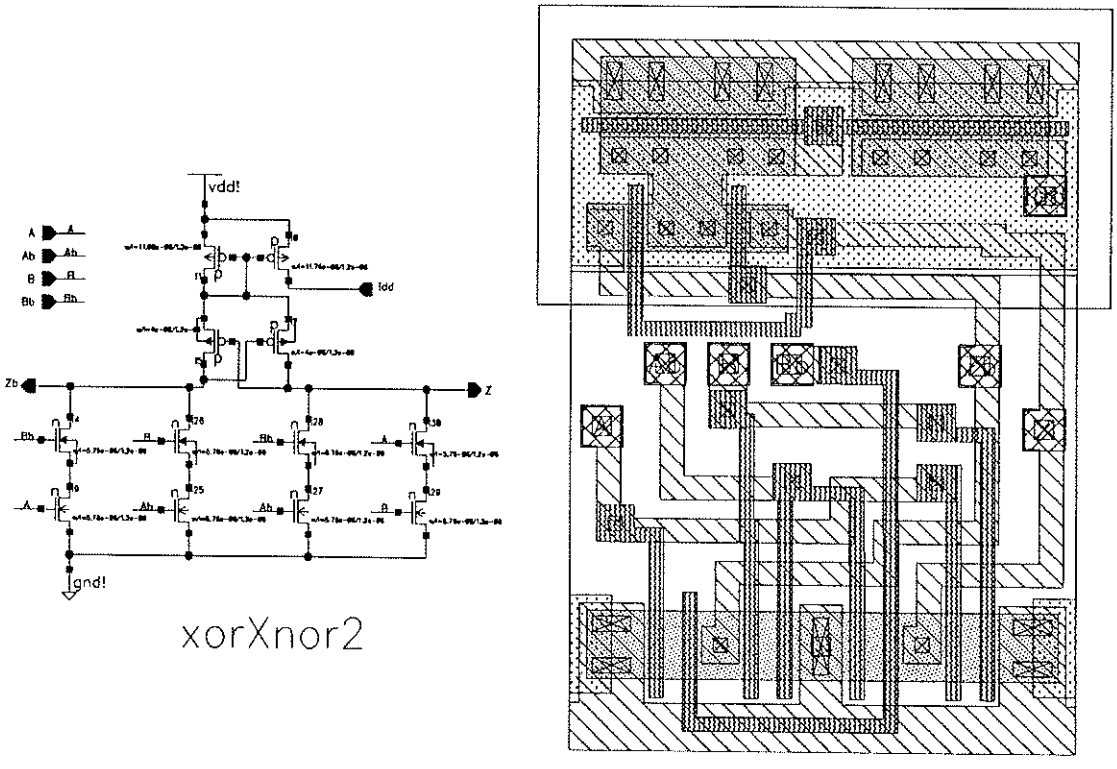
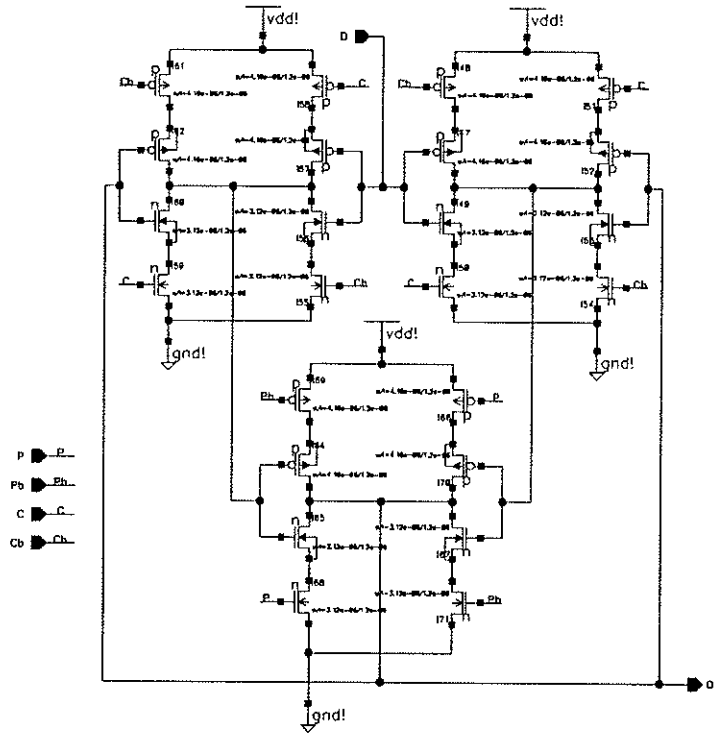


