The McGraw-Hill Companies

Nanocomputing

The Future of Computing

The McGraw-Hill Companies

Nanocomputing

The Future of Computing

Vishal Sahni
Faculty of Engineering
Dayalbagh Educational Institute,
Dayalbagh, Agra

Debabrata Goswami
Femtosecond Laser Lab
Indian Institute of Technology
Kanpur

Tata McGraw-Hill Publishing Company Limited
NEW DELHI

McGraw-Hill Offices
New Delhi New York St Louis San Francisco Auckland Bogotá Caracas
Kuala Lumpur Lisbon London Madrid Mexico City Milan Montreal
San Juan Santiago Singapore Sydney Tokyo Toronto

The McGraw-Hill Companies

Tata McGraw-Hill

Published by Tata McGraw-Hill Publishing Company Limited,
7 West Patel Nagar, New Delhi 110 008.

ISBN (13): 978-0-07-024892-2
ISBN (10): 0-07-024892-3

Managing Director: *Ajay Shukla*
Head—Professional and Healthcare: *Roystan La´Porte*
Publishing Manager—Professional: *R Chandra Sekhar*
Junior Sponsoring Editor—Computing: *Ritesh Ranjan*
Production Executive: *Rita Sarkar*
Manager—Sales and Marketing: *S Girish*
Product Manager—Science, Technology and Computing: *Rekha Dhyani*
Controller—Production: *Rajender P Ghansela*
Asst General Manager—Production: *B L Dogra*

Typeset at The Composers, 260, C.A. Apt., Paschim Vihar, New Delhi 110 063 and printed at Gopsons Papers Ltd., A-2 & 3, Sector 64, Noida 201 301

Cover Design: Kapil Gupta

RXCYCRQXDDDZA

The McGraw-Hill Companies

To

the cherished memories of my Guru and Guide

Most Revered Dr. Makund Behari Lal Sahab

D.Sc. (Lucknow), D.Sc. (Edinburgh)

(1907-2002)

August Founder of Dayalbagh Educational Institute

in the Year of His Birth Centenary

—Vishal Sahni

my parents

Late K P Goswami and Mrs M Goswami

—Debabrata Goswami

Professor C.N.R. Rao, F.R.S.
Linus Pauling Research Professor and Honorary President,
Jawaharlal Nehru Centre for Advanced Scientific Research, Bangalore
Chairman, Nanoscience and Technology Initiative, Govt. of India
Chairman, Scientific Advisory Council to the Prime Minister of India

Foreword

Nobel laureate Richard Feynman had prognosticated in his talk on December 29, 1959 at the annual meeting of the American Physical Society at the California Institute of Technology (Caltech) entitled 'There's plenty of room at the bottom', "In the year 2000, when they look back at this age, they will wonder why it was not until the year 1960 that anybody began seriously to move in this direction.... when our computers get faster and faster and more and more elaborate, we will have to make them smaller and smaller. But there is plenty of room to make them smaller. There is nothing that I can see in the physical laws that says the computer elements cannot be made enormously smaller than they are now."

Nanoscience encompasses a wide array of research areas from quantum structures and nanocrystals to nanoparticles, nanowires, nanotubes, nanobiological systems, DNA as molecular nanowires, characterization and nanomanipulation. The range of fields in the realm of nanoscience are equally stunning—from chemistry to biology, chemical technology to electronics, healthcare, manufacturing, computing and space and military applications. This is just the tip of the iceberg—the potential of nanotechnology is being vigorously pursued.

Currently there is a worldwide effort to spur nanotechnology and it has been receiving increasing attention. The next ten years will see nanotechnology playing a dominant role in the global business environment, and is expected to go beyond the billion dollar estimates and cross the figure of one trillion US dollars.

Focused programs on nanotechnology have been launched by several nations like NSTI (Nanoscience and Technology Initiative) in India and NNI (National Nanotechnology Initiative) in the United States. It is estimated that two million workers will be needed to support nanotechnology industries worldwide within 15 years. We have to train students, teachers and research scholars. Unless we do this, there will not be enough work happening in this area in the near future.

India is emerging as a global leader in the nanotechnology field and is contributing to the development of new technologies besides carrying out basic research at the frontiers. Former President Dr. A.P.J. Abdul Kalam had envisioned in his address at the inauguration of the JN Tata Lecture on February 19, 2006, "Nanotechnology is knocking at our doors. . . molecular switches and circuits along with nano cells will pave the way to the next generation computers. . .With the emergence of nanotechnology, there is convergence of nano-bio-info technologies resulting in new devices which have wider applications in structure, electronics, healthcare and space systems. Potential applications are virtually endless. Progress in nanotechnology is spurred by collaboration among researchers in material science, mechanical engineering, computer science, molecular biology, physics, electrical engineering, chemistry, medicine and aerospace engineering. This is one of the important emerging areas which brings synergy in research and development by combining the strengths of the multiple domain knowledge leading to the creation of a knowledge society."

We have been observing tremendous speed in computing power—a quantum jump unmatched by progress in any other field. Moore's Law has been relentlessly shrinking the size of transistors, but this progress cannot go on forever and will knock at the doors of nanotechnology very soon. The quantum effects that come into play at this stage will be of tremendous potential for computing. Already quantum computers are being billed as the next generation computing and their power will be comparable to nothing that we know of now. It will crack all cyber security on RSA[1] in a

[1]RSA stands for initials of Rivest, Shamir, Adleman (3 scientists who had devised the cryptosystem), R. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, Vol. 21 (2), pp.120–126. 1978.

matter of seconds. Nanotechnology-based quantum computers would revolutionize computing and increase their computing power tremendously.

In spite of the large amount of research and development, there are not many books on this frontier area. It is extremely important to have entry-level textbooks which provide a smooth passage to our students from classrooms to nanotech laboratories. This book elucidates the many ideas and fields that go into nanocomputing. It has an easy-to-understand discussion on nanocomputing with its prospects and challenges, the underlying physics, reliability and most of all nanoscale quantum computing, optical and molecular computing. The book has a dedicated chapter on Quantum Cellular Automata Designer Software provides an opportunity to start working on nanocomputing with nothing more than a desktop PC. It is an ideal way of inculcating interest in the field and should give a fillip to nano-education in the country.

The authors have done a wonderful job of synergizing the various insights on nanocomputing in this text which should be a valuable asset for nano-students, scientists and researchers the world over. I wish the book success.

C.N.R. RAO

Preface

Nanocomputing is defined as, "*The use of the distinct properties of structures on the scale of 100 nanometers or smaller to solve complex mathematical or logical problems, or to accomplish any of the tasks that we expect of modern computers.*"

Many breakthroughs have been achieved in different nanotechnological fields. To start with, general desktop computing has made the transition into nanotechnology with the advent of nanotransistors. Also, carbon nanotubes and silicon nanowires are now being tested for use in computers to speed up connections between nanotransistors. There are also some very interesting theoretical nanocomputing possibilities, the most important of which is quantum computing, which uses the quantum properties of single atoms to process information in ways that are very different from the way current computers work.

Presently efforts towards developing electronic devices at a nanoscale are primarily driven by the limitation of the existing photolithography used in fabricating devices on silicon, and also from the belief that for Moore's Law to hold good, one has to seek answers at the molecular level. Diffraction in photolithographic processes limits the feature size achievable on silicon to about half the wavelength of light. Another problem in scaling down the size of silicon transistors is that at about 50 nm it would be impossible to dope silicon uniformly, and the channels would have so few electrons that the transistor may cease to function reliably. Serious research efforts to seek an alternative to bulk silicon had begun way back in the mid-1980s, but in this decade it has caught the attention of a large number of researchers from physics, chemistry, life sciences, and engineering. A variety of approaches are being tried out to build nano-tubes, nano-wires, logic gates, and memory using an assortment of materials that not only include traditional semi-conductor materials and gold, but also organic molecules and even DNA.

The recent developments in nanotechnology promise immense avenue for exploitation in almost all fields of human interaction.

Keeping in pace with the global nanotechnology competition, Department of Science and Technology, Government of India, assessed the importance of this emerging, highly interdisciplinary field and launched a national program titled 'Nano Science and Technology Initiative (NSTI)' in the tenth Five Year Plan. The program focuses on overall research and development in nanoscience and technology to enable India to become a significant player in the area and contribute to the development of new technologies besides carrying out basic research at the frontiers.

This book provides a holistic view of nanocomputing with sprinkling of optical and molecular computing—two allied and important fields of computing at the nanoscale.

Chapter 1 introduces the subject and discusses prospects and challenges of nanocomputing. Chapter 2 deals with the physics of nanocomputing and Chapter 3 describes nanocomputing in the presence of imperfections. Chapter 4 discusses reliability of nanocomputing and Chapter 5 presents nanoscale quantum computing. The QCADesigner Software is presented in Chapter 6, and Chapter 7 deals with molecular and optical computing and ultrafast pulse shaping.

The CD-ROM with the book serves as a teaching aid which will help students and researchers start off with nanocomputing right away on a PC. It has several learning tools including NANOLAB, a MATLAB based reliability evaluation tool; NANOPRISM, a probabilistic model checking based tool; and QCADesigner, a design and simulation tool for Quantum Dot Cellular Automata (QCA). The utility of these tools along with several examples has been presented all through the book.

Vishal Sahni
Debabrata Goswami

Acknowledgements

An enormous number of people have contributed to bring this book in the present form and it is not possible to list all of them here. Their endless support and encouragement has gone a long way in fructifying this endeavor.

We are extremely grateful to Most Revered Prof. P.S. Satsangi Sahab, Chairman, Advisory Committee on Education, Dayalbagh, Agra, who advised us to take up the subject of 'Nanocomputing'. We express our gratitude for his paternal guidance, incessant encouragement and unbounded grace in all spheres of life which have crowned our humble efforts with success.

We are thankful to Dr Lov K. Grover, distinguished member of technical staff at Bell Laboratories, New Jersey, USA, and founder of the famous Grover's Algorithm, for being so generous all throughout the venture. A visit to his laboratory in summer 2007 unlocked so many things. We are grateful to our colleagues at DEI and IIT Kanpur in the Quantum-Nano Research group including Prof. Ashutosh Sharma, Dr Laxmidhar Behera, Dr Ravi Shankar, and Dr V. Subramaniam who have helped in so many ways. Prof. P.K. Kalra at IIT Kanpur has been a pillar of support all along the journey.

We are thankful to many more people, very active in the nanocomputing field, in India and abroad, who have seen drafts of this book in development and whose valuable suggestions, comments, advice and support have brought the book to its present stage. It is not possible to list all of them here but notable are Prof. Peter Shor, Prof. Edward Farhi and Dr Scott Aaronson, Massachusetts Institute of Technology, USA; Prof. Raymond Laflamme, Prof. Peter H. Roe, Prof. Richard Cleve, Dr. Drew Knight, Dr Ashwin Nayak and Dr Jonathan Walgate, University of Waterloo, Canada; Dr Jean Christian Boileau, University of Toronto, Canada; Prof. Prabhat Hajela, Rensselaer Polytechnic Institute, USA; Prof. Anil Kumar, NMR Research Centre, Prof. Ajay Sood, Department of Physics and Prof. Apoorva Patel, Centre for Theoretical Studies at India n Institute of Science, Bangalore; Prof. Jaikumar Radhakrishnan

and Dr Achanta Venugopal at Tata Institute of Fundamental Research, Mumbai; Prof. Arvind, Indian Institute of Technology, Chennai; Prof. Sudeb P. Pal, Indian Institute of Technology, Kharagpur; Prof. V. Ramgopal Rao, Indian Institute of Technology, Bombay; Prof. Prem K. Kalra, Prof. Huzur Saran, Prof. M. Jagadesh Kumar, Dr Ashok Ganguli and Dr Amit Kumar, Indian Institute of Technology, Delhi; Prof. K.R. Parthasarthy, Indian Statistical Institute, Delhi; Prof. R. Simon, Institute of Mathematical Sciences, Chennai; Dr. Arun Kumar Pati, Institute of Physics, Bhubaneshwar and Dr R. Srikanth at Raman Research Institute, Bangalore. Their contribution in bringing the book to this present form cannot be adequately expressed in words.

We are extremely grateful to Prof. C.N.R. Rao, National Research Professor and Linus Pauling Research Professor at Jawaharlal Nehru Centre for Advanced Scientific Research, Bangalore, for being so kind to us all through the endeavor and his magnanimity to write a foreword for our book. He is the founder of one of the most profound and influential nanotechnology initiatives to date and a world renowned authority on the subject. As Chairman of NSTI, Government of India and Chairman, Scientific Advisory Council to the Prime Minister (SAC-PM), he is the spearhead of the nanotechnology and scientific programs of the country.

We are indebted to several scientists and institutions for being so generous in granting permissions to freely use their material as part of the book. In particular, we are thankful to Dr Doug Smith, Editor of Caltech *Engineering and Science* magazine for permission to reproduce Richard Feynman's historic speech which launched the subject. We are grateful to Dr. Simon Benjamin and Dr. Artur Ekert, Oxford for permission to include portions of their introductory article on nanocomputing, Dr. Sandeep Shukla, Virginia Tech. for NANOLAB and NANOPRISM tools, Dr. Seth Copen Goldstein, Carnegie Mellon for reliability strategies and Dr. Konrad Walus, University of British Columbia for QCADesigner software.

We are thankful to Department of Science and Technology, Government of India, and Indo-US Science and Technology Forum for the generous funding of our quantum and nano computing endeavors.

We are thankful to the editorial and the production teams at McGraw-Hill Education, India, for their efforts. They have done a great job.

We are grateful to our colleagues and friends at Dayalbagh Educational Institute and IIT Kanpur. Prof. Sanjay G. Dhande, Director IIT Kanpur has been a guiding beacon all along our academic pursuits. Prof. V.G. Das, Director DEI, Prof. S.S. Bhojwani, Coordinator, Distance Education Programme, Prof. Satish Kumar, Coordinator, Multimedia Labs, Prof. V. Prem Pyara, Prof. P.K. Saxena and Dr Rahul Swarup Sharma, Faculty of Engineering, all at DEI, Agra, have helped us in so many ways.

We owe a lot to our dear parents, Dr Sudhir Sahni and Dr (Mrs.) Renu Sahni, and Late Mr K.P. Goswami and Mrs M. Goswami whose academic ethos have cast every nano-portion of ours in the quantum mould. We are extremely grateful to our wives, Pushpa Sahni and Sonaly Goswami for their constant encouragement and affection. The arrival of a nanocomputer, Mukti, in our midst has been really lucky for all of us and we are sure she will see the age of quantum teleportation and nanocomputers all around. We are extremely grateful to Mr Asad Pathan, Mrs. Shobha Pathan, Mr S.P. Gupta and Late Mrs S.P. Gupta for all their inspiration and encouragement. We thank our young friends Prakash, Pooja, Arsh, Shabd, Rohit and Amrita for their rock-solid support.

We are home-grown products of Dayalbagh Educational Institute and IIT Kanpur. We feel proud to be a part of these great centres of learning which provide a unique atmosphere for academic pursuits and bring about physical, intellectual, emotional and ethical integration of an individual with a spirit of truthfulness, temperance and courage, and cultivate a spirit of humility, simple living, selfless service and sacrifice, while imparting education of excellence as well of relevance to contemporary needs with a scientific temper.

VISHAL SAHNI
DEBABRATA GOSWAMI

Contents

List of Abbreviations

AOM	Acousto-Optic Modulator
APD	Avalanche Photo Diode
ASIC	Application Specific Integrated Circuit
BJT	Bipolar Junction Transistor
BTBT	Band to Band Tunneling
C^2NOT	Controlled-Controlled NOT
CAD	Computer Aided Design
CAEN	Chemically Assembled Electronic Nanotechnology
CMOS	Complementary Metal Oxide Semi-conductor
CNN	Cellular Nonlinear Network
CNOT	Controlled NOT
CNT	Carbon Nano Tube
CPU	Central Processing Unit
CTMR	Cascaded Triple Modular Redundancy
CVD	Chemical Vapor Decomposition
DAPER	Defect Aware Place and Route
DIBL	Domain Induced Barrier Lowering
DNA	Deoxyribonucleic Acid
DTMC	Discrete Time Markov Chain
DTR	Data Transfer Ratio
DUPER	Defect Unaware Place and Route
DWDM	Dense Wavelength Division Multiplexing
EO	Electro-Optical
FET	Field Effect Transistor
FPGA	Field Programmable Gate Array
FWHM	Full Width at Half Maximum
HDL	Hardware Description Language
I/O	Input/Output
ITRS	International Technology Roadmap for Semi-conductors
LCM	Liquid Crystal Modulator
LLN	Law of Large Numbers
LUT	Look Up Table
MOSFET	Metal Oxide Semi-conductor Field Effect Transistor
MRF	Markov Random Fields

NIST	National Institute of Standards & Technology
NLO	Non-linear Optical
NMR	Nuclear Magnetic Resonance
NMR	N-Modular Redundancy
P & R	Place and Route
PRISM	Probabilistic Model Checker
QCA	Quantum-dot Cellular Automata
RF	Radio Frequency
RH	Reconfigurable Hardware
ROM	Read Only Memory
RZ	Return to Zero
SEC	Single Error Correction
SET	Single Electron Transistor
SLM	Spatial Light Modulators
SQUID	Superconducting Quantum Interference Devices
SRAM	Static Random Access Memory
STM	Scanning Tunneling Microscope
TDM	Time Division Multiplexing
TMR	Triple Modular Redundancy
VCSEL	Vertical Cavity Surface Emitting Laser
VLRF	Very Large Reconfigurable Fabric
VLSI	Very Large Scale Integration
WDM	Wavelength Division Multiplexing

CHAPTER 1

Nanocomputing–Prospects and Challenges

1.1 INTRODUCTION

Let us begin our nanocomputing journey with the classic speech the great physicist, Richard Feynman gave on December 29, 1959, at the annual meeting of the American Physical Society at California Institute of Technology (Caltech).

> "I would like to describe a field, in which little has been done, but in which an enormous amount can be done in principle. This field is not quite the same as the others in that it will not tell us much of fundamental physics (in the sense of, "What are the strange particles?") but it is more like solid-state physics in the sense that it might tell us much of great interest about the strange phenomena that occur in complex situations. Furthermore, a point that is most important is that it would have an enormous number of technical applications.
>
> What I want to talk about is the problem of manipulating and controlling things on a small scale. As soon as I mention this, people tell me about miniaturization, and how far it has progressed today. They tell me about electric motors that are the size of the nail on your small finger. And there is a device on the market, they tell me, by which you can write the Lord's Prayer on the head of a pin. But that's nothing; that's the most primitive, halting step in the direction I intend to discuss. It is a staggeringly small world that is below. In the year 2000, when they look back at this age, they will wonder why it was not until the year 1960 that anybody began seriously to move in this direction.

Miniaturizing the computer

I don't know how to do this on a small scale in a practical way, but I do know that computing machines are very large; they fill rooms. Why can't we make them very small, make them of little wires, little elements—and by little, I mean *little*. For instance, the wires should be 10 or 100 atoms in diameter, and the circuits should be a few thousand angstroms across. Everybody who has analyzed the logical theory of computers has come to the conclusion that the possibilities of computers are very interesting—if they could be made to be more complicated by several orders of magnitude. If they had millions of times as many elements, they could make judgments. They would have time to calculate what is the best way to make the calculation that they are about to make. They could select the method of analysis which, from their experience, is better than the one that we would give to them. And in many other ways, they would have new qualitative features.

If I look at your face I immediately recognize that I have seen it before. (Actually, my friends will say I have chosen an unfortunate example here for the subject of this illustration. At least I recognize that it is a *man* and not an *apple*.) Yet there is no machine which, with that speed, can take a picture of a face and say even that it is a man; and much less that it is the same man that you showed it before—unless it is exactly the same picture. If the face is changed; if I am closer to the face; if I am further from the face; if the light changes—I recognize it anyway. Now, this little computer I carry in my head is easily able to do that. The computers that we build are not able to do that. The number of elements in this bone box of mine is enormously greater than the number of elements in our "wonderful" computers. But our mechanical computers are too big; the elements in this box are microscopic. I want to make some that are submicroscopic.

If we wanted to make a computer that had all these marvelous extra qualitative abilities, we would have to make it, perhaps, the size of the Pentagon. This has several disadvantages. First, it requires too much material; there may not be enough germanium in the world for all the transistors which would have to be put into this enormous thing. There is also the problem of heat generation and power consumption; TVA would be needed to run the computer. But an even more practical difficulty is that the computer would be limited to a certain speed. Because of its large size, there is finite time required to get the information from one place to another. The information cannot go any faster than the speed of light—so, ultimately, when our computers get faster and faster and more and more elaborate, we will have to make them smaller and smaller.

But there is plenty of room to make them smaller. There is nothing that I can see in the physical laws that says the computer elements cannot be made enormously smaller than they are now. In fact, there may be certain advantages."

1.2 History of Computing

Early computers started from Jacquard's loom in 1746 which used punched cards to control weaving (Fig. 1.1a). Charles Babbage's analytical engine (Fig. 1.1b) developed in 1834 was a mechanical computer with punched-card data input. Herman Hollerith developed an electronic tabulator/sorter (Fig. 1.1c) which collated the US 1890 census data.

(a) Jacquard's loom (b) Babbage's analytical engine (c) Hollerith's tabulator

Fig. 1.1: Early Computers

In the twentieth century, the idea of Turing machine was proposed by Alan Turing (Fig. 1.2) which used a two-way tape for data input and storage and finite-state machine for reading/writing on tape. Computers in the twentieth century are based on the von Neumann concept of stored programs and the fetch-execute cycle.

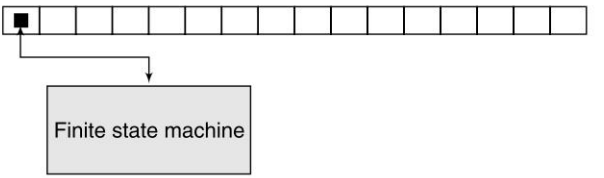


Fig. 1.2: Turing Machine

Early examples of twentieth century's programmable computers include Atanasoff (1940) which was a tube-based linear equation solver, Zuse's Z3 (1941) which was relay-based, Colossus (1943) which was tube-based and broke the German Enigma code in World War II, Mark I (1944) which was a general-purpose, relay-based computer and ENIAC (Electronic Numerical Integrator and Computer)—the first general purpose electronic computer built at University of Pennsylvania in 1946, which was a tube-based computer.

The computer revolution truly began with the invention of the transistor (Fig. 1.3) at Bell Labs in 1947 by Shockley. As a semiconductor switch, it replaced the vacuum tube. By 1958, IBM began selling the 7070, a transistor-based computer.

Fig. 1.3: Shockley's Transistor at Bell Labs, 1947

The IC (integrated circuit) was invented independently in 1959 by Jack Kilby and Robert Noyce wherein transistors and wires were combined on a chip through photolithography. In the words of Jack Kilby:

> "What we didn't realize then was that the integrated circuit would reduce the cost of electronic functions by a factor of a million to one, nothing had ever done that for anything before".[1]

After that came the VLSI (very large scale integration) revolution. In 1969, the Intel 4004 CPU was placed on a chip. By the late 1970s, very complicated chips were being assembled. New challenges like specifying large chip designs simply, simulating the electronics, laying out chips, designing area efficient algorithms and understanding tradeoffs through analysis were encountered.

[1]http://inventors.about.com/library/weekly/aa080498.htm

VLSI emerged as an academic area in the late 1970s with the publication of a book 'Introduction to VLSI' by Mead and Conway in 1980. HDLs (hardware design languages) were developed for specifying large chip designs, electronic simulators such as Spice were developed for simulating the electronics, computer-aided designing emerged for laying out chips, and area-efficient algorithms and theory for VLSI layouts and AT^2 lower bounds were developed. Intel's first microprocessor, the 4004, ran at 108 kilohertz (108,000 hertz), compared to the Intel® Pentium® 4 processor's initial speed of 1.5 gigahertz (1.5 billion hertz). If automobile speed had increased similarly over the same period, we could now drive from San Francisco to New York in about 13 seconds.

In the VLSI model, wires have width and gates occupy area. The feature size of VLSI technology is the size of the smallest feature (wire width/separation). The area of gates is comparable to the square of feature size. The area occupied by wires often dominates the area of gates. With a dead end in sight for Moore's Law (discussed in the next section), devices are so small that electronic models are no longer accurate, and expensive redesign is needed to meet system requirements.

1.3 Nanocomputing

Nanocomputing is defined as, "*The use of the distinct properties of structures on the scale of 100 nanometers or smaller to solve complex mathematical or logical problems, or to accomplish any of the tasks that we expect of modern computers.*"

Many jumps have been made in different nanotechnological fields. To begin with, general desktop computing has made the transition into nanotechnology, with the advent of nanotransistors. Also, carbon nanotubes and silicon nanowires are now being tested for use in computers to speed up connections between nanotransistors. There are also some very interesting theoretical nanocomputing possibilities, the most important of which is quantum computing, which uses the quantum properties of single atoms to process information in ways that are very different from the way current computers work.

The key thing to be kept in mind is that 'bulk materials' have been used up to this point to create computers. Bulk materials are

those materials that are of large size so that they behave predictably. Experiments with bulk materials normally yield empirical data that support the principles of physics and electrical engineering. As we will see in the following chapters, nanomaterials do not always do this.

The technology that has been used so far has been CMOS (complementary metal oxide semiconductor technology) Technology. This involves the use of semi-conductors, which form gates and transistors. Semi-conductors are substances that behave sometimes like a conductor and sometimes like an insulator. When semi-conductors are 'doped' (mixed) with impurities, their electrical conductivity increases. Semi-conductors until now were bulk-sized materials and as a result behaved predictably.

A doped semi-conductor is one that has excess electrons. When an external charge is applied to the n-type material (material with excess electrons), the electrons are attracted to the holes in the p-type material (material with lesser electrons than protons). This generates an electrical current through the junction in between the 2 semi-conductors. When an external charge is applied to the p-type material, the electrons fill in the holes and there is no current generated, hence it is named as semi-conductor. A wafer of semiconducting material is shown in Fig. 1.4.

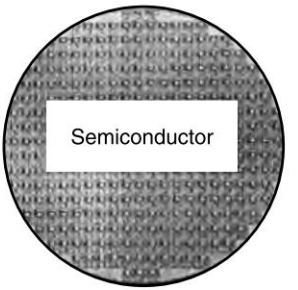


Fig. 1.4: Wafer of Semiconducting Material

A gate is the structure that takes many different inputs and produces one output. In today's computers, an n-type semiconductor is used along with a p-type semi-conductor to form each gate. Gates can be AND, OR, XOR, NOT or a combination of these like NAND, NOR etc. This is where bits (0's and 1's) come into play. A 0 (off position) and 1 (on position) are the directions that pass through these gates. Depending on the condition (AND, OR, etc.)

the combination of 0's and 1's will yield different results. These different results are what make a computer capable of calculating numbers, displaying pictures, playing chess, projecting digital sound and doing an endless number of other operations.

While computers are extremely useful today, they are limited in complexity because a switch could only be in either of the 2 states—On or Off. As discussed later in this chapter, there can be more than 2 states if quantum computing is used, which is important since the increase in computer technology will be exponential with those additional states.

Transistors regulate the electronic signals that are responsible for operations in a computer. They could be composed of a p-type layer sandwiched between 2 n-type layers or an n-type layer sandwiched between 2 p-type layers. When the central semi-conductor has a small change in voltage or current, the whole transistor receives a large change in current. It acts like a switch and can open and close gates many times (in today's case, billions) per second.

1.3.1 Transistors inside the Machine

Integrated circuits, or 'chips', are made up of many different transistors linked with circuits. They are put on silicon microchips connected through silicon wires. Fig. 1.5 shows an integrated circuit mounted on a silicon wafer, followed by a semi-conductor.

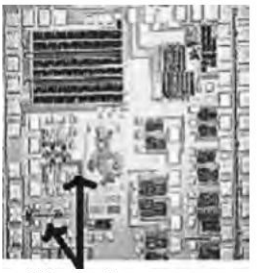


Fig. 1.5: Integrated Circuit

Microprocessors are brain of the computer with all the transistors put together. They can move data from one location to another, perform mathematical operations and are the basis of personal computers today. The rule that the evolution of transistors

follows is called Moore's Law (Fig. 1.6), which is the prediction that the number of transistors per chip will grow exponentially, doubling every couple of years. As the number of transistors on a chip increases, a computer will be able to perform more and more operations per second. Moore's Law has been consistent, but is thought to end around 2016 due to problems with size and heat.

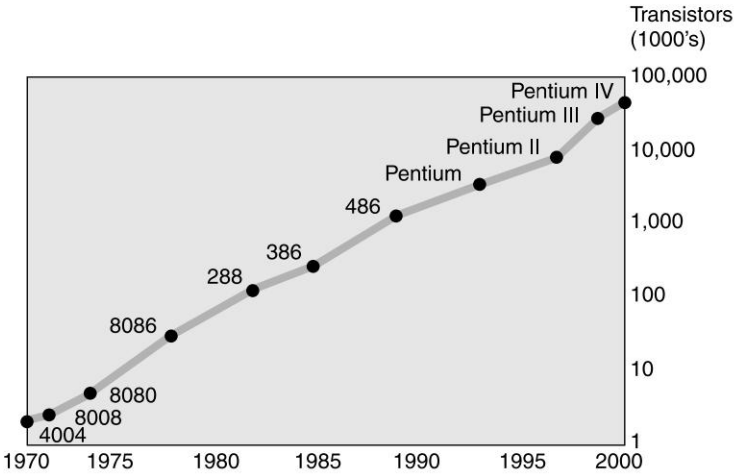


Fig. 1.6: Moore's Law

Heat dissipation and electronic charge are obstacles in making today's computers faster. Heat is generated by electrons moving through the semiconductors. Currently, this is not a problem, as computer fans maintain the temperature. However, as transistors get to the molecular scale and are placed very close together, the heat will be great enough to vaporize the silicon chip.

One might wonder why we even need to improve if we already have adequate ones: why even bother going through the trouble of trying to make computers smaller if there will be problems doing so. The reason technologists and scientists even bother with nanocomputing is because of the potential it holds for the future. The same amount of information that fits on a current 80 gigabyte hard drive will be able to fit in a space that is too small to see with the naked eye. This will revolutionize the computer industry and will take devices such as cell phones and palm pilots to a new level. Number of operations per second could increase by a factor of at least 10^6. Complex math and molecular chemistry problems that have not been solved yet will be able to be solved.

We have officially entered the nanotechnology realm with the introduction of Intel's 70 nanometer gate length in 2000. This was improved to a 50 nanometer gate length in 2002. However, we will not be able to advance further without making some drastic changes. It will probably require using something else other than today's CMOS technology. Instead of using electric charge to operate and store memory we must search for alternate ways for this. Properties of particles, such as particle spin and photon field, have been examined as other sources of building very small and powerful computers. Nanotechnology introduces many new structures that behave in ways that few understand. These structures that are only a few atoms across, will allow the computer revolution to extend Moore's Law beyond 2016.

Over the last few decades computer power has grown at an amazing rate, doubling every couple of years. This increase is essentially due to the continual miniaturization of the computer's most elementary component—the transistor. As transistors became smaller, more could be integrated into a single microchip, and so the computational power increased. However, this miniaturization process is now reaching a limit—a quantum threshold below which transistors will cease to function. Present 'state-of-the-art' components possess features only a few hundreds of nanometers across (a nanometer is a thousandth of a micron, or a billionth of a meter). If these chips were to be miniaturized further to the scale of tens of nanometers, then their operation would be disrupted by the emergence of quantum phenomena, such as electrons tunneling through the barriers between wires. In order for the science of computation to progress further, an alternative to transistor technology must be found. The components of the new technology will have to function based on quantum effects rather than despite them.

As shown in Fig. 1.7, there are ways to redesign transistors to work using quantum effects. The structure on the right is a single-electron transistor (SET) which was carved by the tip of a scanning tunneling microscope (STM). According to classical physics, there is no way that electrons can get from the 'source' to the 'drain', because of the two barrier walls on either side of the 'island'. But the structure is so small that quantum effects occur, and electrons can, under certain circumstances, tunnel through the barriers (but only one electron at a time can do this!). Thus the SET wouldn't work without quantum mechanics.

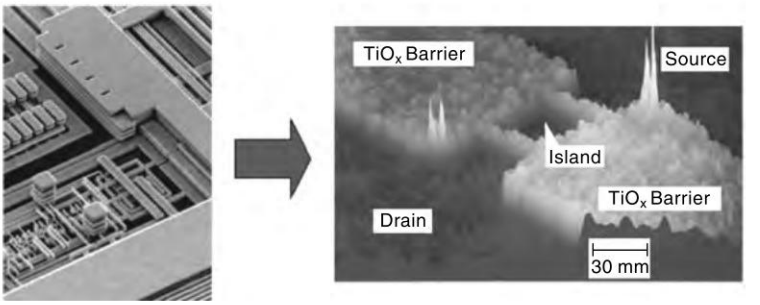


Fig. 1.7: The Transition from Microtechnology to Nanotechnology

But it might be better to give up the idea of transistors altogether, and use a completely new architecture that is more suitable for the nanometer scale as shown in Fig. 1.8. As an alternative to using new kinds of transistors, nanocomputers might have an entirely new type of architecture made up of many simple units called 'cells'. This type of architecture is suitable for the nanometre scale, where simple units form naturally. One way to make the cells would be using structures called quantum-dots, which are also known as 'artificial atoms'.

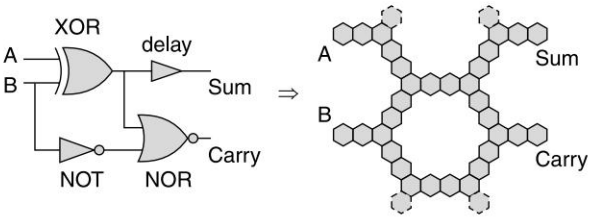


Fig. 1.8: Half-adder using Cells

The first generation of nanocomputers will have components that behave according to quantum mechanics, but the algorithms that they run will probably not involve quantum mechanics. We might call such computers 'nanometer-scale classical computers' (here the word 'classical' means 'not quantum'). But scientists have recently realized that there is another, more exciting possibility—quantum mechanics might be used in an entirely new kind of algorithm that would be fundamentally more powerful than any classical scheme. A computer that could run such an algorithm would be a true 'quantum computer'.

Although both the structures in Fig. 1.9 use quantum mechanics, only the one on the right could be employed in a true 'quantum computer'. The 1H and 13C nuclei in isotopically labeled chloroform behave like small magnets, and interact with an external magnetic field. Nuclear spins can store and process information in the so-called quantum superpositions.

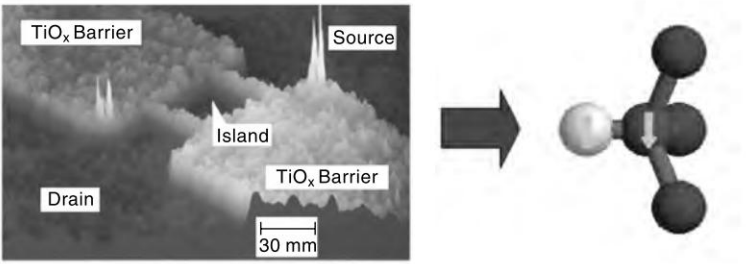


Fig. 1.9: From an SET (on the left) to the Ultimate Computer Element—A Molecule

1.4 Quantum Computers

To explain what makes quantum computers so different from their classical counterparts we begin by having a closer look at a basic chunk of information, namely one bit. From a physical point of view, a bit is a physical system which can be prepared in one of the two different states representing two logical values—no or yes, false or true, or simply 0 or 1. For example, in today's digital computers, the voltage between the plates in a capacitor represents a bit of information: a charged capacitor denotes bit value 1 and an uncharged capacitor bit value 0. One bit of information can also be encoded by using 2 different polarizations of light or two different electronic states of an atom. However, if we choose an atom as a physical bit then quantum mechanics tells us that apart from the two distinct electronic states the atom can also be prepared in a coherent superposition of the two states. This means that the atom is both. in state 0 and state 1. There is no equivalent of this superposition in the classical world; it is a purely quantum mechanical phenomenon. Since we are used to seeing classical physics at work in the every day world, such quantum phenomena often seem counter-intuitive.

Now we push the idea of superposition of numbers a bit further. Consider a register composed of three physical bits. Any classical register of that type can store in a given moment of time only 1 out of 8 different numbers, i.e., the register can be in only 1 out of 8 possible configurations, such as 000, 001, 010, ... 111. A quantum register composed of 3 qubits can store in a given moment of time all 8 numbers in a quantum superposition.

This is quite remarkable that all 8 numbers are physically present in the register but it should be no more surprising than a qubit being both in state 0 and 1 at the same time. If we keep adding qubits to the register we increase its storage capacity exponentially, i.e., 3 qubits can store 8 different numbers at once, 4 qubits can store 16 different numbers at once, and so on; in general, L qubits can store 2^L numbers at once (here 2^L means 2 to the power of L). Once the register is prepared in a superposition of different numbers we can perform operations on all of them. For example, if qubits are atoms then suitably tuned laser pulses affect atomic electronic states and evolve initial superpositions of encoded numbers into different superpositions. During such an evolution each number in the superposition is affected, and as the result we generate a massive parallel computation, albeit in one piece of quantum hardware. This means that a quantum computer can in only one computational step perform the same mathematical operation on 2^L different input numbers encoded in coherent superpositions of L qubits. In order to accomplish the same task, any classical computer has to repeat the same computation 2^L times, or one has to use 2^L different processors working in parallel. In other words, a quantum computer offers an enormous gain in the use of computational resources such as time and memory.

1.5 Nanocomputing Technologies

The current effort in developing electronic devices at a nano-scale is driven primarily by the limitation of the existing photolithography used in fabricating devices on silicon, and also from the belief that for Moore's Law to hold good, one has to seek answers at the molecular level. It is well known that due to diffraction in photolithographic processes, the feature size achievable on silicon would be limited to about one-half the wavelength of light. Another

problem in scaling down the size of silicon transistors is that at about 50 nm it would be impossible to dope silicon uniformly, and the channels would have so few electrons that the transistor may cease to function reliably. Serious research efforts to seek an alternative to bulk silicon had begun way back in the mid-1980s, but in this decade it has caught the attention of a large number of researchers from physics, chemistry, life-sciences, and engineering disciplines. A variety of approaches are being tried out to build nano-tubes, nano-wires, logic gates, and memory by using an assortment of materials that not only include traditional semi-conductor materials and gold, but also organic molecules and even DNA. The seriousness of the effort being made in molecular electronics and the promise of a possible success has led to the creation of the National Nanotechnology Initiative by the US Government as well as Nanoscience and Technology Initiative by Government of India.

A major difference between the existing silicon technology and emerging nanoelectronics is that nanoelectronic systems, such as a nano-computer system, would have to be designed and self-assembled bottom-up whereas for the existing silicon technology, a top-down design using photolithography is the norm. Therefore, this section discusses the shifting design paradigm that will have to be embraced for building nano-computing devices. Defect levels are expected to be high in nano-technology circuits, and to counter this, defect- and fault-tolerant architecture will have to be used. The next section, therefore, discusses the design paradigm for using this technology in building nano-computers.

1.5.1 From Microelectronics to Nanoelectronics

The rapid growth of microelectronics has been based on the continuous miniaturization of electronic components over decades. Since the invention of the transistor, electronic circuits have evolved at an amazing pace from the early integrated circuits (ICs), with tens of components, to the present VLSI systems with hundreds of millions of components. This evolution is commonly referred to as being governed by Moore's Law which states that the number of electronic components per chip doubles every 18 months. VLSI circuits today are based on the CMOS field-effect transistors (FETs),

and the state-of-the-art fabrication process of the CMOS has reached a node dimension of 90 nm. However, as CMOS technology enters the nanoelectronic realm (tens of nanometers and below), where quantum mechanical effects start to prevail, conventional CMOS devices are meeting many technological challenges for further scaling. A variety of non-classical CMOS structures have been invented and investigated worldwide. It is generally believed that these novel structures will extend the CMOS technology to 45 nm nodes by the year 2009. If this scaling continues beyond 2009, however, CMOS technology is anticipated to hit a brick wall and cease to decrease in size around 2019. This will be due to many reasons such as the physical limitations imposed by thermal fluctuations, power dissipations and quantum effects, and the technological limitations in manufacturing models (for e.g., lithography), etc.

