

OPTICAL COMPUTER ARCHITECTURES

**The Application of Optical Concepts
to Next Generation Computers**

Alastair D. McAulay

NCR Distinguished Professor and Chairman
Department of Computer Science and Engineering
Wright State University



A Wiley-Interscience Publication

JOHN WILEY & SONS, INC.

New York

Chichester

Brisbane

Toronto

Singapore

In recognition of the importance of preserving what has been written, it is a policy of John Wiley & Sons, Inc., to have books of enduring value published in the United States printed on acid-free paper, and we exert our best efforts to that end.

Copyright © 1991 by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

Reproduction or translation of any part of this work beyond that permitted by Section 107 or 108 of the 1976 United States Copyright Act without the permission of the copyright owner is unlawful. Requests for permission or further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.

Library of Congress Cataloging in Publication Data:

McAulay, Alastair D.

Optical computer architectures : the application of optical concepts to next generation computers / Alastair D. McAulay.

p. cm.

"A Wiley-Interscience publication."

Includes bibliographical references and index.

1. Computers, Optical. 2. Computer architecture. 3. Optical data processing. I. Title.

TA1630.M43 1991

621.39'1—dc20

ISBN 0-471-63242-2

90-42103

CIP

Printed in the United States of America

10 9 8 7 6 5 4 3

Preface

Rapid advances in optical technology, including the availability of low cost laser diodes and fiber optics, create opportunities for significantly advancing the field of computing. Currently, optics is appearing in computer memories and for computer interconnections. The unique advantages of optics over conventional electronics suggests that optics will become increasingly important in computers.

This book is aimed at those interested in how to use optics for computing. It was used as a text for a first year graduate course in *Optical Computing* in the Department of Computer Science and Engineering at Wright State University. It emphasizes concepts and provides the background to understand them. Specific knowledge in optics and computer architecture is not assumed. Exercises are provided for verifying understanding.

Part I provides background for understanding the rest of the book. Chapter 1 discusses the motivation for considering optics and ways in which optics might evolve to become dominant in computing. Chapter 2 provides a review of basic concepts involving optical wave and lens systems. Chapter 3 reviews the basic principles of coherent processing. Electronic and optical addressable optical devices, suitable for optical computing, are discussed in chapters 4 and 5 respectively.

Part II describes methods of assembling the optical components into computing subsystems. Modularity in computer design is achieved by grouping together bits into words and operations into subsystems. This provides economy of design and manufacture, and permits specific computers to be designed by tailoring combinations of these subsystems. Methods of providing optical interconnections for computing are discussed in chapter 6. Chapter 7 considers optical memory. Chapter 8 shows how logic may be performed optically for use in control or arithmetic units. The synthesis of logic circuits is described in chapter 9 and involves merging parallel logic and interconnections. The construction of highly parallel arithmetic units is described in chapter 10. Matrix computations are discussed in chapter 11. Algorithms for high performance computing are described in chapter 12.

Part III describes optical computing systems constructed according to different architectural models of computation. Most computers today operate sequentially on instructions, updating only one element of the state at a time. Chapter 13 describes the history of such machine design and provides

an all optical implementation of a sequential machine. The other models of computation described are more parallel and may therefore be more suitable for optical implementations. Optical dataflow machines, chapter 14, operate as data becomes available and have no concept of state. Optical cellular automata, chapter 15, represent a model of computation in which the whole state is updated at each time step, and the interconnections are local. The important role of cellular automata in computer theory is reviewed. Optical linear neural network models of computation involve learning and widespread interconnections, chapter 16. Learning is aimed at reducing programming and providing adaptivity. Greater capability is achieved with optical nonlinear neural networks, described in chapter 17. Optical autoassociative and self organizing neural networks are discussed in chapter 18.

I wish to acknowledge the many researchers in this field whose publications I have referenced or with whom I have had discussions, including organizations such as the International Society for Optical Engineering (SPIE), the Optical Society of America (OSA), the IEEE Computer Society, and the Association for Computing Machinery (ACM). I also wish to thank the funding agencies that have supported my activities, including Air Force Office of Scientific Research, Defense Advanced Research Projects Agency, NCR, and Wright State University.

ALASTAIR D. MCAULAY

Kettering, Ohio

Contents

I	Background for Optical Computing	1
1	Why Optical Computers?	1
1.1	Electronic Computer Architectures	1
1.1.1	Requirements for Future Computers	1
1.1.2	Limitations of Current Computer Technologies	2
1.1.3	Direction for Computers	6
1.2	Why Use Optics in Computers?	6
1.2.1	Advantages of Optics	7
1.2.2	Introduction to Functional Capabilities of Optics	8
1.2.3	Strategy for Evolving Optics into Computers	10
1.3	Exercises	12
2	Basic Concepts in Optics	15
2.1	Wave Phenomena	15
2.1.1	Waves	15
2.1.2	Wave Interactions with Media	19
2.2	Polarization and Anisotropic Crystals	21
2.2.1	Polarization	22
2.2.2	Anisotropic Crystals	23
2.3	Lenses and Lens Systems	26
2.3.1	Lenses	27
2.3.2	Optical Lens Systems	27
2.3.3	Lens as a Phase Transformation	29
2.3.4	Imaging Property of Lenses	31
2.4	Coherency	32
2.4.1	Temporal Coherency	32
2.4.2	Spatial Coherency	33
2.5	Exercises	34
3	Fourier Optics	39
3.1	Correlation and Convolution	40
3.1.1	Correlation	40
3.1.2	Convolution	41