Figure A.5: Schematic and Layout of a Two-Input CVSL XOR/XNOR Gate



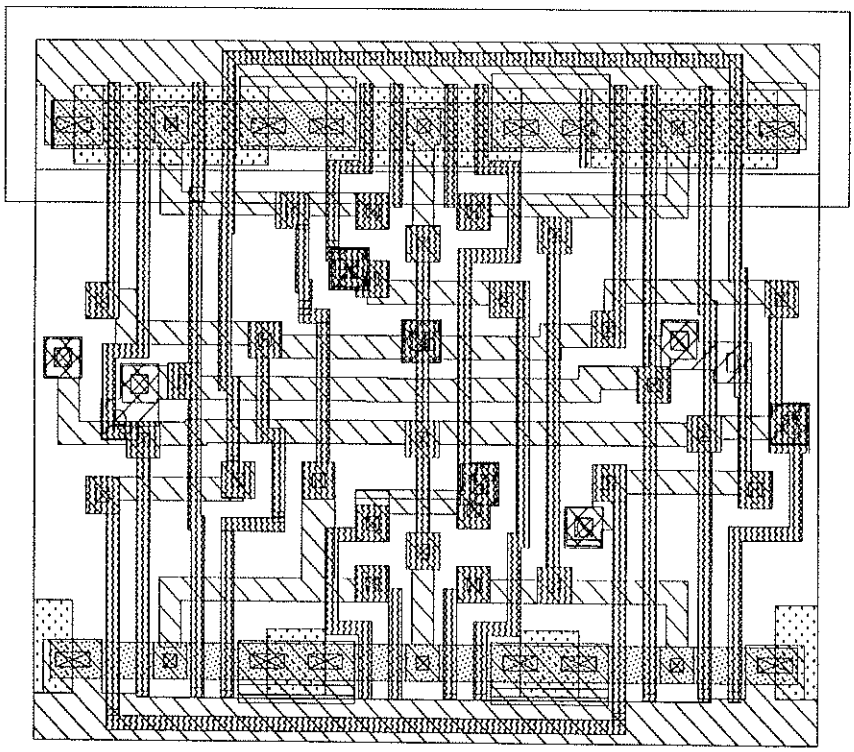


Figure A.6: Schematic and Layout of an Event-Controlled Storage Cell

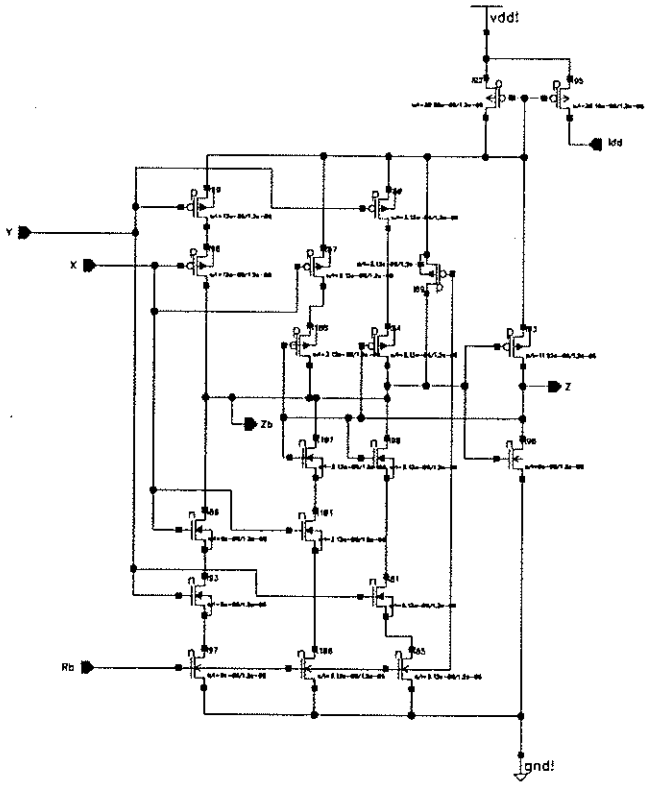