1.5.2 From Nanoelectronics to Nanoelectronic Computers

The advances at device and circuit levels have raised design issues for computer architectures based on nanoelectronic and quantum devices. The developments of nanoelectronics could eventually lead to extremely large scales of integration, of an order of a trillion (10^{12}) devices in a square centimeter. The architectures of the integrated circuits and systems must be suitable for implementations in nanoelectronic devices. In other words, architectures must optimally make use of the proprieties and at the same time deal with the drawbacks of the devices. There are many features in nanoscale devices that impose limitations on nanoelectronic architectures, while the most prominent ones have been recognized as the devices' poor reliabilities, the difficulties in realizing interconnects, and the problem of power dissipation.

The unreliability of nanoelectronic devices comes from two sources. One is the bottom-up manufacturing process of self-assembly, which will be used at dimensions below those for which conventional top-down fabrication techniques can be used. Since imprecision and randomness are inherent in this self-assembly process, it is almost inevitable that a large number of defective devices will appear due to this fabrication process. The other source of errors is the environment in which the devices will be operating.

Due to a reduced noise tolerance of low thresholds of state variables, malfunctions of devices may be induced by external influences such as electromagnetic interference, thermal perturbations, cosmic radiation, etc. Hence, permanent faults or defects may emerge during the manufacturing process, while transient errors may spontaneously occur during operation. The issue of defect- and fault-tolerance is therefore critical for any large integration of unreliable nanoelectronic devices. Several techniques, such as NAND-multiplexing, N-modular redundancy (NMR) and reconfiguration have been investigated for fault-tolerant implementations in nanocomputer architectures.

1.5.3 Expectations from Alternative Technologies

In order for alternative technologies to be acceptable, it is expected that they must be easier and cheaper to manufacture than CMOS. They should be able to drive capacitances of interconnects of any length and have a high level of integration (>10^{10} transistors/circuit). They should have high reproducibility (better than ± 5%) and reliability (operating time >10 years) with very low cost (<1 μcent/transistor) and better heat dissipation characteristics and amenable solutions.

1.5.4 Improving on Current Transistor Technology

Currently, to increase the advancement toward better nanotransistors, research is being done on the use of **carbon nanotubes (CNTs)** and **silicon nanowires** for future transistors and other molecular electronic devices. The appeal of these materials is that their small molecular structures will conceivably enable scaling (miniaturization) beyond current advanced lithographic techniques). This approach could possibly help memory and other similar applications, but the fundamental limit calculation reveals that potential molecular or CNT devices would have to operate slower than scaled CMOS as they are smaller than scaled CMOS.

1.5.5 Carbon Nanotubes

Carbon nanotubes (CNTs) are formed when a sheet of carbon atoms is rolled up to form molecular atoms or cylinders, as seen in Fig. 1.10. These cylinders have a diameter of only about 1 to 20 nanometers, but can have lengths from 100 nanometers to several microns. The fabrication of the tubes is what determines whether they are a semi-conductor material or a metallic material. An example of the type of semi-conductor that the carbon nanotubes can be used in is the zirconium dioxide gate. The tubes would help in their insulation, because they have an efficient charge injection and they reduce current leakage, or losses, due to their high capacitance. Tests conducted on such semi-conductors show that they have good switching characteristics based on their subthreshold slope. The difference in subthreshold values, or subthreshold swing, is an important aspect in the miniaturization of transistors because "it measures how well a small swing in gate voltage can cut off current flow. Low cutoff current directly translates into low standby power." (Bourianoff)[2].

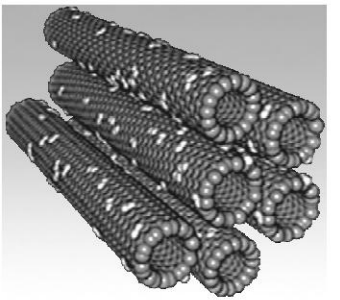


Fig. 1.10: Carbon Nanotubes

Currently, there are two main problems with CNT devices, the first of which is that CNT materials are produced with all shapes, types and sizes mixed together. To actually use the tubes, though, the CNTs "must be separated into groups of similar size and chirality." Chirality is the property of having handedness (different

[2]George Bourianoff, "The Future of Nanocomputing," *Computer*, 36(8), Aug. 2003.

from its mirror image), being non-superimposable with the mirror image of oneself. This process becomes arduous since the only way to separate them is to look at each individual tube through a **scanning tunneling microscope (STM)** and then sort them based on what is seen through the STM. Another problem that CNTs have shown is one of contact resistance. The resistance inside the tubes is too high. Even the theoretically best value of 6 kilo ohms is high and will limit the maximum current. This is a problem when the tubes are used with more conventional CMOS devices.

Silicon Nanowires: Another material being looked at for help in creating future transistors are silicon nanowires, seen in Fig. 1.11. They show promise in that they result in greater mobilities than bulk silicon. This effect, while not completely understood, is thought to be related to the quantum-confined nature of the wire, which limits the density of available phonon states and hence reduces the probability of an electron-scattering event—that is, it reduces drag. Since silicon is the popular material inside computers already, this observation makes replacing bulk silicon channels with silicon nanowires an appealing option.

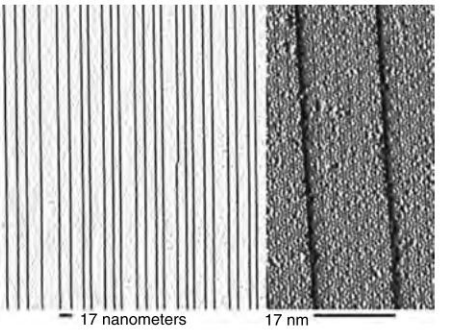


Fig. 1.11: Silicon Nanowires

There are three silicon nanowire devices that have already been fabricated. The silicon nanowire connects the source and drain contact points, but with different gate structures: a back gate, a metallic gating structure separated by an oxide and a coaxial structure. Currently, experiments are being performed to tell which of these is the best, and how close they can get to the theoretical productivity limit by using silicon nanowires.

Silicon nanowires can also probably help in other applications that rely on certain length to diameter ratios. Cross-bar arrays can be used to explain this sort of application. In such structures, one array of parallel nanowires is overlaid on a second array of nanowires oriented at right angles to the first array. The cross-points of the arrays can be used to either store or switch information depending on the device details. For example, they can act as a switch with bistable positions open or closed. The mechanical equilibrium of the wires maintains the neutral (open) position. Applying opposite charges to the wires pulls them toward each other until they touch, at which time molecular forces hold them in the closed position. Applying similar charges to the two wires forces them apart, resetting them to the initial position.

However, these isolated devices are not as fast as scaled silicon. Their advantages are increased density and reduced fabrication costs. To see this progression, note the difference in the sizes of computer components as shown in Fig. 1.12. However, problems with these cross-bar architectures include the creation of signal restoration and fan out and the connection of self-assembled modules to global control lines.

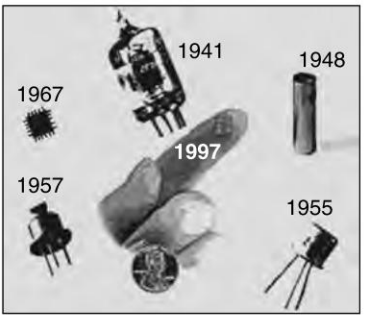


Fig. 1.12: Shrinking Size of Computer Components over the Years

Currently, there are nanotransistors about 35 nanometers wide with a 1.2 nm thick gate oxide. Ballistic electron transport over the distance of 30 nm minimizes electron-defect scattering and permits very high current efficiencies. Transistors now allow only 35 percent of the current that flows through them to get from the source to the drain, losing the rest to the insulator surfaces around it. At 35 nm though, the electrons migrate ballistically, meaning that they do not bump into anything. This causes only a 15 percent loss

from start to finish. This nanotransistor, made of Tungsten and Silicon, appears in Fig. 1.13.

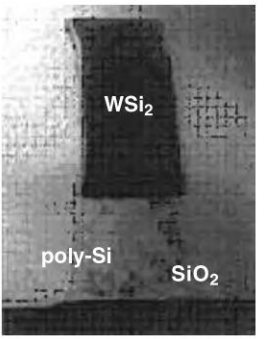


Fig. 1.13: Nanotransistor

1.5.6 Alternatives to Transistor Technology—Quantum Computing

The next step is to go on a whole new route altogether with computing—quantum computing, which is very different from the computing that we do today. While it is very different from any computing we do today, it is still a part of nanocomputing, because it uses the properties of single atoms to do computing, which is certainly within the realm of nanocomputing. Fig. 1.14 shows a model of a possible architecture for quantum computation.

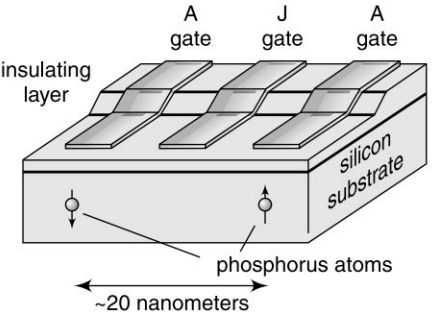


Fig. 1.14: Proposed Quantum Computer Architecture

What makes quantum computing so different is that it is based on the laws of quantum mechanics, which are fundamentally different from classical mechanics. Classical computers use transistors, with states of 1 or 0, which represent either the flow of current or

the lack of current, respectively. A single 1 or 0 is called a bit. In a quantum computing system, a single piece of information is called a qubit. This is inherently different from a normal bit. Here, the number 0 or 1 represents the state of an atom, where 0 is at rest, and 1 is excited. The main difference from a normal bit lies in the following: the laws of quantum mechanics allow an atom to be arranged in a coherent superposition of states of 0 and 1. **Coherent superposition** or **quantum superposition** is defined as a state where an atom has an equal probability to be at excited or at rest state.

Again, the idea is that the atom is in both states at the same time. This allows for some interesting computational applications. A normal register that is 3 bits wide can hold a single number out of 8 possibilities. At the same time, 3 qubits can hold all 8 of the possible numbers at the same time. This is because it can contain each qubit in superposition between the states, and therefore contain all possible numbers. Also, if you increase the number of qubits by 1, this doubles the number of numbers that can be stored, as well as the size of the numbers, whereas adding an extra bit would only double the size of the number that can be stored.

From the humble beginnings that are currently implemented with transistors just dipping into nanoscale to more complex nanostructures, such as silicon nanowires and carbon nanotubes to the theoretical quantum computing, nanocomputing is something that is very important to computers. Even though the possibilities look exciting, we still have a long way to go before we reach any of the exciting theoretical possibilities that lie out there. With the continuing efforts of researchers, the future of nanocomputing looks bright.

Nanocomputers have the potential to revolutionize the twenty-first century in the same way that the transistor and the internet led to the information age. Increased investments in nanotechnology could lead to breakthroughs such as molecular computers. Billions of very small and very fast (but cheap) computers networked together can fundamentally change the face of modern information technology and computing in corporations that are today using mighty mainframes and servers. This miniaturization will also spawn a whole series of consumer-based computing products: computer clothes, smart furniture, and access to the internet that is a thousand times faster than today's fastest technology.

Several of the technologies are strongly tied to a single application area or niche where the technology is particularly effective. For example, quantum computing can be used to find prime factors very efficiently by means of Shor's Algorithm, but is much less efficient when used for other applications. In this case, an 'effective' time per operation is defined as the time required by a classical device in a classical architecture by using a classical algorithm to do the calculation. Therefore the 'effective' operation time of an N-qubit quantum computer factoring a large number is very much faster than the operation time of an N-gate classical computer because of the inherent parallelism associated with quantum computing. Similar arguments can be made for biologically inspired and optical computing.

Figure 1.15 represents initial estimates for the comparison of these very disparate technologies. It conveys meaningful information about the relative positions of the emerging technologies in this application space. It shows that few of the new technologies are directly competitive with scaled CMOS and most are highly complementary. It also shows very clearly the benefit to be derived from heterogeneous integration of the emerging technologies with silicon to expand its overall application space.

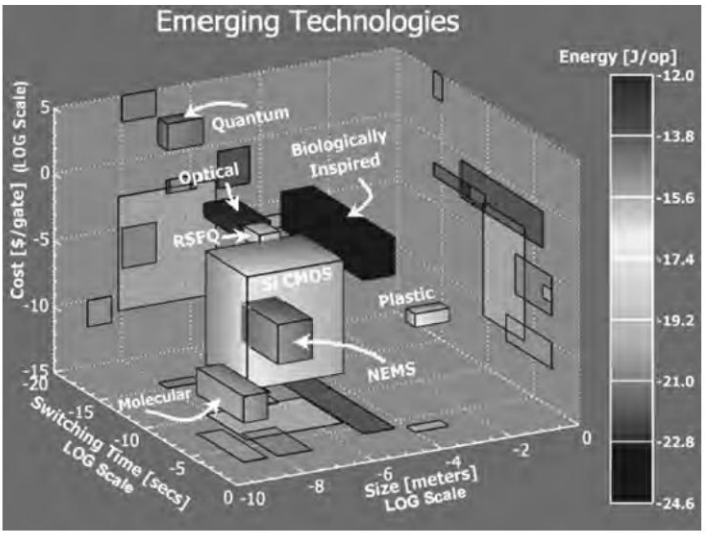


Fig. 1.15: Emerging Computer Technologies

1.6 Nano Information Processing

Information processing to accomplish a specific system function, in general, requires several different interactive layers of technologies. A comprehensive list of these layers begins with the required application or system function, leading to system architecture, micro or nanoarchitecture, circuits, devices, and lastly, would-be materials. As shown in Fig. 1.16, a different representation of this hierarchy begins with the lowest physical layer represented by a device and ends with the highest layer represented by a computational model. In this more schematic representation focused on generic information processing, a fundamental unit of information (for example, a bit) is represented by a computational state variable, for example, the position of a bead in the ancient Abacus calculator, or the charge or voltage state of a node capacitance in CMOS logic. A device provides the physical means of representing and manipulating a computational state variable among its two or more allowed states. The device is a physical structure resulting from the assemblage of a variety of materials possessing certain desired properties obtained through exercising a set of fabrication processes. An important layer not shown in this hierarchy is the classes of materials and processes necessary to fabricate the required device structure. Architecture, or in this instance nanoarchitecture,

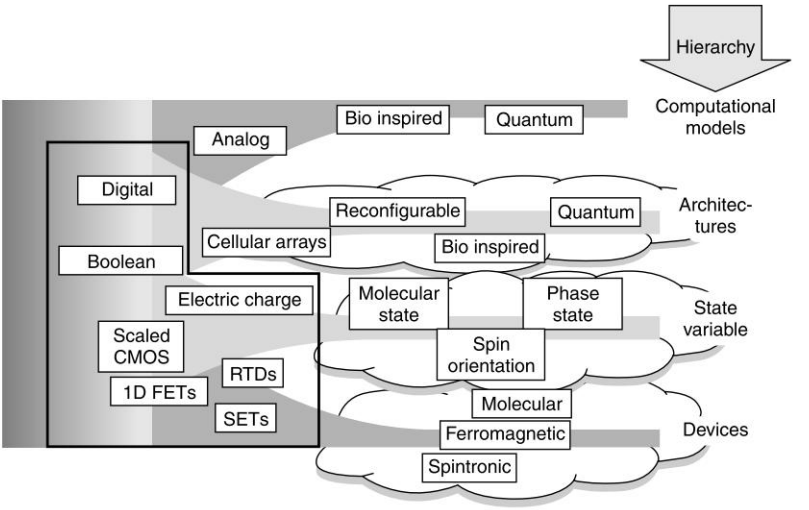


Fig. 1.16: Information Processing Technology

is the physical means of organizing higher level functional primitives formed by using devices to represent and enable execution of a computational model. A computational model is the means by which information is processed, for example, logic, arithmetic, memory, CNN (cellular non-linear network), or bio-inspired neuromorphic functions using digital, analog, or bio-inspired methods.

In Fig. 1.16, the elements shown in the outline box represent current CMOS and other technologies based on charge as the computational state variable used in Boolean architecture enabling a digital computational model. The entries to the right of the outline box grouped in the 4 categories summarize possible approaches to new device structures enabling some of the indicated new state variables to achieve the new nano-architectures and computational model. A new information processing technology will likely require an innovative and interactive combination of new elements in each of these layers.

1.7 Prospects and Challenges

Contrary to popular belief, the marriage of chemistry, computing, and microscopic engineering known as nanotechnology is not a new phenomenon; scientists have been working on the possibilities for decades. Nanotechnology today is an emerging set of tools, techniques, and unique applications involving the structure and composition of materials on a nanoscale—that is, billionths of a meter. This research has the potential to usher in a golden era of self-replicating machinery and self-assembling consumer goods made from cheap raw atoms. The following list presents just a few of the potential applications of nanotechnology:

- Expansion of mass storage electronics to huge multi-terabit memory capacity, increasing by a thousand fold the memory storage per unit. Recently, IBM's research scientists announced a technique for transforming iron and a dash of platinum into the magnetic equivalent of gold: a nanoparticle that can hold a magnetic charge for as long as 10 years. This breakthrough could radically transform the computer disk-drive industry.
- Making materials and products from the bottom up; that is, by building them from individual atoms and molecules.

Bottom-up manufacturing should require fewer materials and pollute less.

- Developing materials that are 10 times stronger than steel, but a fraction of the weight, for making all kinds of land, sea, air, and space vehicles lighter and more fuel-efficient. Such nanomaterials are already being produced and integrated into products today.
- Improving the computing speed and efficiency of transistors and memory chips by factors of millions, making today's chips seem as slow as the dinosaur. Nanocomputers will eventually be very cheap and widespread. Supercomputers will be about the size of a sugar cube.
- Using gene and drug delivery to detect diseased cells; nanoagents will target organs in the human body, providing molecular repair and cell surgery.
- Removing the finest contaminants from water and air to promote a cleaner environment and potable water.

Many more applications will be recognized or identified over time.

There are ample opportunities for those working in semi-conductor devices and materials to develop better materials and processes before nanoelectronics could be commercialized. However, material development, especially for nanoelectronics, would require highly sophisticated equipment, instruments, and laboratory facilities. Therefore, research in this area would be primarily confined to government laboratories, large corporations, and a few major universities that are already doing so. Another important need would be in the area of device modeling and simulation. This new technology would require developing models and those models into possibly a very different breed of CAD (computer-aided design) tools. A few modeling and CAD tools are being developed at the University of Purdue (www.nanohub.org), and are worth considering for those interested in this aspect of nanoelectronics.

Development of commercial devices for nano-computing would require solving several 'hard' problems. Some of these problems are discussed here. The 'interconnect problem' is one of the most challenging issues. There are two aspects of the interconnect problem. First, the minimum contact resistance to make connection between nanodevices and the external world is 6 K ohms or more,

which is very high. The second aspect concerns the large number of wires that would be needed to connect such complex devices. Adequate spacing between these wires would be needed to prevent crosstalk and capacitive coupling. Current efforts are focused mostly on developing basic nano-devices that could serve as the basic building-blocks in assembling larger nano-systems. However, the nature of such integration is not yet known. Mapping of the functionality of the traditional silicon-based circuits into nano-electronics paradigm would be another challenge. Developing circuit models for nanodevices that could be used for integration into CAD tools for design verification and simulation will require significant effort. It would be quite a challenge to develop design and test strategies for such dense systems. Cost-effective manufacturing processes will have to be developed for mass production of nano-computers based on nano-technology. Cost-effective self-assembly of nano-devices would have to be developed. However, these challenges also present opportunities to e1ectrical and computer engineers.

Designing nanocomputers would require an entirely different bottom-up design approach and tools. New design management and verification paradigms will have to be developed. Significant opportunity exists in the area to account for this new design philosophy. The stuck-at fault models, most commonly used to test the functionality of bulk silicon circuits, might not apply to the new nano-circuits. Therefore, new fault models, test algorithms, and test strategies will have to be developed.

Microelectronics has come a long way in a short period of time from power-hungry NMOS and BJT devices to faster and leaner CMOS devices. There was enough skepticism just 20 years ago in promoting CMOS technology because of latchup, slow speed, and many related fabrication problems. However, all those problems were gradually solved resulting in the CMOS being the technology of choice. It is hoped that in the same way researchers would find solutions to the problems that appear to be formidable at this juncture in time and make nanoelectronics and nanocomputing a reality and the technology of choice in the coming decades.

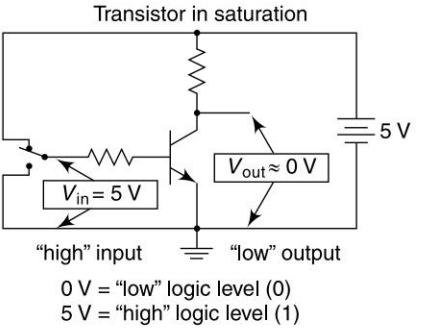
CHAPTER 2

Physics of Nanocomputing

2.1 Digital Signals and Gates

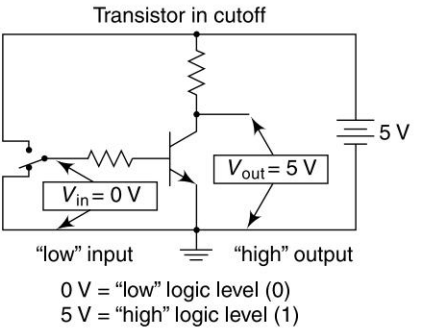
While the binary numeration system is an interesting mathematical abstraction, it has got interesting practical application to electronics. This section explains practical application of the concept of binary bits to circuits. What makes binary numeration so important to the application of digital electronics is the ease in which bits may be represented in physical terms. Because a binary bit can only have 1 of 2 different values, either 0 or 1, any physical medium capable of switching between two saturated states may be used to represent a bit. Consequently, any physical system capable of representing binary bits is able to represent numerical quantities, and potentially has the ability to manipulate those numbers. This is the basic concept underlying digital computing.

Electronic circuits are physical systems that lend themselves well to the representation of binary numbers. Transistors, when operated at their bias limits, may be in 1 of 2 different states: either cutoff (no controlled current) or saturation (maximum controlled current). If a transistor circuit is designed to maximize the probability of falling into either one of these states (and not operating in the linear, or active, mode), it can serve as a physical representation of a binary bit. A voltage signal measured at the output of such a circuit may also serve as a representation of a single bit, a low voltage representing a binary '0' and a (relatively) high voltage representing a binary '1'. Note the following transistor circuit:



In this circuit, the transistor is in a state of saturation by virtue of the applied input voltage (5 volts) through the two-position switch. Because it is saturated, the transistor drops very little voltage between collector and emitter, resulting in an output voltage of (practically) 0 volts. If we were using this circuit to represent binary bits, we would say that the input signal is a binary '1' and that the output signal is a binary '0'. Any voltage close to full supply voltage (measured in reference to ground, of course) is considered a '1' and a lack of voltage is considered a '0'. Alternative terms for these voltage levels are high (same as a binary '1') and low (same as a binary '0'). A general term for the representation of a binary bit by a circuit voltage is logic level.

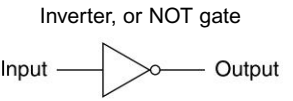
Moving the switch to the other position, we apply a binary '0' to the input and receive a binary '1' at the output:



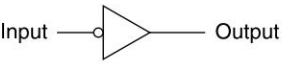
What we have created here with a single transistor is a circuit generally known as a logic gate, or simply gate. A gate is a special type of amplifier circuit designed to accept and generate voltage signals corresponding to binary 1's and 0's. As such, gates are not

intended to be used for amplifying analog signals (voltage signals between 0 and full voltage). Used together, multiple gates may be applied to the task of binary number storage (memory circuits) or manipulation (computing circuits), each gate's output representing one bit of a multi-bit binary number. Right now it is important to focus on the operation of individual gates.

The gate shown here with the single transistor is known as an inverter, or NOT gate, because it outputs the exact opposite digital signal as what is input. For convenience, gate circuits are generally represented by their own symbols rather than by their constituent transistors and resistors. The following is the symbol for an inverter:



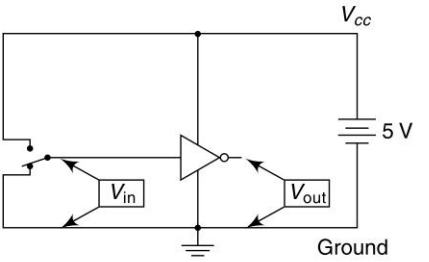
An alternative symbol for an inverter is shown here:



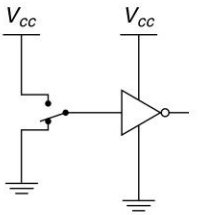
Notice the triangular shape of the gate symbol, much like that of an operational amplifier. As was stated before, gate circuits actually are amplifiers. The small circle, or 'bubble' shown on either the input or output terminal is standard for representing the inversion function. As you might suspect, if we were to remove the bubble from the gate symbol, leaving only a triangle, the resulting symbol would no longer indicate inversion, but merely direct amplification. Such a symbol and such a gate actually do exist, and it is called a buffer, the subject of the next section.

Like an operational amplifier symbol, input and output connections are shown as single wires, the implied reference point for each voltage signal being 'ground'. In digital gate circuits, ground is almost always the negative connection of a single voltage source (power supply). Dual, or 'split' power supplies are seldom used in gate circuitry. Because gate circuits are amplifiers, they require a source of power to operate. Like operational amplifiers, the power supply connections for digital gates are often omitted from the symbol for simplicity's sake. If we were to show all the necessary

connections needed for operating this gate, the schematic would look something like this:



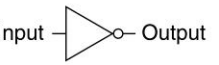
Power supply conductors are rarely shown in gate circuit schematics, even if the power supply connections at each gate are. Minimizing lines in our schematic, we get this:



'V_{cc}' stands for the constant voltage supplied to the collector of a bipolar junction transistor circuit, in reference to ground. Those points in a gate circuit marked by the label 'V_{cc}' are all connected to the same point, and that point is the positive terminal of a DC voltage source, usually 5 volts.

One common way to express the particular function of a gate circuit is called a truth table. Truth tables show all combinations of input conditions in terms of logic level states (either 'high' or 'low', '1' or '0', for each input terminal of the gate), along with the corresponding output logic level, either 'high' or 'low'. For the inverter, or NOT, circuit just illustrated, the truth table is very simple indeed:

NOT gate truth table



Input	Output
0	1
1	0

Truth tables for more complex gates are, of course, larger than the one shown for the NOT gate. A gate's truth table must have as many rows as there are possibilities for unique input combinations. For a single-input gate like the NOT gate, there are only two possibilities, 0 and 1. For a two input gate, there are four possibilities (00, 01, 10, and 11), and thus four rows in the corresponding truth table. For a three-input gate, there are 8 possibilities (000, 001, 010, 011, 100, 101, 110, and 111), and thus a truth table with 8 rows are needed. The mathematically inclined will realize that the number of truth table rows needed for a gate is equal to 2 raised to the power of the number of input terminals.

2.2 Silicon Nanoelectronics

MOS (metal oxide semiconductor) devices have been scaled for more than 30 years to achieve higher density and performance at lower power consumption. Transistor delay times have reduced by more than 30 percent per technology generation resulting in doubling of performance every 2 years. Supply voltage (V_{DD}) has been scaling down at the rate of 30 percent per technology generation in order to keep power consumption under control.

2.2.1 Short Channel Effects

A short channel effect in scaled MOSFET (metal oxide semi-conductor field-effect transistor) devices is the lowering of the threshold voltage V_{th} with decreasing channel length. In long channel devices the source and drain are separated far enough that their depletion regions have no effect on the potential or field pattern in most part of the device, and hence, the threshold voltage is virtually independent of the channel length and drain bias. In a short-channel device, however, the source and drain depletion width in the vertical direction is comparable to the effective channel length. This causes the depletion regions from the source and the drain to interact with each other. The consequence of this is lowering of the potential barrier between the source and the channel.

The drain voltage also has a significant effect on the potential barrier for short channel devices. Under off conditions, this potential

barrier between the source and the channel prevents electrons from flowing to the drain. For a long-channel device, the barrier height is mainly controlled by the gate voltage and is not sensitive to drain-source voltage V_{ds}. However, when a high drain voltage is applied to a short-channel device, the barrier height is lowered, resulting in further decrease of the threshold voltage. The source then injects carriers into the channel surface without the gate playing a role. This is known as DIBL (drain-induced barrier lowering). It is enhanced at higher drain voltage and shorter effective lengths.

2.2.2 Leakage Current in Scaled Devices

A major concern in silicon MOSFETs is the increasing drain control over the channel. Fig. 2.1 shows how leakage power is keeping pace with active power in scaled CMOS.

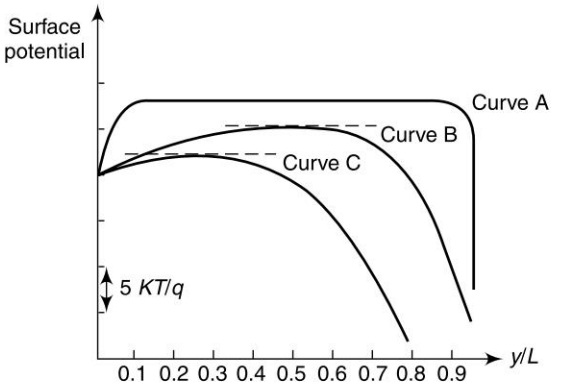


Fig. 2.1: Plot of Surface Potential versus Lateral Distance: (a) Curve A– Long Channel MOSFET (b) Curve B–Long Channel MOSFET (c) Curve C–Short Channel MOSFET at High Drain Bias

As DIBL increases, the V_{th} of the device gets significantly lowered, resulting in a higher subthreshold leakage current. Subthreshold or weak inversion current is the drain current of the MOSFET when the gate is biased at a voltage less than V_{th}. The minority carrier in the weak inversion region is small but not zero. This results in a diffusion current from the drain to the source of the device even when the gate to source voltage (V_{gs}) is at zero potential. As the threshold voltage is lowered, the current increases

exponentially. The problem of lowered V_{th} in MOSFETs is reduced by increasing the channel doping in a region below the drain and the source (retrograde well) and near the source-bulk and drain-bulk junctions referred to as halo implants.

For scaled devices ($l_{eff} < 50$ nm) the increased halo doping creates a high electric field across the reverse biased drain-bulk junction. This causes a junction BTBT (Band-to-Band Tunneling) current to flow from the drain to the source of an NMOS (n-type Metal Oxide Semiconductor) device. A similar current flows across the source-body junction too depending on the biasing conditions. Thus the halo doping increases the subthreshold current at the cost of higher BTBT leakages in scale devices as shown in Fig. 2.2.

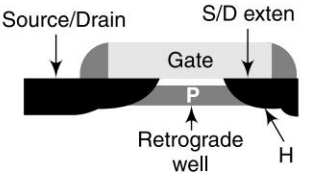


Fig. 2.2: Channel Profile Showing Retrograde Well and Halo Regions

For scaled devices, the oxide thickness is scaled commensurately to increase the gate control over the channel. This results in another significant leakage current called gate tunneling leakage. Gate tunneling current is the current due to the tunneling of electrons from the conduction band of bulk silicon and the source/drain overlap regions through the potential barrier of the oxide into the gate of the device. In scaled silicon devices the leakage current increases almost exponentially with scaling. This reduces the 'on' current to 'off 'current ratio of transistors and also consumes considerable amount of power even in the standby mode. Some popular methods to reduce it include using multiple V_{th} transistors in design, low V_{dd} (or drowsy) state in standby mode, and adaptively control threshold voltage during various operation modes.

2.2.3 Process Variation

Process parameter variation has also been identified as one of the principle bottlenecks in scaling of silicon MOSFETs beyond

100 nm. As the device dimensions continue to shrink, it is becoming increasingly difficult to control the critical process parameters, like gate length, oxide thickness, dopant concentration and random dopant fluctuation. This has resulted in significant variation of the threshold voltage of the device thereby causing a considerable spread in the switching delay of logic gates.

With process variation, production yield has gone down drastically and new design methodologies like statistical timing analysis, statistical sizing for yield are becoming popular. In the regime where the gate lengths are scaled below 50 nm, predictable current design with tolerable power budgets may become uneconomical for production.

2.3 Carbon Nanotube Electronics

Carbon nanotubes have been called the "wonder material of the twenty-first century", "the building blocks for the future of electronics", and the "replacement for silicon circuits"[1]. While it is debatable whether such grand predictions will come true, carbon nanotubes have unarguably generated tremendous interest amongst chemists, physicists and electrical engineers alike by virtue of their unique properties and potential to offer solutions to several problems as conventional technology approaches fundamental limits. This section focuses on the unique electronic properties of carbon nanotubes and strives to understand how these electronic properties arise from energy band theory considerations.

2.3.1 Band Structure of Carbon Nanotubes

A carbon nanotube (CNT), simply, is the honeycomb lattice of a graphene sheet rolled into a cylinder. MWCNT (multi-walled CNTs), concentric cylinders of hollow carbon nanotubes, were first discovered by S. Iijima in 1991, and SWCNT (single-walled CNTs), single, hollow CNTs, were discovered two years later by S. Iijima and D. Bethune. SWCNTs are typically several micrometers (μm) long and only a few nanometers (nm) in diameter, resulting in a

[1]Avouris, P., "Supertubes", *IEEE Spectrum*, August 2004, p. 41.

large aspect ratio that allows approximation of the CNT as a 1D system. Depending on the CNT chirality (degree of twist) and diameter, a CNT can be either metallic or semiconducting. It can be theoretically predicted that 1/3 of rolled CNTs will be metallic and 2/3 will be semiconducting. Moreover, the bandgap of semiconducting nanotubes can be 'tuned' by adjusting the diameter of the nanotube; the larger the diameter, the smaller the bandgap. The fact that the electronic structure of a carbon nanotube depends only on its physical geometry, without any doping, is unique to solid-state physics and the basis for many of its proposed applications in electronics.

This section investigates the theoretical reasoning behind the above observations. Specifically, the electronic structure of graphene is used to study how energy gaps are formed when the graphene sheet is rolled into a cylinder. It is hoped that the analysis and discussions below will help in understanding the origins of the electronic properties of CNTs and remove the perception of 'magic' surrounding this 'wonder' material.

If silicon electronics reach the limits of scalability, it will only be prudent to look for other materials to replace silicon. Carbon nanotube has emerged as a promising replacement to silicon in future nanoelectronic designs. Carbon nanotube transistors are predicted to have about 10 times the current density of silicon MOSFETs, and maintain an on-current to off-current ratio of more than 10^3. Further, carbon nanotubes will allow successful integration of gates with high dielectric strength because there is no dangling bond in carbon nanotubes. This would result in the possible use of thicker gate dielectrics thereby reducing gate leakage at no performance penalty. Several other molecular diodes that are being currently investigated show enormous promise as ultra-scaled switches for future technologies.

In pursuit for novel materials in a post-silicon electronics era, scientists and engineers worldwide have already started active research in carbon nanotube (CNT) electronics. Although carbon filaments of nanoscale diameters (~10 nm) were extensively grown in the 1970s and 1980s, it was only after the pioneering work of Iijima in 1991 that the potential of carbon nanotube as a possible device material has been recognized and extensively studied. Owing to their excellent electrical, mechanical and thermal properties, researchers have identified an array of potential applications for

carbon nanotubes. In the short span since their inception, field-effect transistors, diodes, optical and cathode ray emitters, bio-sensors and energy storage elements have been demonstrated. Fig. 2.3 shows some graphite forms and strips rolled into tubes.

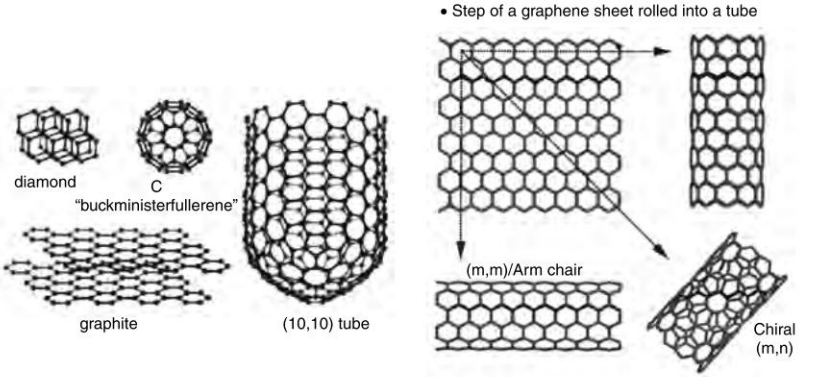


Fig. 2.3: Graphite Forms and Strips Rolled into Tubes

2.3.2 Carbon Nanotube Properties

CNT is a tubular form of carbon with diameter as small as 1 nm. Its length is a few nm to microns. CNT is configurationally equivalent to a two-dimensional graphene sheet rolled into a tube. CNT exhibits extraordinary mechanical properties as listed here.

- The strongest and most flexible molecular material because of C-C covalent bonding and seamless hexagonal network architecture.
- Young's modulus of over 1 TPa vs 70 GPa for aluminum, 700 GPa for C-fiber, strength to weight ratio 500 time >for Al; similar improvements over steel and titanium; one order of magnitude improvement over graphite/epoxy.
- Maximum strain ~10 percent much higher than any material.
- Thermal conductivity ~ 3000 W/mK in the axial direction with small values in the radial direction.
- Electrical conductivity six orders of magnitude higher than copper.
- Can be metallic or semiconducting depending on chirality with 'tunable' bandgap.
- Electronic properties can be tailored through application of external magnetic field, application of mechanical deformation.

- Very high current carrying capacity.
- Excellent field emitter; high aspect ratio and small tip radius of curvature are ideal for field emission.
- Can be functionalized.

CNTs are grown by CDV from an iron containing catalyst that is patterned on a Si support as shown in Fig. 2.4.

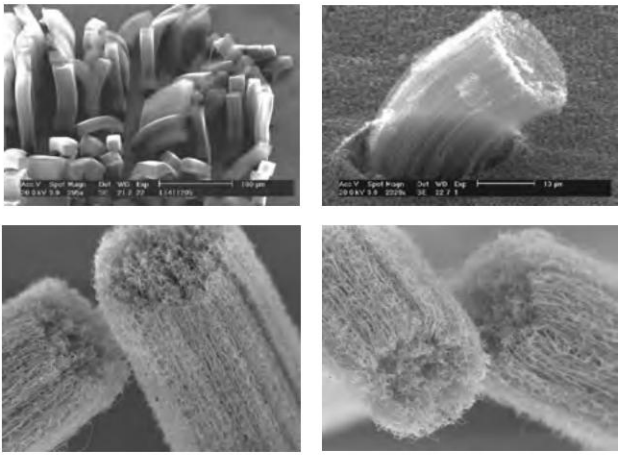


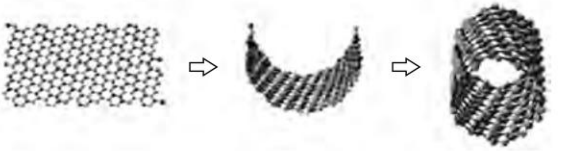
Fig. 2.4: Growth of Carbon Nanotubes[2]

2.3.3 Molecular Structure

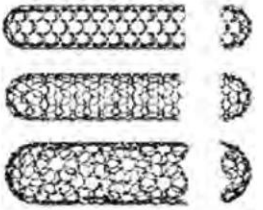
The carbon nanotube is the fourth stable structure of carbon after diamond, graphite and fullerene. An ideal nanotube can be thought of as a hexagonal network of carbon atoms (that form crystalline graphite) that has been rolled up to make a seamless cylinder (Fig. 2.5). Just a nanometer across, the cylinder can be tens of microns long, and each end is 'capped' with half of a fullerene molecule. Single-wall nanotubes can be thought of as the fundamental cylindrical structure, and these form the building blocks of both multi-wall nanotubes and the ordered arrays of single-wall nanotubes called ropes.

Since carbon nanotubes are constructed of hexagonal networks, the carbon atoms contain an sp^2 hybridization. Among the 4 valence electrons of carbon atom the first 3 electrons belong to the σ-orbital and are at energies ~2.5 eV below the Fermi level;

[2]http://pages.unibas.ch/phys-meso/Pictures



(a) Formation of carbon nanotube from a graphite sheet



b) Nanotube structure with fullerene 'cap'

(c) A fullerene (C_{60}) molecule

Fig. 2.5: Molecular Structure of Carbon Nanotube

therefore, they do not contribute to conduction. The fourth valence electron, however, is located in the π-orbital, which is slightly below the Fermi level; therefore, it does not contribute to conduction, but controls conduction and transport properties. This corresponds to the valence bond of the energy diagram. The anti-bonding π-orbital is slightly above the Fermi level, which corresponds to the conduction band in an energy diagram.

2.3.4 Chiral Vector

The structure of single-wall carbon nanotube (except for cap region on both ends) is specified by a vector of original hexagonal (also called honeycomb) lattice called the chiral vector. The chiral vector corresponds to a section of nanotube perpendicular to the tube axis. In Fig. 2.6, the unrolled hexagonal lattice of the nanotube is shown, in which $\overrightarrow{OB}$ is the direction of the nanotube axis, and $\overrightarrow{OA}$ corresponds to the chiral vector, C_h.

By considering the cyrstallographically equivalent sites O, A, B and B', and by rolling the honeycomb sheet so that points O and A coincide (and points B and B' coincide), a paper model of carbon nanotube can be constructed. The vector $\overrightarrow{OB}$ defines another vector namely translational vector, T. The rectangle generated by

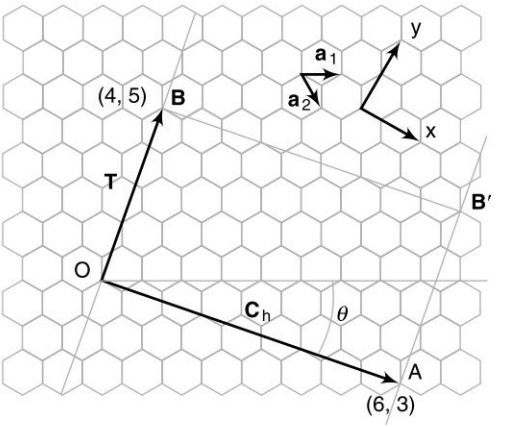


Fig. 2.6: Definition of Chiral Vectors in the Hexagonal Lattice

the chiral vector C_h and translational vector T, i.e., the rectangle $OAB'B$ in Fig. 2.6 is called the unit cell for the nanotube. The chiral vector of the nanotube is defined as,

$$C_h = na_1 + ma_2 \tag{2.1}$$

Where n, m are integers ($0 \leq |m| \leq n$) and a_1, a_2 are the unit vectors of the hexagonal lattice. In Fig. 2.6, a_1 and a_2 can be expressed using the Cartesian coordinate (x, y) as

$$a_1 = \left(\frac{3}{2} a_{cc}, \frac{\sqrt{3}}{2} a_{cc} \right) \tag{2.2}$$

$$a_2 = \left(\frac{3}{2} a_{cc}, -\frac{\sqrt{3}}{2} a_{cc} \right) \tag{2.3}$$

Here, a_{cc} is the bond length of carbon atoms. For graphite a_{cc} = 1.42 Å. This same value is often used for nanotubes. But, a_{cc} = 1.44 Å is a better approximation for nanotubes. It should really depend on the curvature of the tube. A slightly larger value for more curvature is known.

We see from equations (2.2) and (2.3), that the lengths, a_1, a_2, i.e., $|a_1|$, $|a_2|$ are both equal to $\sqrt{3}\, a_{cc} = a$. Therefore, a is the unit length and this is also the lattice constant. Hence a_1, a_2 can be expressed in terms of lattice constant,

$$a_1 = \left(\frac{\sqrt{3}}{2}, \frac{1}{2} \right) a \tag{2.4}$$

$$a_2 = \left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)a \tag{2.5}$$

Length of the chiral vector is the peripheral length of the nanotube

$$L = |C_b| = a\sqrt{n^2 + nm + m^2} \tag{2.6}$$

The angle between the vectors C_b and a_1 is called chiral angle θ. It denotes the tilt angle of the hexagons with respect to the direction of the nanotube axis, and it specifies the spiral symmetry. The chiral angle is defined by taking the inner product of C_b and a_1, to yield an expression for cos θ:

$$\cos\theta = \frac{C_b \cdot a_1}{|C_b||a_1|} = \frac{2n + m}{2\sqrt{n^2 + nm + m^2}} \tag{2.7}$$

From this expression it can be shown that the chiral angle,

$$\theta = \tan^{-1}\left[\frac{\sqrt{3}m}{2n + m}\right] \tag{2.8}$$

The tube diameter is then given by

$$d_1 = \frac{L}{\pi} = \frac{a}{\pi}\sqrt{n^2 + nm + m^2} \tag{2.9}$$

The translational vector, T, which is perpendicular to the chiral vector, is expressed as

$$T = [(2m + n)a_1 - (2n + m)a_2]/d_R \tag{2.10}$$

The length T is the unit lattice length along the tube axis direction:

$$T = \sqrt{3}C_b/d_R = 3a_{c-c}\sqrt{n^2 + nm + m^2}/d_R \tag{2.11}$$

Here,

$$d_R = \begin{cases} d \text{ if } n\text{-}m \text{ is not a multiple of } 3d \\ 3d \text{ if } n\text{-}m \text{ is a multiple of } 3d \end{cases}$$

and d is the highest common divisor of (n,m).
The number of hexagons in a unit cell is given by:

$$N = \frac{2(n^2 + nm + m^2)}{d_R}$$

In Fig. 2.7, some chiral vector directions with different values of (n, m) are shown.

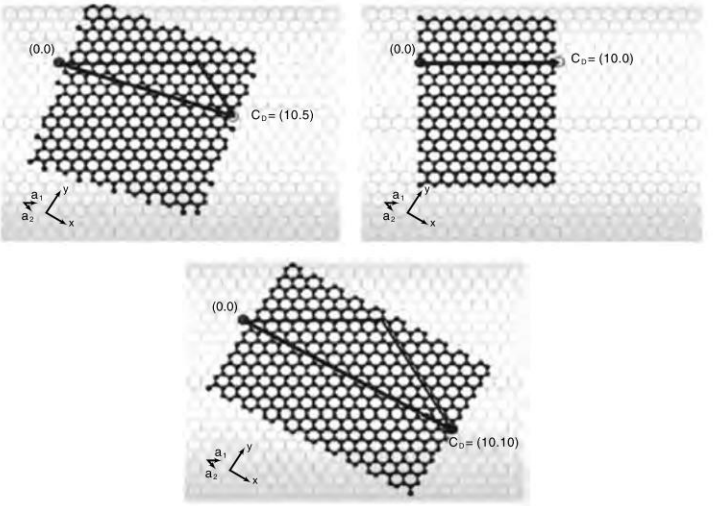


Fig. 2.7: Different Chiral Vectors in Unfolded Carbon Nanotube Lattice

Fig. 2.8 shows a real space lattice of graphene and reciprocal space representation of graphene with the first Brillouin zone.

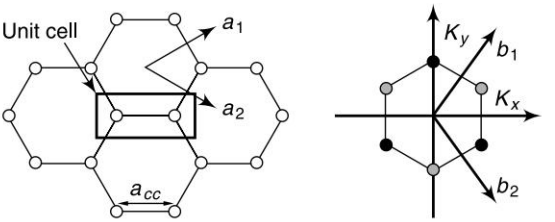


Fig. 2.8: (a) Real Space Lattice of Graphene, (b) The Reciprocal Space Representation of Graphene Showing the First Brillouin Zone

2.4 Carbon Nanotube Field-effect Transistors

Some of the pioneering work in carrier transport in carbon nanotubes has given insight into the behavior of carbon nanotube based transistors. The studies extend to both metallic as well as semiconducting nanotubes experimentally and theoretically but the principle transport mechanism has not yet been established without reasonable doubt. Two of the important devices experimentally used are Schottky Barrier Carbon nanotube FETs and MOSFET-like carbon nanotube FETs. Fig. 2.9 shows a computer simulated picture of a nanotube transistor.

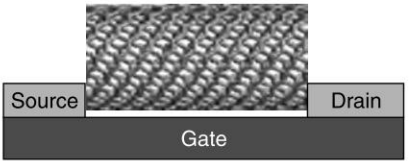


Fig. 2.9: Computer Simulated Picture of a Nanotube Transistor

2.4.1 Schottky Barrier Carbon Nanotube FETs

It has been experimentally shown that semiconducting nanotubes (CNTs) can work as channel material of field-effect transistors (FETs). Significant progress has been made in understanding and modelling the principle transport properties of these transistors. With ultra thin gate dielectrics, low voltage operation of carbon nanotube based transistors was demonstrated. It has been observed that the contact plays an important role in determining the performance of these nanotransistors. It has been predicted that in metallic source-drain carbon nanotube transistors, a potential barrier exists between the source/drain and the channel. The current in these devices is determined by the amount of tunneling through this potential barrier, which is modulated by the gate voltage.

Two important aspects of these nanotube transistors are worth mentioning. Firstly, the energy barrier at the Schottky barrier severely limits the transconductance of the nanotube transistors in the 'ON' state and reduces the current delivery capability—a key metric to transistor performance. Secondly, Schottky barrier carbon nanotube FETs exhibit strong ambipolar characteristics and this constraints the use of these transistors in conventional CMOS logic families. Fig. 2.10 shows the I-V characteristics of a Schottky barrier carbon nanotube FET.

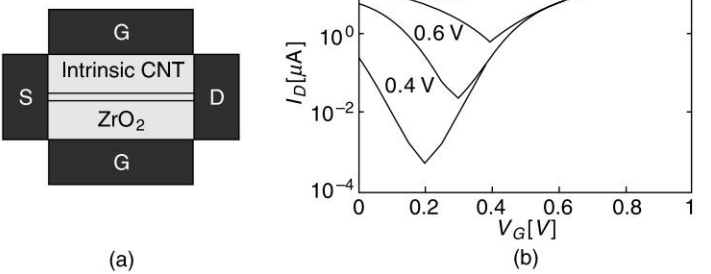


Fig. 2.10: (a) A Schottky Barrier Carbon Nanotube FET (the source and drain are metallic and a high K dielectric has been used), (b) The I_D-V_G Characteristics of the Schottky Barrier FET showing Ambipolar Conduction

2.4.2 MOSFET-like Carbon Nanotube FETs

Attempts have been made to develop carbon nanotube FETs which would behave like normal MOSFETs (Fig. 2.11). In this MOSFET-like device, the source and drain regions are heavily doped and it operates on the principle of barrier height modulation by application of the gate potential. It is important to note that doping in carbon nanotubes is not substitutional doping as in silicon. The required doping of the source/drain extension may be achieved either chemically or electrically. Carbon nanotubes are intrinsically p-type. With deposition of highly electropositive materials like potassium on a carbon nanotube, the Fermi level inside the nanotube can be shifted causing it to behave like n-type. In this case, the on-current is limited by the amount of charge that can be induced in the channel by the gate. It is obvious that the MOSFET-like device will give a higher 'on' current, and hence would define the upper limit of performance. Figure 2.12 shows a 3D model of carbon nanotube.

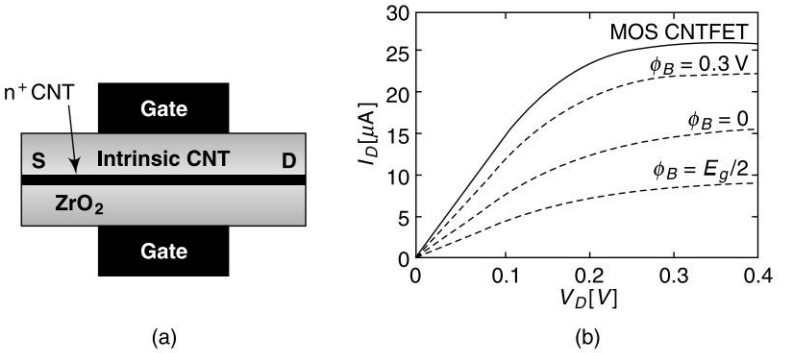


Fig. 2.11: (a) MOSFET-like Carbon Nanotube FET having n+ Source and Drain Regions, (b) The I_D-V_G Characteristics of the MOSFET-like Device Showing a Higher 'on' Current than a Corresponding Schottky Barrier Device (for different source drain metal work-functions)

2.5 Nanolithography

Nanolithography is a term used to describe a number of techniques for creating incredibly small structures. The sizes involved are on the order of tens of nanometers (nm). The word lithography is used because the method of pattern generation is essentially the same as writing, only on a much smaller scale.

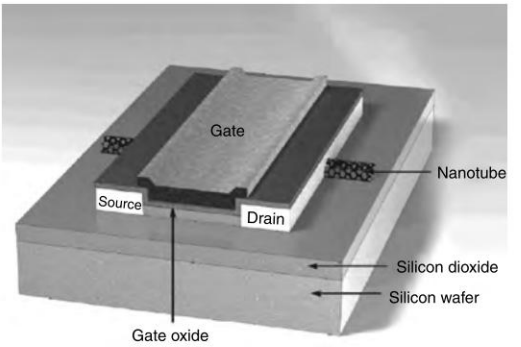


Fig. 2.12: 3D Model for CNFET Created by the IBM WRL (IBM Research News, 2002)

One common method of nanolithography, used particularly in the creation of microchips, is known as photolithography. This technique is a parallel method of nanolithography in which the entire surface is drawn on in a single moment. Photolithography is limited in the size it can reduce to, however, because if the wavelength of light used is made too small the lens simply absorbs the light in its entirety. This means that photolithography cannot reach the super-fine sizes of some alternate technologies.

A technology that allows for smaller sizes than photolithography is that of electron-beam lithography. Using an electron beam to draw a pattern nanometer by nanometer, incredibly small sizes (of the order of 20nm) may be achieved. Electron-beam lithography is much more expensive and time consuming than photolithography, however, making it a difficult sell for industry applications of nanolithography. Since electron-beam lithography functions more like a dot-matrix printer than a flash-photograph, a job that would take five minutes using photolithography will take upwards of five hours with electron-beam lithography.

New nanolithography technologies are constantly being researched and developed, leading to smaller and smaller possible sizes. Extreme ultraviolet lithography, for example, is capable of using light at wavelengths of 13.5 nm. While hurdles still exist in this new field, it promises the possibility of sizes far below those produced by current industry standards. Other nanolithography techniques include dip-pen nanolithography, in which a small tip is used to deposit molecules on a surface. Dip-pen nanolithography can achieve very small sizes, but cannot currently go below 40 nm.

2.6 Conclusions

Carbon nanotubes have caught the fancy of physicists, device engineers and circuit designers. Research has begun to harness the potential of these nano devices and use carbon nanotube based transistors in integrated circuit design for the future generations. Although, the present day understanding of the devices needs to be furthered and a considerable portion of the theoretical work has not yet been demonstrated in experiments, the promise is enormous. Like any other device that is in its premature state, reliable production of these devices is definitely an issue and an enormous amount of research is necessary to build carbon nanotube FETs with performance matrices comparable to the modern-day silicon MOSFETs. However, with their super-scaled dimensions, reliable and high current carrying capabilities and strong mechanical properties, CNTs have emerged as champions among the different revolutionary non-silicon devices that are being explored worldwide.

With the need for higher and higher integration density and complex on-chip functionality, the laws of physics would be taken to their limits, and circuit and system designers would have an increasing important role to play. Understanding the principles of operation of such ultra-scaled devices and using them in the ICs of the future generation would be a hurdle that the device, circuit and the architecture communities have to overcome together.

Electronic properties of carbon nanotubes have been explored with a variety of experimental techniques, over a wide range of temperature, by using magnetic fields and depending on contact geometry and materials.

The specificity of carbon nanotubes physics comes first from their reduced dimensionality, since diameters range from 1nm to few tens of nanometers, while nanotube length can be upscaled up to several microns or more. Physics in quasi-one dimensional systems is known to become more complex in regards to conventional materials for plenty of reasons. First, the contribution of electronic confinement, localization phenomena, electron-phonon or electron-electron interaction due to the reduction of screening effects is enhanced. This makes the description of intrinsic transport at the same time more complex but also challenging since carbon nanotubes on the other side offer relatively simple systems

for realistic modelling of both real space orbitals distribution and consequent electronic spectra.

For instance, it is possible to challenge the joint contribution of bandstructure effects and quantum interference effects in multi-walled nanotubes, or to investigate the possibility to unveil signatures of non-Fermi liquid (such as Luttinger Liquid) by analyzing current-voltage abnormalities beyond the linear response. An important observation is the fact that one can not easily elaborate on some universal framework for carbon nanotubes physical properties, since single-wall nanotubes, multi-walled or hybrid nanotubes will manifest different signatures due to severe change of transport dimensionality, specific resonances, conductance patterns and different interfacing properties with contact materials.

The second specificity when exploring nanotube electronics is the contact issue. Indeed, depending on nanotube geometrical features (for instance band-gap value of semiconducting tubes), and the nature of metallic contacts (Pd, Ti, Au) and contact geometry, the resulting charge injection properties at nanotube/metal interfaces will be very different. For metallic nanotubes, the contact nature ranges from ohmic and transparent contacts with little backscattering, to highly resistive contacts dominating electrical response as a consequence of interface quality and orbital bonding phenomenon. Contacting semiconducting nanotubes with metals also yield the possibility to get Schottky contacts that will dominate the transport physics of the nanotube-based field-effect transistor.

All this demands to explore electronic phenomena in carbon nanotubes based systems and devices with care, and to dedicate effort for showing how some particular physical properties suffer from upscaling either the tube diameter, the number of walls constituting the nanotube, or by changing the nature of contact material, and other environmental circumstances.

CHAPTER 3

Nanocomputing with Imperfections

3.1 INTRODUCTION

Before venturing into nanocomputing in the presence of defects and faults it is essential to have a framework for discussing the specific ideas being pursued in the nanocomputing community.

A defect or more specifically a manufacturing defect is a physical problem with a system that appears as a result of an imperfect fabrication process. By contrast, a fault is an incorrect state of the system due to manufacturing defects, component failures, environmental conditions, or even improper design. The faults can be categorized as:

- **Permanent:** Permanent faults are in the case of physical defects or permanent device failures during the lifetime of the systems.
- **Intermittent:** Intermittent faults may periodically stop and start, but are potentially detectable and repairable.
- **Transient:** Transient faults are due to temporary environmental conditions.

With this context, defect tolerance is the ability of a system to operate correctly in presence of manufacturing defects while fault tolerance is the ability of the system to operate correctly in the presence of permanent, intermittent and transient faults. Clearly, fault tolerance encompasses defect tolerance but also implies the ability to withstand temporary faults as well. Generally, both defect and fault tolerance requires redundancy to overcome problems within the system. This redundancy may be in terms of the replication of functions temporally or physically, or by using techniques such as error-control coding, which uses a redundancy in

the code space for the data to detect and correct faults. Often, the system must be able to reconfigure its resources to take advantage of redundant components.

Tolerance to defects and faults can be accomplished at several different levels of abstraction—the physical device level, the architectural level and the application level. The physical device level refers to specific features of nanoscale devices that provide tolerance to defects or device failures. At the architectural level, defect and fault tolerance is achieved through techniques of assembling collections of these nanoscale devices. Finally, defect and fault tolerance at the application level involves features of the computing applications themselves that allow them to operate correctly despite defects and faults in the computing systems on which they execute. Though nanoscale devices have some features that may make them tolerant to some defects and faults and though some applications themselves may have some inherent defect or fault tolerance, most of the techniques relate to the architectural level.

3.2 Nanocomputing in the Presence of Defects and Faults

To maximize the reliability of a system based on nanoscale devices for a given or minimal cost, we expect that it may require a combination of techniques at various levels of abstraction—from device to application. As an example, an extremely high degree of hardware reliability may not be necessary if the application itself can handle a certain degree of noise in its data. In such a case, it may not be cost effective to use extreme levels of redundancy when the application does not require it.

Some of the more traditional approaches are described in the following sections.

3.2.1 Triple and N-Modular Redundancy

In TMR (triple-modular redundancy), 3 copies of the same hardware are executed with common inputs so that, ideally, they produce the same outputs if all modules are defect or fault free. It is assumed that at most, any one module may either have a defect or

fault during operation, the outputs of the 3 modules are then combined by using a majority vote circuitry, which selects the output which is in a majority, i.e., produced by 2 or 3 of the models.

TMR with a single voter is shown in Fig. 3.1. This technique is fairly easy to apply to digital logic at the cost of increased circuit area and power and decreased circuit speed. One of the strengths of TMR is that it can tolerate multiple failures in a single module. Faults or defects in 2 or more of the 3 modules will cause the logic to fail.

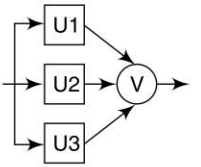


Fig. 3.1: Triple-Modular Redundancy with a Single Voter

NMR(N-modular redundancy) uses N copies of the hardware, N being an odd number so no 'tie' votes are possible. NMR is shown in Fig. 3.2. Again, the output of the hardware is determined using a majority ($floor\left(N/2 + 1\right)$ or more outputs) voting scheme.

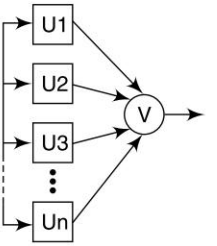


Fig. 3.2: N-Modular Redundancy

The advantage of NMR over TMR is that NMR will correctly compute the output with multiple defects or faults in $floor\,(N/2)$ of the modules. As with TMR, the NMR technique can take advantage of redundant voters to reduce the probability of system failure due to a single defective or faulty voter. Other variations of this approach include cascade NMR.

3.2.2 NAND Multiplexing

Von Neumann, the originator of TMR also developed a theory that has been termed 'NAND multiplexing' which can be used to produce the expected function in the presence of a high number of defects and faults in its components—up to a failure probability of about 0.0107 for each component. This theory was developed during an era when the reliability of individual components used in building computers was low; this required designers to consider both defect and fault tolerance in their designs.

The scheme involves replicating the function to be multiplexed N times. N wires are used to carry the signal of each input and output. Processing is performed in two stages: an executive stage and a restorative stage (as shown in Fig. 3.3). The executive stage performs the function by using the N copies of the original function unit. For each bundle of N wires, the bundle is considered "stimulated" (logic '1') if at least $(1 - \Delta) \cdot N$ of the wires are "stimulated" where $0 \leq \Delta \leq 0.5$; likewise, an input is considered "unstimulated" (logic '0') if no more than $\Delta \cdot N$ wires are "stimulated". Stimulation levels in between these two values are considered undecided and would indicate that the circuit has failed or malfunctioned. Based on the probability of function failure and the probability of correctness of the inputs to the executive stage, the percentage of wires in the output bundle that are in the correct stimulated or unstimulated states may be lower than the fractions of the input-wire bundles that were correct.

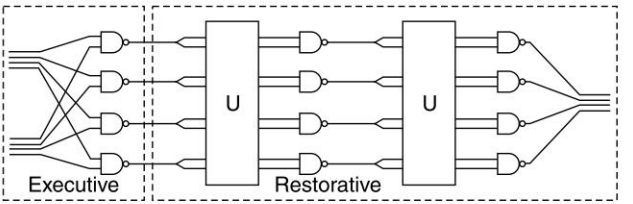


Fig. 3.3: NAND Multiplexing with $N = 4$

As a result of this potential reduction in effective signal strength, von Neumann developed the restorative stage as a part of this multiplexing technique. The purpose of the restorative stage is to increase the number of wires in the output bundle that are in the majority state, whether the bundle is in the stimulated or unstimulated state.

Several research groups in the nanotechnology community as well as elsewhere have described Neumann's technique as 'NAND multiplexing' because one of his analyses used NAND gates for both the executive and restorative stages. The interest in this theory with regards to nanocomputing is two-fold. Firstly, systems designed with this approach can withstand a high probability of failure for their components. Secondly, nano-assembly techniques could provide the number of components needed to reach the redundancy levels required for usable systems. Of course, the high cost of redundancy greatly reduces the number of 'usable' components provided by any nanoscale implementation technology. In this regard, Neumann realized the impracticality of the approach for his time and acknowledged that it might be useful in the future, saying

> "This implies, that such techniques are impractical for present technologies of componentry (although this may perhaps not be true for certain conceivable technologies of the future), but they are not necessarily unreasonable (at least not on grounds of size alone) for the micro-componentry of the human nervous systems."[1]

Subsequent analyses of NAND multiplexing have refined the bounds for the maximum fault probability for each device and the reliability of NAND multiplexing approaches.

3.2.3 Error-Control Coding

Another conventional technique to mask the presence of defects and faults is to use error-control coding. With this technique, the redundancy exists in how the data itself is encoded, not in replicating hardware having the same function. The extra bits required for error-control coding are used to help hardware distinguish between error-free data and data with errors. Ideally, if too many errors do not occur, the hardware can also use the encoding redundancy to locate and fix the data errors, if they exist. A variety of error-control codes have been developed, ranging from the well-known SEC (single-error correcting) Hamming codes to Reed-Solomon and convolutional codes. This variety of codes is used to

[1]John Von Neumann, "Probabilistic logics and the synthesis of reliable organism from unreliable components", Automata Studies, *Annals of Mathematics Studies* (34), Princeton Univ. Press, 1956.

handle a variety of different error conditions—single-bit; multiple, independent; multiple, consecutive (or burst), etc.

3.2.4 Reconfiguration

Often in conjunction with redundancy and self-assembly, reconfiguration has been explored as a defect and fault mitigation method for molecular-scale computers. The basic idea of reconfiguration is that the capability exists within a system to modify the functionality after manufacture. Reconfiguration is a widely recognized defect and fault management technique in conventional electronics. Examples at the computer system level include the ability to de-activate chip or cores within a chip upon error diagnosis; the ability to switch to spare bits for single cell failures in cache memories; to delete cache lines to map out bad bits; to bypass a cache or an entire memory card and to mark I/O resources unavailable upon diagnosis of I/O failure. Diagnostic hardware must exist to detect the failure. Once a failure has been detected, the failed unit must be by-passed and, if a redundant resource exists, the redundant unit is activated and wired in. This process is shown in Fig. 3.4 where the black node has failed and is replaced by the spare in its row.

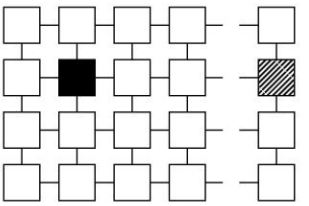


Fig. 3.4: Fault Avoidance through Redundancy and Reconfiguration

An interesting reconfiguration technique is the cell matrix. The cell matrix architecture facilitates dynamic defect and fault discovery/recovery. The CM (cell matrix) is a fine-grained reconfigurable fabric composed of simple, homogenous cells and nearest-neighbor interconnect. Like FPGAs, the CM cells are based on LUTs (look-up tables). There are no critical, irreplaceable elements whose failure could cause the entire system to fail. The homogeneity of cell structure and interconnect as well as the ability of the cell matrix to self-reconfigure makes the architecture inherently fault tolerant.

Each cell can receive configuration commands from adjacent cells, and can, in turn, send configuration commands to neighboring cells. This allows a cell or collection of cells to:

- monitor the neighbor's activities,
- detect erroneous behavior,
- disable defective neighboring cells, and
- relocate damaged segments of the circuit to other locations.

3.2.5 Fault Simulation

Few tools exist to help study defect and fault behavior in fault-prone circuits, especially at the architectural and system levels. An obvious technique to study faults in a system is to inject faults and then observe system behavior. This concept was exploited by the Space-Based Reconfigurable Computing project at the Los Alamos National Laboratory. In this work, SRAM-based FPGAs were used as computing engines in order to meet the size, weight and power constraints of satellite-based processing. The work was motivated by a need to compute in the presence of on-orbit radiation effects without resorting to fully radiation-hardened electronics. In many ways, the problem mirrors the trade-offs between nanocomputing and conventional micro-scale computing: commercial components have many times the density of radiation-hardened electronics but suffer a high degree of faults in a radiation environment.

In satellite-based processing, it is desirable to use commercial electronics for several reasons. Radiation hardened parts cost an order of magnitude more than conventional ones. The radiation hardened systems are too slow to do real-time data processing. In addition, the only fully radiation hardened FPGAs available cannot be reconfigured to hold different data processing algorithms. The available radiation-tolerant SRAM FPGAs use a configuration memory, so that the part may be repeatedly re-configured with new algorithms. In addition, the configuration data may be read out and repaired while the parts are active.

To explore the feasibility of using fault-prone high density devices for computing, a simulation—or more properly, emulation-environment has been developed. The emulator allows artificial injection of faults into an FPGA by dynamically reconfiguring the FPGA with corrupted configuration data. Fig. 3.5 illustrates the

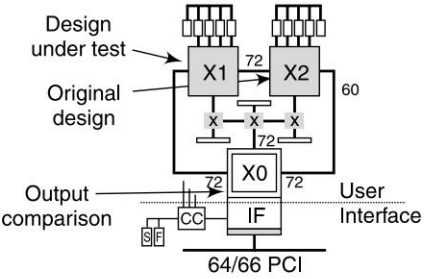


Fig. 3.5: Transient Fault Emulation Testbed

mechanism. The emulator uses 3 FPGAs, all with a common clock. *X*1 and *X*2 initially hold identical designs. As the circuit is clocked, *X*3 monitors outputs from *X*1 and *X*2 and signals the processor when outputs differ. During operation, *X*1's configuration is selectively modified while the results from *X*1 are compared to those from *X*2 on a clock-by-clock basis. Through repeated and extensive testing, it is possible to correlate a single-bit upset in the configuration data with an output error, yielding for a specific circuit the probability of output failure attributable to each bit in the configuration. This tool helps an application designer to understand the fault behavior of an application as well as where to insert redundancy or other error detection and correction circuitry to improve reliability. In a nanocomputing context, such a tool would be useful for characterizing an application's reliability for varying degrees of fault rates and types.

3.3 Defect Tolerance

Until now defects in manufacturing have primarily been the concern of process engineers, not circuit designers or architects. In the era of nanoelectronics, delivering chips which can be viewed as defect free will be prohibitively expensive. In fact, this is already happening. As an example, state-of-the-art FPGA chips with known defects can be purchased at a discount. The 'defective' chips can be used because the defects on the particular chip are determined not to affect the customer's design. In the future, defect tolerance will have to be designed at the circuit and architectural levels. Defect tolerance can be achieved by combining reconfigurable fabrics with new tools. Reconfiguration provides defect tolerance by configuring the desired circuit around the defects, thus creating

a reliable system from an unreliable substrate. Before the fabric is shipped its defects are mapped. When the chip is used, the desired circuit is configured around the defects. The two main challenges are to develop architectures and tools which can find the defects quickly and then—in the field—quickly place and route (P & R) circuits around the defects. Final P & R needs to be done in the field so that a single configuration can be shipped for all devices, in spite of the fact that each device will have a different set of defects. Such defect-tolerant reconfigurable fabrics made from future-generation technologies are referred to as VLRFs.

The high-defect densities in VLRFs require a completely new approach to manufacturing computing systems. No longer will it be possible to test a chip and throw it away if it has only a handful of defects, since we expect that every chip will have a significant number of defects. Instead, we must develop a method to use defective chips. The ability to tolerate defects in the final product in turn eases the requirements of the manufacturing process. In some sense, this introducers a new manufacturing paradigm: one which trades off post-fabrication programming for cost and complexity at manufacturing time.

Modern memory chips and hard drives are able to achieve some degree of defect-tolerance by leveraging redundancy and post-manufacturing adaptiveness that allows them to substitute spare, working resources for defective ones. In large, high-density memory chips, extra rows and columns are built into the chip. After manufacturing, a testing phase locates failing rows and columns, and these are replaced by the spare rows or columns by using a laser to burn a bypass path. Some modern operating systems go a step further: when they detect a memory error, a testing tool is used to detect the failing rows and columns, and these are then replaced by the spare rows or columns by using a laser to burn a bypass path. Some modern operating systems go a step further: when they detect a memory error, a testing tool is run to detect the failing memory regions; the operating system then remembers not to use those regions when it stores data to memory. With VLRFs, techniques based on simple row or column replacement will not be sufficient. It is unlikely that a portion of the fabric of any appreciable size will be defect free. Moreover, these devices are being projected as a replacement, not just for memories but also for logic, where simple techniques such as row-replacement will not work since logic is less regular.

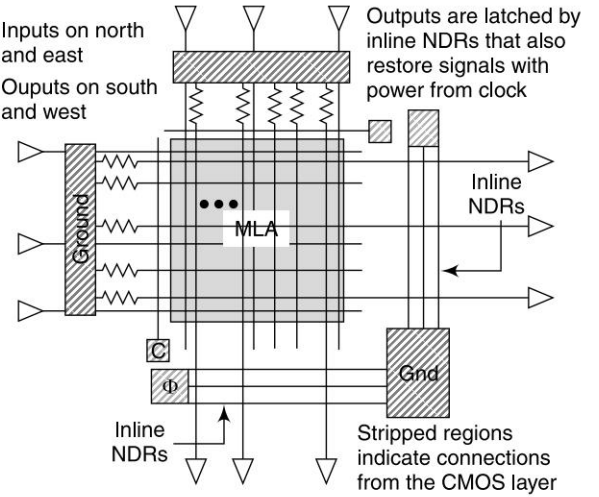
One approach to achieve defect tolerance would be to use techniques developed for fault-tolerant circuit design. Such circuit designs range from simple ones involving triple-modular redundancy or other relatively simple forms of majority logic to more complex circuits that perform computation in an alternative, sparse code space, so that a certain minimum number of errors in the output can be corrected. A novel architectural approach for nanoscale computing is based on using probabilistic models of computation based on Markov Random Fields and seeks to maximize the probability of correct Boolean state configurations by minimizing the entropy of a suitable energy distribution that depends on neighboring nodes in the Boolean network. A natural solution for achieving defect tolerance in VLRFs is suggested by looking at reconfigurable fabrics, for example, field-programmable gate arrays (FPGAs). An FPGA is an interconnected set of programmable logic elements. Both the interconnect and logic elements may be programmed, or configured, to implement any circuit. The key idea behind defect tolerance in FPGAs is that reconfigurability allows one to find the defects and then to avoid them.

3.3.1 Nanotechnology and Molecular Circuits

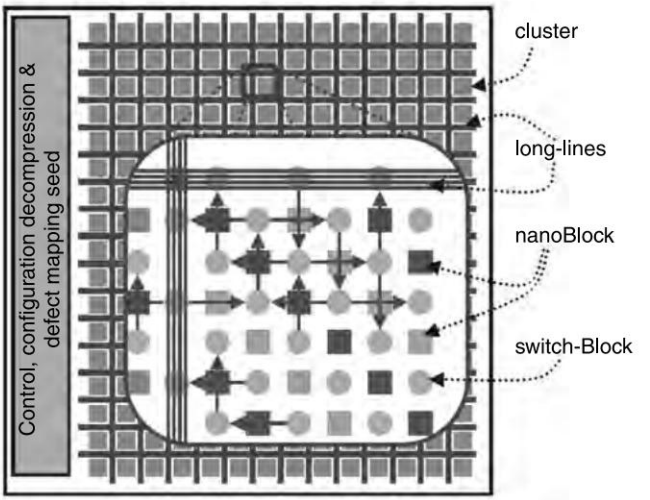
Significant progress has been made in developing molecular scale devices. Molecular scale FETs, negative differential resistors, diodes and non-volatile switches are among the many devices that have been demonstrated. Advances have also been made in assembling devices and wires into larger circuits. In addition to the increases in density, molecular electronics also promises to introduce devices with characteristics not found in silicon-based systems. One example is the non-volatile programmable switch, which holds its own state without using a memory cell, and can be configured using signal wires; such a switch has the same area as a wire-crossing. This contrasts with reconfigurable fabrics made today by using standard CMOS, where a reconfigurable switch has the same area as a memory cell and is 2 to 3 orders of magnitude bigger than a wire-crossing.

The requirements imposed by the manufacturing process as well as the area advantages presented by the molecular reconfigurable crosspoints have prompted a number of researchers to propose

regular, mesh-based reconfigurable architectures for VRLFs. One such architecture is the *nanoFabric*[2] (Fig. 3.6), which is fine-grained reconfigurable and is designed to overcome the limitations of



(a) A *nanoBlock* is the smallest configurable logic block of a *nanoFabric*



(b) A *nanoFabric* consists of many regularly tiled *nanoBlocks*, interspersed with routing resources

Fig. 3.6: The *nanoFabric*, an Example of a Reconfigurable CAEN-based Architecture

[2]Seth Copen Goldstein and Mihai Budiu, "nanoFabrics: Spatial Computing Using Molecular Electronics", Proc. of The 28th Annual International Symposium on Computer Architecture, June 2001.

self-assembly of molecular scale components. The basic unit of the *nanoFabric* is the programmable molecular switch, which can be configured either as a diode or an open switch. This molecular switch eliminates much of the overhead needed to support reconfiguration in traditional CMOS circuits, since the switch holds its own state and can be programmed without extra wires. These switches are organized into 2-D meshes called *nanoBlocks*[3], which can be configured to implement logic functions. The *nanoBlocks* in turn are organized into clusters which can be connected by using long lines which run between the clusters. Within a cluster, each logic block is connected locally to 4 neighbors. In addition to the functionality of the logic blocks, the connections to the interconnect are also all programmable.

3.3.2 Reconfigurable Hardware

RH (reconfigurable hardware) shares features of both custom hardware and microprocessors. It not only offers the promise of increased performance but it also amortizes the cost of chip manufacturing across many users by allowing circuits to be configured after they are fabricated. Its computational performance is close to custom hardware, yet, because it is programmable, its flexibility approaches that of a processor. Because of their enormous potential as computational elements, there has been much research into using RH devices for computing.

Differences between Reconfigurable Hardware and Processors

1. **Unbounded computational bandwidth:** A microprocessor is designed with a specific number of functional units. The computational bandwidth of a processor is thus bounded at the time of manufacturing. Moreover, it is unusual for a processor to reach its peak performance, because the parallelism available in the program rarely has the exact same profile as

[3]Seth Copen Goldstein and Mihai Budiu, "nanoFabrics: Spatial Computing Using Molecular Electronics", Proc. of The 28th Annual International Symposium on Computer Architecture, June 2001.

the available functional units. In contrast, RH can support a virtually unbounded number of functional units. Not only can highly parallel computational engines be built, they can exactly fit the application requirements, since the configuration is created post-fabrication.

2. **Unlimited register bandwidth**: Another subtle but important difference between a processor and an RH device is the way they handle intermediate computation results. Processors have a predetermined number of registers. If the number of manipulated values exceeds the number of registers, then they have to be spilled into memory. Additionally, the fixed number of internal registers can throttle parallelism.
3. **Full out-of-order execution:** While superscalar processors allow instructions to execute in orders different from the one indicated by the program, the opportunity to do so is actually restricted by several factors, such as limited use window, generic exception handling and structural hazards. None of these constraints exists in RH implementations.

Other advantages of reconfigurable hardware include

1. They exploit all of an application's parallelism: task-based, data, instructional, pipeline and bit-level.
2. They create customized function units and data-paths, matching the application's natural data size.
3. They eliminate a significant amount of control circuitry.

3.3.3 Very Large Reconfigurable Fabrics (VLRFs)

There are two scenarios in which VLRFs can be used:

1. **Factory-programmable-devices:** Factory-programmable devices are configured by the manufacturer to emulate a processor or other computing device. User applications treat the device as a fixed processor (or potentially as a small number of different processors). Processor designers will use traditional CAD tools to create designs by using standard cell libraries. These designs will then be mapped to a particular chip, taking into account the chip's defects. A finished product is therefore a VLRF chip and a ROM containing the configuration for that chip. In this scenario, the configurability of the VLRF is used only to accommodate a defect-prone manu-

facturing process. While this provides the significant benefits of reduced cost and increased densities, it ignores much of the potential in a VLRF. Defect tolerance and limitations of the manufacturing process require that a VLRF be reconfigurable, so it can exploited to build application-specific processors.

2. **Reconfigurable computing devices:** Reconfigurable fabrics offer high performance and efficiency because they can implement hardware matched to each application. However, this extra performance comes at the cost of significant work by the computer. A conservative estimate for the number of configurable switches in a 1 cm^2 VLRF including all the overhead for buffers, clock, power, etc., is in the order of 10^{11}. Even assuming that a compiler manipulates only standard cells, the complexity of mapping a circuit design to a VLRF will be huge, creating a compilation scalability problem. Traditional approaches to place-and-route in particular will not scale to devices with billions of wires and devices.