Figure A.7: Schematic and Layout of a Muller-C Element with Reset

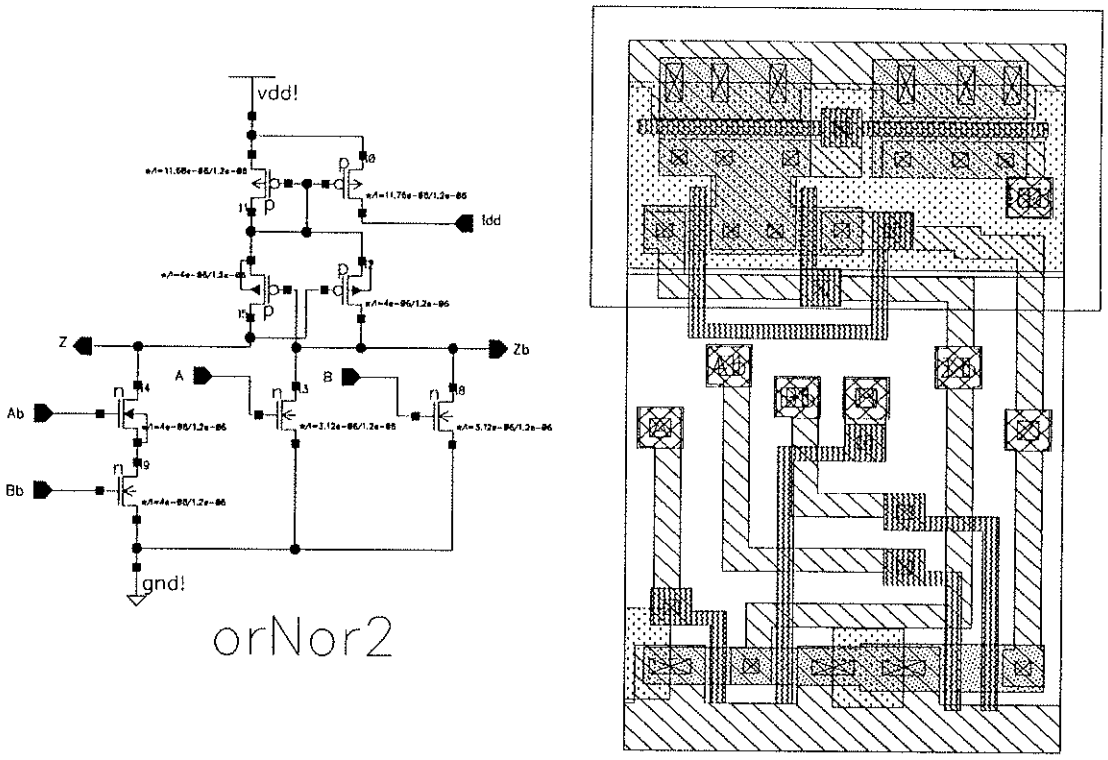


Figure A.8: Schematic and Layout of a Two-Input CVSL OR/NOR Gate

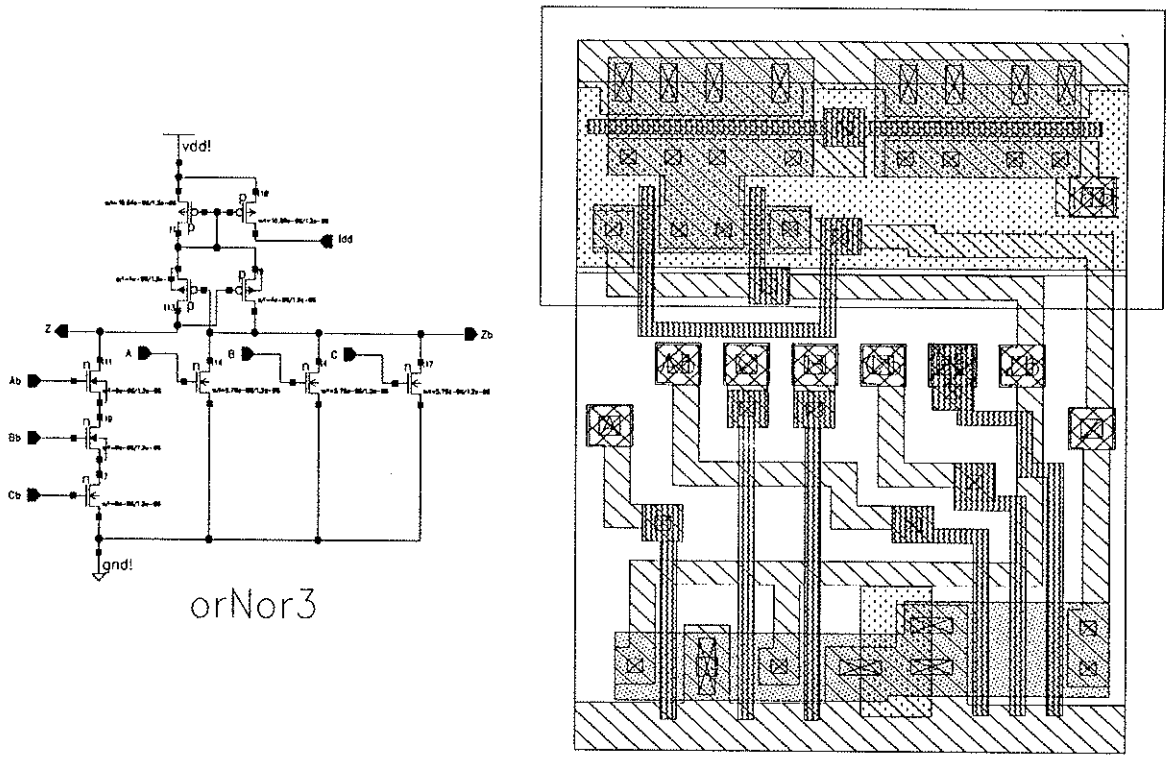


Figure A.9: Schematic and Layout of a Three-Input CVSL OR/NOR Gate