A two-fold approach to defect-tolerance is to first construct a map of the defects. Then, while configuring the device to implement a particular circuit, the defects are avoided by using only the good components of the device. The three requirements of a programmable device for this approach are:

- it must be programmable,
- it must have a rich fine-grained interconnect, and
- it should allow us to implement a particular logic function in many different ways.

All of these attributes are necessary for both defect detection and defect avoidance. During defect detection, the different test circuits on the device are reprogrammed. Each different instance of a test structure gives us information about different sets of components on the device. The latter two attributes are most necessary during defect avoidance. They allow a particular circuit to be implemented without requiring us to use any of the defective components. Fig. 3.7 shows the inter-operability of a new set of tools, such as fast testers to generate defect maps and place-and-route tools to convert circuit description into fabric configurations taking into account the defect map for the fabric.

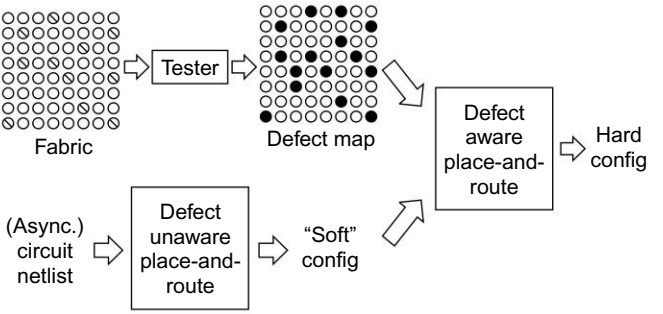


Fig. 3.7: The Tool-flow for using Molecular Reconfigurable Fabrics for Computation

3.3.4 Testing

VLSI testing is a much-studied area of research. A large number of testing strategies and design methodologies have been proposed over the years to improve the speed and accuracy of VLSI testing, and hence to enhance manufacturing yield. Most such techniques have been designed around the assumption that a single, to at most very few faults exist in the portion of the circuit under test. The problem we wish to tackle is significantly harder, since a large fraction of the resources under test may be defective. A key advantage we enjoy over traditional VLSI testing is that since the fabric is reconfigurable, we have the freedom to implement a circuit of choice to carry out the testing, rather than being limited to passing input vectors to the fabricated circuit.

The approach adopted here configures sets of fabric components into test circuits whose output is used to infer the defect status of individual components. Unlike the dedicated built-in self test structures often incorporated in current digital designs, the test circuits placed on the fabric during this self-diagnosis phase will utilize resources that will be available later for normal fabric operation; our expectation is that testing in this way should not require any dedicated fabric resources, and so supporting such a testing methodology should not incur either an area or a delay penalty.

As an example, consider the situation in Fig. 3.8(a). Five components are configured into one test-circuit, so that defects in one or more circuit components would cause the circuit output to be incorrect. By comparing the circuit's output with the correct result, it can be determined if any of the circuit's components were

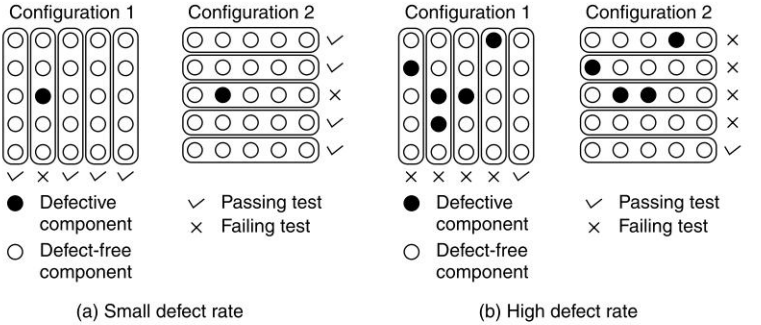


Fig. 3.8: Identification of Defective Components

defective. In the first run, the circuits are configured vertically, and test circuit 2 detects a default. In the next run, the circuits are configured horizontally, and test circuit 3 fails. Since no other errors are detected, we can say that the component at the intersection of these 2 circuits is defective, and all others are good.

However, if the defect rate is higher, as it is likely to be in VLRFs made with future-generation technologies, this method no longer produces very good results. For example, in Fig. 3.8(b) with more faults than Fig. 3.8(a), only one vertical and one horizontal circuit return the correct results and we gain no information about defect locations in the rest of the fabric. Also, the test circuits used will have a much larger number of components than in the simple examples shown here. This will be true for two reasons:

1. Controlling and observing a small set of resources in the interior of the fabric will require fabric interconnect resources, which may themselves be defective: an incorrect circuit result would mean a defect in the circuit's parts, or in the wires and switches used to observe the circuit's output. These interconnect resources will therefore have to be considered part of the 'test circuit', thus imposing a limit on how small these circuits can be made.
2. For high-density fabrics, small test circuits would imply a long testing time, so much so that the fabrics may become economically unviable.

Pseudo-code of a testing algorithm by Mishra and Goldstein[4] is presented here. It consists of two phases: the probability assignment phase (lines 1-10) and the defect location phase (lines 11-22). The

[4]Mishra and Goldstein, "Defect Tolerance at the End of the Roadmap", Proc. of the International Test Conference (ITC), Charlotte, NC, 2003.

probability assignment phase assigns each component a probability of being defective, and discards the components which have a high probability. This should result in a large fraction of defective components being identified and eliminated from further testing. The remaining components are now likely to have a small enough defect rate that they can be tested in the defect location phase by using the simple method employed above to identify all the defect free components. In each phase, the fabric components are configured into test circuits in a particular orientation, or tiling; since each circuit uses only a small number of components, many such circuits can be configured in parallel, or tiled, across the fabric. For example, the circuits in Fig. 3.8 are arranged in two tilings—vertical and horizontal.

```
    // Probability Assignment Phase
1   mark all fabric components not suspect
2   for iteration from 1 to N1 do
3       while probabilities not stable do
4           for all fabric components marked not suspect do
5               configure components into type-I test-circuits
                using a particular tiling
6               compute defect probability for each component
                using circuit results from current iteration
7           done
8       done
9   mark components with high defect probability as suspect
10  done
    // Defect Location phase
11  for iteration from 1 to N2 do
12      while results improve do
13          for all fabric components marked not suspect or
            not defective do
14              configure components into type-2-test-circuits
                using a particular tiling
15              for all circuits with correct output do
16                  mark all circuit components not defective
17              done
18          done
19      done
20  mark some suspect components not suspect
21  done
22  Mark all remaining components as defective
```

Once the results of the test circuits have been obtained, they are used to determine the probability of each individual component being defective. It can be done by sorting analysis or Bayesian analysis.

3.3.4.1 Sorting Analysis

Let a component c be part of n different circuits. Based on the result of these n circuits, we calculate a fault-value for the component, as follows: if the test circuits are counter circuits, the fault-value is simply the sum of the number of defects in each of the n circuits. If the circuit is an LFSR-based none-some-many circuit, numerical weights are assigned to each result (for example, 2 to many defects, 1 to some and 0 to none) and sum up all n weights for the component. Once this calculation has been performed for all the components under test, they are sorted according to their fault-values and components with higher fault-values are assigned a higher probability of being defective. This method involves simple calculations and places no specific restrictions on the shape or nature of the tilings.

3.3.4.2 Bayesian Analysis

Again, let a component c be a part of n different test circuits. Let p be the *a priori* known defect rate in the fabric, obtained through some initial testing or from knowledge of the manufacturing process. Let $a_1, a_2, \ldots a_n$ represent numerical results for each of these circuits (these can be actual defect counts for *counter* circuits, or numerical weights for the *none-some-many* circuits as described above). We need to find the posterior probability of component c being defective given our knowledge of the circuit results. Let A be the event that c is good, and let B be the event of obtaining the circuit results that we have obtained for the n circuits. Therefore, we need to find

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A \cap B)}{P(A \cap B) + P(\bar{A} \cap B)}$$

If c is the only component that the n circuits share, this equation simplifies to the following:

$$P(A \mid B) = \frac{1}{1 + \dfrac{(1-p)^{n-1} n^n}{p^{n-1}(n-a_1)(n-a_2)\ldots(n-a_n)}}$$

This equation is solved for each component to obtain its probability of being good (the probability of being bad, which is required by the algorithm, is simply this value subtracted from 1).

A short testing time is crucial for the usability and low cost of these fabrics, so it is important to ensure that the testing procedure scales with fabric size. The reconfigurability of the fabric can be leveraged to reduce the time spent on an external tester significantly. Once a part of the fabric is tested and defect-mapped, it can be configured to act as a tester for the other parts. Also, there is nothing to prevent us from having multiple testers active simultaneously. In such a scenario, the first area to be tested is its adjacent ones, which further test their adjacent ones and so on, and the testing can move in a wave through the fabric. For large fabrics, multiple such waves may grow out from different externally-tested areas. Now, as the fabric size increases, testing time grows linearly with the distance this wave has to traverse through the fabric, which is proportional to the length of the fabric's edge, and to the square root of the number of components of the fabric. Fig. 3.9 shows *nanoFabric* testing in a wave-like manner.

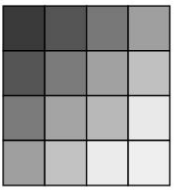


Fig. 3.9: Testing a *nanoFabric* in a Wave-like Manner

3.3.5 Placement and Routing

The place-and-route process generates fabric configurations that avoid using defective components. The place-and-route tools have to deal with the large size of the fabrics, as well as with the fact that each individual fabric has a unique set of defects and therefore requires some effort from the place-and-route tools to configure around its particular set of defects. There are two stages in the process:

- An initial, fabric-and-defect-independent step (DUPER, Defect Unaware Place and Route) which includes technology mapping and global placement and routing, and generates

what we call a 'soft' configuration. The 'soft' configuration is guaranteed to be place-and-routable on a fabric that has a defect level below a certain threshold.

- A final, defect-aware step (DAPER, or Defect Aware Place and Route) that transforms the 'soft' configuration into a 'hard' configuration, taking into account the target fabric's defect map. At this step the final place-and-route is performed, using only non-defective resources. If the defect-map is imprecise or coarse-grained, this step requires the use of an on-the-fly defect mapper to pin-point the precise location of defects. The hard configuration is specific to each fabric.

In general, the place-and-route tool should be able to take in a map of defects (as generated by our testing tool) and produce fabric configurations that avoid using these defective resources. The tool should address the following constraints:

1. As opposed to current hardware design methodologies where the place and route tools are run once per design, the approach here requires running them once per fabric. This is because each fabric would have a defect map that is unique to it, requiring some place and route effort to avoid the defects. To make this process practical, ways have to be found to minimize the per-fabric effort, while still achieving the desired functionality.
2. Adjusting the design to avoid defects may have an unpredictable effect on the timing of various signals. It has to be ensured that in spite of these adjustments, the implemented circuits are able to meet all the timing and functionality guarantees.
3. Since VLRFs are envisioned for general-purpose computation rather than only as ASIC replacements, the execution time of the place-and-route tool should be comparable to software compilation and installation runtimes—measured in seconds and minutes, rather than in hours and days, as is the case for current hardware design or reconfigurable logic tool flows. This should be true in spite of significantly more available resources compared to current hardware design tools.
4. For significantly high defect rates, the complete defect map may not be available. This is because storing a detailed defect map for the full fabric may require resources comparable to those available on the fabric itself.

3.4 Towards Quadrillion Transistor Logic Systems

Hardware and logic designs have come a long way. The transistors used in a modern single-chip CPU are several hundred million times smaller than the original transistor built in 1947. If a contemporary CPU were built with the original transistor technology, it would take up a space of roughly one square kilometer. Current ways to produce logic designs pack many more transistors into hardware than their predecessors ten years ago, and ten to twenty years from now there may be ways to produce hardware devices with a billion times more transistors or switches.

There has been and continues to be strong economic incentive for miniaturization of logic designs and electronics. Although for some products, this has been used to simply reduce the footprint, designers have also been freed to create larger and more complex designs as transistor density has increased. Technical breakthroughs over the next ten to twenty years could come gradually, but may instead exhibit sudden leaps in progress as problems are solved and discoveries are made. There are several interesting research questions at this juncture that are appropriately addressed both in academics and industry. Some of them are:

1. Getting involved in the analysis of nascent switch-production methods to model how well they fit the engineering requirements of integrated circuits and what the outcomes could be from using a particular new method. Considerations include the number of switches per unit area or volume inside the device (density), total number of switches inside the device (volume), operating condition limitations, operating speeds, power requirements, production costs, and the reliability of production methods and of the product during its lifecycle.
2. Coming up with designs, and design tool capacity, for effective use of 10^{17} transistors or switches, such as designs that will scale up gracefully or even seamlessly as density increases, and ways to produce designs that readily lead to production of larger switch counts.
3. Coming up with more powerful design and verification tools to handle logic designs with many orders of magnitude greater scope and complexity.

4. Coming up with more flexible processes for the product path, from definition of a new product's requirements, through logic-design, test and verification, and implementation in hardware. Processes should be flexible enough to permit things like co-development of design, test and build, development of design and verification methods and blurring of test and build into a more iterative process that accounts for the imperfect nature of a build process.

3.4.1 Cell Matrix

The above mentioned abilities—through molecular engineering, quantum dots, processes evolved from CMOS, etc., are still under development, and a picture of the manufacturing techniques that will be used at the 1 to 10 nanometer scale is not yet fully formed. However, from progress to date, a number of independent researchers have begun to describe what is attainable from nanoelectronics and desirable from a computer architecture standpoint. These descriptions read like a description of the Cell Matrix™ architecture: a regular, homogeneous, three- or two-dimensional array of simple reconfigurable elements with local-only interconnections, fault tolerance, dynamic fault handling, and better than linear configuration times.

If cell matrix technology is combined with nanotechnology, it benefits both technologies. The construction of Cell Matrices™ lets nanotechnology enter the digital circuit and system market sooner, because it permits the construction of a single, simple physical structure that is then electronically differentiated into any desired digital circuit or system through software. Nanotechnology benefits cell matrix technology because it provides inexpensive manufacture of cell matrices that contain enough cells to be useful for solving the kinds of large, difficult problems that people would love to solve today but cannot do so with today's technology. As an example, we have designed high performance parallel searchers that trade materials for time: much faster solutions by using a lot more materials. But to build the machines we have designed, we need significantly less expensive materials with a unit size orders of magnitude smaller than what is available today, and to end up with a machine with, say, 10^{18} (a trillion trillion) components that takes up a reasonable amount of space and that can be configured

or set up quickly, we need cells arrayed in three dimensions rather than just two.

The cell matrix architecture acts as a bridge between nanotechnology and the world of electronic components: nanotechnology can be used to build cell matrix hardware; cell matrices can then be used to implement electronic components, circuits, and systems, and this three-step process is significantly easier than going straight from nanotechnology to arbitrary electronic components, circuits and systems.

A bridge may not be needed in the distant future. Humans may figure out how to build any electronic component they need in an extremely compact representation at the atomic scale. In the meantime, the cell matrix architecture makes it possible to make steady progress toward atomic scale computing elements and to make useful and saleable products sooner.

For certain types of hardware and for certain classes of problems, the physical features of cell matrix hardware, such as low power requirements and ability to withstand manufacturing defects and later damage, will make the use of cell matrices preferable over literal physical translations of the component. And for other classes of problems such as Avogadro scale computers—that is, computers that efficiently use on the order of 10^{23} components—both the physical features such as the need for little external interconnection, and the functional features of the cell matrix architecture, such as fast configuration times and dynamic, self-configuring hardware, cause the cell matrix hardware to become an end in itself rather than a means, or bridge.

The Cell Matrix^{TM} is an architecture for a novel type of reconfigurable hardware system. Similar to an FPGA, the cell matrix is composed of a large number of simple reconfigurable elements (cells). Unlike most FPGAs though, there are essentially no internal structures besides the cells themselves. Moreover, each cell is connected to only a small set of neighboring cells. These two characteristics mean that the cell matrix architecture is inherently fault isolating: defects in a cell will generally have limited scope.

In contrast to an externally-controlled FPGA, the cell matrix is a self-configurable system. This means that the cells within the system are able to analyze and modify other cells, without intervention or guidance from outside the matrix. The cell matrix architecture does not specify a particular topology or the system's cells.

Cells may be three-sided, four-sided, six-sided, or any other number of sides greater than two. Cells may be interconnected in two-dimensional or three-dimensional topologies—topologies greater than three dimension are also possible.

Regardless of these specifics, cells and their resultant matrix all operate along identical principles. Each cell has two inputs on each side, labeled *D* and *C*. Each cell has a corresponding set of outputs (*D* and *C*). Cells are interconnected according to the matrix-wide definition of a neighborhood, with inputs and outputs connected in the obvious fashion. Additionally, each cell contains an internal lookup table (LUT). The LUT maps every possible combination of *D* inputs to a set of *C* and *D* outputs.

Each cell within the matrix operates in one out of two ways, depending on the *mode* in which the cell is operating. If a cell is in *D mode*, it continually samples its *D* inputs, looks up a set of outputs in its internal LUT, and sends those LUT values to its *C* and *D* outputs. This happens continuously, without any clocking or synchronization.

If, instead, a cell is in *C* mode, it samples its *D* inputs as specified by a system-wide clock, and loads the sampled bits into its internal LUT. Moreover, before a LUT bit is overwritten, it is sent to the cell's *D* outputs. *C* Mode is thus the mode in which a cell's LUT is read or written, while *D* Mode is the mode in which a cell is able to perform data processing functions via its LUT.

Finally, a cell's mode is specified by its *C* inputs. If any *C* inputs are asserted, then the cell is in *C* Mode. Otherwise the cell is in *D* Mode. Three key consequences of this are:

1. The mode of any cell *C*1 can be specified by any of its neighbors *C*2 (since *C*2 has a *C* output connected to one of *C*1's *C* inputs);
2. The mode of a cell can change over time, since the value of its *C* inputs can change over time; and
3. A cell's mode is more or less independent of that of other cells—it is not a system-wide property, but a property of each cell.

The interaction of *D* and *C* mode cells thus allows cells within the matrix to read, modify and write other cells' LUTs. The LUTs can be processed as ordinary data, shared among cells as data, and then later used to configure cells, i.e., treated as code. This can be used to yield a number of powerful functions, including the testing of cell behavior, the creation of dynamic circuitry under

the direction of the matrix itself, and the configuration of a large numbers of cells in parallel. Figures 3.10 and 3.11 show the structure of hardware layer and logic processor, respectively.

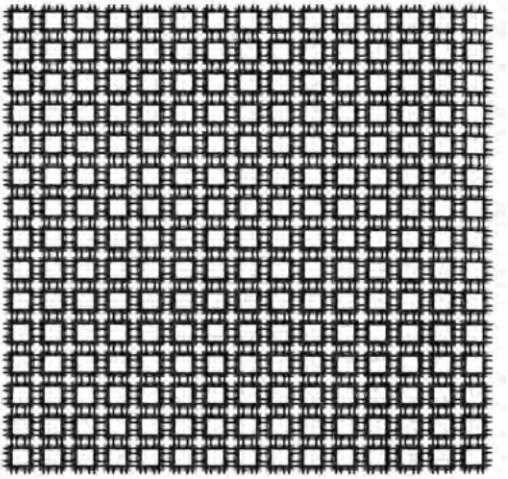


Fig. 3.10: Hardware Layer Y Definition

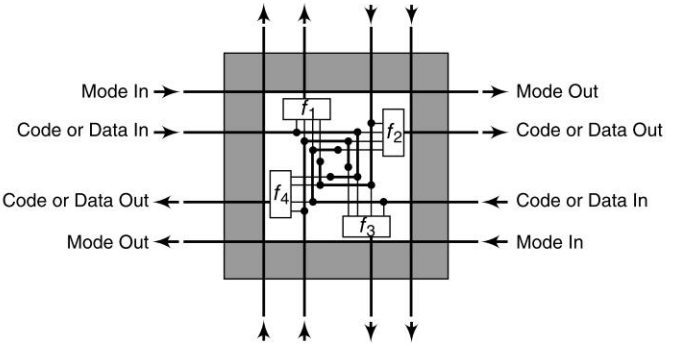


Fig. 3.11: Structure of Each Logic Processor in the Hardware Layer Y in Fig. 3.10

3.4.2 Overcoming Manufacturing Defects

Manufacturing defects must be refined and improved, but perfection is a different goal to hold, particularly when the desire to continue to miniaturize switches persists in driving manufacturing onward to new challenges. The term manufacturing defect is used for those failures that are turned up during initial testing of the hardware. All other hardware failures that turn up later in the product

life cycle are referred to as operating errors. There are at least five conceivable ways to handle defects in the construction of logic designs in hardware.

- Discard any hardware that is not perfect.
- Repair, remove, and replace the individual defects.
- Build redundancy into the hardware and a means to use only perfect resources within the hardware.
- Use logic designs for Layer *X* that can function despite defects or runtime faults.
- Perfect the manufacturing technique so that it creates no defects.

An approach adopted by Durbeck and Macias[5] is to take two layers to a logic design, the normal logic design layer *X*, and the lower level implementation layer *Y*. It is possible to incorporate redundancy into the logic systems design and implementation, as it is done today for systems onboard satellites and space-ships, via modifications to the logic design layer *X*. The lower layer *Y* can also be used quite effectively to safeguard products against manufacturing defects. The basic mechanism offered here for safeguarding a logic system's perfect function against manufacturing defects is either to enlarge the system hardware to include redundant copies of resource, or to go in and repair, remove, or replace defects. Layer *Y* can achieve both of these models.

The benefit of the approach is that some measure of both fault tolerance and redundancy is automatically provided for all logic designs *X* by the nature of layer *Y*. *Y* contains low-cost redundancy already, because the logic cells themselves are resources that can be used to implement transistors, wires, flip flops, truth tables, gates, logic blocks, state machines or any other digital circuit component. If one cell is bad, a design layout tool can use the one next to it. This is a great way to provide redundancy to put in beforehand, and to where exactly focus the extra resources. Instead resources are pulled from a general pool and used as cleverly as the design layout tool or diagnostic system is designed to utilize them.

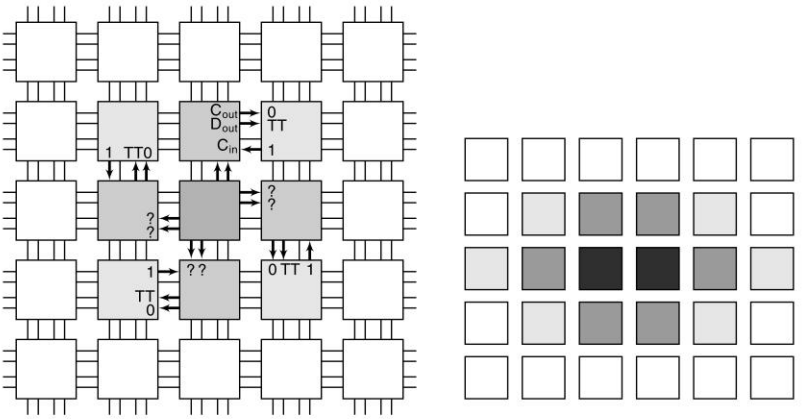
One way to achieve this design flow process that can lay out perfect systems on top of an imperfect *Y* layer is to implement

[5]Durbeck and Macias, "The Cell Matrix: An Architecture for Nanocomputing", *Nanotechnology*, Vol. 12, Institute of Physics Publishing, Bristol.

something analogous to what is used for memory, or disks today, such as SCANDISK type of process that checks each region of hardware and constructs a map of all bad regions, which is then used during writes to strictly prevent the copying of data into bad regions of hardware. The system that performs the scanning must itself be non-defective, and it should have the goal of marking as little extra hardware off-limits as is possible.

If the intention is to use hardware despite the presence of manufacturing defects, then no defective hardware can be used in the layout of logic designs, and the defects present in the hardware must not be allowed to affect the logic design's function. How we approach this in the *X*-*Y* approach is that the layout of layer *X* on layer *Y* must be one that avoids placing any logic or wires within the defective regions. This preventive step could be done by either physically removing the defects from layer *Y*'s hardware, or by logically removing them from layer *Y*'s functioning. This effectively stops the spread of defects and guarantees that they will not alter the operation of logic design *X* in any way. This situation is a promising one, because it means that defects are naturally limited in their locality and effects.

Figure 3.12(a) shows how a guard wall is erected. It is assumed that analysis of each cell for defects has already been performed, and the centre cell has been determined to be defective. Layout tools need to have knowledge of both the defects and the guard



(a) The construction of a guard wall

(b) Example of a guard wall for a 2-cell-wide defect

Fig. 3.12

walls and are responsible for ensuring that they do not attempt to lay down a part in this region: if they try, they will fail, because the guard wall is already in place and irrevocable. They may use only the unutilized resources and sides within the outermost level of the guard wall. The defective cell is logically isolated from the functioning of the rest of the matrix by ensuring that its outputs will be explicitly ignored. This is done by putting its immediate neighbors, the plus shape of dark gray cells around the centre, permanently into *C* mode, which causes them to send out low signals on the rest of their output lines no matter what signals they receive. They are put into *C* mode by the light gray cells, as indicated by the 1 on the C_in lines of the dark gray cells. This strategy completely contains the signals from the defective centre cell. It uses up the four neighboring cells completely, and the edge of one of each neighbor one level further out in the adjacency that is used to send the C mode signal. Figure 3.12(b) shows that the guard wall grows compactly around the defect with a two-cell defective region.

Black cells are defective, dark gray cells are used to isolate the defective cells' signals, and light gray cells are used to orchestrate this isolation

Before defects can be examined and explicitly walled off, they must be found and pinpointed as precisely as possible. Because cells have the ability to exchange data with other cells, as well as the ability to change a neighboring cell's function, it is possible for one cell to perform a series of tests on a neighboring cell. For example, a cell can be configured to always output 0, and by then verifying that the output is zero, one can confirm the output is not stuck-at-1. Similarly, a cell can be configured to always output 1, to detect a stuck-at-0 fault in the output. By configuring a cell as a wire that outputs its input, one can partially verify the cell's configuration mechanism. Configuring a cell as an inverter allows one to check for a short between the input and the output of the cell, as well as to further test the cell's configuration mechanism. More complex test patterns can be used to further exercise the cell under test.

Another approach to manufacturing defects is not to accept them and handle them, but to go back in after manufacture and testing, and physically remove or replace them, or to affect repairs to them. This second defect-handling process could be done by the same

process that built the hardware, by a sort of "try again" model, or by another process that specializes in removal and repairs, or by a combination thereof.

Efforts to perfect the manufacturing technique even if it does not meet with complete success are also desirable because they result in the highest density and volume for a fixed manufacturing process. This effort may also help to reduce the operating errors. The construction of a simpler target is often easier to perfect. Accordingly, the construction of the *Y* layer is easier to perfect because it is simply a task of repeatedly constructing the same small design for the logic cell and repeating the same interconnection pattern throughout.

CHAPTER 4

Reliability of Nanocomputing

Silicon-based devices are fast approaching their practical limits and Moore's Law will no longer be sustainable. As a result, many alternatives to silicon-based devices are being explored for the basis of developing new nanoelectronic systems. In the process, it is expected that the past approach of using global interconnections and assuming error-free computation may no longer be possible, thereby presenting new challenges to computer engineers. It is likely that nanoscale computing will be dominated by communication, where processing is based on redundant and adaptive pathways of error-prone connections.

Till date, the fabrication of nanocircuits has been limited to a few devices intended to demonstrate simple logic or memory operations. There are no actual data to measure the characteristics of large networks of devices. However, it is possible to pose two likely characteristics that will have to be confronted in the development of computational architectures that use these devices.

1. **A high and dynamic failure process:** It can be expected that a significant fraction of the devices and their interconnections will fail. These failures will occur both during fabrication and at a steady rate after fabrication, thus precluding a single test and repair strategy.
2. **Operation near the thermal limit:** As device sizes shrink, the energy difference between logic gates will approach the thermal limit. Thus, the very nature of computation will have to be probabilistic in nature, reflecting the uncertainty inherent in thermodynamics.

Nanocomputing architecture research has taken two approaches. The first approach simply increases existing machine resources while the second approach uses modular and hierarchical architectures to improve the performance of traditional single-thread

architectures. In particular, the highly regular, locally connected, peripherally interfaced, data-parallel architectures of the second approach offer a good match to the characteristics of nanoelectronic devices.

4.1 Markov Random Fields

An approach presented here by Bahar, et al.[1], is based on Markov Random Fields (MRF). The MRF provides a formal probabilistic framework so that computation can be directly embedded in a network with immunity to both device and connection failures. Since logic states are computed probabilistically, the computation is also robust to the logic signal fluctuations that will arise as the operation approaches the thermal limit of computation. Furthermore, the MRF is general and directly programmed without learning.

The MRF has been widely used in computer vision, physics, computational biology, and communications and is proposed as a model for uncertain and noisy computation. The MRF represents the relationship $X = \{X_1, X_2, \ldots X_n\}$. Each X_i can take on values from a range set L. In some MRF treatments, the random variables are called sites and the set L is called the label set.

The joint probability of variable assignments is denoted as,

$$p(x_1, x_2, \ldots, x_n) = p(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) \quad (4.1)$$

where, $x_i \in L$. The conditional probability of a particular variable, say x_2, is in general,

$$p(x_2 | x_1, x_3 \ldots, x_n) = \frac{p(x_1, x_2, \ldots, x_n)}{p(x_1, x_3, \ldots, x_n)} \quad (4.2)$$

In this case, the random variable set $\mathbf{X}$ is completely statistically independent. If all the random variables are independent then,

$$p(x_1, x_2, \ldots, x_n) = p(x_1)p(x_2)\ldots p(x_n), \text{ and} \quad (4.3)$$

$$p(x_2 | x_1, x_3 \ldots, x_n) = p(x_2) \quad (4.4)$$

The MRF defines the concept of a neighbourhood, N_i to represent the conditional dependence of a variable, X_i, on a subset of $\mathbf{X}$. The neighborhood can vary from complete dependence (the

[1]Bahar et. al., "*A* Probabilistic-Based Design for Nanoscale Computation*"*, *Nano, Quantum and Molecular Computing*, Kluwer Academic Publishers, 2004.

entire set **X**) to complete independence (the null set). As an example, suppose the neighborhood of X_2 is $N_2 = \{X_3, X_5, X_7\}$. Then,

$$p(x_2 | x_1, x_3 \ldots, x_n) = p(x_2 | x_3, x_5, x_7) \tag{4.5}$$

The formal definition of a Markov random field can now be stated. Let **X** be a family of random variables defined over a set of values from L. **X** is said to be a Markov random field on L with respect to a neighborhood system N if and only if the following two conditions hold :

$$p(x_i) > 0, \quad \nabla X_i \in X \qquad (\textit{Positivity}) \tag{4.6}$$

$$p(x_i | \{X - x_i\}) = p(x_i | N_i) \qquad (\textit{Markovianity}) \tag{4.7}$$

A remarkable key property of Markov random fields is that $p(x_i | N_i)$ can always be expressed in terms of a function of the cliques formed from a site and its neighborhood. In this context, the sites are considered to be nodes in a graph and the conditional dependencies between nodes are the graph edges. This graph interpretation of a MRF neighborhood is shown in Fig. 4.1. In this interpretation, the edges of the graph indicate elements of a neighborhood. The influence of each clique on the probability of the entire set of random variables can be expressed in terms of a set of terms, U_c called clique energy functions. The variable, c, indexes the cliques over the entire set of nodes, C. The use of the energy concept relates to the historical origins of the MRF model in physics.

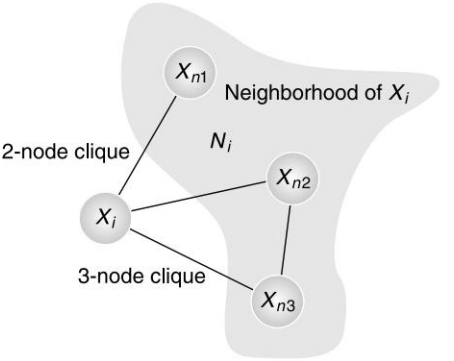


Fig. 4.1: An MRF Neighborhood and Example Cliques

The probability of a particular label assignment is given by,

$$p(x_1, x_2, \ldots, x_n) = \frac{1}{Z} e^{\frac{-1}{k_b T} U_C(x_{c1}, x_{c2}, \ldots, x_{cm})} \tag{4.8}$$

Equation (4.8) is called the Gibbs formulation. The fact that a general MRF is equivalent to the Gibbs form was established by the Hammersley and Clifford Theorem[2]. For a given clique, c, $U_c(x_{c1}, x_{c2}, \ldots, x_{cm})$ is defined on the set of m random variables (nodes) in the clique. The term k_bT can be interpreted as thermal energy from the physical point of view, but in the following calculations it is merely treated as a constant in proportion to the clique energy that controls the sharpness of the Gibbs probability distribution. The term Z is called the partition function and is a constant required to normalize the probability function to [0, 1]. It is the sum over all possible label assignments of the exponential term in the numerator.

The great power of the Gibbs representation is that the problem of finding a global site label assignment with maximum probability can be decomposed into that of minimizing clique energies. In most practical problems, the neighbourhoods are small and the cliques involve only a few nodes.

The goal in mapping logic circuits onto the MRF is to map noisy and faulty circuit operations to probability maximization (clique energy minimization) on the MRF. In this application, the nodes of sites correspond to logic signal terminals. The neighbourhoods of the MRF correspond to logic interactions. An example is shown in Fig. 4.2.

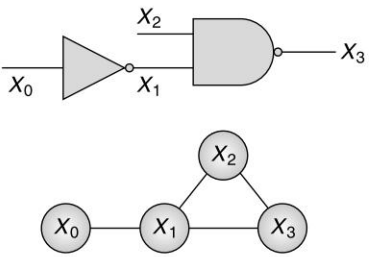


Fig. 4.2: An Example Mapping from a Logic Circuit on to an MRF. The Graph Edges Indicate Neighborhood Relations

The input and output of the inverter are considered to be statistically dependent as indicated by the graph edges between the two nodes. The graph edge does not explicitly represent causality but just that there is a joint probability relationship between X_0

[2]J. Hammersley and P. Clifford, "Markov Fields on Finite Graphs and Lattices", Technical Report, University of California, Berkeley, 1968.

and X_1, i.e., $p(x_0, x_1)$. That is, one doesn't think of X_0 causing X_1, instead their joint assignments must be maximally probable. Thus in the case of an isolated inverter with logic gates taken from {0, 1}, there are two equally probable assignments ($X_0 = 0$, $X_1 = 1$) and ($X_0 = 1$, $X_1 = 0$). For the NAND gate, the graph structure indicates that cliques up to size three are required to represent the statistical dependence among the three logic terminals.

The MRF is a completely general computational framework and in principle any type of computation could be mapped on to the model. It can be illustrated by an example of combinatorial logic. The programming of the MRF is straightforward in this case, and will permit some analysis of the fault tolerance of the architecture.

Combinatorial architecture can be implemented by using a simple, yet powerful, form for the clique energy, called the automodel. For cliques up to order three, the energy function is given by:

$$U_C = k + \sum_{i \in c_0} \alpha_i x_i + \sum_{i,j \in c_1} \beta_{ij} x_i x_j + \sum_{i,j,k \in c_2} \gamma_{ijk} x_i x_j x_k \quad (4.9)$$

The constants, α_i, β_{ij} and γ_{ijk} are called interaction coefficients. The constant k acts as an energy offset.

In order to relate the logic compatibility function to a Gibbs energy form in Eq. 4.8, it is necessary to use the axioms of the Boolean ring. The Boolean ring expresses the rules of symbolic Boolean logic in terms of algebraic manipulation as follows:

$$X' \rightarrow (1 - X)$$

$$X_1 \wedge X_2 \rightarrow X_1 X_2 \qquad \text{(multiplication)}$$

$$X_1 \vee X_2 \rightarrow X_1 + X_2 + X_1 X_2$$

The logic variables are treated as real valued algebraic quantities and logic operations are transformed into arithmetic operations. Additionally, it is desired that valid input/output states of computational logic should have lower clique energies than invalid states so as to maximize the probability of being in a correct (i.e., valid) state as expressed in Eq. 4.8. Thus, the clique energy expression is obtained by a negative sum over minterms from the valid states,

$$U_C = -\sum_i f_i(x_0, x_1, \ldots, x_n)$$

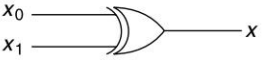
where, $f_i = 1$, and the minterms are transformed by using the Boolean ring rewrite rules. This form exploits the simplification

that cross-products of minterms vanish. For instance, the Boolean ring conversion for the minterm $(x_0, x_1, x_2) = 000$ is,

$$\begin{aligned} x_0' \wedge x_1' \wedge x_2' &= (1 - x_0)(1 - x_1)(1 - x_2) \\ &= (1 - x_0 - x_1 + x_0x_1)(1 - x_2) \\ &= 1 - x_0 - x_1 - x_2 + x_0x_1 + x_0x_2 + x_1x_2 - x_0x_1x_2 \end{aligned}$$

Example 4.1

Exclusive-OR Gate



i	x_0	x_1	x_2	f
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Fig. 4.3: The Logic Compatibility Function for an Exclusive OR Gate with All Possible States

The effect of structure-based errors, or errors on the coefficients in the clique energy, is shown by using an XOR example. There are three nodes in the network: the inputs x_0, x_1 and the output x_2 of the gate. Successful operation of the gate is designated by the computability function, $f(x_0, x_1, x_2)$ as shown in Fig. 4.3. Here we list all possible states (valid states with $f = 1$ and invalid state with $f = 0$) because our approach adapts to errors and we make no assumption about the occurrence of errors. For the exclusive OR example, by summing over the valid states based on the Boolean ring axiom, $000 = (1 - x_0)\,(1 - x_1)\,(1 - x_2)$, $011 = (1 - x_0)x_1x_2$, $101 = x_0(1 - x_1)\,x_2$, and $110 = x_0x_1(1 - x_2)$, we can compute the clique energy as follows:

$$\begin{aligned} U_C = &-1 - (1 - x_0)(1 - x_1)x_2 - (1 - x_0)x_1(1 - x_2) \\ &\qquad - x_0(1 - x_1)(1 - x_2) - x_0x_1x_2 \\ = &-1 + x_0 + x_1 + x_2 - 2x_0x_1 - 2x_0x_2 - 2x_1x_2 + 4x_0x_1x_2 \end{aligned} \quad (4.10)$$

If we take the structural errors into consideration in our design, the clique energy in Eq. 4.9 can be rewritten as:

$$U_C = k + Ax_0 + Bx_1 + Cx_2 - 2Dx_0x_1 - 2Ex_0x_2 - 2Fx_1x_2 + 4Gx_0x_1x_2 \quad (4.11)$$

where, k is a constant, and the nominal weight values for the coefficients are: $A = B = C = D = E = F = G = 1$, as derived above for the error-free case. In the modified equation, the energy coefficients have been replaced by variables to indicate that their values can deviate from the ideal setting due to failures. The variables A, B, C stand for the first-order clique energy coefficients, and D, E, F are second coefficients. The third order coefficient, G, constrains the values of all the lower order coefficients as will be shown shortly. In the nanoarchitecture being described here, the coefficient values are determined by a set of gate connections, so coefficient error is caused by structure-based failure. For successful operation of the logic, it is necessary that the energy of correct logic configurations always be less than the invalid state configurations.

LEMMA: For any combinational logic, the energy of a correct logic state is always less than that of an invalid state by a constant.

Proof

For example, in a simple exclusive-or design shown in Fig. 4.3, the clique energy is

$$U_C = -1 + x_0 + x_1 + x_2 - 2x_0x_1 - 2x_0x_2 - 2x_1x_2 + 4x_0x_1x_2$$

By substituting the invalid and valid states into this energy equation, we get the energy for valid states is always '-1' while that of invalid states is always '0'. The energy difference is a constant (in this case, it is one).

The reason is embedded in the definition of clique energy:

$$U_C(x_0, x_1, x_2) = -\sum_i f_i(x_0, x_1, x_2)$$

For a valid state of any logic, the summation of valid states, f_i, is always one. Or, clique energy U_C is always $U_C = -1$. On the other hand, for any invalid state, the summation of valid states is always zero or $U_C = 0$. Therefore, the energy of a valid logic state is always less than an invalid state by a constant.

Half-Adder

As a first example of useful combinational logic, let us build a device that can add two binary digits together. We can quickly calculate what the answers should be:

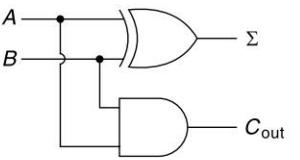
`0 + 0 = 0` `0 + 1 = 1` `1 + 0 = 1` $1 + 1 = 10_2$

So we well need two inputs (a and b) and two outputs. The low order output will be called **Σ** because it represents the sum, and the high order output will be called C_{out} because it represents the carry out.

The truth table is

A	B	Σ	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

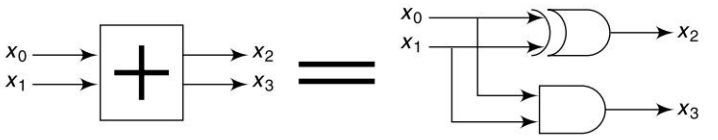
Simplifying Boolean equations or making some Karnaugh map will produce the same circuit shown below, but start by looking at the results. The **Σ** column is our familiar XOR gate, while the C_{out} column is the AND gate. This device is called a half-adder.



Example 4.2

Half-Adder

The effect of errors on the coefficients in the clique energy is illustrated by using another more complicated example – a half-adder. There are four nodes in the network: the inputs x_0, x_1, the sum x_2, and the carry bit, x_3, of the gate (without considering the carry from the previous stage). The successful operation of the gate is designated by the compatibility function $f(x_0, x_1, x_2, x_3)$ as shown in Fig. 4.4.



i	x_0	x_1	x_2	State
0	0	0	0	Valid
1	0	0	1	Invalid
2	0	1	0	Invalid
3	0	1	1	Valid
4	1	0	0	Invalid
5	1	0	1	Valid
6	1	1	0	Valid
7	1	1	1	Invalid

(a) For summation

i	x_0	x_1	x_3	State
0	0	0	0	Valid
1	0	0	1	Invalid
2	0	1	0	Valid
3	0	1	1	Invalid
4	1	0	0	Valid
5	1	0	1	Invalid
6	1	1	0	Invalid
7	1	1	1	Valid

(b) For carry

Fig. 4.4: The Logic Compatibility Function for a Half-Adder Gate

The clique energy for summation is the same as the exclusive-or case in Eq. 4.9 while the clique energy for the carry bit is:

$$U_C = -(1-x_0)(1-x_1)(1-x_3) - (1-x_0)x_1(1-x_3)$$
$$x_0(1-x_1)(1-x_3) - x_0x_1x_3$$

The reason why the clique energy for summation and carry bits is computed is that the summation (x_2) and carry (x_3) are independent outputs. Their results only depend on inputs x_0 and x_1. Based on such a design, we can drastically reduce the computational complexity of mixing both x_2 and x_3 into clique energy computation. The clique energy can then be expressed as follows:

$$U_C = -1 - (1-x_0)(1-x_1)(1-x_2)(1-x_3) - (1-x_0)x_1x_2(1-x_3)$$
$$-x_0(1-x_1)x_2(1-x_3) - x_0x_1(1-x_2)x_3$$

If device errors are taken into consideration in the design, the clique energy for the summation portion is the same as in Eq. 4.11 while the clique energy for the carry portion is:

$$U_C = k + Hx_3 + Dx_0x_1 - 2Ix_0x_1x_3$$

Here the same error coefficient D is assumed for connection x_0x_1.

4.2 Reliability Evaluation Strategies

New technologies for building nanometer scale devices are expected to provide the means for constructing much denser logic

and thinner wires. These devices will have high defect density due to their miniscule dimension, quantum physical effects, reduced noise margins, system energy levels reaching thermal limits of computation, manufacturing defects, aging and many other factors. In the near future, we will be unable to manufacture large, defect-free integrated circuits. Thus, designing reliable system architectures that can work around these problems at run-time becomes important. The key challenges in determining reliability-redundancy trade-off points are:

- The arbitrary augmentation of unreliable devices could result in the decrease of the reliability of a specific architecture.
- For each specific architecture and a given failure distribution of devices, once an optimal redundancy level is reached, any increase or decrease in the number of devices may lead to less reliable computation.
- Redundancy may be applied at different levels of granularity, such as gate level, logic block level, functional unit level, etc.
- Determining the correct granularity level for a specific Boolean network is crucial in such trade-off analysis.

4.3 NANOLAB

NANOLAB consists of a library of functions implemented in MATLAB. The library consists of functions based on the probabilistic non-discrete model of computation discussed earlier, and can handle discrete energy distributions at the inputs and interconnects of any specified architectural configuration. It has the libraries that can compute energy distribution at the outputs given continuous distributions at the inputs and interconnects, introducing the notion of signal noise. Therefore, this tool supports the modelling of both discrete and continuous energy distributions.

NANOLAB automates the probabilistic design methodology by computing energy distribution and entropy at the primary/intermediate outputs and interconnects of Boolean networks, and by implementing Belief Propagation. The logic compatibility functions (similar to a truth table) for the different component gates of the Boolean network and the energy distribution at the primary inputs are specified to the tool. The tool has the capability to model uniform and Gaussian noise at the primary inputs and intercon-

nects of combinational blocks, and analyzes such systems in terms of entropy at the outputs. Such modeling features in NANOLAB expedite and enhance the analysis of reliability measures for different defect tolerant architectures. An example will illustrate the power of the methodology.

Example 4.3

Fig. 4.5 shows a cascaded triple modular redundancy (CTMR) configuration with three TMR blocks working in parallel with a majority gate logic. The code listing in Fig. 4.6 is a MATLAB script that uses NANOLAB functions and the Belief Propagation algorithm to evaluate the probability of the energy configurations at the output of the CTMR.

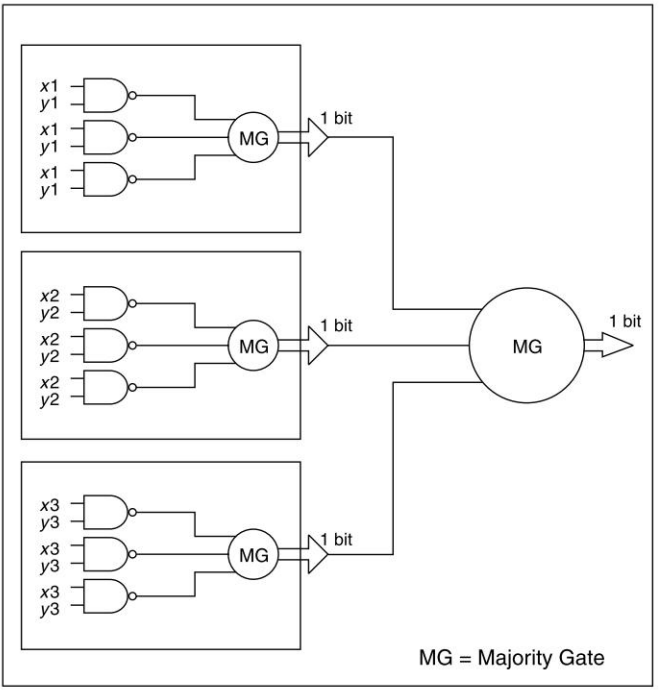


Fig. 4.5: Cascaded Triple Modular Redundancy with Triple Voters: Multi-layer Voting

The probability distributions for x_1, y_1, x_2, y_2, x_3 and y_3 for the NAND gates in Fig. 4.5 are specified as vectors. These vectors specify the probability of the inputs being at a discrete logic state

of low or high. The input probability distributions for all the TMR blocks are the same in this case but these can be varied by having separate input vectors for each TMR block.

Table 4.1: Probability of the output z of a logic gate being at different energy levels for values of KT $\in$ {0.1, 0.25, 0.5, 1.0}

z=0.0	z=0.2	z=0.5	z=0.8	z=1.0
0.809	0.109	0.006	0.025	0.190
0.798	0.365	0.132	0.116	0.201
0.736	0.512	0.324	0.256	0.263
0.643	0.547	0.443	0.379	0.356

Figure 4.6 shows the MATLAB script for first order CTMR with discrete input distribution.

```
no_of_blocks = 3;                          % number of TMR blocks
prob_input1 = [0.1   0.9];                 % prob distbn of input1 of NAND gate
prob_input2 = [0.1   0.9];                 % prob distbn of input2 of NAND gate
BT-values = [0.1   0.25   0.5   1.0];      % different kbT values

for TMR_block = 1 : no_of_blocks
        counter = 1;
        % energy_2_input_gates_function is a NANOLAB function and takes in as
                                                                  parameters.
        % The output gives the state configurations of the primary output of the logic
        % gate at different kbT values.

prob1 = energy_2_input_gates_function (input1, prob_input1, prob_input2, BT_Values);
prob2 = energy_2_input_gates_function (input1, prob_input1, prob_input2, BT_Values);
prob3 = energy_2_input_gates_function (input1, prob_input1, prob_input2, BT_Values);
[a,b] = size (prob1);

% req_pb1, req_pb2, req_pb3 are vectors which contain probabilities
% of the output being a 0 or 1 for a particular kbT value for Belief Propagation

  for i = 1:a
        req_pb1 = [prob1(i,1)       prob1(i,b)];
        req_pb2 = [prob2(i,1)       prob2(i,b)];
        req_pb3 = [prob3(i,1)       prob3(i,b)];
        BT = BT_Values(i);
        % energy_3_input_gates_function is a part of NANOLAB and takes in input
        % and output parameters similar to the previous 2 input gates function.
        t_p = energy_3_input_gates_function (input2, req_pb1, req_pb2, req_pb3,
                                                              BT_Values(i));
        prob(TMR_block, counter) = t_p(1,1);
        counter = counter + 1;
        prob(TMR_block, counter) = t_p(1,b);
        counter = counter + 1;
  end
end
```

Fig. 4.6: MATLAB Script for First Order CTMR with Discrete Input Distribution

The NANOLAB functions return vectors similar to the one shown in Table 4.1. These indicate the probability of the output of a logic network being at specified energy levels for different *KT* values. In the CTMR configuration, for each TMR block, the energy configurations at the outputs of each of the three NAND gates are obtained from the function for a two-input gate. Then these probabilities are used as discrete input probability distributions to the function for a three-input gate. Similarly, the probabilities of the final output of the CTMR are calculated.

4.4 NANOPRISM

NANOPRISM is a probabilistic model checking based tool that applies probabilistic model checking techniques to calculate the likelihood of occurrence of transient defects in the devices and interconnections of nano architectures. NANOPRISM is based on the conventional Boolean model of computation and can automatically evaluate reliability at different redundancy and granularity levels, and most importantly show the trade-offs and saturation points. At saturation point, the granularity-based redundancy versus reliability reaches a plateau meaning that there cannot be any more improvements in reliability by increasing redundancy or granularity levels. It consists of libraries built on PRISM (probabilistic model checker) for different redundancy-based defect-tolerant architectural configurations. These libraries also support modeling of redundancy at different levels of granularity, such as gate level, logic block level, logic function level, unit level, etc. Arbitrary Boolean networks are given as inputs to these libraries and reliability measures of these circuits are evaluated. This methodology also allows accurate measurements of variations in reliability on slight changes in the behavior of the system's components, for example, the change in reliability as the probability of gate failure varies. An example will make this clear.

Example 4.4

The DTMC model of the TMR configuration of a NAND gate is shown in Fig. 4.7.

```
prob p_err = 0.1;                    // probability that gate has error
prob p_in = 0.9;                     // probability an input is logic high

const R = 3;                         // number of redundant processing units const
const R_limit = 1;

module TMRNAND

        x : bool;
        y : bool;
        s : [ 0 . . 3  ] init 0;     // local state
        z : [ 0 . . R ] init 0;      // number of outputs that are stimulated
        z_output : [0 . . 1] init 0; // output of majority logic
        e : [0 . .  4  ] init 0;     // count of the redundant unit being processed

        [ ] s = 0 & c>R -> (s' = 0);

        //  processed all redundant units
        [ ] s=0 & c=R-> (s' = 3) & (c' = c + 1);

        // initial choice of x and y
        [ ] s = 0&c< R -> p_in : (x' = 1) & (s' = 1) & ('c=c+1) + (1– p_in) : (x' = 0) & (s' = 1)
                                                                              & ('c=c+1);
        [ ]s=1->p_in : (y' = 1) & (s' = 2) + (1-p_in) : (y'=0)&(s'=2);

        // NAND operation
        [ ]s=2->p_err : (z' = z + (x&y))&(s'=0) + (1-p_err) : (z' = z+(!(x&y)))&(s'=0);

        // majority logic
        [ ]s = 3 & z>=0 & z<=R_limit-> (s'=0) & (z_output'=0);
        [ ]s=3 & x>R_limit & z<=R->(s'=0) & (z_output'=1)

end module
```

Fig. 4.7: PRISM Description of the TMR Configuration of a Single NAND Gate

It is assumed that the inputs X and Y have identical probability distribution (probability of being logic high is 0.9), and the failure (inverted output) probability of NAND gates is 0.1. However, the input probability distributions and failure distribution of the NAND gates can be changed easily by modification of the constants given at the start of the description. The probabilistic state machine for this DTMC model built by PRISM has 115 states and 182 transitions. Also, model checking is performed to compute the probability distribution of the TMR configuration's output being in an invalid state for different gate failure probabilities. Furthermore, since PRISM can also represent non-deterministic behavior, one can set upper and lower bounds on the probability of gate failure and then obtain (best and worst case) reliability characteristics for the system under these bounds. The CTMR configuration uses three TMR logic units and majority voter. The probability distribution obtained for the TMR block of a single NAND gate can be used directly in the CTMR configuration, thus reducing the state space.

4.5 Reliable Manufacturing and Behavior from Law of Large Numbers

Till now, electronics has relied on the 'Law of Large Numbers' (LLN) below the device level to guarantee deterministic device behavior (for example, dopant ratios, transition timing, electron stage storage). However, at the nanoscale, we hope to build devices with small numbers of atoms or molecules (like wires which are 3-10 atoms wide, diodes built from 1-10 molecules) and we hope to store state with small numbers of electrons (for example, 10's). If we are to build devices at these scales, we will no longer be able to rely on the 'Law of Large Numbers' **below** the device level. We must, instead, employ the 'Law of Large Numbers' **above** the device level in order to obtain predictable behavior from atomic-scale phenomena which are statistical in nature. At the same time, the 'Law of Large Numbers' can also help us by providing statistical differentiation at scales smaller than those we can pattern directly or economically by using lithography.

The 'Law of Large Numbers' (LLN) is said to hold for a sequence of random variables with finite expected values when the mean value for the sequence converges to the expected value. That is, for a sequence of independent, identically distributed random variables, y_i:

$$\lim_{n\to\infty} \text{Prob}\left[\left|\frac{1}{n}\sum_{i=1}^{n} y_i - E(y)\right| \geq \varepsilon\right] = 0$$

This implies that even though each individual y_i is probabilistic in nature, aggregate properties of a large number of the y_i's are quite predictable. As we increase the number of events over which we aggregate, the likelihood of deviating more than a tiny percentage from this mean is smaller and smaller.

Most of the physical phenomena we rely upon in electronic systems are statistical at the atomic scale. In the construction of silicon devices, doping is a good example. We build transistors in MOS devices by mixing in a percentage of impurities (donors, acceptors) to change the band structure of the devices. We do not place 999 Silicon atoms and then one Boron atom. Instead, we arrange to impact the Silicon with a given intensity of Boron atoms

to replace the Silicon atoms. We don't guarantee exactly where each Boron atom ends up. Nor do we guarantee that every bond in the crystalline lattice is perfectly made.

During operation, we typically think about charges on capacitors and gates and current flows across devices. However, current flow is simply an aggregate view of the behavior of individual electrons. Individual electrons travel across a device or region of silicon probabilistically depending on the fields and the thermal energy. We only know statistically what each electron will do.

Similarly, we can isolate charge on a node such as a memory element or gate input. We can construct electrostatic barriers to hold the charge in place. Nonetheless, thermal energy and quantum tunneling give the individual electrons some probability of surmounting the barrier and leaving, or netting the node.

The law of large numbers is needed to know how we can effectively get very reliable device manufacture and device properties even though each individual atom or electron behaves probabilistically. Referring to the above example, while we cannot guarantee that we have exactly 999 Silicon atoms and then one Boron atom in our Silicon crystal, the LLN assures us that we can have high confidence that there are close to 10^6 Boron atoms in a doped region with 10^9 crystal cites. The LLN can be exploited above the device level through some of the techniques described here:

4.5.1 Tolerate Variations in Manufacture by Selecting which Devices to Use

DRAM row and column sparing is perhaps the most familiar case of using LLN design to deal with manufacturing defects. These defects may arise from the probabilistic assembly of atomic-scale structures as described earlier, or, more likely from the lack of perfect purity and calibration in the design of our manufacturing systems. In either case, if we are unhappy with the probability that our manufacturing process produces a perfect memory bank where we require every row, column, and memory bit to perform, perfectly, we add spare rows and columns {Fig. 4.8a}. When the expected number of defects per bank is less than one, it is highly unlikely that we will see many defects per bank. Providing one or

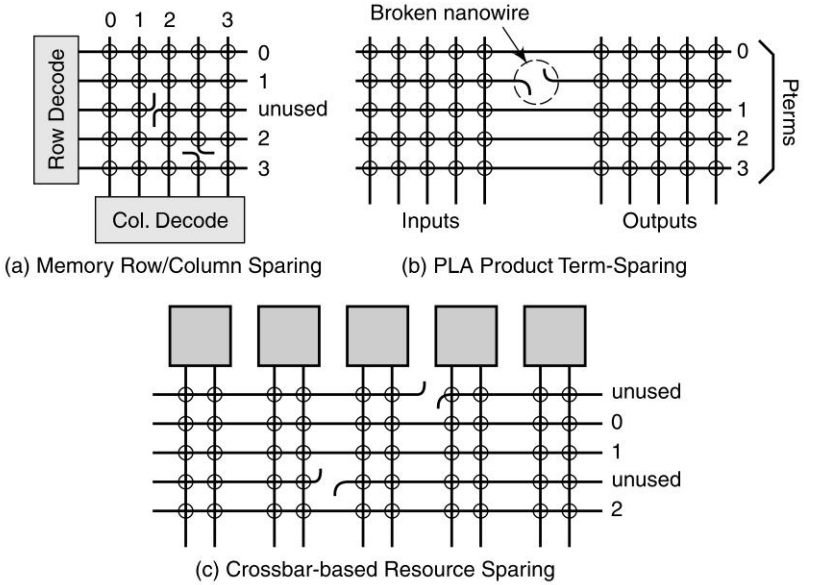


Fig. 4.8: Crossbar-based Resource Sparing

a few spares per bank guarantees a very high probability that each bank can be made perfect. This shows that use of LLN is already a well established practice in some of the systems employed every day.

Row sparing in DRAMs is a special case of *M*-of-*N* sparing. That is, we fabricate or assemble *N* equivalent items in our system but only require that *M* of them function in order to have a correct system. This way, rather than requiring that *M* things yield perfectly; we simply require that at least *M* items out of *N* yield. Here the term *yield* is used to broadly mean that the device or component has been manufactured adequately to perform its intended role.

Statistically, if the probability that each item yields is *P*, then the probability that every one of *M* items will yield is:

$$P_{allyield}(M) = P^M$$

We can calculate the probability that exactly *i* items will yield by using a binomial distribution:

$$P_{allyield}(N, i) = ({}^{N}C_i)P^i(1-P)^{N-i}$$

That is, there are ${}^{N}C_i$ ways to select *i* good items from *N* total items, and the yield probability of each case is $P^i(1-P)^{N-i}$. We yield an

ensemble with M items whenever M or more items yield, so our system yield is actually the cumulative distribution function:

$$P_{MofN} = \sum_{M \le i \le N} \left(({}^{N}C_{i}) P^{i} (1 - P)^{N-i} \right)$$

To exploit this M-of-N sparing, we need designs structured with a large number of identical items which are cheaply interchangeable. The crossbar organization used in memory arrays, interconnects, and programmable logic arrays is a prime example of structure which has this property.

- In a memory, all the rows (or columns) are identical. As long as we can program up the addressing for each row and column and configure non-used lines so they do not interfere with operation, we can use any M of the N row lines to serve as our desired row lines.
- In a Programmable Logic Array (PLA), all of the programmable terms (for example, product terms) are logically equivalent. We simply need to be able to allocate enough product terms to cover our logic function {Fig. 4.8(b)}.
- In a routing channel, all the wires which span the same source – sink distance are identical. Any good wire which can be programmed to provide the source to sink connection is adequate. If we have a full crossbar set of connections between our sources and sinks, we have the desired property that any M of the lines can serve to provide each connection {Fig 4.8(c)}.

The key element in all of these examples is that they can be configured to select the M useful components from the total N total components *post fabrication*. That is, after fabrication, we can test the device and program it to use only the functional resources.

Large sets of parallel, nanowires assembled into crossbars are one of the things that can be built at sublithographic scales. Further, while the full connectivity of the crossbar is quite expensive for interconnect in conventional CMOS where programmable switchpoints are large compared to wire crossings, full connectivity is relatively cheap in many of the emerging nanotechnologies. In particular, several technologies offer the prospect of non-volatile crosspoints that fit within the space of a wire crossing. As a result, this full M-of-N wire/row/product-term section is relatively inexpensive.

A convenient feature of the row/wire/product-term sparing is that defect remapping is a local operation. We simply need to configure the producer and consumer to connect to a unique, functional wire to serve in this role. The full crossbar interconnect at the ends of the mapping prevents this choice from effecting the mapping of resources elsewhere in the design.

Fig. 4.9 shows a simple example of how full crossbar choice of alternate resources allows us to localize the impact of remapping. The left side shows a sparse interconnect scheme in the spirit of traditional FPGA designs. The right side shows the design with full connectivity for sparing. The middle row shows a broken line. In the crossbar case, this can be accommodated simply by shifting

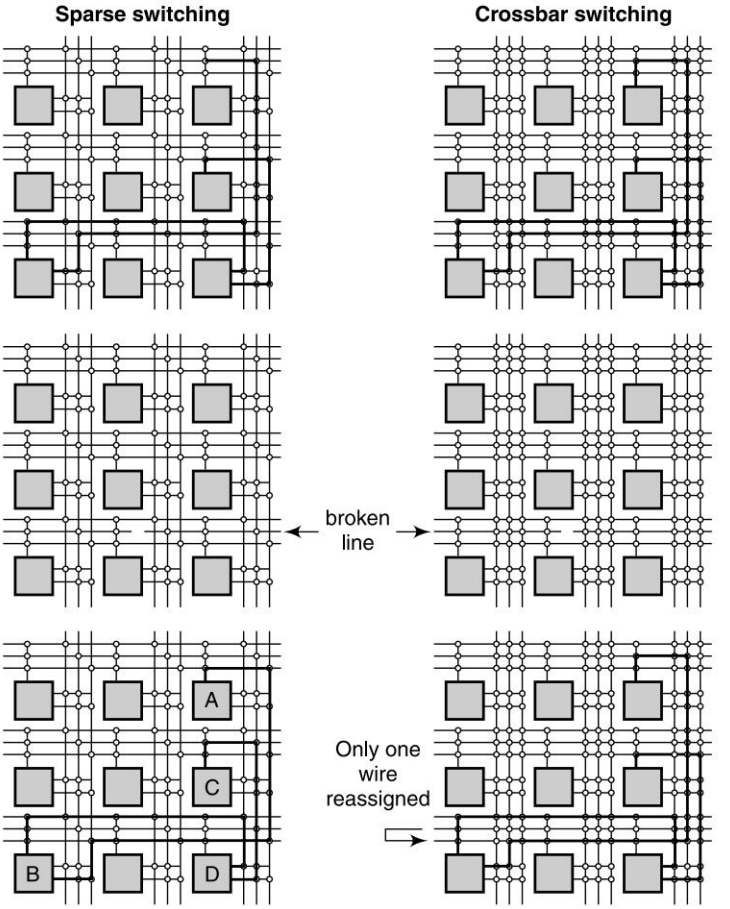


Fig. 4.9: Local Defect Remapping Example

the net segment which previously used the broken wire to a free track in the same channel. This change is contained locally to this one channel; the segments of the net in different channels do not need to change, nor does the routing of any of the other nets. In the sparse case, however, we are forced to change the track assignment of this $A \rightarrow B$ net in all the channels it traverses due to the limited switch-box population; we are further forced to reroute the $C \rightarrow D$ net in order to accommodate this change.

4.5.2 Tolerate Variation in Manufacture by Selecting which Device to Use for what Role

The model presented above is that a resource is simply all good or bad. However, we may not need to use all the potential functionality of a resource. This gives us the opportunity to select the resource which is simply 'good enough' to serve for the purpose at hand.

As an example, consider the molecular-switch crosspoints in Fig. 4.10. The assembly techniques allow us to, statistically, place a number of switchable molecules between a pair of crossed conductors. Any particular junction may get fewer or no molecules in the junction. While the percentage of junctions that could be programmed will increase as the technology matures, the nature of the assembly process suggests we will always have some statistical gaps in the molecular coverage and we will get some statistical variation in the number of molecules in a junction. Junctions with fewer molecules may have too high a resistance to perform properly, or may simply perform slower than junctions with the expected number of molecules in the junction.

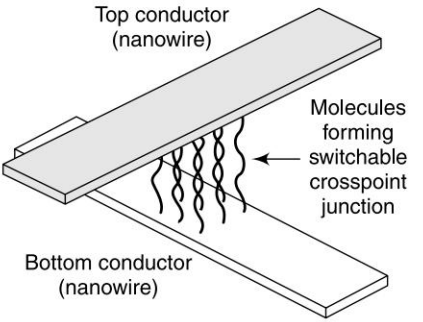


Fig. 4.10: Molecule-based Switchable Crosspoint

4.5.3 Exploit Variations to get Differentiation at the Nanoscale

Constructing differentiated patterns at the nanoscale is a key challenge associated with nanoscale engineering. Conventional lithographic approaches may not extend economically to the nanometer scale. There are a number of bottom-up assembly techniques that can provide interesting nanoscale features. These techniques, however, generally give us one of two things:

1. **Regular structures such as the crossbars, memory cores, and PLAs:** A key challenge is addressing the nanoscale wires from conventional microscale wires. We do not expect all of our nanoscale components to be perfect and they may start out as regular, undifferentiated arrays. If we can address our nanoscale wires from reliable, microscale wires, we can test resources, configure the system to use the functional resources, and programmably differentiate the regular structure.

   Because of the scale difference between the microscale wires (for example,100 – 200 nm pitch) and our nanoscale wires (for example, < 20 nm pitch), it is not desirable to directly connect each nanoscale wire to a single microscale wire. A direct connection would prevent us from packing the nanoscale wires at their tight pitch. A natural solution to bridging between the micro and nano scale is to use a demultiplexer. The demultiplexer decodes a densely coded input and uses that to address one of its outputs. Using a demultiplexer here allows a small number of reliable, microscale wires to address a large number of nanoscale wires. From an information theory standpoint, the demultiplexer only needs $\log_2(N)$ input wires in order to specify which of the N nanowires it should address. Since $\log_2(N)$ will be much smaller than N, for sufficiently large N, this allows us to minimize the cost of the interface microscale wires and maintain the density benefit of the nanoscale wires.
2. **Statistical structures:** A statistical scheme will be sufficient to construct such a multiplexer. A physical process can be used to produce a random distribution of gold particles in a layer between the microscale and nanoscale wires. These particles effectively provide sparse, random addressees for the nanowires. LLN can be used with statistical effects to engineer the desired property.

4.5.4 Tolerate Variations in Behavior by Performing Redundant and Self-checking Computations

The three methods focused on the statistical nature of fabrication and assembly at the nanoscale. Since we will also have a small number of electrons holding the state and driving logic transitions, we must be also concerned about active device behavior. Here also, to protect against the small number effects below the device level, we will have to exploit LLN effects above the device level. To successfully exploit atomic-scale devices, we must find the right level and hierarchy for the deployment of these LLN techniques to assure the correct dynamic behavior of the computations.

CHAPTER 5

Nanoscale Quantum Computing

The scale of quantum physical phenomena is so vast that even a super computer built on von Neumann's style of computing cannot realistically model quantum physics at the atomic and subatomic levels. On the other hand, quantum computers, which mimic quantum physics themselves, are capable of vast parallelism and could theoretically simulate such phenomena. In 1985, seminal work by Deutsch[1] showed that quantum computers can create a quantum superposition of states allowing each of them to follow a distinct computational path until a final output is obtained. Such free access to parallelism is unprecedented if a classical model of computation is used.

With the advent of nanoscale technologies, quantum computing is viewed as more than a source of large-scale parallelism. It is likely that in the near future we can exploit quantum entanglement and quantum mechanical reversible transformations to build new kinds of computing systems.

In 1973, Bennett's article entitled, "The Thermodynamics of Computation"[2] discussed reversibility in quantum computations steps. This work motivates us to ask if we can actually realize computing logic that is reversible and hence would free us from the growing power concerns in CMOS-based computing.

---

[1]David Deutsch, "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer", Proceedings of the Royal Society of London. *Series A, Mathematical and Physical Sciences*, Volume 400, Issue 1818, pp. 97-117, July 8, 1985.

[2]Charles H. Bennett, "The Thermodynamics of Computation—A Review", *International Journal of Theoretical Physics*, Vol. 21, pp. 905-940, 1982.

5.1 Quantum Computers

To understand how quantum computers work it is helpful to start off with how conventional computers work. Conventional digital computers—the ones on our desks—are at their very core a collection of millions of electrical switches that are either on or off. Each such electrical switch is binary in nature – it can exist as either on or off (0 or 1) but never as both. All the wonderful computer applications we are familiar with are achieved through manipulation of these switches in a consistent fashion. The logic behind the manipulation comes from the work of a nineteenth century English mathematician named George Boole; the circuits inside a computer that implement Boolean logic are called "logic gates". The way a digital computer performs additions—or for that matter, any other function—is by passing bits through logic gates. There are basic logic gates with names such as AND, OR and NOT that are used to build computer circuits. For example, the NOT gate takes an input bit and inverts it; thus, an "on" bit is converted to "off" (0 is converted to 1) if it passes through a NOT gate. In a computer, this output is passed to another gate for more processing. Each step of a calculation keeps time to the computer's internal clock; each step means passage through a logic gate.

For a quantum computer however the switches do not have to be either on or off; they are somewhere in between—a combination of both on and off. Counter-intuitive as this statement may sound, it is consistent with the rules of quantum mechanics. A quantum mechanical object exists simultaneously in several states; it can inhabit parallel worlds, which is to say, by proper design it can carry out multiple computations in the same hardware. It is this property that gives quantum computing its power. This will become clearer once we understand a few quantum phenomena.

5.1.1 Classical Gates

Consider a hydrogen atom that has a single electron orbiting around a nucleus. If we solve the quantum mechanical equations of motion for the electron, we find that the electron can inhabit certain orbits that are separated from each other like the rungs of a ladder. Each different orbit it can inhabit has a discrete energy associated

with it. The lowest energy orbit is called the ground state—in the absence of any other excitation, the electron prefers to reside here. A shining light of the right frequency will cause the electron to move to an orbit with higher energy—an excited state. The electron may stay in the excited state, but if another pulse of light of the same frequency that drove the atom to the excited state comes along, it causes the electron to go back to the ground state by a phenomenon called stimulated emission. If we denote the excited state by a "1" and the ground state by a "0", we can use this behavior to build a NOT gate, since the pulse of light that hits the electron effectively flips its state.

5.1.2 Reversible Operations

What other operations can we perform through quantum mechanical systems? It turns out that the classes of transformations that are possible are those that are reversible, that is, if we know the output of the transformation, we can deduce what the inputs should have been. An example of this is the NOT operation described earlier: if the output is a 1, the input should have been 0, and if the output is a 0, the input should have been 1. On the other hand, the AND and OR gates are not reversible, for example, if the output of an AND gate is 0, there is no way of deducing what the inputs were.

This seems to suggest that quantum computers might be less powerful than classical computers since we have to satisfy the reversibility condition. Indeed, initially there were concerns that as device sizes became smaller and computer operations became more and more quantum mechanical, it might pose limitations on what computers could do. But it was soon realized that everything that was possible classically would still be possible quantum mechanically, only it would have to be done differently.

There indeed exists a set of reversible gates through which it is possible to synthesize any Boolean circuit using approximately the same number of gates as a circuit synthesized with AND and OR gates. These are the CNOT (controlled NOT also called an XOR gate) and C^2NOT (controlled controlled NOT also called a "Toffoli" gate).

5.1.3 Beyond Classical Gates

As mentioned earlier, the real advantage of quantum computers comes from the fact that it is possible to design certain applications that take advantage of quantum parallelism. It is easily possible to design an application where multiple computations are carried out in parallel, the problem is that this produces multiple results—the difficulty is in getting the desired result out of the system. Unlike a classical parallel computer where we can examine each of these multiple results, in a quantum computer, if we try to observe the state it simply produces each result with a certain probability.

5.1.4 Superposition

Consider a simple harmonic oscillator such as a pendulum. If we push the bob (weight) of the pendulum, we transfer energy to the pendulum and it starts to oscillate. In classical physics this transfer of energy is a continuous function. On the other hand, if we try to analyze the simple harmonic oscillator quantum mechanically, something quite strange happens. As we push the pendulum, energy is transferred to the pendulum in discrete lumps. The energy levels are discrete in a manner analogous to the hydrogen atom. In fact, the two systems are similar: the electron in the hydrogen atom is confined in a potential well created by the nucleus which is similar to the potential of the harmonic oscillator. Just as in the hydrogen atom, there is a state with a minimum energy known as the ground state. This is the lowest rung of the energy ladder. When one solves the energy equation for a quantum mechanical harmonic oscillator, one gets characteristic solutions called wavefunctions (or basis functions) for each level. The amplitudes of the wavefunctions, when squared, describe the probability of the oscillator's presence at different positions.

Could one use a system of classical oscillators to carry out computing similar to quantum computing? The answer is that one could certainly design a system of coupled classical oscillators that would evolve in a way similar to a quantum system; however, the hardware required would be considerably larger. To synthesize a system with N modes of oscillation requires N classical oscillators. No

matter how we couple classical oscillators, each degree of freedom gives just one mode of oscillation. In contrast, a single quantum oscillator has multiple degrees of oscillation each with its own wavefunctions. In general, the evolution in time of a particle will be described by the combination of the evolution of each of these wavefunctions. In quantum mechanical jargon, this phenomenon is known as superposition. According to this, a quantum mechanical system, such as an electron, behaves as if it is in multiple places and occupies multiple states at the same time. A quantum computer exploits superposition to carry out computations in multiple states simultaneously in the same hardware.

This behavior allows a single quantum structure with (at least) two clearly identifiable states, such as the nucleus of an atom in the presence of a magnetic field, to be used as a quantum bit or a qubit. Several qubits can be put together to build a quantum computer. The state of the quantum computer will be described by the superposition of each of the qubits of the system. Unlike a digital bit, which has to be a 0 or 1, each qubit simultaneously takes on values 0 and 1 and in between with certain probabilities. This may sound like an analog variable, but it is indeed digital, for example, whenever you measure a qubit, you will get either 0 or 1—nothing in between.

5.1.5 The Sqrt(NOT) Operation

When we shine light on an electron in a ground state for less than the time needed to excite it to a higher energy state, it inhabits both states simultaneously—it exists in a superposition of the ground and excited states. For example, if we shine light for half the time required to excite the electron from the ground to the excited state, and then observe the state, we get the ground and the excited states with equal probabilities.

As described before, if we apply a pulse of light of the proper intensity and time to an electron in the higher energy state, it gets driven to the lower energy state through stimulated emission. However, if we apply a pulse of only half this length, again, just as in the previous paragraph, it gets driven to a superposition of the ground and excited states—if we observe the electron we will be equally likely to observe either state.

Therefore, applying a half pulse is a randomizing operation. Irrespective of which state, 0 or 1 we start with, if we apply a half-pulse and then observe the system we get both states with equal probabilities. But what if we do not observe the system and subject it to further quantum operations? Physicists know well that quantum systems behave very differently depending on whether or not they are observed (for example, the Schrodinger's Cat Paradox[3]). For example, consider applying two successive half pulses one after another. This is just like a full pulse and applies a NOT operation. Therefore, a half pulse is like a sqrt(NOT). What seemed like a randomizing suddenly becomes predictable—this has no classical analog—it is just quantum mechanics at its most mysterious! To harness the computational power of this operation, we need to go to machine language and use this as one of the primitive operations.

5.1.6 Quantum Algorithms—Necessity of Quantum Software in Conjunction with the Hardware

Software often does not get the recognition it deserves. The non-specialist frequently underestimates the importance of software. One should hardly pick on non-specialists when fully trained lawyers at the world's leading computer maker have been guilty of the same mistake. In one of the biggest blunders in corporate

[3]Schrödinger, Erwin, "Die gegenwärtige Situation in der Quantenmechanik (The present situation in quantum mechanics)". Naturwissenschaften, November 1935.

(Schrödinger's cat, often described as a paradox, is a thought experiment devised by Erwin Schrödinger. It attempts to illustrate what he saw as the problems of the Copenhagen interpretation of quantum mechanics when it is applied beyond just atomic or subatomic systems.

A cat is penned up in a steel chamber, along with the following device (which must be secured against direct interference by the cat): in a Geiger counter there is a tiny bit of radioactive substance, so small, that perhaps in the course of the hour one of the atoms decays, but also, with equal probability, perhaps none; if it happens, the counter tube discharges and through a relay releases a hammer which shatters a small flask of hydrocyanic acid. If one has left this entire system to itself for an hour, one would say that the cat still lives if meanwhile no atom has decayed. The psi-function of the entire system would express this by having in it the living and dead cat (pardon the expression) mixed or smeared out in equal parts.)

history, IBM gave Microsoft an unrestricted license to develop and market the operating system for the IBM PC.

The IBM-Microsoft deal took place almost two decades ago. When computers were first being developed, hardware was the stumbling block; programmers wrote software around the hardware. Nowadays, most often software is the bottleneck: it is generally the hardware that is planned around the software. Successive generations of Intel microprocessors are designed to be able to run existing software. Often, most of the time taken to develop a new high-technology product that involves computers also involves writing the software that enables the product.

Today, in the new paradigm of quantum computing it is important to think about software issues of quantum computers in conjunction with the hardware. It takes considerable effort to devise software for quantum computers that harnesses their power. One of the hottest fields in physics and computer science is to devise hardware techniques for implementing quantum computers and to devise applications in which they greatly outperform classical computers. The power of quantum computers lies not in providing faster hardware to carry out the same applications but in enabling an entirely new class of applications by using fundamentally new programming procedures, known to computer scientists as algorithms.

> "Don't diddle code to make it faster, find a better algorithm."
>
> —The Elements of Programming Style, Kernighan and Plaguer[4]

Algorithms are basically recipes that tell a computer how to go about solving a particular problem quickly. A good algorithm drastically reduces the time taken to solve a problem. Finding good and efficient algorithms is very important because algorithms tend to be used over and over again for tasks routinely performed by computers. In the last 25 years, the theory of algorithms has been quantitatively developed by computer scientists using extremely sophisticated mathematics. It was only in the last decade that two powerful algorithms were discovered for quantum computing—one of these algorithms is for factorization and another for searching unsorted databases. Peter Shor discovered the factorization algo-

[4]Kernighan, B. and Plauger, K. (1974), *The Elements of Programming Style*. McGraw-Hill.

rithm in 1994[5]; it is often called the first 'killer application' of quantum computing. Much of today's excitement in quantum computing was triggered by Peter Shor's factorization algorithm. The searching algorithm[6] was invented by Lov Grover at Bell Labs a couple of years later in 1996. Rudimentary versions of both have been recently implemented in hardware and tested.

In mathematics, there is a class of problems that is difficult to solve, but given the answers, it is easy to check if they are correct. Factorization is one such problem. It is the opposite of multiplication, which is something that every school child understands. If someone gives you two numbers, say 131 and 133, and a piece of paper and pen and asks you to multiply them, very soon you can deduce that the answer is 17423. On the other hand, if someone gives you the number 17423 and asks you to figure out which two numbers have to be multiplied to get this; the problem—called factorization—is much harder. One needs to try out several possibilities before one hits on the correct answer. Given a paper and pen it would take most people at least few minutes. A conventional computer would have to follow a procedure similar to what a person with a pen and paper would have to do. Mathematicians, despite centuries of work, have not been able to find any short cuts. This problem is believed to be intrinsically difficult, so much so that it forms the basis of cryptographic protocols that are used for banking and commerce. If someone were to find an efficient way for factorization, many of the cryptographic transmissions in the last two decades would suddenly become public. So it was a big surprise when Peter Shor discovered a technique that made factorization almost as easy as multiplication, given a quantum computer. The heart of the factorization algorithm is estimating the 'periodicity' of repetition of a sequence.