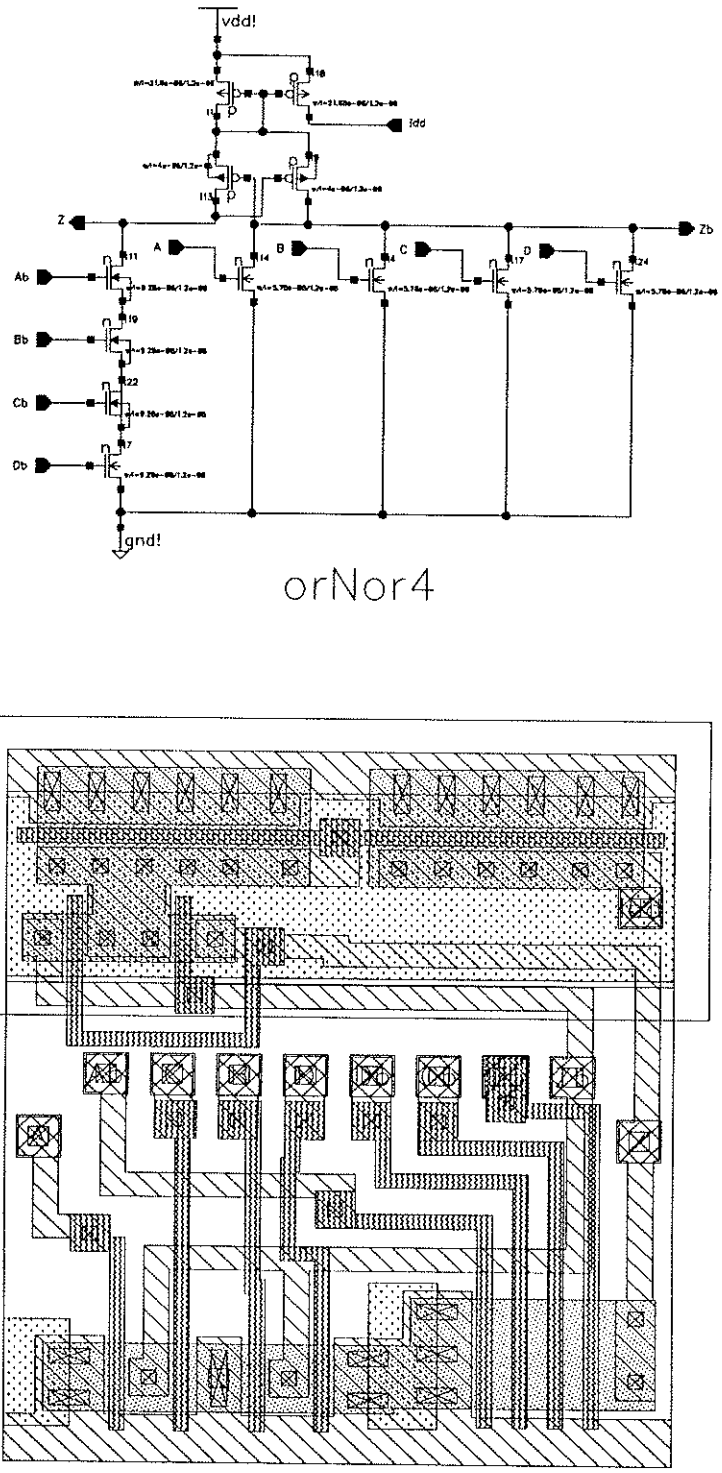


Figure A.10: Schematic and Layout of a Four-Input CVSL OR/NOR Gate

Appendix B

The Multiplier Circuit

The schematic diagrams for the four-bit Booth-encoded multiplier design are provided in this appendix.