Factorization is a kind of a search in a well-defined context. The next major development in quantum algorithms was an algorithm that allows general searches. The search algorithm allows an item

[5]Peter Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Algorithms on a Quantum Computer", Proceedings of the 35th Annual Symposium on Foundations of Computer Science, pp. 124-134, Santa Fe, NM, Nov. 20-22, 1994, IEEE Computer Society Press.

[6]Lov K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search", Proceedings of 28th Annual ACM Symposium on Theory of Computing (STOC), pages 212-219, May 1996.

to be retrieved from a large unsorted database quickly. This algorithm is much simpler than the factorization algorithm. Unlike the factorization algorithm that provides an exponential speed-up over the best-known classical algorithm, the search algorithm only provides a square-root speed-up. On the other hand, it is considerably more widely applicable. The search algorithm has formed the building block of several other important algorithms and can be used in a number of tasks, for example, optimization, game theory, and even precision measurements.

Let us take a look at the software behind quantum computers and see where we are today on the hardware front. Because it is widely applicable to many computer science scenarios, we will look at one algorithm—the search algorithm—in detail, as it will allow us to understand a quantum computer's superiority. But before that, let us spend some time understanding how quantum computers work.

5.1.7 Searching by using Sqrt(NOT)

Let us start with a classical search algorithm called simulated annealing. This algorithm uses an analogy with the process of annealing and searches for state with the lowest cost for a combinatorial optimization problem. The idea is to carry out a thermal diffusion over the various possible states of the combinatorial optimization problem in the presence of a potential function. The potential function is chosen so that the desirable states have a lower potential. As the temperature is lowered, in analogy with the annealing phenomenon, the system reaches the lowest cost states. The algorithm works very well in practice—its limitation being that if the problem to be solved has too many local optima then it can get stuck in one of them and not reach the global optimum.

Let us next think of a similar quantum mechanical algorithm. The advantage of a quantum mechanical system is that it would have the ability to tunnel out of local optima. Unfortunately, even for the classical algorithm, no one has been able to carry out a theoretical analysis of the algorithm to see how well it would converge with a realistic cooling schedule for a problem of practical interest. The quantum situation is even more complicated. Not

only are there local optima where the system can get stuck, but there is all sorts of tunneling going on between various local optima. Simulations of a related algorithm have recently been carried out by Edward Farhi[7] at Massachusetts Institute of Technology in which the ground state is slowly changed. These suggest that quantum mechanics may indeed offer some advantage over classical algorithm although the evidence is preliminary.

Apart from tunneling the other direction quantum mechanics offers an advantage in is the fact that because of superposition a single quantum particle can explore multiple states simultaneously. This is the principle that the quantum search algorithm makes use of. There is an equal probability of the particle making a transition from one state to another. In addition, there is a potential that attracts the particle to the desired state and nudges it out of the initial starting state.

We can implement the random diffusion by means of a sqrt(NOT) or some similar operation. In order to apply the same principle to a larger system think of multiple sqrt(NOT) operations independently applied to a number of qubits. In case we take $\log_2 N$ qubits and we start it from any basis state, the system gets driven to an equal superposition of all N basis states so that if we were to do a measurement we would get all states with equal probabilities. Denote this composite operation by W. Again as in the case of a single qubit, if we apply W twice, the system gets "derandomized".

A potential function is implemented by a phase rotation operation, another strictly quantum mechanical operation. Those familiar with Schrödinger's Equation will recognize this operation as the evolution of a wavefunction $\psi(x)$ according to a diffusion equation, where there is a phase rotation of each state that is proportional to its potential $V(x)$. Similarly, in the discrete state situation, one implements a potential function by means of selective phase rotation of the desired state. The question that arises is whether it is possible to implement a selective phase shift that rotates the phase of the desired state without knowing which one it is. It is indeed possible to implement such a selective phase rotation based on just the desired properties of the state we are looking for, say the t state, even without knowing which one this

[7]Edward Farhi et. al., "A Numerical Study of the Performance of a Quantum Adiabatic Evolution Algorithm for Satisfiability", available online : arXiv:quant-ph/0007071, 2000.

is. This is like saying that our t state is one that extends most in the say, x direction. Even without knowing which one this is, we can implement a selective phase shift of this state by having a potential that has a high x gradient. Similarly, in the discrete situation we can implement a selective phase shift provided we can design a quantum circuit that can evaluate whether or not a given state is the t state.

Let us denote a transformation that inverts the phase of an item in the t state by I_t. By diffusion in the presence of such a phase rotating potential function, as in simulated annealing, one can drive the system into the target state (note that unlike a classical algorithm no observations in intermediate steps are permitted). Indeed, the framework over here is simple enough that one can analytically show that by starting with each qubit in the 0 state and applying the following we obtain the t state. An observation then reveals which the t state is

$$W(I_0 W I_t\ W)\ldots\text{sqrt}(N)\text{ repetitions}\ldots(I_0 W I_t\ W)\ (I_0 W I_t\ W),$$

Here W is like a diffusion operation, I_t is a selective phase rotation transformation that changes the potential of the t state, and I_0 is a selective phase rotation transformation that changes the potential of the 0 state. The analysis of the algorithm is mathematical and based on properties of non-commuting matrices.

The power of the algorithm comes from the fact that it needs only about sqrt(N) steps to find an item in an unsorted database of N items. Since the database is unsorted, any classical algorithm would need to go through each item of the database and would need about N steps. The reason that the algorithm needs sqrt(N) steps can be understood by first thinking of a classical algorithm where we successively examine randomly chosen items until we find the desired one. The classical algorithm needs about N tries. This is because in each try the probability of getting the desired item is $1/N$, and so the cumulative probability of getting the right answer increases by this amount in each try. Therefore, in about N tries it increases to close to 1. In quantum mechanics, it is amplitudes, not probabilities that add in intermediate steps. The amplitude in each state is the square-root of the probability and hence equal to 1/sqrt(N). Therefore, after each iteration the total amplitude in the target state increases by about 1/sqrt(N), and hence in about sqrt(N) iterations, the total amplitude in the t state rises to about 1.

To appreciate the power of the search algorithm, consider a database that contains a million items—let us say, the telephone book for a fair-sized city. In the phone book, the entries are sorted alphabetically by name. Someone knew a particular number and wanted to find the name associated with this number. Since the database is not sorted by phone numbers, a classical computer would typically require 500,000 queries to the database (half the number of total entries) to find the correct match. A quantum computer using the search algorithm, by examining multiple items simultaneously and using the power of quantum interference is able to identify the correct entry in only about 1,000 queries to the database (the square root of the number of total entries), a fairly remarkable improvement in speed. Of course this would only be possible if the database was available on a quantum computer. This improvement becomes more and more dramatic as the size of the database increases.

Can we do better than this? The somewhat surprising answer is no. It has been analytically proved based on fundamental properties of quantum mechanical transformations, that it is not possible to improve the quantum search algorithm even by a little bit. This proof is considerably more complicated than the analysis of the algorithm and there is no simple explanation why this is the case.

The search algorithm was originally designed for the exhaustive search problem, it has been shown to be surprisingly versatile and indeed it can yield a square-root advantage for problems from several different fields. To be able to implement the search algorithm for a real-life problem (let us say the database of all telephone numbers in the United States) requires a quantum computer with hundreds of qubits. At present, such a large, many-qubit quantum computer does not exist—there are daunting challenges; the largest number of qubits working together is seven, although there are groups which are working on quantum computers with about a dozen qubits.

5.2 Hardware Challenges to Large Quantum Computers

Existing computers use quantum mechanics to design devices to process the information, the information itself is stored classically.

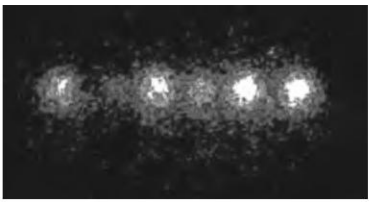
The structure of a quantum computer is different from any known device in the sense that information itself is stored quantum mechanically. No one knows what structure would be most suitable for a quantum computer—this is the focus of intense research. There are a number of possible implementations being considered. Controlled quantum operations have been experimentally demonstrated in the following structures—ion traps, quantum optics using single photons as qubits, solid-state structures and nuclear magnetic resonance (NMR) in organic liquids.

The biggest problem is that qubits are extremely sensitive to the external environment— the slightest disturbance like a stray photon hitting the system causes the calculations to break down. This is known as decoherence. To prevent this, a quantum computer has to be almost completely isolated from its environment. One of the consequences is that the entities of the system must be designed to be microscopic since the decoherence times of macroscopic objects are too short to carry out any controlled quantum operation. For example, a football gets perturbed by stray particles roughly once every 10^{-21} seconds; in contrast, a free electron can stay isolated for almost a day. This leads to the problem that it is difficult to make sub-atomic particles interact in a controlled way. Unlike (classical) VLSI circuits, whose size can be scaled down as semi-conductor fabrication technology becomes more sophisticated, the components of a quantum computer would need to be microscopic right from day one.

In classical computation, error correction techniques are well known (one does not hear very much about these since present-day semi-conductor technology is reliable enough that these are rarely necessary). For a long time it was not clear whether or not it was possible to carry out error correction in quantum computation. Unlike classical bits, qubits cannot be externally observed in intermediate steps of the calculation. However, just in the last five years, it has been shown that it is indeed possible to correct errors in quantum circuits through carefully designed quantum operations. However, the requirements are much more demanding than classical computation. It is possible to design classical error correcting circuits, provided the error rate is less than 1 in 3; in quantum computation, the error rate would need to be less than 1 in 10^4. This is several orders of magnitude more demanding than what today's quantum circuits can deliver.

5.2.1 Ion-traps

One of the first demonstrations of quantum computation was a CNOT gate by the NIST group in which a group of ions trapped by means of a radio frequency ion-trap was used. Information is stored in individual ions using two internal (electronic states). Lasers, individually directed at each ion, are used to accomplish single qubit operations. Different qubits can be made to interact through their collective vibrational motion in the trap following an ingenious proposal by Cirac and Zoller[8] that couples the center of mass motion to the internal energy levels. Fig. 5.1 shows some trapped ion quantum computers.



Six ions confined in a linear rf trap. The four bright ions (#1,#3,#5,#6) are ^{113}Cd , the other two (#2, #4) are ^{111}Cd.

Three trapped $^{112}Cd^{+}$ ions exhibit four modes of oscillation in an asymmetric rf (Paul) ion trap.

Fig. 5.1: Ion Trap Quantum Computer

Bruce Kane's scheme of phosphorus in silicon[9] builds upon modern semi-conductor fabrication and transistor design, drawing upon understood physical properties. Kane proposed that the nuclear spin of a phosphorus atom coupled with an electron embedded in silicon under a high magnetic field and low temperature can be used as a quantum bit, much as nuclear spins in molecules have been shown to be good quantum bits for quantum computation with nuclear magnetic resonance. This quantum bit is classically controlled by a local electric field as shown in Fig. 5.2, where two phosphorus atoms are spaced 15-100 nm apart. This inter-

[8]Cirac and P. Zoller, "Quantum Computations with Cold Trapped Ions", *Phys. Rev. Lett.* 74, 4091 (1995).

[9]Bruce E. Kane, "A Silicon-based Nuclear Spin Quantum Computer", *Nature* 393, May 1998.

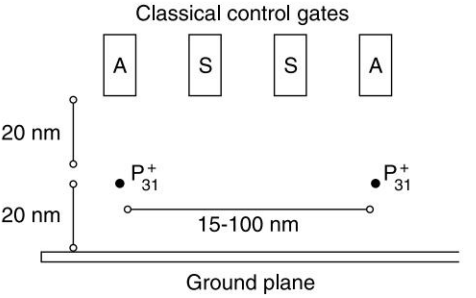


Fig. 5.2: The Basic Quantum Bit Technology Proposed by Kane

qubit spacing is currently a topic of debate within the physics community, with conservative estimates of 15 nm, and more aggressive estimations of 100 nm. What is being traded off is noise immunity versus difficulty of manufacturing.

As many as 20 nanometers above the phosphorus atoms lie three classical wires that are spaced 20 nm apart. By applying precisely timed pulses to these electrodes, arbitrary one and two quantum gates can be realized. Four different sets of pulse signals must be routed to each electrode to implement a universal set of quantum operations.

5.2.2 Solids

After the experience of solid-state integrated circuits, it seems most attractive to think of solid-state quantum computers: the hope is that just as in classical solid-state circuits, we would be able to replicate devices to obtain large circuits. The problem is that the noise (and hence decoherence) problem is most severe in solids. There is a tradeoff as to what to use as our qubits. Spin (magnetic) degrees of freedom are typically much more isolated than charge (electrical) degrees of freedom—as a result, coherence times are much longer. On the other hand, precisely because of this isolation, it becomes difficult to get different spin qubits to interact. There are a number of proposals and ongoing experiments, however, only single qubit operations have been demonstrated in solids, so far.

Bruce Kane suggested an approach using nuclear spins of donor atoms, which are implanted close to the surface in a semiconductor substrate. These spins have naturally long coherence

time; the challenge is to get the different spins to interact. This is arranged by coupling these to a single electron spin. This is a relatively weak interaction overall and places severe demands on the noise tolerance (the system needs to be cooled to extremely low temperatures) and the fabrication technology (the donor spins are separated by about 100 Angstroms).

Another possibility is to use the spin or charge of an electron as qubits—electrons interact more strongly than nuclear spins and are hence easier to couple together. Advances in semi-conductor technology have enabled the fabrication of structures called quantum dots and quantum wells, which can be used to confine and manipulate individual electrons. In an experiment,[10] reported in *Nature*, David Awschalom's group at the University of California at Santa Barbara, has shown that spins can be controlled in a specially designed well like potential landscapes or "quantum well" in a aluminum-gallium-arsenide crystal. The researchers used electrons in quantum wells that were 100 nanometers across. By applying voltages between wells, the flow of electrons between them and their spins could be controlled.

Superconducting structures offer an exciting possibility for solid-state implementations. In a superconductor, the charge carriers decouple from the host solid environment at low temperatures. Two types of structures have been demonstrated, those with the charge as qubits and those with the magnetic flux as the qubits. Due to the strength of the interaction, charge-based structures would be easier to couple together but have shorter coherence times. There were two exciting recent developments in this direction: first Nakamura's group in NEC, Japan, demonstrated coherence times of a few nanoseconds for Cooper pairs in a superconductor;[11] then, Devoret's group in France[12] improved this to about microsecond. The other approach is to use the magnetic flux in superconducting quantum interference devices (SQUIDs) as qubits. SQUIDs are tiny loops of a superconductor that, when exposed to

[10]Y. Kato, R. C. Myers, A. C. Gossard, and D. D. Awschalom, "Coherent Spin Manipulation without Magnetic Fields in Strained Semiconductors", *Nature* 427, 50 (2004).

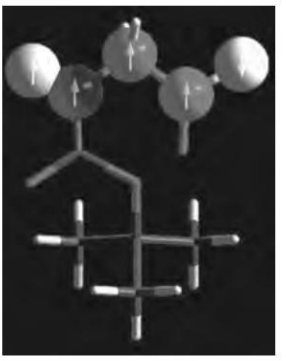
[11]Y. Nakamura et al., "Coherent Control of Macroscopic Quantum State in a Cooper-pair Box", *Nature* 398, 786-788 (1999).

[12]M.H. Devoret et. al., "Single Cooper Pair Electronics", *Applied Superconductivity*, Volume 6, Issues 10-12, October 1999, pp. 491-494.

a magnetic field, carry an electrical current. Recently, such flux quanta qubits have been demonstrated.

5.2.3 NMR in Organic Liquids

This uses an organic molecule as the quantum computer with nuclei of the various atoms constituting the qubits. The unique feature of this scheme is that instead of using a single molecule, it uses an ensemble of molecules in the form of a liquid. The advantage is that it no longer requires ultra-precise operations to be carried out at the sub-atomic level, instead it uses RF pulses to control the interaction between various nuclear spins. This approach easily leads to a few qubit quantum computers. The first implementation of the search algorithm was done by using a NMR quantum computer. In the year 2000, a team at Los Alamos carried out a quantum calculation that used seven qubits. In 2001, a team at MIT-IBM-Stanford also used a seven-qubit structure to implement the factorization algorithm. The Los Alamos team is in the process of building a ten-qubit quantum computer. Fig. 5.3 shows molecular hardware and NMR spectrometer for NMR quantum computing.



Molecular "hardware" of the first five-qubit NMR quantum computer

Nuclear magnetic resonance (NMR) spectrometer at the TU Münch

Fig. 5.3: NMR Quantum Computing

The problem with using an ensemble of molecules is that it is not possible to precisely initialize the ensemble since this is a liquid—hence the fraction of molecules initialized to the proper states drops exponentially with the number of qubits. As a result, this approach may not scale to more than ten qubits or so.

5.2.4 Optics

This scheme uses the polarization states of single photons as qubits. The challenge is to get single photons to interact. Unlike electrons that interact through the Coulomb interaction, it is very difficult to get single photons to interact. This was first demonstrated by Jeff Kimble's group in 1996 by using the Kerr rotation in the presence of Cesium atoms.

An exciting new development is the idea of accomplishing nonlinear effects through the detectors—as a result most of the computation would only require classical linear optics which is well understood. Several groups around the world are working to demonstrate this. The bottleneck is the single photon sources and detectors.

In reality, these hardware approaches will take more time, and it seems unlikely that we will have a quantum computer sitting on our desk in this decade. When a large quantum computer does get built, the potential changes in the software side will be major. Factoring numbers and searching are two known applications where a quantum computer would yield a tremendous advantage; scientists are scrambling to find out what other applications might exist.

5.3 Fabrication, Test, and Architectural Challenges

5.3.1 Fabrication Challenges

The most obvious difficulty in fabricating quantum computers is the small scale of the components and the precision with which they must be placed in the system. Since reliable quantum operations are already challenging, given a fabricated system with perfect spacing and alignment, variations should be minimized and probably need to be detected. Furthermore, the use of quantum operations to test components should also be minimized.

For the Kane technology, the first hurdle is the placement of the phosphorus atoms themselves. The leading work in this area has involved precise ion implantation through masks and manipulation of single atoms on the surface of silicon. For applications where substantial monetary investment is not an issue, slowly

placing a few hundred thousand phosphorus atoms with a probe device may be possible. For bulk manufacturing, the advancement of DNA or other chemical self-assembly techniques may be developed. While new technologies may be developed to enable precise placement, the key is only the spacing (60 nm) of the phosphorus atoms themselves and the number of control lines (3) per qubit. The relative scale of quantum interaction and the classical control of these interactions is what will lead to the fundamental constraints on quantum computing architectures.

A second challenge is the scale of classical control. Each control line into the quantum data path is roughly 10 nm in width. While such wires are difficult to fabricate, it is expected that other electron beam lithography or phase-shifted masks will make such scales possible.

Another challenge is the temperature of the device. In order for the quantum bits to remain stable for a reasonable period of time, the device must be cooled to less than one degree Kelvin. The cooling itself is straightforward, but the effect of the cooling on the classical logic is a problem. There are two issues that arise. Firstly, conventional transistors stop working as the electrons become trapped near their dopant atoms, which fail to ionize. Secondly, the 10 nm classical control lines begin to exhibit quantum-mechanical behavior such as conductance quantization (conductance quantization is a phenomenon associated with quantum transport in nanowires) and interference from ballistic transport.

Many researchers are already working on low-temperature transistors. For instance, single-electron transistors (SETs) are the focus of the intense research due to their high density and low power properties. SETs, however, have been problematic for conventional computing because they are sensitive to noise and operate best at low temperatures. For quantum computing, this predilection for low temperatures is exactly what is needed.

5.3.2 Testing Challenges

Once fabricated, qubits and control will be difficult to test. Tolerances are tight, and it may be necessary to avoid using qubits in the system that are spaced incorrectly or have control signals that are misaligned.

It is likely that the most effective test for the spacing of control signals is to inspect, using an SEM or other device, the pattern of small 10 nm vias (channels) above each ion before they are covered by subsequent layers of metal. Connectivity from wide control wires to the vias will have to be verified via a quantum test program.

The spacing and alignment of the ions that implement the qubits is also problematic. Defects could be caught via quantum test programs, but the test would have difficulty distinguishing between ion spacing errors, misalignment between control vias and ions, and control via spacing errors. Efficient test patterns will be needed to test individual qubits and the two-qubit operations between neighbouring qubits.

5.3.3 Architectural Challenges

There are two fundamental difficulties in taking these individual quantum components and structuring them into a working, large-scale quantum computer. Firstly, because quantum computing requires very low temperatures, and classical circuits (required for control of quantum gates) are designed for higher temperatures, the design must be adjusted to allow classical circuits to work at very low temperatures. Secondly, because quantum operations are so error-prone, and error correction circuits themselves so large, a reliable communication mechanism is required.

The quantum mechanical behavior of the control lines presents a subtle challenge that is often overlooked. At low temperatures and in narrow wires the quantum nature of electrons begins to dominate over normal classical behavior. For example, in 100 nm wide polysilicon wires at 100 millikelvin, electrons propagate ballistically like waves, through only one conductance channel, which has an impedance given by the quantum of resistance, h/e^2 $\approx$ 25 kΩ. Impedance matches to these and similar metallic wires make it impossible to properly drive the AC current necessary to perform qubit operations.

Avoiding such limitations mandates a geometric design constraint: narrow wires must be short and locally driven by nearby wide wires. By using 100 nm as a rule of thumb for a minimum metallic wire width sufficient to avoid undesired quantum behavior at these

low temperatures, we obtain a control gate structure such as shown in Fig. 5.4. Here, wide wires terminate in 10 nm vias that act as local gates above individual phosphorus atoms. Note how access lines quickly taper into upper layers of metal and into control areas of a classical scale. These control areas can then be routed to access transistors that can gate on and off the frequencies (in the 10's to 100's of MHz) required to apply specific quantum gates.

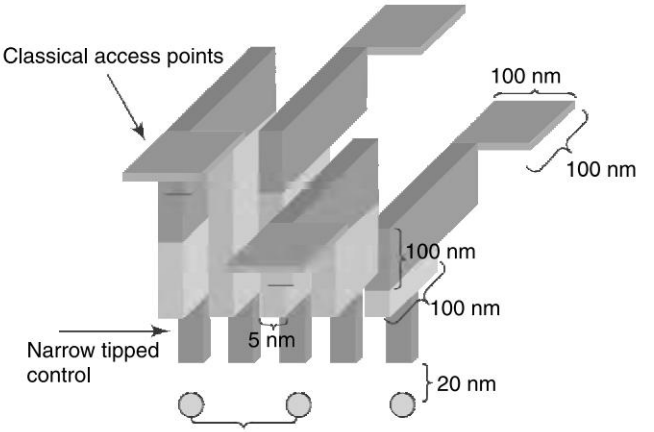


Fig. 5.4: Control Gate Structure

5.4 Quantum-dot Cellular Automata (QCA)

This section outlines the basic concepts of quantum-dot cellular automata (QCA) for familiarization with this new nanotechnology, and will develop the concepts to help design and simulate our own Quantum-dot Cellular Automata (QCA) circuits by using QCADesigner, a free design and simulation tool in the nest chapter.

5.4.1 Background

The seemingly endless progress of microelectronics has been a result of the semi-conductor industry's ability to continuously scale down the transistor, which is the fundamental computing component of the modern computer. Clearly, this scaling cannot continue forever. One of the challenges to continuous transistor scaling is being noticed even today, namely leakage currents through the gate oxide. This leakage current results from quantum mechanical

tunneling of electrons from the gate electrode through the oxide and into the transistor channel. As transistors continue to shrink, more and more of these quantum effects will start to overwhelm their operation. Because of the legacy of the transistor, researchers are trying many different approaches to maintain the functionality of the transistor at ever smaller scales.

Instead of continuously fighting to maintain transistor functionality at smaller scales we wish to find a device that works on a different principle, such that it gets better as its features are reduced rather than worse, like the transistor. A novel idea has been proposed originally in 1993 by researcher, Craig Lent at the University of Notre Dame that may just fit the bill. This technology, called QCA, consists of planar arrays of so-called QCA cells. These cells have features on the very low nanometer scale, much smaller than the smallest transistor, and actually get better as the features are reduced. These devices rely on the quantum mechanical effects such as electron tunneling that are starting to hinder transistor operation.

5.4.2 What is a Quantum-Dot?

Quantum dots are nanostructures created from standard semi-conductive materials such as InAs/GaAs. These structures can be modeled as three-dimensional quantum wells. As a result, they exhibit energy quantization effects even at distances several hundred times larger than the material system lattice constant.

A quantum dot can indeed be visualized as a well. Electrons, once trapped inside the dot, do not alone possess the energy required to escape. We can use quantum physics to our advantage because the smaller a quantum dot is physically, the higher the potential energy necessary for an electron to escape. Fig. 5.5 shows an example of a quantum dot.

5.4.3 Quantum-Dot Cellular Automata

QCA is a novel nanotechnology that attempts to create general computational functionality at the nanoscale by controlling the position of single electrons. The fundamental unit of QCA is the

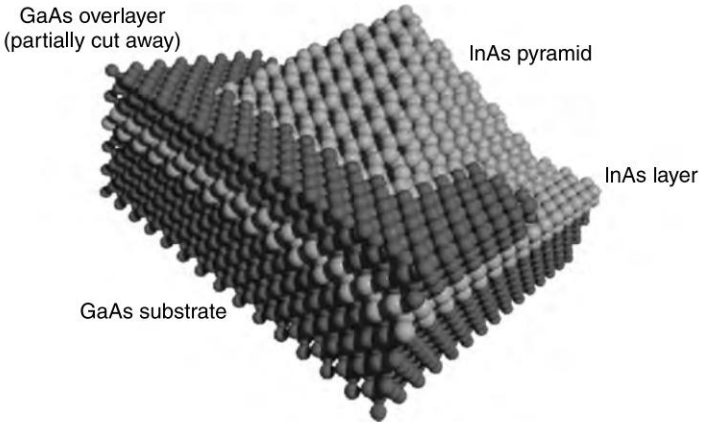


Fig. 5.5: Example Quantum Dot Pyramid Created with InAs/GaAs.
(Photo from: University of Newcastle: Condensed Matter Group)

QCA cell created with four quantum-dots positioned at the vertices of a square as shown in Fig. 5.6. The cell is loaded with two extra electrons which tend to occupy the diagonals of the cell. Binary information is encoded in the two possible polarizations. The cell will switch from one polarization to the other when the electrons quantum mechanically tunnel from one set of dots to the other.

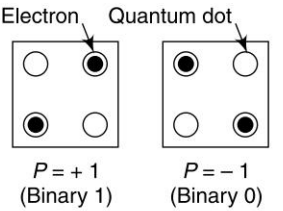


Fig. 5.6: Two QCA Cells Showing the Four Quantum Dots Arranged in a Square Pattern

The bounding box shown around the cell is used only to identify one cell from another; they do not represent any physical system. Because the electrons are quantum mechanical particles they are able to tunnel between the dots in a cell. The electrons in cells placed adjacent to each other will interact. As a result, the polarization of one cell will be directly affected by the polarization of its neighbouring cells. This interaction is shown in Fig. 5.7 with the corresponding non-linear cell-to-cell response function. The first

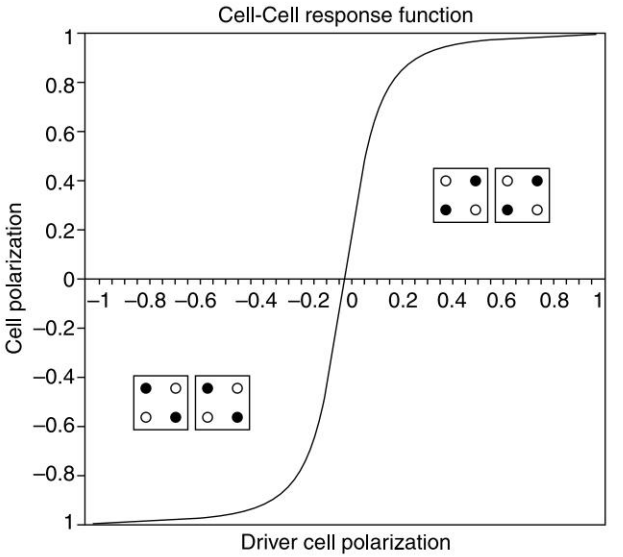


Fig. 5.7: The Non-linear Response Function of One Cell on to its Neighbor

cell acts as a driver and its polarization is varied from -1 to 1. The graph shows the resulting polarization of its neighbor. It can be seen that the driver cell will force an almost complete polarization in its neighbor even if its own polarization is not saturated.

This interaction forces neighboring cells to synchronize their polarization. Therefore, an array of QCA cells acts as a wire and is able to transmit information from one end to another; i.e. all the cells in the wire will switch their polarizations to follow that of the input or driver cell (Fig. 5.8). The cells in the wire will synchronize their polarization to follow the input or driver cell. In this way, information arriving at the input is reflected at the output after some short propagation delay.

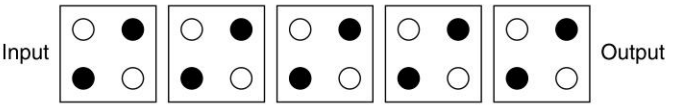


Fig. 5.8: QCA Cells Lined up in this Way Create a QCA Wire

5.5 Computing with QCA

How exactly do we build something that can perform computing using these cells? In order to perform general computation we

require a universally complete computing logic set. In other words, we need a set of Boolean logic gates that can perform the AND, OR, NOT, and FANOUT operations. The combination of these functions is considered universal because any general Boolean function can be implemented with a combination of these logic primitives. The fundamental logic primitive that can be created with QCA is a majority gate or majority voter shown in Fig. 5.9. The output of the majority gate reflects the majority of the inputs.

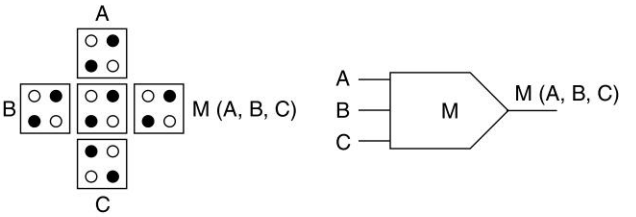


Fig. 5.9: QCA Majority Gate or Majority Voter

The truth table for the majority gate is shown in Table 5.1.

Table 5.1: Truth Table for Fig. 5.9

ABC	M
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

The majority gate produces an output which reflects the majority of the inputs. The majority function is a part of a larger group of functions called threshold functions in which a certain threshold number of the inputs must be set to '1' before the output produces a '1'. Threshold functions are commonly used in the construction of neural networks, where the sum of inputs must reach a certain threshold before the output is asserted. This makes QCA potentially interesting with respect to Neural Network design although this has not yet been fully explored.

Creating a majority gate using CMOS consumes several transistors, and as a result they have not gained much popularity. With QCA, the majority function is the most fundamental logic gate, and

it turns out that AND and OR gates are easily created using a majority. In order to create an AND gate we simply fix one of the majority gate inputs to 0 ($P = -1$). To create an OR gate we fix one of the inputs to 1 ($P = +1$) as shown in Fig. 5.10. By fixing one of the inputs to the majority gate to 0 ($P = -1$) we create an AND gate. Alternatively, if we fix one of the inputs to a 1 ($P = +1$) we produce an OR gate.

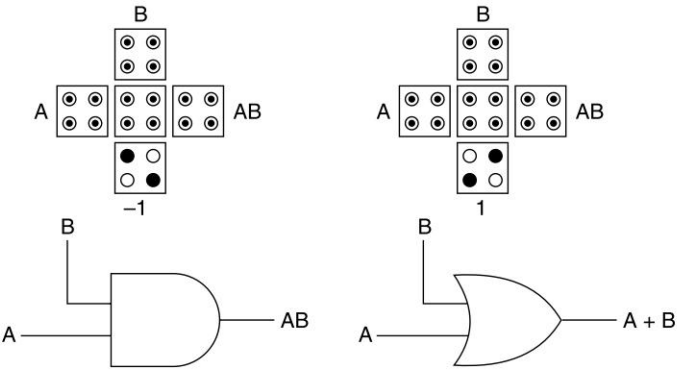


Fig. 5.10: An OR Gate

The inverter or NOT gate is also simple to implement using QCA. It turns out that if you place two cells at 45 degrees with respect to each other they interact inversely. Fig. 5.11. shows one of many different QCA inverter layouts. The cells positioned at 45 degrees with respect to each other interact inversely; i.e. their polarization is always inverted.

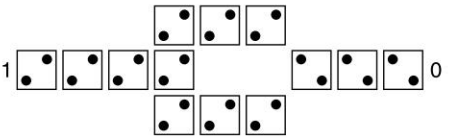


Fig. 5.11: QCA NOT or Inverter Gate

The last thing we need is FANOUT; i.e. one signal comes in and several copies go out. In standard electronic circuits a FANOUT is just a connection of several metal wires. It is therefore taken for granted but other technologies may not have this important capability. Fortunately for us, the FANOUT is no problem to create with QCA. In Fig. 5.12, we see the FANOUT is just the opposite of a majority gate.

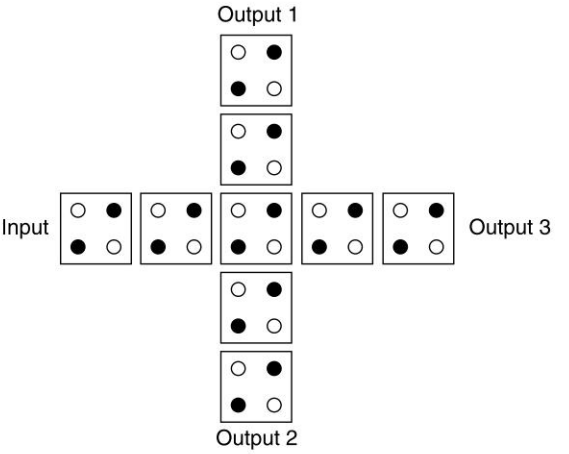


Fig. 5.12: QCA Fanout. The Input Signal Appears at Each of the Outputs

What is the difference between the majority and the fanout? And how do the circuits tell which one is which? They don't. These circuits are exactly the same, and the only way to tell is to know which way signals travel in a circuit as discussed below. If there are three input signals which collide at the center cell, then a majority function will be computed. If, on the other hand, one input arrives at the center cell, then its value will be copied to the outputs. The flow of information in a QCA circuit determines everything.

There is another interesting possibility which involves rotating the dots in a cell by 45 degrees. If you consider the interaction of such cells you will find that they interact inversely (note: this is not the same as the inverter where the cells were placed at 45 degrees. Here the dots in one cell are rotated). As a result, a wire created with these 45 degree cells forms what is called an 'inversion chain' and each cell takes on the opposite polarization of its neighbours as shown in Fig. 5.13. Each cell in the chain takes on the opposite polarization of its nearest neighbours.

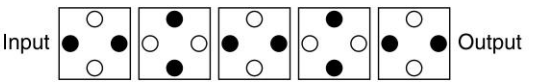


Fig. 5.13: QCA Inversion Chain

Interestingly, when a wire of regular cells crosses a wire of these 45 degree cells, the two wires do not interact! In other words, we

can cross signals directly over each other as shown in Fig. 5.14. The information traveling along the vertical wire does not interact with the horizontal wire.

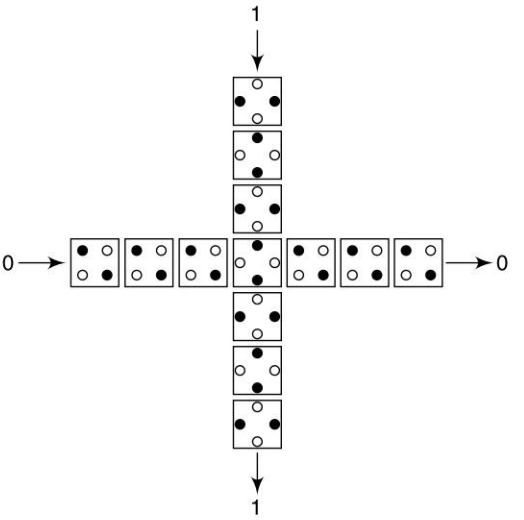


Fig. 5.14: QCA Crossover

5.6 QCA Clocking

Clocking is important in most computational technologies and a requirement for the synchronization of information flow in QCA. Several techniques are available for asynchronous computation, but are far less common than their synchronous counterparts. Presently, all QCA circuit proposals require a clock not only to synchronize and control information flow but the clock actually provides the power to run the circuit. The cells are not powered from any other external source apart from the clock. Therefore it is difficult to imagine a QCA circuit that can avoid using a clock.

QCA clocks have been proposed to control the potential barriers between the quantum dots. As such, they control the rate at which electrons are able to quantum mechanically tunnel between the dots in the cell and therefore switch the polarization of the cell. When the clock signal is **high** the potential barriers between the dots are **low** and the electrons effectively spread out in the cell and no net polarization exists; that is, $P = 0$. As the clock signal is switched **low**, the potential barriers between the dots are raised

high and the electrons are localized such that a polarization is developed based on the interaction of their neighbours; that is, they take on the polarization of their neighbours. Basically all one has to remember is that clock high means cell is unlatched, clock low means cell is latched.

In order to pump information down a circuit in a controllable manner four clocking zones are available as shown in Fig. 5.15. Each clock signal lags in phase by 90 degrees with respect to the one before it. This way, the cells connected to successive clocking signals are latched in series and propagate information in the same direction; that is, *C*0-*C*1-*C*2-*C*3-*C*0-*C*2…

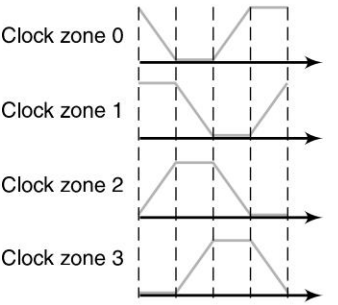


Fig. 5.15: 4 QCA Clocking Signals/Zones

Therefore, to pump information down lets say a QCA wire, we connect different parts of the wire to the different clock signals. Figure 5.16 shows a wire connected to different clock zones. The decreasing shades of gray represent increasing clocking zones. Since the cells in one clock zone get latched and stay latched until the next group of cells gets latched, they can be considered a *D*-latch. This is not a regular *D*-latch because a group of cells connected to *C*1 will only transmit information to cells connected to *C*2, never *C*0 nor *C*3! We see that information flows from one

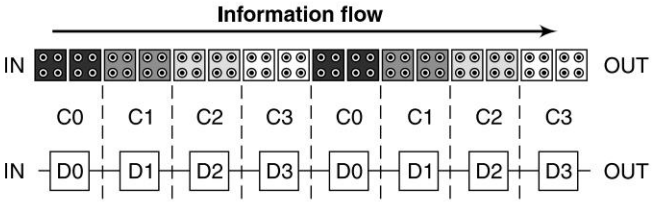


Fig. 5.16: Clocked Wire. Each Group of Cells Connected to a Particular Clocking Zone can be Described Schematically as a D-latch

clocking zone to another. In this snapshot *C*2 cells are latched while others are in the relaxed state. As shown in Fig. 5.16 we can number these *D*-latches with the appropriate clock zone to obtain a schematic representation of the QCA wire.

5.7 QCA Design Rules

Some initial guides for QCA design rules are:

- Minimum width to guarantee flow: There is no direct analog in molecular QCA circuits until you begin to consider "thicker" wires. As with simple cell interactions, the error rates associated with molecular wires will be affected by the amount of energy required to excite one cell in it into a mistake state. Cell placements, stray particles, etc., can all contribute to kink energy.
- Minimum wire spacing for separations: As with metal wires in CMOS circuits, molecular QCA wires will also have to be a certain distance apart from one another to ensure that there is no cross talk or short circuits between them. Distance between individual cells in a wire will also have to be defined to ensure that a value is propagated. Additionally, clock wires must be laid out as well to generate required electric fields.
- Overlap rules to create devices and contacts: When considering overlap, we must ensure that all cells are "clocked" by an electric field and thus space silicon wires accordingly. QCA analogs to overlap also include crossovers between 45-degree and 90-degree cells and the inputs of a majority gate.