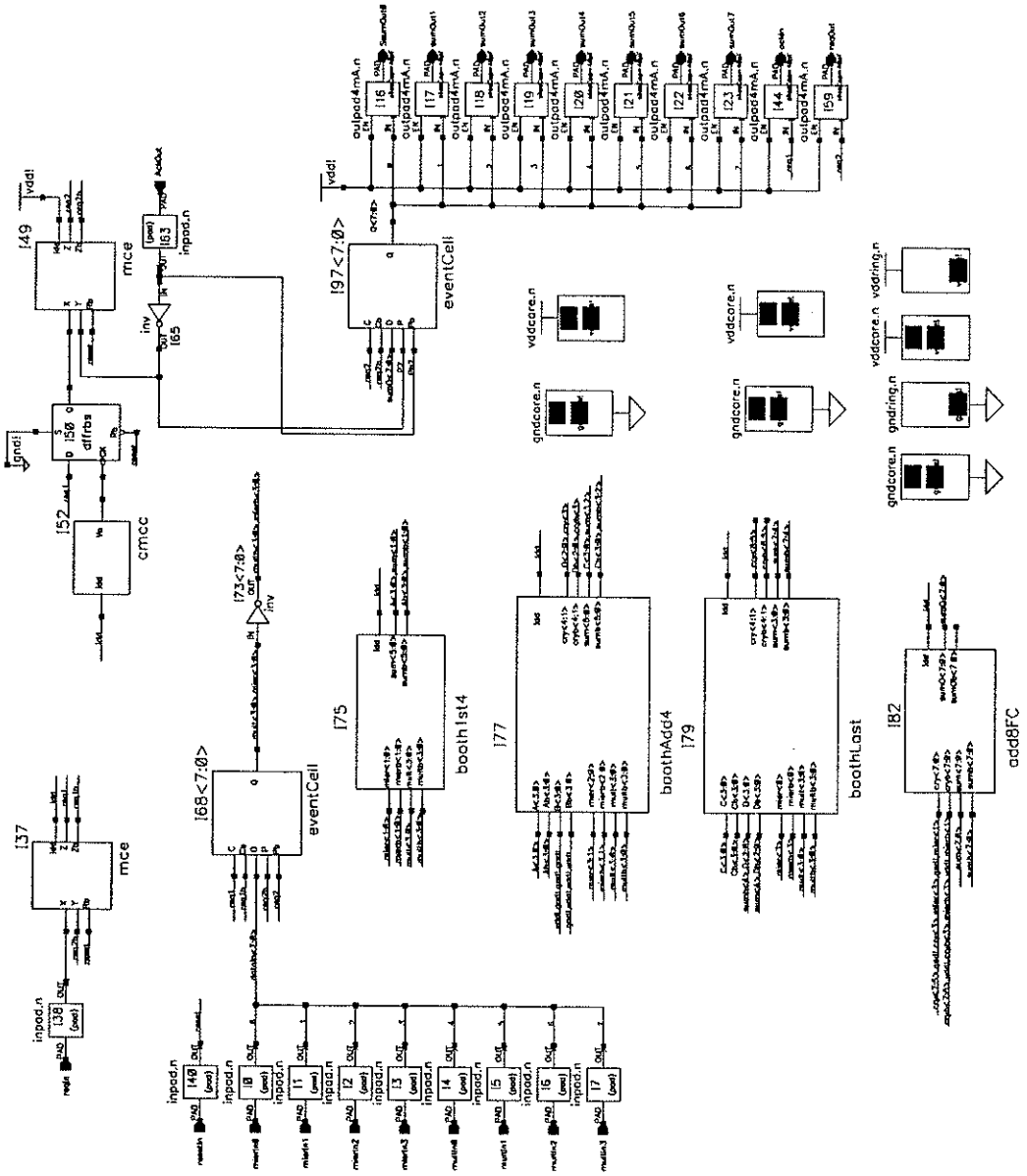


Figure B.1: Root Schematic Diagram of the Four-Bit Booth-Encoded Multiplier Circuit

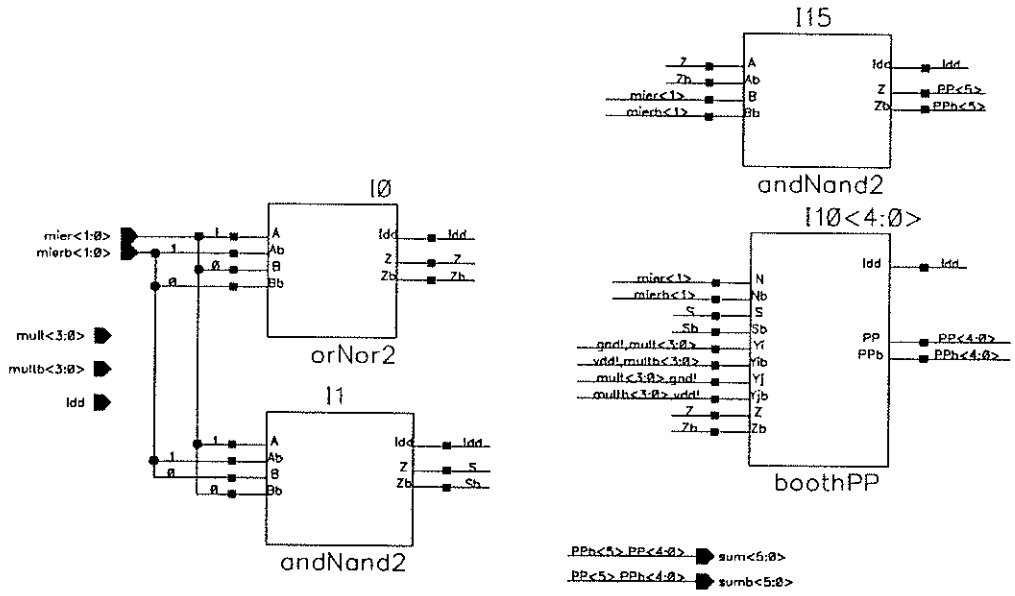


Figure B.2: Schematic Diagram of the First Stage in the Booth Multiplier (booth1st4)

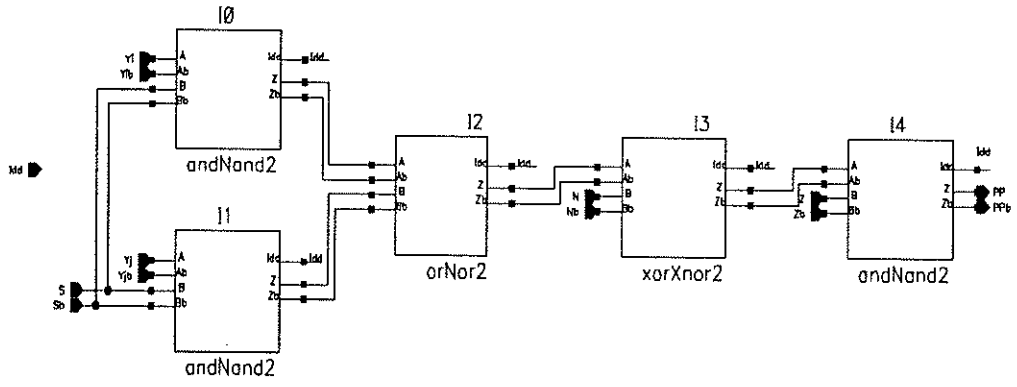


Figure B.3: Schematic Diagram of the Partial Product Generator Circuit (boothPP)

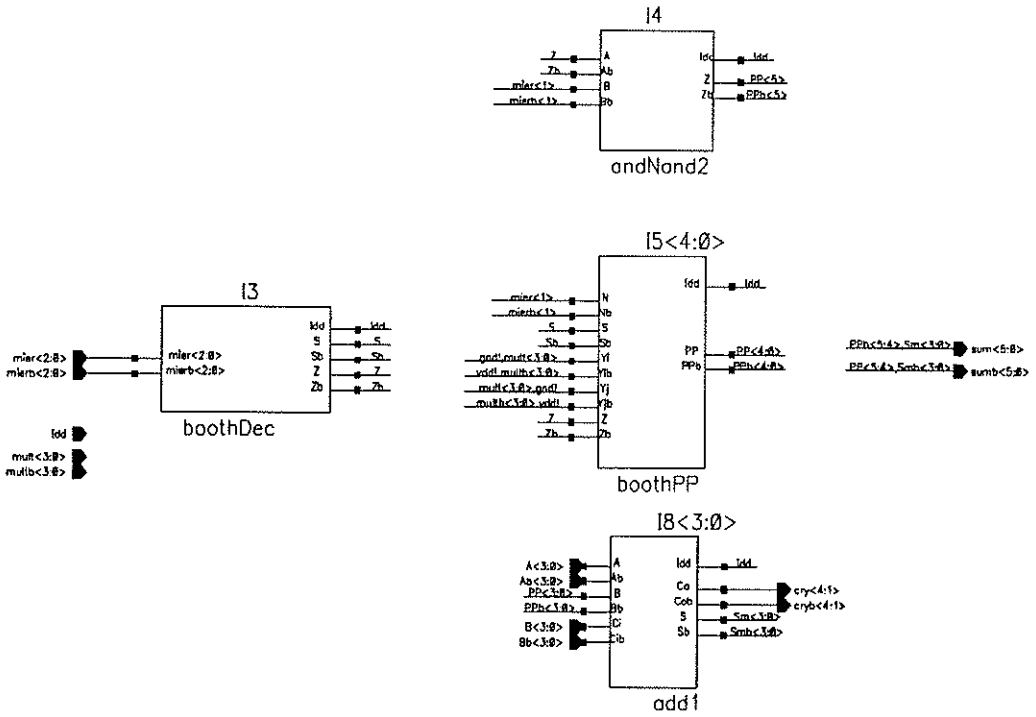


Figure B.4: Schematic Diagram of the Second Stage in the Booth Multiplier (boothAdd4)