As an example design rule for molecular QCA, consider spacing between two molecular QCA cells. Specifically what is the maximum allowed and minimum required distance between two cells such that they will still transmit data? In Figure 5.17, these distances are labeled as x_{max} and x_{min} and specific values would be

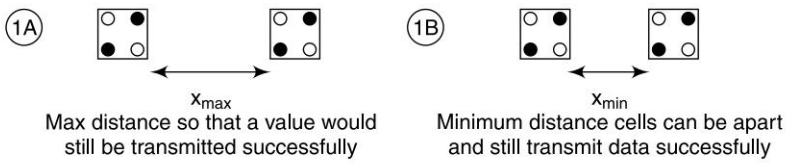


Fig. 5.17: Design Rules 1A and 1B for Molecular QCA Cells Spacing

governed by: substrates to which QCA cells can attach, E_{kink} (that is, background charge with energy of interaction proportional to $1/d^2$ could cause it), and dipole interactions between cells (proportional to $1/d^3$. Also, x_{min} will provide an initial upper bound on maximum device densities.

5.7.1 QCA CAD and Placement

Nanotechnology and devices will have revolutionary impact on the computer-aided design (CAD) field. Similarly, CAD research at circuit, logic and architectural levels for nano devices can provide valuable feedbacks to nano research and illuminate ways for developing new nano devices. CAD can help research to move from small circuits to small systems of QCA devices. It is time for CAD researchers to play an active role in nano research.

As QCA is being considered as an alternative to silicon-based computation, it is appropriate to enumerate what QCA's "wins" over silicon-based systems could be. Table 5.2 lists obstacles to CMOS-based Moore's Law design, their effects on silicon-based systems, and how they will affect QCA.

From the information in Table 5.2, it is apparent that QCA faces some of the same general problems as silicon-based systems (timing issues, lithography resolutions and testing), that QCA does not experience some of the same problems as silicon-based systems (quantum effects and tunneling), and that silicon-based systems can address one problem better than QCA currently can (I/O). However, if the I/O problem is resolved, QCA can potentially offer significant "wins" with regard to reduced power dissipation and fabrication. Additionally, QCA can also offer orders of magnitude in potential density gains when compared to silicon-based systems.

Table 5.2: Comparing Characteristics of Silicon-based Systems to QCA Based Systems

Obstacle	*Effect on CMOS Circuits*	*Effect on QCA*
Quantum effects and tunneling	A gate that controls the flow of electrons in a transistor could allow them to tunnel through small barriers—even if the device is supposed to be off.	No effect: QCA devices are charge containers, not current switches and actually leverage this property.

(*Contd*)

(*Contd*)

Obstacle	*Effect on CMOS Circuits*	*Effect on QCA*
High power dissipation	Chips could melt unless problems are overcome for which the silicon roadmap says, 'there are no known solutions'. 2014 projection: Chip with 10^{10} devices dissipates 186 W of power.	10^{11} QCA devices with 10^{-12} switching times dissipate 100 W of power. QCAs silicon-based clock will also dissipate power. Still, clocking wires should move charge adiabatically, greatly reducing power consumption.
Slow wires	Wires continue to dominate the overall delay. Also, projections show that for feature size of 60 nm, less than 10% of the chip is reachable in 1 clock cycle.	The inherent pipelining cased by the clock makes global communication and signal broadcast difficult. Problems are similar to silicon-based systems but for different reasons.
Lithography resolutions	Shorter wavelengths and larger apertures are needed to provide finer resolutions for decreased feature sizes.	QCA's clock wiring is done lithographically, which is subject to the same constraints as silicon-based systems. However, closely spaced nanowires could also be used.
Chip I/O	I/O count continues to increase as the technology advances (Rent's rule), but pin counts do not scale well. With more processing power, we will need more.	I/O remains under investigation with one approach being to include "sticky ends" at the ends of certain DNA tiles in order to bind nanoparticles or nanowires.
Testing	Even if designs are verified and simulated, defects caused by impurities in the manufacturing process, misalignment, broken interconnections, etc., can all contribute to non-functional chips. Testing does not scale well.	We must find and route around defects caused by self-assembly and/or find new design methodologies to make circuits robust. Defects for self-assembled systems could range from 70-95%. Structures such as thicker wires could help.
Cost	Fabrication facility cost doubles approximately every 4.5 years, and could reach 200 billion dollars in 2015.	Self-assembly could be much more inexpensive.

5.7.2 CMOS vs QCA Placement

Although QCA and CMOS have considerable technological differences, CMOS VLSI placement algorithms have been modified to

satisfy the design constraints imposed by QCA physical science. There are many reasons for using this approach. Notably, VLSI design automation algorithms work on graph-based circuits, and it has been found to be advantageous to represent QCA circuits as graphs—especially because at present, only two-dimensional circuits have been proposed and are seen as technically feasible. Existing algorithms can be fine-tuned to meet QCA's constraints and objectives. Additionally, physical design issues for CMOS have been widely studied, optimized and proved to be NP-complete[13]. This, it makes sense to leverage this existing body of knowledge and apply it to a new problem. Finally, because so few design automation tools and methodologies exist for QCA, using VLSI algorithms as a base will allow us to compare and set standards for QCA place and route methodologies.

Specifically, the following similarities and differences exist between CMOS and QCA placement.

- Similarities: In CMOS placement, partitioning, floor planning, and placement are performed in this order (a hierarchical approach) to efficiently handle the design complexity. A similar approach is used in QCA placement: zone partitioning, zone placement, and cell placement. The following objectives are common in both CMOS and QCA partitioning and work towards achieving the same goal: cut size and performance. The area, performance, congestion, and wire length objectives are common to both CMOS and QCA placement.
- Differences: Two major differences are QCA clocking and QCA single-layer routing resource. Minimizing the total number of QCA wire crossings is critical in QCA placement as QCA layouts must be done in a single layer (unlike the multi-layer CMOS layout). This node duplication in CMOS targets area and performance optimizations while QCA duplication targets minimizing wire crossings to conform to QCA's clocking requirements.

[13]In computational complexity theory, the complexity class NP-complete, is a subset of NP ("non-deterministic polynomial time"). Problems are designated "NP-complete" if their solutions can be quickly checked for correctness, and if the same solving algorithm used can solve all other NP problems.

CHAPTER 6

QCADesigner Software and QCA Implementation

The seemingly endless progress of microelectronics has been a result of the semi-conductor industry's ability to continuously scale down the transistor, which is the fundamental computing component of the modern computer. Clearly, this scaling cannot continue forever. One of the challenges to continued transistor scaling is being noticed even today, namely leakage currents through the gate oxide. This leakage current results from quantum mechanical tunneling of electrons from the gate electrode through the oxide and into the transistor channel. As transistors continue to shrink, more and more of these quantum effects will start to overwhelm their operation. Because of the legacy of the transistor, researchers are trying many different approaches to maintain the functionality of the transistor at ever smaller scales.

Instead of continuously fighting to maintain transistor functionality at smaller scales we wish to find a device that works on a different principle, such that it gets better as its features are reduced rather than worse, like the transistor. A novel idea has been proposed originally by Craig Lent at the University of Notre Dame that may just fit the bill. This technology, called quantum-dot cellular automata or QCA, consists of planar arrays of so called QCA cells. These cells have features on the very low nanometer scale, much smaller than the smallest transistor, and actually get better as the features are reduced. As well, these devices rely on the quantum mechanical effects such as electron tunneling that are starting to hinder transistor operation.

QCA is an emerging nanotechnology concept for the realization of a computer built with arrays of nanoscale QCA cells. These QCA cells are capable of performing all complex computational functions required for general-purpose computation. QCA has been listed as one of the six emerging nanotechnologies with applications in future computers by the International Technology Roadmap for Semi-conductors (ITRS). QCADesigner facilitates rapid design, layout, and simulation of QCA circuits by providing powerful CAD features available in more complex circuit design tools. Fig. 6.1 shows a QCADesigner screenshot showing a simple 4-bit processor layout

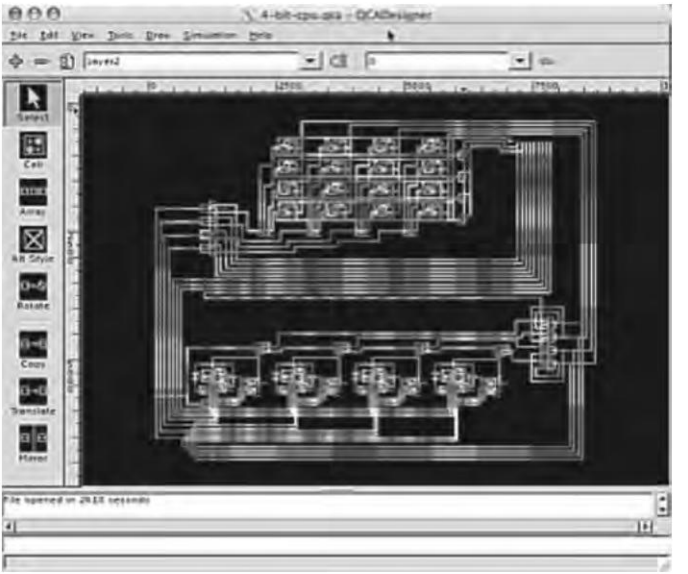


Fig. 6.1: QCADesigner Screenshot Showing a Simple 4-bit Processor Layout

6.1 Basic QCA Circuits using QCADesigner

By combining the basic circuit elements described above we can in fact create general purpose computing circuits. The circuits and simulation results shown in this chapter have been created by using QCADesigner.

6.1.1 QCA Full-adder

The full-adder is one of the most fundamental arithmetic building blocks. The adder was one of the first complex circuits designed with QCA originally by the Notre Dame group. Recently, they have shown how to create an adder with fewer majority gates. This layout for the full-adder is shown in Fig. 6.2.

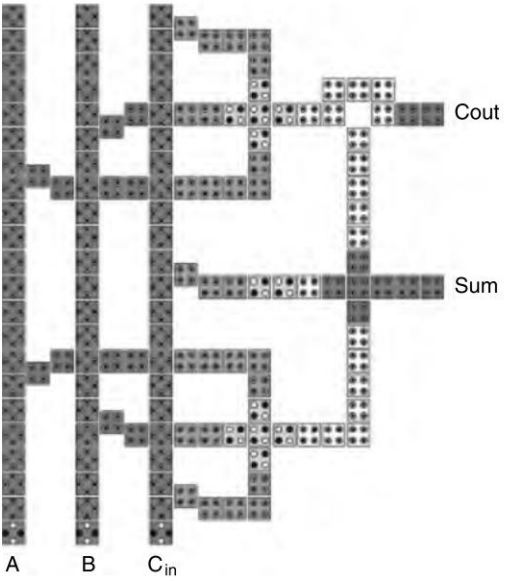


Fig. 6.2: Layout of QCA Full-adder
(The different shades of gray represent connections to the different clock phases.)

Alternatively, we can represent this full-adder schematically as shown in Fig. 6.3. The number next to the D represents the clocking zone to which the cells making up the D-latch are connected to.

We can simulate this circuit in QCADesigner. The simulation results using the bistable simulation engine are shown in Fig. 6.4. The first three curves represent the input waveforms here, exhaustively tested. The next two represent the outputs "Sum" and "Cout" or carry output. Notice that each time the clock C0 goes low the output cells get latched to some value "$P = -1$" or "$P = 1$". The first time the outputs are latched they fall to a random polarization

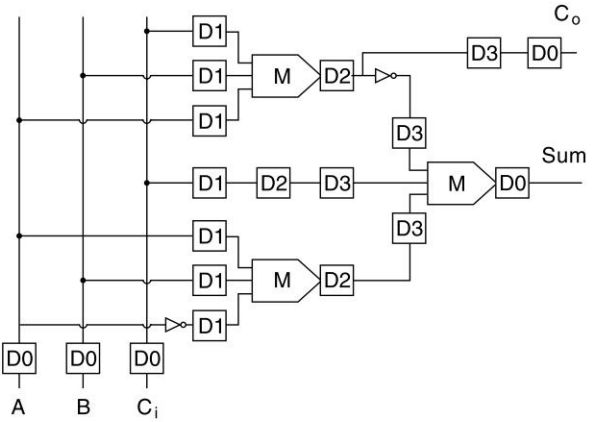


Fig. 6.3: QCA D-latch Schematic of Full-adder

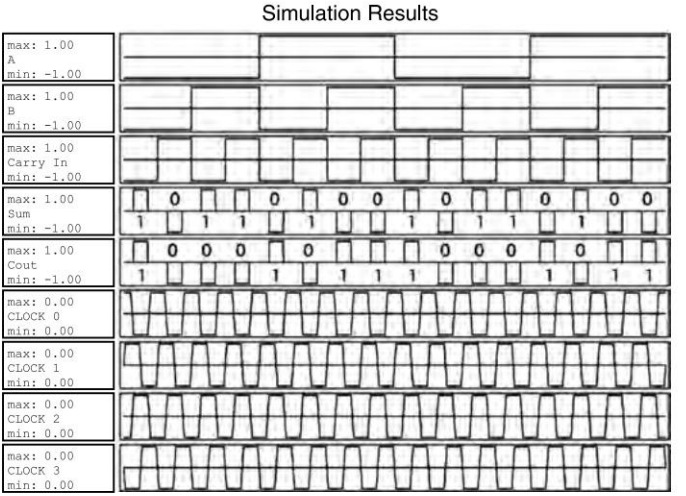


Fig. 6.4: QCADesigner Simulation Results for the Full-adder in Fig. 6.3

(in this case Sum = 1 and Cout = 1) not determined by the inputs because the inputs have not had a chance to propagate through the circuit yet. Therefore, the correct output is delayed by 1 clock cycle with respect to the input.

If we examine the outputs and take into consideration the one cycle delay, we will find that the circuit does replicate full-adder functionality as given in the truth table in Table 6.1.

Table 6.1: Truth Table for Full-adder

ABCin	*Sum*	*Cout*
000	0	0
001	1	0
010	1	0
011	0	1
100	1	0
101	0	1
110	0	1
111	1	1

6.2 QCA Implementation

The concept of a QCA cell is generic in that it can be implemented in several different ways and there have been proposals for:

- Semi-conductor implementation (discussed here).
- Molecular implementation.
- Magnetic implementation.

Each of these implementations has certain advantages/disadvantages and none have yet been completely developed.

The semi-conductor implementation is advantageous because of the success of semi-conductors in microelectronics for which many tools and techniques have been developed. It would be easier to use existing facilities and methods to create a viable QCA solution. Unfortunately, the dot size required to create room temperature devices is still outside the range of semi-conductor fabrication techniques.

The molecular implementation benefits from the high regularity of individual molecules and the small dots size (redox sites) available with molecules. Molecular QCA has the potential to operate easily at room temperature and at very high operating speed. As well, because of their extremely small size, it is easy to imagine that the circuit device density would by very large, close to some fundamental limits. The problem is that the placement of molecules required to create large circuits is probably still far off.

The magnetic implementation uses nanoscale magnets to act as the cells, and encodes the polarization in the magnetic vector of each of the nanomagnets. It has been shown that these nanomagnetic QCA could easily operate at room temperature and are within present fabrication techniques. Unfortunately, magnetic

QCA does not appear to have the necessary switching speed to compete with today's computers but may be an alternative for creating memory.

6.2.1 The Basic Device and Circuit Elements

A high level diagram of a "candidate" four-dot metal QCA cell is shown in Fig. 6.5. It shows four quantum dots that are positioned to form a square. Quantum dots are small semi-conductor or metal islands with a diameter that is small enough to make their charging energy greater than k_bT, where k_b is Boltzmann's constant and T is the operating temperature. The charging energy is the potential energy needed to overcome the electrostatic repulsion from the other electrons in the dot—or in other words, the energy required to add an electron to a dot. If this energy is greater than the thermal energy of the environment (k_bT), dots can trap individual charges.

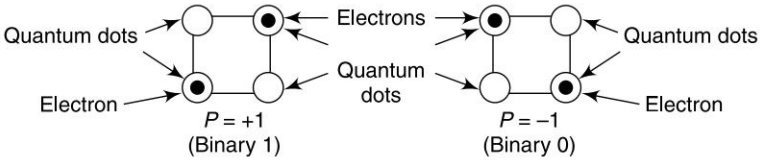


Fig. 6.5: QCA Cell Polarizations and Representations of Binary 1 and Binary 0

Exactly two mobile electrons are loaded into this cell and can move to different quantum dots by means of electron tunneling. Tunneling paths are represented by the lines connecting the quantum dots in Fig. 6.5. Coulombic repulsion will cause "classical" models of the electrons to occupy only the corners of the QCA cell, resulting in two specific polarizations. These polarizations are configurations where electrons are as far apart from one another as possible, in an energetically minimal position, without escaping the confines of the cell. Here, electron tunneling is assumed to be completely controllable by potential barriers that can be raised and lowered between adjacent QCA cells by means of capacitive plates parallel to the plane of the dots.

In addition to these two "polarized" states, there also exists a decidedly non-classical unpolarized state. Briefly, in an unpolarized

state, inter-dot potential barriers are lowered to a point which removes the confinement of the electrons on the individual quantum dots, and the cells exhibit little or no polarization as the wave functions of the two electrons smear themselves across the cell.

6.2.2 The Majority Gate

The fundamental QCA logical gate is the three-input majority gate which appears in Fig. 6.6. Computation is performed with a majority gate by driving the device cell (Cell 4) to its lowest energy state, which will occur when it assumes the polarization of the majority of the three input cells (1, 2 and 3). We define an input cell simply as one that is changed by a logical signal propagating toward the device cell. The device cell will always assume the majority of the polarizations of the input cells because in that polarization, the electrostatic repulsion between the electrons in the three input cells and the electrons in the device cell will be at a minimum.

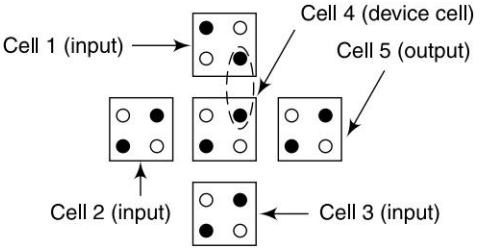


Fig. 6.6: The Fundamental QCA Logical Device—the Majority Gate

6.2.3 A Wire

Figure 6.7 shows a "90-degree" wire. The wire is called "90-degrees" as the cells from which it is made are oriented at a right angle. The wire is a horizontal row of QCA cells and a binary signal propagates from left to right because of electrostatic interactions. Initially, cell 1 has polarization $P = -1$ and cell 2 has polarization $P = +1$. It is assumed that charges in cell 1 are trapped in polarization $P = -1$ but those in cells 2-9 are not. Because the driving cell is "trapped", there is no danger that this wire could "reverse directions" and have a polarization propagate in a direction from which it came. Initially, electron repulsion between cell

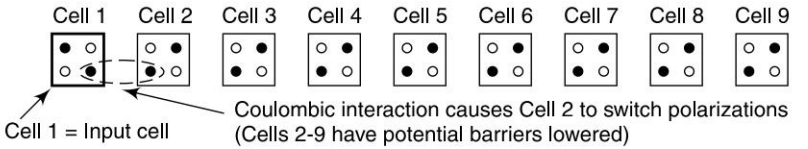


Fig. 6.7: A QCA "wire"

1 and 2 will cause cell 2 to change polarizations. Then, electron repulsion between cell 2 and 3 will cause cell 3 to change polarizations. This process will continue down the length of the QCA "wire". When electrons in all the cells settle in an energetically minimal position, it implies that the system is in a ground state. Energetically minimal positions simply mean that electrons are in positions such that the Coulombic repulsions between them are as low as possible.

6.2.4 A 45-degree Wire

It is also possible to form what is called a "45-degree" wire. As shown in Fig. 6.8, a binary value propagates down the length of such a wire, alternating between polarization $P = +1$ and polarization $P = -1$. It is this orientation of electrons within QCA cells that represents the minimum energy configuration for each cell. Interestingly, with this orientation of wire, both a complemented or uncomplemented signal value can be ripped off of the wire by placing a 90-degree "ripper" cell at the proper location between 45-degree cells (Fig. 6.9).

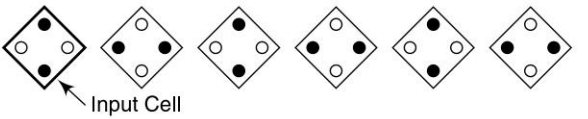


Fig. 6.8: A 45-degree Wire

6.2.5 Off-centre Wires

In theory, QCA cells do not have to be exactly aligned to transmit binary signals correctly. Cells with a 90-degree orientation could be placed next to one another but off-centre, and a binary value could still be transmitted successfully (Fig. 6.10). However, successful transmission is subject to the exact positioning of the off-centred cell.

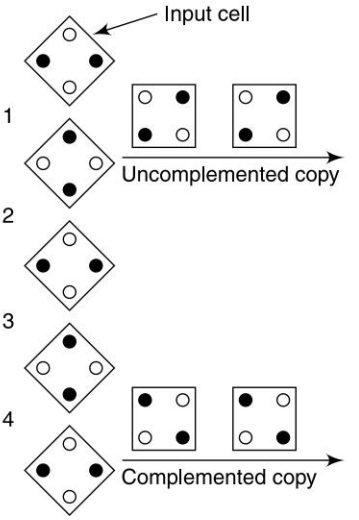


Fig. 6.9: Ripping off a Binary 0 and 1 from a 45-degree Wire

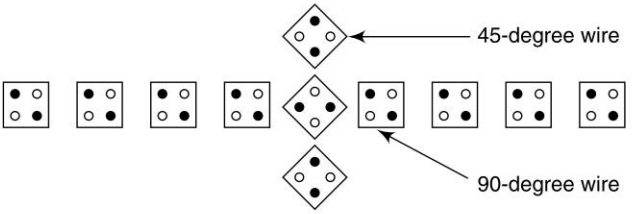


Fig. 6.10: An Off-centre Binary Wire

6.2.6 Wire Crossings in the Plane

QCA wires possess the unique property that they are able to cross in the plane without the destruction of a value being transmitted on either wire. However, this property will hold only if the QCA wires are of different orientations such that one wire is comprised of 45-degree cells and another comprised of 90-degree cells (Fig. 6.11). However, while theory tells us that this property should

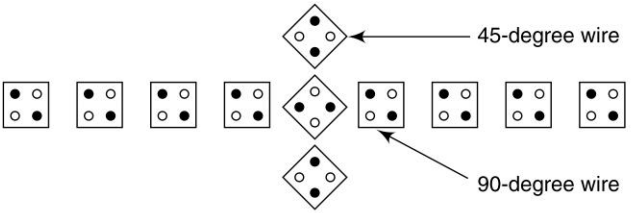


Fig. 6.11: Two Wires Crossing in the Plane

hold, the problem of engineering devices to realize such functionality has not yet been completely solved.

Example 6.1

It would be possible to implement certain circuits in QCA with just majority gates (at least without inverters). A value's complement can be obtained simply by ripping a signal value off of a 45-degree wire at the proper location. Implementing the logical AND and OR functions is also quite simple. The logical function the majority gate performs is $Y = AB + BC + AC$.

The AND function can be implemented by setting one value (A, B or C) in the majority gate equation to a logical 0. Similarly, the OR function can be implemented by setting values (A, B or C) in the majority gate equation to a logical 1. Similarly, the OR function can be implemented by setting one value (A, B or C) in the majority gate equation to a logical 1. This results in the logical AND/OR equations. It is worth noting that because this property exists, and given the fact that it is possible to obtain the inverse of a signal value, the QCA logic set is functionally complete, and any logical circuit can theoretically be generated with only QCA devices.

More complex logical circuits (such as the multiplexer in Fig. 6.12) can then be constructed from majority-gate converted AND gates,

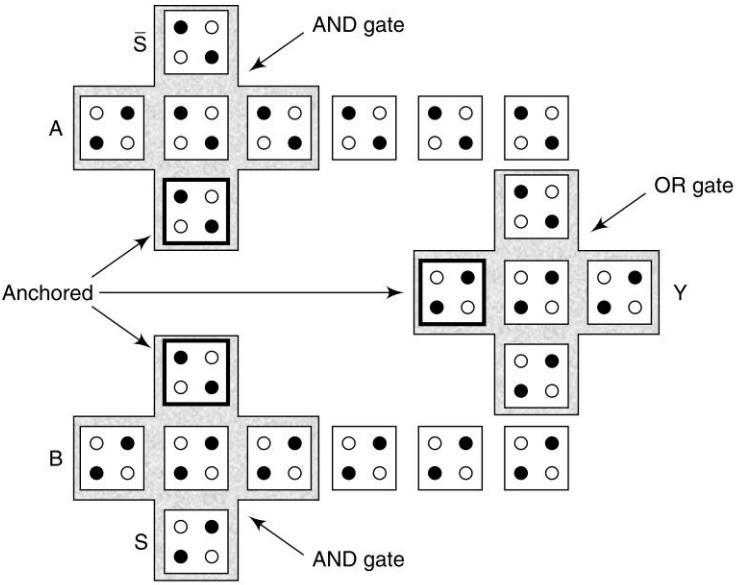


Fig. 6.12: A 2 × 1 Multiplexer with Logical Equation $Y = AS' + BS$

OR gates, and inverters, if not more clever combinations of simply majority gates. In this figure, QCA cells labeled "anchored" are considered to have their electron polarization permanently frozen to successfully implement the AND and OR functions.

CHAPTER 7

Molecular and Optical Computing

7.1 Molecular Computing

The notion of harnessing individual molecules at nanoscales for computational purposes is an idea that can be traced back at least to the time when electronic computers were being constructed in the 1940s. Electrons are, in fact, orders of magnitude smaller than molecules but over 10^{15} are required just to communicate a carriage return to a conventional processor. The idea of improving the efficiency of hardware utilization by using biomolecules is attractive for several reasons. Firstly, hardware is inherently parallel, and parallelism is a good way to handle computational bottlenecks. Secondly, biomolecules occur abundantly in nature, for example, inside all known living cells with (eukaryote) and without (prokaryote) nuclei, and constitute the basic substratum of life. Consequently, they have developed a structure that enables them to solve a number of difficulties for parallel computing, such as massive communication over noisy media and load balancing problems, by mechanism that we may not even be aware of. Furthermore, short biomolecules can now be synthesized at low cost. Thirdly, their physical implementation is therefore relatively simple compared to the demanding and costly fabrication processes used in VLSI. Consider the following figures in terms of sheer space, not to mention performance. A human brain consists of about 10^{12} neurons, and a human body comprises over 10^{15} cells; each cell contains a copy of the entire genetic code consisting of over three billion nucleotide pairs to perform living functions, all that nicely packed in a few double helices about 3.4 mm wide and

a few microns long. Therefore, computing based on molecular media would really tip the scales in terms of miniaturization. On the other hand, these advantages are obtained at the expense of complications that are non-issues for conventional computers, as will be seen presently. The basic problem for computation remains: how to trick a piece of matter (biomolecules in this case) evolved to have a "mind" of its own following predetermined physical and/or chemical laws, to perform an anthropomorphic task typical of what we understand today as computation?

The use of molecules for electronic devices was suggested in 1974 in a seminal paper by Avi Aviram of IBM and Mark A. Ratner of North Western University[1]. By tailoring the atomic structures of organic molecules, they proposed, it should be possible to concoct a transistor-like device. But their ideas remained largely theoretical until a recent confluence of advances in chemistry, physics and engineering.

Of all the groups that have turned Aviram and Ratner's ideas into reality, two teams—one at the University of California at Los Angeles and Hewlett-Packard, the other at Yale, Rice and Pennsylvania State University have demonstrated that thousands of molecules clustered together can carry electrons from one metal electrode to another. Each molecule is about 0.5 nm wide and one or more nanometers long. Both groups have shown that the clusters can behave as on/off switches and might thus be usable in computer memory; once on, they will stay on for 10 minutes or so. That may not be a long time, but computer memory typically loses its information instantly when the power is turned off; even when the power is on, the stored information leaks away and must be "refreshed" every 0.2 second or so. Although the details differ, the switching for both molecules is believed to involve a well-understood chemical reaction—oxidation reduction—in which electrons shuffle among atoms within the molecule. The reaction puts a twist in the molecule, blocking electrons as surely as a kink in a hose blocks water.

In the 'on' position, the clusters of molecules may conduct electricity as much as 1,000 times better than in the 'off' position. The ratio is actually rather low compared with that of typical semiconductor transistors, whose conductivity varies a million-fold.

[1]Aviram, A. & Ratner, M.A. "Molecular Rectifiers", *Chem. Phys. Lett.*, 29, 277 (1974).

Researchers are now looking for other molecules with even better switching properties, and are also working to understand the switching process itself. Fig. 7.1 shows a molecular electronic switch.

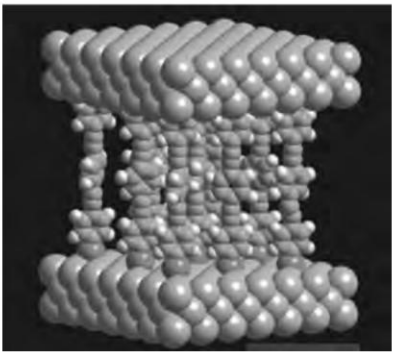


Fig. 7.1: A Molecular Electronic Switch

7.1.1 Brief Background of Molecular Electronics

Molecular electronics uses primarily covalent bonded molecular structures, electrically isolated from a bulk substrate. Devices of this description, wires and switches composed of individual molecules and nanometer-scale supramolecular structures, some times are said to form the basis for 'intramolecular electronics'. This is to distinguish them from organic microscale transistors and other organic devices that use bulk materials and bulk-effect electron transport just like the semi-conductor devices.

7.1.1.1 Molecular Electronic Switching Devices

There are four broad classes of molecular electronic switching devices.

- Electronic field-controlled molecular electronic switching devices include molecular quantum-effect devices.
- Electromechanical molecular electronic devices employing electrically or mechanically applied forces to change the conformation, or to move a switching molecule or group of atoms to turn a current on and off.

- Photoactive/photochromic molecular switching devices which use light to change the shape, orientation, or electron configuration of a molecule in order to switch a current.
- Electrochemical molecular devices which use electrochemical reactions to change the shape, orientation or electron configuration of a molecule and hence to switch a current.

The first two categories of molecular electronic devices, the electric field-controlled molecular electronic switches are most closely descended from the solid-state microelectronics and nanoelectronic devices described in the earlier sections, and promise to be the fastest and most densely integrated of the four categories; the electromechanical molecular switching devices are also promising, since they too could be laid down in a dense network on a solid substrate. Each of the other two categories, while quite promising in general, has a major drawback for use in nanocomputers. Photoactive devices in a dense network would be difficult to switch individually, since light cannot be easily confined on length scales very much below its wavelength (~ 500 nm to 1000 nm). Electrochemical molecular devices would likely require immersion in a solvent to operate.

7.1.2 Origins of Molecular Computing

Recent advances in computer science have been characterized by the computational implementation of well-established biological paradigms. Notable advances are artificial neural nets, inspired by the brain, and its obvious connection to natural intelligence, and evolutionary computation, inspired by the Darwinian paradigm of natural selection. Early ideas of molecular computing attempted to emulate conventional electronic implementations in other media, for example, implementing Boolean gates in a variety of ways. A fundamental breakthrough characteristic of a new era was made by Adleman[2] (1994) where he conducted an experiment performed with molecules of fundamental importance for life, DNA (deoxyribonucleic acid) molecules, to solve a computational problem known to be difficult for ordinary computers, namely the Hamiltonian

[2]Leonard M. Adleman, "Molecular Computation of Solutions to Combinatorial Problem", *Science*, 266: 1021-1024, Nov. 1994.

Path Problem (HPP)[3]. This problem is typical of an elite set of problems in the well-known complexity class NP that exemplify the computational difficulty of search procedures that plague a number of very important applications in combinatorial optimization, operations research, and numerical computation. Adleman's experiment ushered in a new computational paradigm in molecular computing for several reasons. Firstly, it showed that it is indeed possible to orchestrate individual molecules to perform computational tasks. Secondly, it showed the enormous potential of DNA molecules for solving problems beyond the reach of conventional computers that have been or may be developed in the future based on solid-state electronics. Shortly after, in 1995, the first conference on DNA-based computing was organized at Princeton University after which several events have been held.

7.1.3 Some Techniques of Molecular Computing

A. Adleman's Landmark Experiment

The Hamiltonian Path Problem (HPP) is defined as follows:

Instance: A directed graph Γ and two vertices, source and destination;

Question: Yes/no, there is a path following arcs in the graph connecting the source to the destination vertices and passing through each other vertex exactly once.

This problem is NP-complete, i.e., it is representative of many difficulties that afflict conventional computers for solving very important problems in combinatorial optimization and operations research. Each complete problem in NP contains all problems in the class NP as special cases after some rewording, and is characterized by the fact that their solutions are easily verifiable, but extremely difficult to find in a reasonable amount of search time.

[3]Rubin, Frank, "A Search Procedure for Hamilton Paths and Circuits", *Journal of the ACM*, Volume 21, Issue 4. October 1974.

Hamiltonian Path Problem (HPP): In the mathematical field of graph theory, the Hamiltonian path problem and the Hamiltonian cycle problem are problems of determining whether a Hamiltonian path or a Hamiltonian cycle exists in a given graph (whether directed or undirected).

The best-known general techniques to apply to these problems amount essentially to an exhaustive search through all possible solutions, looking for satisfaction of the constraints required by the problem. It is therefore an ideal candidate for a brand new computational approach using molecules.

Adleman's brilliant insight was to carefully arrange a set of DNA molecules so that the chemistry that they naturally follow would perform the brunt of the computational process. The key operations in this chemistry are sticking operations that allow the basic nucleotides of nucleic acids to form larger structures though the processes of ligation and hybridization. Adleman assigned well-chosen unique single-stranded molecules to represent vertices, used Watson-Crick complements of the corresponding halves to represent edges joining two vertices, and synthesized a picomol of each of the 21 resulting molecules. Taking advantage of the fact that molecular biologists have developed an impressive array of technology to manipulate DNA, he designed a molecular protocol (one would say algorithm in computer science) that enabled the molecules to stick together in essentially all the possible ways. The edge molecules splinter nearby vertex molecules to construct longer and longer molecules representing paths in the original graph. If there exists a Hamiltonian path called for in the problem specification, one representative molecule would thus be created by the chemistry on its way to equilibrium. By using more of the same biotechnology, he could then determine the presence or absence of the molecule in the final test tube and respond accordingly to the original problem.

This breakthrough experiment provided a very appealing argument for molecular computing. The most important point is perhaps that biotechnology is mature enough to stop dreaming about experiments for solving hard computational problems, and that it is time to begin thinking about specific experimental setups to solve them.

B. DNA Computation in Ciliates

Landweber and Kari[4] presented a different version on the origin of DNA computing in 1998. They provide a convincing argument that

[4]L.F.Landweber, and L. Kari., "The Evolution of DNA Computing: Nature's Solution to a Combinatorial Problem", Proceedings of the 3rd Annual Genetic Programming Conference, July 22-25, 1998, Morgan Kaufmann Publishers, San Francisco, pp.700-708.

several million years earlier and unknown to all of us, the ciliated protozoa *oxytricha nova* and *oxytricha trifallax* of the genus *oxytricha* solved a problem similar to HPP while unscrambling genes as part of their reproductive cycle. Ciliate cells possess, in general, two nuclei, an active macronucleus and a functionally inert micronucleus. The macronucleus forms from the micronucleus after sexual reproduction. The process requires more than simple copying, however, because intervening non-protein coding sequences that shatter it into nearly a hundred pieces must be removed, and moreover, the relevant protein coding sequences sometimes appear scrambled and must be restored to their natural order. This process is essentially identical to the problem one faces in HPP, namely to arrange the cities in the right order for a Hamiltonian path. The analogy goes further since the protozoa seem to rely on short repeat sequences that act as sort of matching tags in recombination events. If the mechanisms underlying this type of information processing can be fully attributed to the same kinds of processes present in Adleman's experiment, then molecular computation is certainly millions of years old. Therefore, the origins of molecular computing are still buried in the evolution of genetic complexity in the biological kingdom.

C. Bacteriorhodopsin

An older alternative to DNA molecules that support optical computing is the protein bacteriorhodopsin, which contains the light sensitive rhodopsin present in vertebrate retinas. In essence, this molecule consists of seven alpha-helical segments that span the purple membrane of a microorganism commonly known as *halobacterium halobium* (Fig. 7.2). This organism grows in salt marshes at higher salt concentrations than in sea water, where exposure to high thermal fluctuations and photochemical damage has made it capable, for the sake of metabolic energy, of switching chemically among a few atomic states a thousand times more efficiently than similar synthetic materials. Switching can take place by absorption of green and blue light as many as 10 million times before wearing out. The switching property has been used in combination with lasers to create a storage medium for optical computer memories that is almost in commercial stage now. The possibility exists that it might become a core memory for a molecular

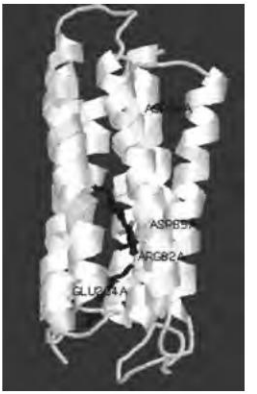


Fig. 7.2: Structure of Bacteriorhodopsin

computer. Although certainly involving amino acids at the protein-building sites, this type of computation is more passive than the earlier methods.

7.1.4 Challenges before Molecular Computing

Molecular computing has generally aimed, so far, at solving the same ordinary algorithmic problems that are commonly posed for conventional VLSI-based computers, albeit by an entirely different type of operational process. None of them has exhibited the kind of practical success that can qualify for a "killer application". Such an application would suit well the nature of biomolecules, beat current and perhaps even future solid-state electronics, and would establish beyond the shadow of doubt the power of the new computational paradigm. The root of the difficulties for molecular computing lies in our relatively poor ability to control the physical chemistry involved in the context of information processing, despite impressive progress in biotechnology that has made it thinkable.

Some of the challenges before molecular computing are as follows:

A. Reliability, Efficiency and Scalability

Reliability, efficiency and scalability are perhaps the three most important issues for molecular computing. Reliability refers to the

degree of confidence with which a lab experiment provides a true answer to the given problem. The efficiency of the protocol refers to the intended and effective use of the molecules that intervene in it. The scalability of a lab experiment is the effective reproducibility of the experiment with longer molecules that can encode larger problem instances while still obtaining equally reliable results under comparable efficiency. These are three distinct but clearly interrelated problems. Biologists have not really faced these problems in their work because in that field the definition of success is different than in computer science. When a biologist clones an organism, the contention is that one experiment was successful, regardless of how many carried out previously were not, or whether only one clone was actually produced.

B. Encoding Problem

Once the encoding molecules for the input of a problem have been chosen, a molecular computer scientist is at the mercy of the chemistry, even though he may still have some control over the protocols that he may perform with them in the laboratory execution. If the encodings are prone to errors, the experiment can be repeated any number of times and always provide the same erroneous results. This fact lessens the effectiveness of the standard method of increasing the reliability of a probabilistic computation with a non-zero probability of errors by iteration.

A mismatched hybridization is a bound pair of oligonucleotides that contains at least one mismatched pair. In addition to frame shift errors in which the n-mers are shifted relative to each other, mismatches leading to false positives include hairpin mismatches, bulges and partial hybridizations. The encoding problem for DNA computing thus consists of mapping the instances of an algorithmic problem in a systematic manner on to specific molecules so that, a) the chemical protocols avoid all these sources of error, and b) the resulting products contain, with a high degree of reliability, enough molecules encoding the answers to the problem's instances to enable a successful extraction.

An optimal encoding would maximize the likelihood of desired hybridizations, and furthermore, would lead to equilibrium reaction conditions that are favorable for retrieving the solution of the problem in the extraction phase.