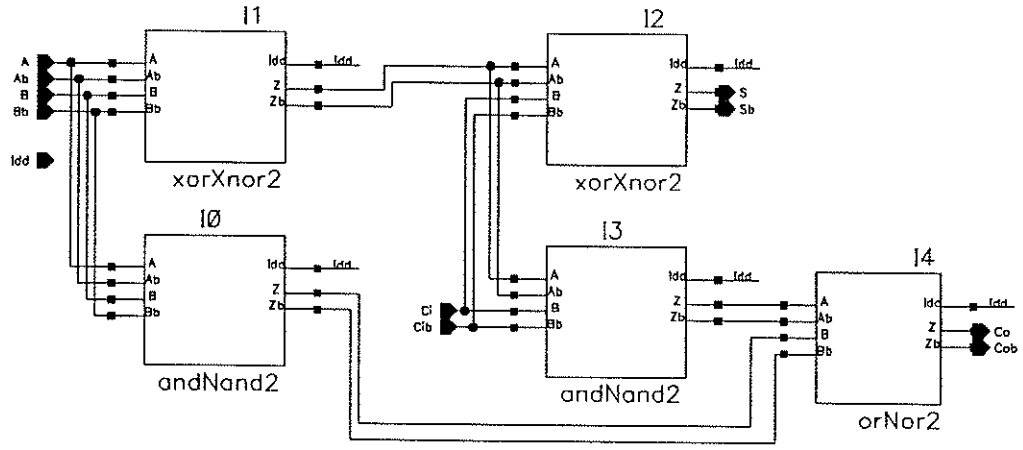


Figure B.5: Schematic Diagram of the One-Bit Adder Circuit (add1)

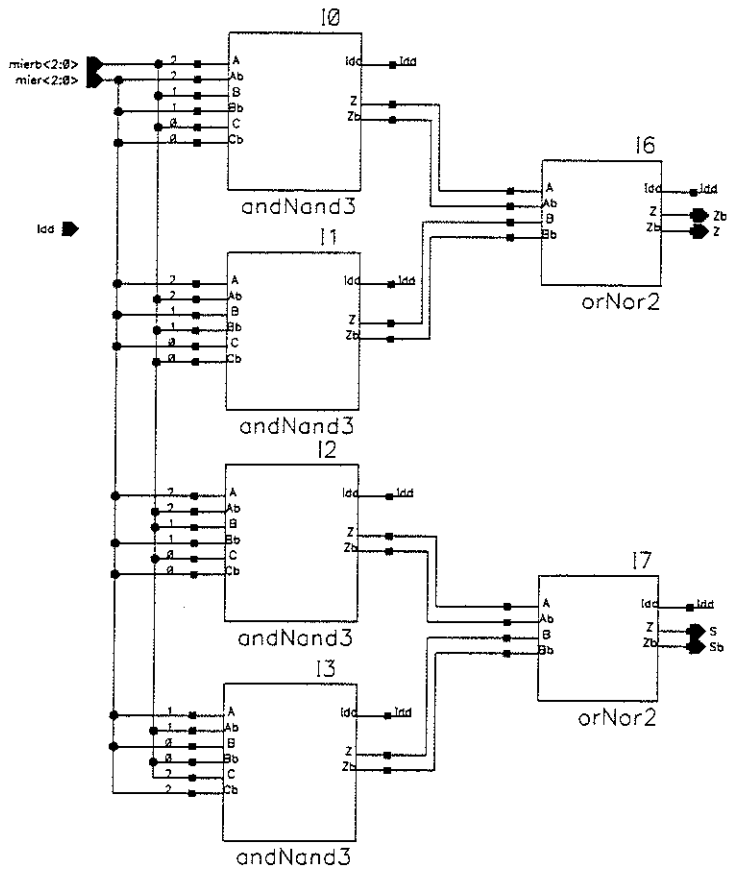


Figure B.6: Schematic Diagram of the Booth Decoder Circuit (boothDec)

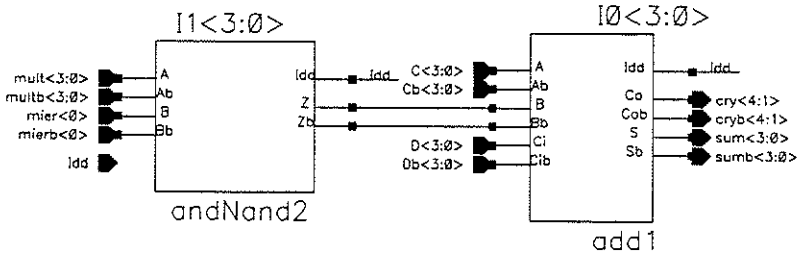


Figure B.7: Schematic Diagram of the Last Stage in the Booth Multiplier (boothLast)

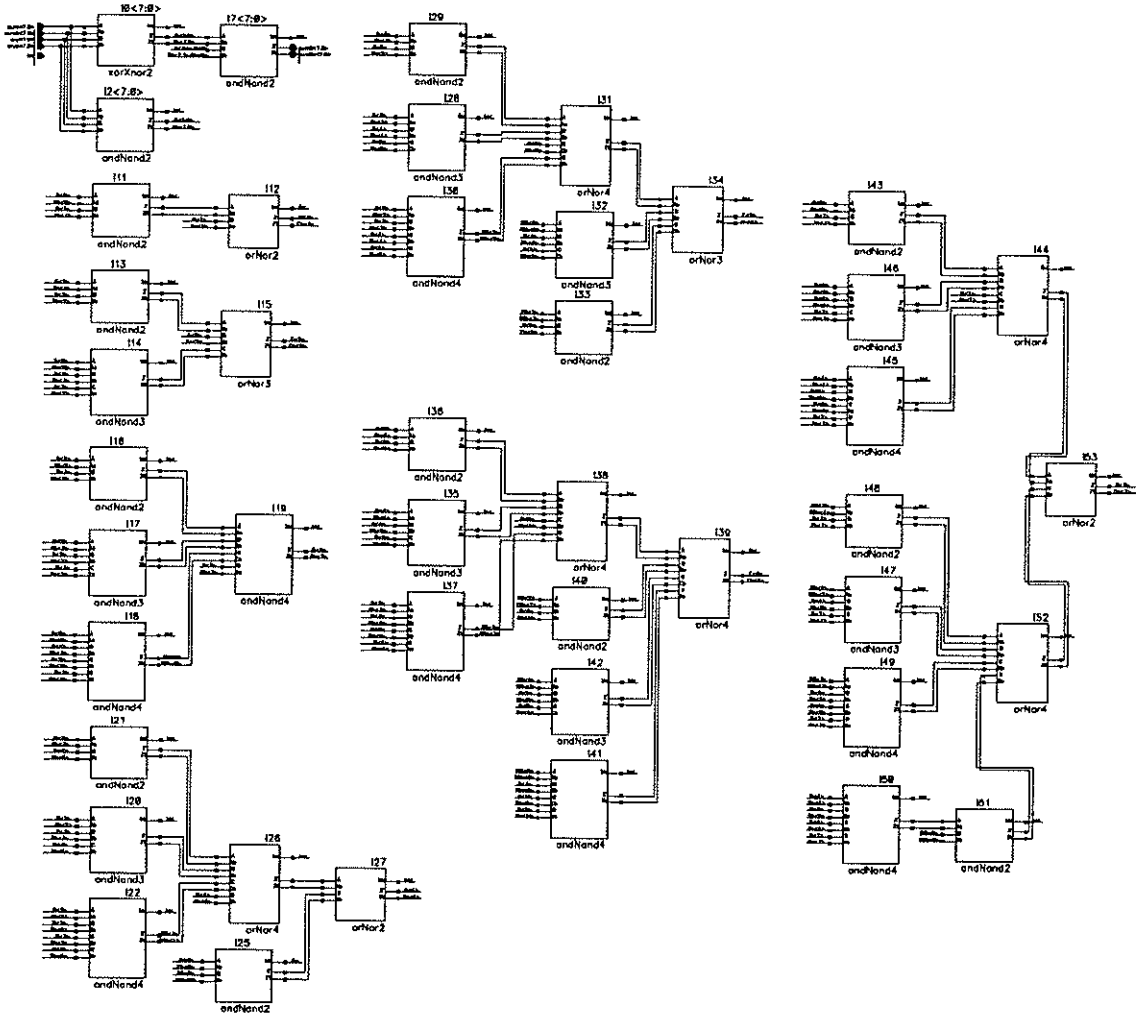


Figure B.8: Schematic Diagram of the Eight-Bit Fast Carry Adder (add8FC)