C. Error-Preventing Codes

It is conceivable that a more principled computational approach can produce solutions of the encoding problem that capture physico-chemical conditions that are good enough to be validated by lab experiments. Perhaps the best example is the combinatorial approach proposed by the computing group in Memphis. The crux of the approach is to regard an experiment for a molecular computation as the transmission of a message from the protocol to the experimentalist through a noisy channel, namely the tube in which the reactions take place. The theory of communication introduced by Shannon[5] has found effective ways to handle this problem by introducing redundancy to protect against noise. The solutions are the so-called error-correcting codes for data transmission that information theorists have spent the last 50 years designing. The mathematical framework is the metric space of Boolean hypercubes with the standard binary Hamming metric[6].

D. Building and Programming Molecular Computers

For several reasons, the greatest engineering and technological challenge posed by molecular computing is perhaps the construction of a molecular computer. In a molecular computer, one would expect to find the basic features that are evident in a conventional electronic computer in an integrated system, namely information storage, programmability, and information processing. Such features are obviously desirable, but whether they are actually realizable is not very clear.

Given the difficulties with implementing traditional algorithms in DNA and their potential for evolutionary-style computation, DNA computers apparently follow Michael Conrad's tradeoff principle[7]:

---

[5]Shannon, C.E. (1948), "A Mathematical Theory of Communication", *Bell System Technical Journal*, 27, pp. 379–423 & 623–656, July & October, 1948.

[6]In information theory, the Hamming distance between two strings of equal length is the number of positions for which the corresponding symbols are different. Put another way, it measures the minimum number of substitutions required to change one into the other, or the number of errors that transformed one string into the other.

[7]Michael Conrad, "On design principles for a molecular computer", *Communications of the ACM*, Volume 28, Issue 5, May 1985, pp. 464–480.

"a computing system cannot at the same time have high programmability, high computational efficiency and high evolutionary adaptability". He describes programmability as the ability to communicate programs to the computing system exactly with a finite alphabet in a finite number of steps. The efficiency of a computing system is defined as the ratio of the number of interactions in the systems that are used for computing and the total number of interactions possible in the system, while evolutionary adaptability is defined as the ability of the system to change in response to uncertainty. It is clear that biomolecules offer, by the nature of their function, a good answer to adaptability. If Conrad's principle holds here, there is good evidence that molecular programming will be a great challenge.

E. Implementing Evolutionary Computation

Evolutionary computation is based on analogies of biological processes implemented in electronics to arrive at computer programs that sometimes outperform software designed by standard methodologies. The most common analogy used is natural selection, or survival of the fittest. The various methodologies include genetic algorithms, genetic programming, evolution strategies, evolutionary programming and immune systems. These algorithms use a generate-and-evaluate strategy: a population of possible solutions is maintained (usually generated at random); individuals are then selected from a population based on their fitness, that is, how well they satisfy an external constraint; the population is then updated by replacing less-fit individuals by combinations of hopefully fitter individuals through some variation operations such as crossover and mutation. A major problem faced by evolutionary algorithms is the strain they place on computational resources and running time. Large clusters of PCs have difficulty supplying the computational power required. Molecular computing offers a great challenge but also great potential for the implementation of evolutionary algorithms.

A new emerging class of biologically inspired systems is based on the immune system. An immune system is capable of combating a large number of different types of invading pathogenic microorganisms. An artificial immune system based on molecules duplicated the ability of a natural immune system to recognize self

from the non-self in order to protect a computer system from computer viruses and other unwanted agents. For discrimination of self from the non-self in a computer, the entities of interest are not molecules or microorganisms, but strings composed from a finite alphabet. These strings can be bit strings, data strings or strings of machine instructions.

A distinction worth making here is between molecular and biological computing. Molecular computing is based on the progress of organic molecules and at first sight molecular and biological computing might seem to be equivalent. However, in biological computing the actual biological processes are used for computation. In molecular computing the organic molecules are used in a way which, although possibly related to their biological functioning, does not directly mimic that functioning. Thus the use of bacteriorhodopsin is an example of molecular computing, whilst real neural computing and DNA computing can be regarded as forms of biological computing.

Molecular computing is not the only technology threatening the dominant position of silicon. There are a number of new technologies based on inorganic materials which make competing claims. Holographic memory, using crystals of lithium niobate offers storage capability in excess of that promised by bacteriorhodopsin. Another memory technology currently subject of research is the 'single electron' memory which uses between 1 and 10 electrons per bit compared with around one million electrons per bit for conventional silicon devices. The result is a massive reduction in power consumption. Work is also being done on controlling neutral atoms in free space. Overlapping laser beams can be used to control the location of a neutral atom, and this opens the way to use quantum phenomena such as spin and coherence to store information in single atoms.

7.2 Optical Computing

Optics, which is the science of light, is already used in computing, most often in the fibre-optic glass cables that currently transmit data down Internet lines much more quickly than traditional copper wires. Thus, optical signals would be the ticket for the fastest supercomputers ever. Compared to light, electronic signals in chips

travel at a snail's pace. Moreover, there is no such thing as a short circuit with light, so beams could cross with no problem after being redirected by pinpoint-size mirrors in a switchboard. Optical computing was a hot research area in the 1980s. But the work tapered off because of material limitations that seemed to prevent optochips from getting small and cheap enough to ever be more than laboratory curiosities. Now, optical computers are back. Researchers are using new conducting polymers to make transistor-like switches smaller and thousand times faster than silicon transistors. And electricity-conducting organic molecules much thinner than semi-conductor wires are being teased into self-assembling. These advances promise super-tiny all-optical chips. In addition, progress in optical storage devices can now shrink an entire library's book collection down to sugar-cube size. Optical computers could leave silicon number crunchers choking by the end of the decade.

7.2.1 Introduction

The pressing need for optical technology stems from the fact that today's computers are limited by the time response of electronic circuits. A solid transmission medium limits both the speed and volume of signals, and builds up heat that damages components. For example, a one-foot length of wire produces approximately one nanosecond of time delay. Extreme miniaturization of tiny electronic components also leads to 'cross-talk'—signal errors that affect the system's reliability. These and other obstacles have led scientists to seek answers in light itself. Light does not have the time response limitations of electronics, does not need insulators, and can even send dozens or hundreds of photon signal streams simultaneously by using different color frequencies. They are immune to electromagnetic interference, and free from electrical short circuits. They have low-loss transmission and provide large bandwidth, i.e., multiplexing capability and they are capable of communicating several channels in parallel without interference. They are capable of propagating signals within the same or adjacent fibers with essentially no interference or cross-talk. They are compact, lightweight, and inexpensive to manufacture, and more facile with stored information than magnetic materials. By replacing

electrons and wires with photons, fiber optic, crystals, thin films and mirrors, researchers hope to build a new generation of computers that work 100 million times faster than today's machines.

Optical interconnections and optical integrated circuits will provide a way out of the current limitations to computational speed and complexity inherent in conventional electronics. Optical computers will use photons traveling on optical fibers or thin films instead of electrons to perform the appropriate function. In the optical computer of the future, electronic circuits and wires will be replaced by a few optical fibers and films, making the systems more efficient with no interference, more cost effective, lighter and more compact. Optical components would not need to have insulators as those needed between electronic components because they do not experience cross-talk. Indeed, multiple frequencies (or different colors) of light can travel through optical components without interfering with each other, allowing photonic devices to process multiple streams of data simultaneously.

Much progress has been achieved, and optical signal processors have been successfully used for applications such as synthetic aperture radars, optical pattern recognition, optical image processing, fingerprint enhancement and optical spectrum analyzers. The early work in optical signal processing and computing was basically analog in nature. In the past two decades, however, a lot of effort has been expended on the development of digital optical processors. The major breakthroughs have been centered around the development of devices such as vertical cavity surface-emitting lasers (VCSELS) for data input, spatial light modulators (SLMs) such as liquid-crystal and acousto-optic devices for putting information on the light beams, and high-speed avalanche photo-diodes (APDs) or so-called smart pixel devices, for data output. Much work remains before digital optical computers will be widely available commercially, but the pace of research and development has increased in the 1990s.

One of the problems optical computers have faced is a lack of accuracy; for instance, these devices have practical limits of 8 to 11 bits of accuracy in basic operations. Recent research has shown ways around this difficulty. Digital partitioning algorithms, which can break matrix-vector products into lower-accuracy sub-products, working in tandem with error-correction codes, can substantially improve the accuracy of optical computing operations. Many

problems in developing appropriate materials and devices must be overcome before digital optical computers will be in widespread commercial use. In the short run, at least, optical computers will most likely be hybrid optical/electronics systems that use electronic circuits to preprocess input data for computation and to post-process output data for error correction before outputting the results. The promise of all-optical computing remains highly attractive, however, and the goal of developing optical computers continues to be a worthy one. Quite a few scientists feel that an all-optical computer will not be the computer of the future but opto-electronic computers will rule where the advantages of both electronics and optics will be used.

7.2.2 Current Use of Optics for Computing

We are in an era of daily explosions in the development of optics and optical components for computing and other applications. The business of photonics is booming in industry and universities worldwide. The data traffic is growing worldwide at a rate of 100 percent per year, while the US data traffic is expected to increase 300 percent annually. The requirement for high data rate transfer equipment is also expected to continue increasing drastically. Electronic switching limits network speeds to about 50 Gigabits per second (1 Gigabit (Gb) is 10^9, or 1 billion bits). Terabit speeds (1 Terabit, abbreviated 'Tb', is 10^{12}, or 1 trillion bits) are needed to accommodate the growth rate of the Internet and the increasing demand for bandwidth-intensive data streams.

Most of the components that are currently very much in demand are electro-optical (EO). Such hybrid components are limited by the speed of their electronic parts. All-optical components will have the advantage over EO components. Unfortunately, there is an absence of known efficient non-linear optical (NLO) materials that can respond to low power levels. Most all-optical components require a high level of laser power to function as required. A group of researchers from the University of Southern California, jointly with a team from the University of California, Los Angeles, have developed an organic polymer with a switching frequency of 60 GHz. This is three times faster than the current industry standard, lithium niobate crystal-based devices. The California team has

been working to incorporate their material into a working prototype. Development of such a device could revolutionize the information superhighway and speed data processing for optical computing. Another group at Brown University and the IBM Almaden Research Centre (San Jose, CA) has used ultrafast laser pulses to build ultrafast data-storage devices. This group was able to achieve ultrafast switching down to 100 ps. The results are almost 10 times faster than currently available "speed limits". Optoelectronic technologies for optical computers and communication hold promise for transmitting data as short as the space between computer chips. NEC (Tokyo, Japan) has developed a method for interconnecting circuit boards optically by using VCSEL arrays. Researchers at Osaka University (Japan) reported on a method for automatic alignment of a set of optical beams in space with a set of optical fibers.

Researchers at NTT (Tokyo, Japan) have designed an optical back plane with free-space optical interconnects using tunable beam deflectors and a mirror. This project had achieved 1000 interconnections per printed-circuit board with throughput ranging from 1 to 10 Tb/s. Optics has a higher bandwidth capacity over electronics, which enables more information to be carried, and data to be processed because electronic communication along wires requires charging of a capacitor that depends on length. In contrast, optical signals in optical fibers, optical integrated circuits, and free space do not have to charge a capacitor and are therefore faster.

Another advantage of optical methods over electronic ones for computing is that optical data processing can be done much easier and is less expensive in parallel than can be done in electronics. Parallelism is the capability of the system to execute more than one operation simultaneously. Electronic computer architecture is, in general, sequential, where the instructions are implemented in sequence. This implies that parallelism with electronics is difficult to construct. Parallelism first appeared in Cray Super Computers in the early 1980s, when two processors were used in conjunction with the computer memory to achieve parallelism and to enhance the speed by more than 10 Gb/s. It was later realized that more processors were not necessary to increase computational speed, but could in fact be detrimental. This is because as more processors are used, there is more time lost in communication. On the other hand, using a simple optical design, an array of pixels can be

transferred simultaneously in parallel from one point to another. To appreciate the difference between both optical and electronic parallelism one can think of an imaging system of as many as 1000 x 1000 independent points per mm^2 in the image plane. For this to be accomplished electrically, a million non-intersecting and properly isolated conduction channels per mm^2 would be required. Parallelism, therefore, when associated with fast switching speeds, would result in staggering computational speeds. Assume, for example, there are only 100 million gates on a chip, much less than what was mentioned earlier (optical integration is still in its infancy compared to electronic). Further, conservatively assume that each gate operates with a switching time of only 1 nanosecond (organic optical switches can switch at sub-picosecond rates compared to maximum picosecond switching times for electronic switching). Such a system could perform more than 10^{17} bit operations per second. Compare this to the gigabits (10^9) or terabits (10^{12}) per second rates which electronics are either currently limited to, or hoping to achieve. In other words, a computation that might require one hundred thousand hours (more than 11 years) on a conventional computer could require less than one hour by an optical one.

Since photons are uncharged and do not interact with one another as readily as electrons, light beams may pass through one another in full-duplex operation, for example, without distorting the information carried. In the case of electronics, loops usually generate noise voltage spikes whenever the electromagnetic fields through the loop changes. Further, high frequency or fast switching pulses will cause interference in neighboring wires. Signals in adjacent fibers or in optical integrated channels do not affect one another nor do they pick up noise due to loops. Finally, optical materials possess superior storage density and accessibility over magnetic materials.

The field of optical computing is progressing rapidly and shows many dramatic opportunities for overcoming the limitations described earlier for current electronic computers. The process, whereby optical devices have been incorporated into many computing systems, is already underway. Laser diodes as sources of coherent light have dropped rapidly in price due to mass production. Also, optical CD-ROM discs have been very common in home and office computers.

7.2.3 Some Roles for Optics

There are certain roles optics can play that electronics cannot, either qualitatively or quantitatively.

A. 2D Array Mapping

Only optics can implement massive, parallel, arbitrary mapping from an $N \times N$ input array to an $N \times N$ output array by using N weighted interconnections. Such a function can serve as the backbone of a truly massively parallel neural network or fuzzy programmable logic array. It can also implement any 2D to 2D mapping, whether or not that mapping is one-to-one.

Optics can handle N on the order of 256 now, to provide parallel, weighted interconnections. Providing such interconnects electronically would lead to an absurd tangle of wires. If we have a continuous function that changes continuously with time, sampling can be used to represent such a function. But we can never achieve time-continuous transformation of time-continuous signals with digital technology.

B. Garbage-free Operations

In a traditional Turing computer, a physical device must perform each operation. Each device operation consumes time and energy and is subject to error. If we could make a system that did not have to tackle all of the steps in terms of actually measuring device operations, it would save the corresponding amount of time, energy, and cost. Some optical processors can achieve this, but the price to pay is that we can never know the values of the unmeasured operations. Garbage bits are information we generate en route to the desired answer, but we really do not need to know their values. As an example, if we need to know the sum of the first six integers, we add 1 to 2 to get 3. Then we would add 3 to 3 to get 6, and so on. All of the intermediate sums (3, 6, and so forth) are garbage. In optics, it is often the case that we do not measure the garbage terms. For example, Fourier filtering multiplies each pixel of the Fourier transformer of the input pattern with a pixel from the mask and then uses a lens to sum those products. But we never measure even one of those products or even one of the

partial sums. We can correctly say that an optical processor performs the intermediate steps virtually, and virtual events carry no price, not in speed energy, error or money.

7.2.4 Optical Computing Paradigms

Optical computing paradigms can enhance conventional computing and at the same time remain hidden from the end user. These systems can be attached to an electronic host, undertake a subset of operations, and relieve the host from a corresponding computational load. Keeping a new technology hidden from the end user is an important factor in the technology's acceptance. If novel architectures are introduced seamlessly, users will adopt them without fear of tearing down their established investments.

Some of the optical computing paradigms are:

- Analog optical computing.
- Digital optical processing.
- Analog-digital hybrid optical computing.

Analog optical processing entails analog operations on sets of analog data, such as continuous tome pictures, real-time images obtained by a camera lens, or light beams modulated by analog electronic signals. The functionality of analog optical processing is unique to optics; its advantages have laid the foundation for the early successes of optical systems. Apart from its historic significance, analog optical processing continues to be a thriving field.

Digital optical processing—the use of light to perform digital logic—targets a class of applications that are currently performed by electronic computers and can be enhanced by optoelectronic architectures. The ability of multiple optical beams to propagate in free space with minimal interaction led to attempts to replace wires with optical links. The logical next step was the development of all-optical or optoelectronic switching techniques, in an effort to replace electronics functionality and take advantage of free-space optics.

Digital-analog optical processing involves hybrid techniques and systems that can process both types of data. This paradigm may be better suited for optoelectronic architectures than electronic ones. As an example, consider an associate processor, a system that can access and process any record in storage on the basis of the record's

content, not its address. Electronic associate systems tend to be costly and are usually limited in size and speed. Furthermore, they are incapable of handling analog data such as images or time signals.

7.3 Ultrafast Pulse Shaping and Tb/sec Data Speeds

There has been a sustained interest in the quest to generate ultrashort laser pulses in the picosecond (10^{-12} s) and femtosecond (10^{-15} s) range. Rapid programmable ultrafast optical pulse shaping at 1550 nm wavelength, the important wavelength range for applications to optical communications have been developed. In particular to this optical information channel, what is achieved with ultrafast optical pulse shaping can be viewed as an optical spectral encoder with rapid update rate. One of the important and promising applications of this spectral encoder is into the high-speed optical communication. Shaping ultrafast pulses is nontrivial since there are no electronic devices that can work on these timescales. If the optical pulse that we wish to shape has a temporal duration of femtosecond or picosecond, then we will need a modulator that works on this time scale. The idea of shaping a pulse by sending it through a modulator, such as Mach-Zehnder, is referred to as direct pulse shaping. Current modulators can operate at 60 Hz, which is much slower than necessary to shape a femtosecond pulse. Therefore, the technique of indirect pulse shaping, which includes liquid crystal modulators (LCM pulse shaping), acousto-optic modulator (AOM pulse shaping), and time-stretched pulse shaping, is used. The choice of which pulse-shaping apparatus to use may depend on the particular application; each technique has different advantages to it.

The overall concept is shown in Fig. 7.3. A gating spreads the pulse, so that each different spectral component maps onto a different spatial position. The collimating lenses and grating pair are set up in a 4F configuration (F being the focal length of the collimating lenses), and in the centre of the 4-F systems, an element is placed that will modulate the spectrum. In case of the AOM as the encoding element, there is a huge difference between speed of sound and speed of light in the AOM crystal. Since the ratio

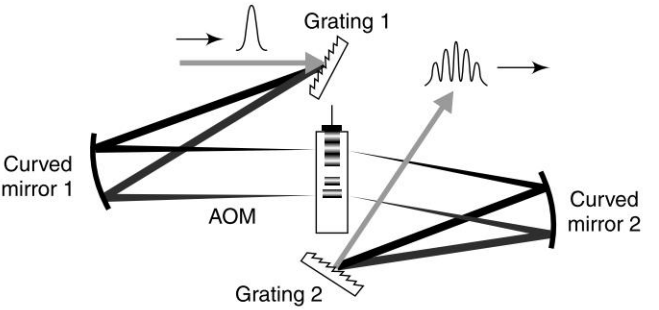


Fig. 7.3: Schematic of a Pulse Shaper

between the two is about 1 is to 1 million, we can use MHz electrical signal to achieve THz programmable modulation of an optical signal and still keep a reasonable update speed. In practice, high resolution spectral encoding is, by definition, a variation of the dense wavelength division multiplexing (DWDM) and can be used to significantly improve the bandwidth efficiency. The idea can be illustrated in the following way: If we start with a 100 fs full-width at half maximum (FWHM) optical pulse and encode, for example, 16 amplitude on-off keying return-to-zero (RZ) format bits in its spectrum, which in the worst possible case would broaden the pulse by a factor of 16—to about 1.6 ps FWHM. The encoded pulses, can, therefore, be well confined in a 4 ps optical switching window, without much distortion to the encoded spectrum. By doing this, the time division multiplexing (TDM) system can benefit from spectrum encoding by a factor of 16 and achievable data translation rate (DTR) be as high as 4 Tbps.

Experimentally, an AOM-based ultrafast optical pulse shaper can implement complex shaped pulses that are necessary to implement logic gates in simple systems. With the progress of solid-state laser technology and commercialization of non-linear processes like optical parametric amplification, an experimental ultrafast optical pulse shaping system with extremely wide wavelength tunability—ranging from UV to far-IR (Fig. 7.4) has been developed such that the choice of material systems is no longer a constraint.

Such ultrafast pulse shaping technology also has the capability to transmit terabit/sec data-bits through optical fibers. Currently, the best possible throughput available from the commercial systems is gigabit/sec, and these use fiber-optic technology. In optical fibers, the information transfer process essentially involves

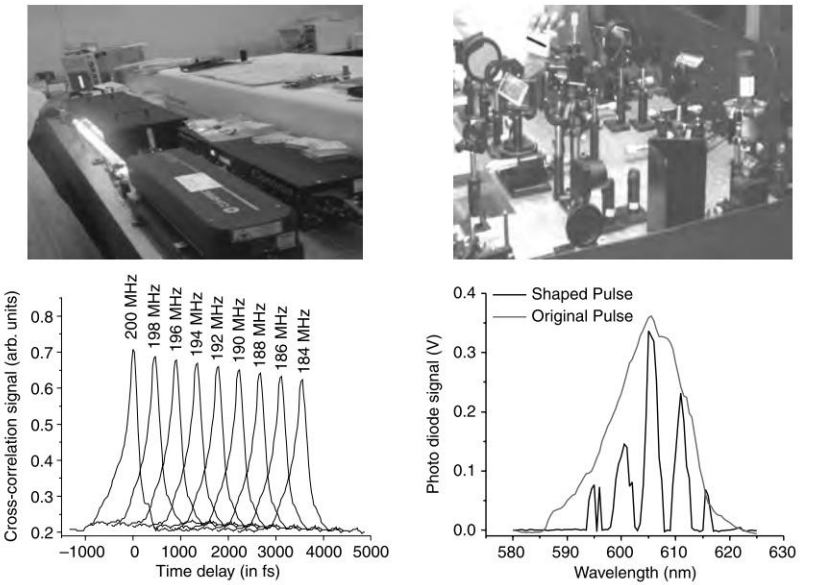


Fig. 7.4: An Acousto-Optic-Modulator based Pulse Shaper Setup with the Help an Amplified Laser System. A Couple of Representative Graphs of the Pulse Shaping Capability Show the Data that are Collected in the Wavelength and Time-domain Respectively.

encoding the data by light modulation techniques. Two different modes of light modulation are commonly used: One is the wavelength domain modulation and the other is the time domain modulation {technically, wavelength division multiplexing (WDM) and time division multiplexing (TDM)}. Each of these schemes has asymptotically approached a maximal data transmission limit of gigabit/sec; however, recent studies show that with the help of ultrafast lasers even more information could possibly be packed. This is because the extremely short pulses from ultrafast lasers are inherently associated with large bandwidths. If the broad spectral content of such ultrafast pulses were wavelength coded (WDM) while the pulses remain confined to a short time, several such time-bursts could be transmitted (TDM) within a second to achieve over a trillion bits of data per second (terabit/sec) transmission. However, taking such advantage of both the WDM and TDM techniques requires the ability to keep the coherence of the short pulses available after modulation. The AOM-based ultrafast pulse shaping technology has the capability for achieving this feat. Terabit/sec transmission is thousand times faster than the best commercially

available fiber optic system. The very high transmission capacity of a terabit can perhaps be well appreciated from the fact that a terabit contains enough bandwidth to transmit about a million movies simultaneously.

Achieving quantum computing goals with optical pulse shaping is thus quite attractive because the shaped pulses can be transmitted over the existing optical transmission hardware to the user locations for terminal computations. Computation and optical information transfers are thus tied on to the same basic infrastructure requirement. In our proposed scheme, the shaped pulses alone should be able to carry all the relevant information about a quantum computation. Thus, the interaction of such optical pulses with the bulk material of choice even at a remote site would enable quantum computation, providing the possibility of transmitting an equivalent of quantum software over physical transmission lines, like optical fibers! Our recent thrust on terabit/sec optical communication efforts with ultrafast pulse shaping technology, therefore, seamlessly integrates into this approach towards quantum computing. Such a pulse shaping scheme has the added advantage to correct for the transmission non-idealities through a closed loop feedback system, where the RF pulse is constantly updated according to the resulting optical pulses to get to the desired shape.

7.3.1 The Role of Non-linear Optics in Optical Computing: Need for New Materials

The field of optical computing is considered to be the most multidisciplinary field and requires for its success, collaborative efforts of many disciplines, ranging from device and optical engineers to computer architects, chemists, material scientists, and optical physicists. On the material side, the role of non-linear materials in optical computing has become extremely significant. Non-linear materials are those, which interact with light and modulate its properties.

For example, such materials can change the color of light from being unseen in the infrared region of the color spectrum to a green color where it is easily seen in the visible region of the spectrum. Several of the optical computer components require efficient non-linear materials for their operation. What in fact

restrains the widespread use of optical devices is the inefficiency of currently unavailable non-linear optical materials, which require large amounts of energy for responding or switching. In spite of new developments in materials, a great deal of research by chemists and material scientists is still required to enable better and more efficient optical materials. Although, organic materials have many features that make them desirable for use in optical devices, such as high non-linearities, flexibility of molecular design and damage resistance to optical radiation, their use in devices has been hindered by processing difficulties for crystals and thin films. Still, some organic materials belonging to the class of phthalocyanines and polydiacetylenes are promising for optical thin films and waveguides. Phthalocyanines are large ring-structured porphyrins for which large and ultrafast non-linearities have been observed. These compounds exhibit strong electronic transitions in the visible region and have high chemical and thermal stability up to 400°C. The third-order susceptibility of phthalocyanine, which is a measure of its non-linear efficiency, has been found to be more than a million times larger than that of the standard material, carbon disulphide. This class of materials has good potential for commercial device applications, and has been used as a photosensitive organic material, and for photovoltaic, photoconductive, and photo-electrochemical applications.

Polydiacetylenes are zigzag polymers having conjugated (alternating) mobile p-electrons for which the largest reported nonresonant (purely electronic) susceptibility for switching has been reported. Consequently, polydiacetylenes are among the most widely investigated class of polymers for non-linear optical applications. Their subpicosecond response to laser signals makes them candidates for high-speed optoelectronics and information processing. Some of these materials can be intrinsically bistable when deposited in thin-film layers. Optical bistable devices and logic gates are the equivalent of electronic transistors. They switch light ON and OFF. They are also useful as optical cells for information storage.

7.3.2 Advances in Photonic Switches

Logic gates are the building blocks of any digital system. An optical logic gate is a switch that controls one light beam by an-

other; it is 'ON' when the device transmits light and 'OFF' when it blocks the light. NASA/Marshall Space Flight Center have demonstrated two fast all-optical switches by using phthalocyanine thin films and polydiacetylene fiber. The phthalocyanine switch is in the nanosecond regime and functions as an all-optical AND logic gate, while the polydiacetylene one is in the picosecond regime and exhibits a partial all-optical NAND logic gate.

To demonstrate the AND gate in the phthalocyanine film, they have guided two focused collinear beams through a thin film of metal-free phthalocyanine. The film thickness was ~ 1 mm and it was a few millimeters in length. They used the second harmonic at 532 nm from a pulsed Nd:YAG laser with pulse duration of 8 ns along with a CW He-Ne beam at 632.8 nm. The two collinear beams were then focused at from a microscopic objective and sent through the phthalocyanine film. At the output a narrow band filter was set to block the 532 nm beam and allow only the He-Ne beam. The transmitted beam was then focused on a fast photodetector and to a 500 MHz oscilloscope. It was found that the transmitted He-Ne cw beam was pulsating with a nanosecond duration and was synchronous with the input Nd:YAG nanosecond pulse. The setup discussed demonstrated the characteristic table of an AND logic gate. A schematic of the setup is shown in Fig. 7.5.

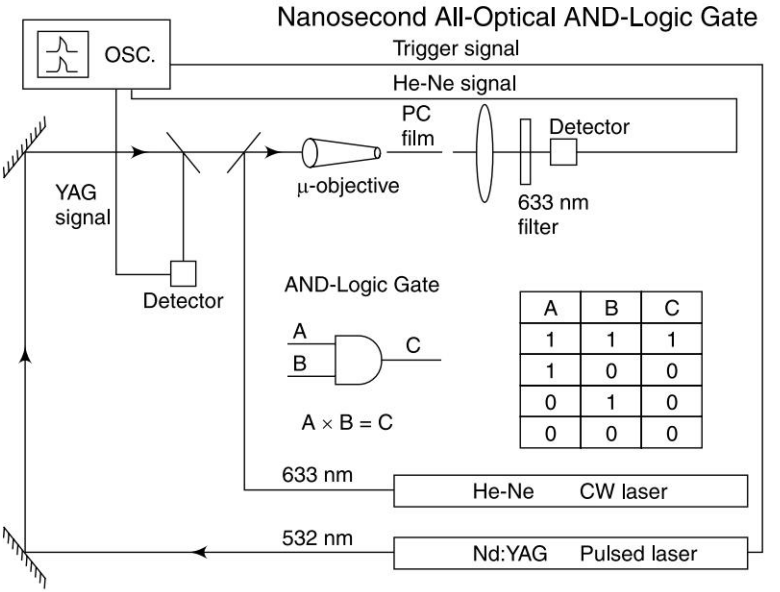


Fig. 7.5: A Schematic of the Nanosecond All-optical AND Logic Gate Setup

The setup for the picosecond switch was very much similar to the setup in Fig. 7.6 except that the phthalocyanine film was replaced by a hollow fiber filled with a polydiacetylene. The polydiacetylene fiber was prepared by injecting a diacetylene monomer into the hollow fiber and polymerizing it by UV lamps. The UV irradiation induces a thin film of the polymer on the interior of the hollow fiber with a refractive index of 1.7 and the hollow fiber is of refractive index 1.2. In the experiment, the 532 nm pulse from a mode locked picosecond laser was sent collinearly with a CW He-Ne laser and both were focused on to one end of the fiber. At the other end of the fiber a lens was focusing the output on to the narrow slit of a monochrometer with its grating set at 632.8 nm. A fast detector was attached to the monochrometer for sending the signal to a 20 GHz digital oscilloscope. It was found that with the He-Ne beam OFF, the Nd:YAG pulse is inducing a week fluorescent picosecond signal (40 ps) at 632.8 nm that is shown as a picosecond pulse on the oscilloscope. This signal disappears each time the He-Ne beam is turned on. These results exhibit a picosecond respond in the system and demonstrated three of the four characteristics of a NAND logic gate as shown in Fig. 7.6.

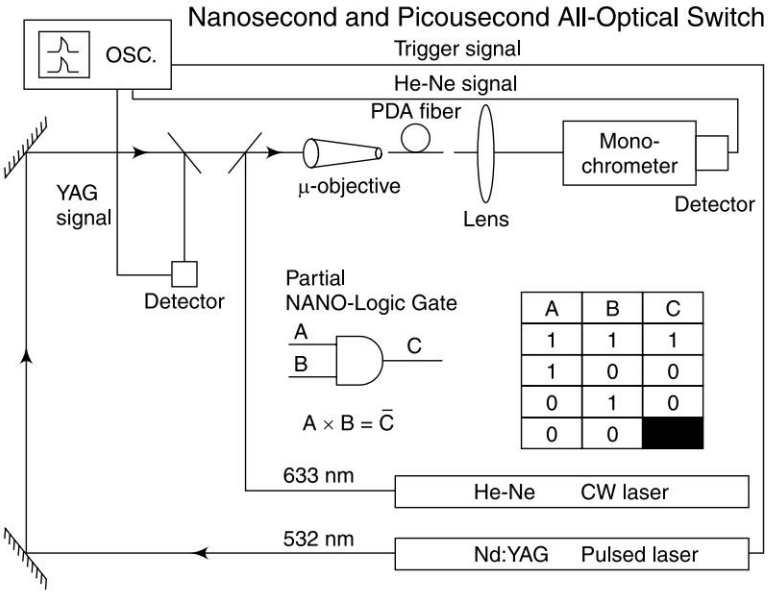


Fig. 7.6: A Schematic of the Nanosecond All-optical NAND Logic Gate Setup

7.4 Conclusions

Research in molecular and optical computing has opened up new possibilities in several fields related to high-performance computing, high-speed communications, and parallel algorithm design. To design algorithms that execute faster for applications, specific properties, such as massive parallelism, and global interconnections must be considered. As optoelectronic and smart pixel devices mature, their software development will have a major impact in future, and the ground rules for computing will be rewritten.

Further Readings

Some useful links on Nanotechnology and Nanocomputing

BC Crandall's Molecular Realities: http://www.well.com/user/bcc/MolecularRealities.html

Brad Hein's Nanotechnology page: http://www.public.iastate.edu/~bhein/nanotechnology1.html

Caltech Materials and Process Simulation Center: http://www.wag.caltech.edu/

Carol Shaw's Molecular Assembly Sequence Software: http://www.carol.com/mass.shtml

DARPA's ULTRA program: http://esto.sysplan.com/ETO/ULTRA/index.html

Hello Nanotechnology, Bye, Bye Money!: http://bcx.ind.wpi.edu/Quincy/Documents/Nanotechnology/hello_nanotechnology.html

Homebrew STM page: http://www.skypoint.com/members/jrice/STMWebPage.html

Institute of Atomic-Scale Engineering: http://www.speakeasy.org/~forrestb/home.html

Institute for Quantum Computing, University of Waterloo: http://www.iqc.ca

John Michelsen's Homepage: http://sugar.ps.uci.edu/~jmichels/

JosH's sci.nanotech archives: http://nanotech.rutgers.edu/nanotech/

Markus Krummenacker's Homepage: http://www.ai.sri.com/~kr/

MITRE Corp. nanoelectronics and nanocomputing site: http://www.mitre.org/research/nanotech/

Molecular Nanotechnology and the WorldSystem: http://www-bcf.usc.edu/~tmccarth/main.htm

Nano Mission, Govt. of India: http://nanomission.gov.in

Nanocomputer Dream Team: http://www.nanocomputer.org/

NanoCon Proceedings: http://www.halcyon.com/nanojbl/NanoConProc/nanocon1.html

Nanosource: http://www.nanosource.org/

Nanotechnology Institute, University of Waterloo: http://www.nanotech.uwaterloo.ca/research/ni.html

Nanotechnology Magazine: http://planet-hawaii.com/nanozine/

Nanothinc: http://www.nanothinc.com/NanoHome.html

NASA Ames NAS Computational Nanotechnology: http://science.nas.nasa.gov/Groups/Nanotechnology/

Ned Seeman's DNA nanotechnology site: http://seemanlab4.chem.nyu.edu/homepage.html
Perimeter Institute, Waterloo: http://www.perimeterinstitute.ca
Red Herring magazine article on Eric Drexler: http://www.herring.com/mag/issue22/world.html
STM work at IBM Almaden: http://www.almaden.ibm.com/vis/stm/stm.html
SunSITE Singapore Nanolink site: http://sunsite.nus.sg/MEMEX/nanolink.html
Transition to Tomorrow: Society at the Cusp of Nanotechnology: http://home.netscape.com/people/jamie/jwd_nanoTTT_page.html
UNC nanoManipulator project: http://www.cs.unc.edu/Research/nano/
US National Nanotechnology Initiative: http://www.nano.gov
USC Laboratory of Molecular Robotics: http://alicudi.usc.edu/~lmr/
Will Ware's freeware NanoCAD: http://world.std.com/~wware/ncad.html

A few nanotechnology groups

Accelrys Inc. (formerly Molecular Simulations Inc.)
Andres Group (Nanocluster Synthesis & Cluster-Assembled Materials), Department of Chemical Engineering, Purdue University
Atom Sciences, Inc. - Elemental surface analyses and imaging with submicron resolution
ATOMA - Software for Nanotechnology
Bucky News Service - Abstracts of recent papers on Fullerene Research
Buckyball Home Page, Department of Physics, State University of New York at Stony Brook
California Molecular Electronics Corporation (CALMEC)
Center for Functional Nanomaterials, Brookhaven National Laboratory
Center for Integrated Nanotechnologies, A U.S. DoE Nanoscale Science Research Center
Center for Molecular Design, Institute for Biomedical Computing, Washington University in St. Louis
Center for Nanoscale Science and Technology, Rice University
Centre for Quantum Computation, University of Oxford

Centre for Self-Organising Molecular Systems (SOMS Centre), University of Leeds
Cientifica - The Nanobusiness Company
Computational Materials Science, NCSU Department of Materials Science and Engineering
EMBnet: European Molecular Biology Network
Energenius Centre for Advanced Nanotechnology (ECAN), University of Toronto
ExPASy Molecular Biology Server
Expert Reviews in Molecular Medicine
Femtosecond Laser Laboratory, SL-216, IIT Kanpur, India.
HP Labs: Quantum Science Research (QSR)
IBM Almaden Research Center Visualization Lab
IBM Research - Nanotechnology
Information Mechanics Group at MIT Laboratory for Computer Science
Institute for Molecular Manufacturing
Institute for Quantum Computing, University of Waterloo
Institute of Biophysics & Center Interdepartimental on Biophysical-chemical and Biomedical Technologies (IBF & CITBB)
Institute of Physical and Chemical Research (RIKEN), Japan
Institute of Physics Publishing
International Society for Molecular Electronics and BioComputing (ISMEBC)
IP Nanoker - Structural Ceramic Nanocomposites for top-end Functional Applications
Laboratory for Molecular Robotics, University of Southern California
Laboratory for Molecular Sciences, Caltech
Laboratory of Mathematical Biology, National Cancer Institute
Laboratory of Supramolecular Photonics, Hunter College of CUNY
Lightyear Technologies Inc.
London Centre for Nanotechnology
Materials and Process Simulation Center (MSC), The Beckman Institute at Caltech
Materials Research Society
Michigan Molecular Institute
Microfabrication Applications Lab (MAL), University of Illinois at Chicago
Mirkin Group

Molecular and Electronic Nanostructures, The Beckman Institute for Advanced Science and Technology at the University of Illinois (Urbana-Champaign, UIUC)
Molecular Manufacturing Enterprises Incorporated (MMEI)
Molecular Manufacturing Shortcut Group, National Space Society
Molecular Modelling Group, at the University of Veszprem, Hungary
Molecular Neuroscience Program at Caltech
Molecular Structure Laboratory, Department of Chemistry, State University of New York at Stony Brook
Nano Letters: American Chemical Society Publication
Nanodot: News and Discussion of Coming Technologies
Nanoelectronics & Nanocomputing Home Page (Mitre)
Nanoelectronics Laboratory (NANOLAB) at University of Cincinnati
NanoInk, Inc.
NanoLab, Inc.
NanoLine at Cornell University
NanoLogic
Nanomanipulator Project, Department of Computer Science at University of North Carolina, Chapel Hill, NC
Nanomechanics LLC
Nanometer Pattern Generation System
NanoPowders Industries
Nanoscale Physics, Purdue University
NanoStructures Laboratory (NSL) at MIT (formerly Submicron Structures Laboratory)
NanoStructures Laboratory (NSL) at Princeton University
Nanotechnik—Piezo Motors, Micro Positioner, Micro Manipulator, Scanning Tunneling Microscope
Nanotechnology, IOP Journal
Nanotechnology Institute, University of Waterloo
Nanotechweb.org - Nanotechnology news, products, jobs, events, and information
Nanothinc Corporation
NanoTools: The Homebrew STM Page
Nanotube Site
NASA-JSC Area Nanotechnology Study Group
National Institute for Nanotechnology / Institut national de nanotechnologie
National Institutes of Health - Molecular Biology Databases
National Institutes of Health - Molecular Modeling Home Page

National Nanofabrication Facility at Cornell University
National Nanofabrication Users' Network (NanoNet Information Central)
National Nanofabrication Users' Network at Penn State University
National Nanofabrication Users' Network at University of California, Santa Barbara
Pedro's Biomolecular Research Tools
Purdue NanoTechnology Initiative (NTI)
Quantum-Nano Centre, University of Waterloo
Quantum and Nano Computing Systems Lab (QANSLAB), Dayalbagh Educational Institute, India
Rice Quantum Institute (RQI)
Roukes Group (Caltech)
Small Times: News about MEMS, Nanotechnology and Microsystems
Solid State Spectroscopy at the Institute of Materials Physics of the University of Wien
Stanford Mesoscopic Quantum Optics Homepage
Stanford NanoFabrication Facility
Technology Review - Nanotech + More
Tsukada Group, Solid State Physics
Virtual Journal of Nanoscale Science and Technology
Weatherall Institute of Molecular Medicine, University of Oxford (formerly Institute of Molecular Medicine)
Zyvex

Index