3.1.3	Fourier Transform Computation of Correlation and Con- volution	41
3.1.4	Coherent Versus Incoherent Processing	42
3.2	Fourier Transforms with Lenses	43
3.2.1	1-D Fourier Transform	43
3.2.2	2-D Fourier Transform and Frequency Filtering	44
3.2.3	Optical Scheme for Coherent Filtering	49
3.3	Grating Filters for Addition and Subtraction	50
3.3.1	Grating Filters	50
3.3.2	Addition and Subtraction with Grating Filters	51
3.4	Complex Transform Filters	54
3.4.1	Storing Complex Functions on Film	54
3.4.2	Using Complex Filters for Convolution and Correlation	55
3.4.3	Generating a Complex Filter Optically	57
3.5	Holograms	58
3.5.1	Equations for Thin Amplitude Holograms	59
3.5.2	Thick Holograms for Volume Memory	60
3.5.3	Phase Holograms	62
3.5.4	Generating a Complex Filter or Hologram by Computer	62
3.6	Exercises	63
4	Devices for Opto-Electronic Interface	67
4.1	Information on SLMs	68
4.1.1	Classification	68
4.1.2	Characteristics	72
4.1.3	Digital Versus Analog Computing	73
4.2	Deformable Mirror Device	74
4.2.1	Construction and Addressing	75
4.2.2	Modeling	77
4.2.3	Converting Phase to Intensity	79
4.3	Double Heterostructure Opto-Electronic Switch	82
4.3.1	Advantages of Gallium Arsenide	82
4.3.2	Device Description	82
4.4	Magneto-Optic Spatial Light Modulators	84
4.5	Acousto-Optic Devices	86
4.5.1	Operation	86
4.5.2	Using Acousto-Optic Cells	89
4.6	Lasers and Optical Detectors	90
4.6.1	Lasers	90
4.6.2	Optical Detectors	93
4.7	Integrated Optics	94
4.7.1	Waveguide Switches	94
4.7.2	Applications to Spectrum Analysis and Filtering	95
4.8	Exercises	97

5	Optically Addressable Spatial Light Modulators	101
5.1	Liquid Crystal Devices	102
5.1.1	Electrooptic Effect	104
5.1.2	Interfacing to Liquid Crystal Devices	104
5.2	Self-Electrooptical Effect Device	107
5.2.1	SEED description	107
5.2.2	Symmetric-SEED	111
5.2.3	Interfacing to S-SEED Arrays Using Patterned Mirrors	112
5.3	Spatial Light Rebroadcasters	113
5.3.1	Electron Trapping Device Description	113
5.3.2	Arithmetic Operations	115
5.3.3	Application to Template Matching	115
5.4	Microchannel Spatial Light Modulator	117
5.5	Fabry-Perot Based Optical Transistors	119
5.5.1	Principle	119
5.5.2	Thin Film Fabry-Perot Devices	121
5.6	Devices for Real-Time Holograms	121
5.6.1	Photorefractive Crystals	121
5.6.2	Four-Wave Mixing or Dynamic Holograms	122
5.7	Exercises	124
II	Subsystems for Optical Computing	127
6	Optical Interconnections	129
6.1	Interconnection networks	129
6.1.1	Use of Networks in Parallel Computers	129
6.1.2	Simple Interconnections	130
6.1.3	Systolic Arrays	133
6.2	Crossbar Switch Interconnection Networks	134
6.2.1	Spatial Light Modulator Crossbar Switches	136
6.2.2	Polarizing Beam Splitter and Variable Grating	138
6.2.3	Holographic Interconnections with 2-D Images	141
6.3	Regular Limited Interconnections	144
6.3.1	Shuffle Power of Two Interconnections	144
6.3.2	Patterned Mirror Banyan Power of Two Interconnection	150
6.3.3	Space Invariant Holographic Interconnections	152
6.4	Multistage Interconnection Networks	154
6.4.1	Two-by-two Switches for Multistage Networks	154
6.4.2	Omega (Multistage Shuffle) Network	156
6.4.3	Integrated Optic Crossbar	157
6.4.4	Reverse Order Beam Splitter Power of Two Networks	158
6.5	Exercises	158

7	Optical Memory	161
7.1	Introduction	161
7.1.1	Nature of Information Stored	162
7.1.2	Associative Versus Random Access Memory	162
7.2	Current Memory Management	165
7.2.1	Heirarchies	165
7.2.2	Cache	168
7.2.3	Virtual Memory	170
7.3	Optical Word Pattern Matching	171
7.3.1	Optical Word Template Matching AM	172
7.3.2	Bit-Slice Associative Memory	176
7.4	Holographic Memory	181
7.4.1	Writing and Reading Holographic Memories	181
7.4.2	Optical Beam Deflection Scanning	183
7.4.3	Mechanical Scanning	185
7.5	Exercises	187
8	Optical Logic	193
8.1	Basic Logic Concepts	193
8.1.1	Logic Elements and Classification	194
8.1.2	Propositional Logic	197
8.1.3	Spatial Logic and Optical Logic Gates	197
8.2	Logic Programming	201
8.2.1	Constructs for Prolog	201
8.2.2	Reduction to Normal Form	202
8.3	Optical Array Logic with Encoded Inputs	203
8.3.1	All Logic Operations in Parallel with Shadow Casting	203
8.3.2	All Logic by Correlation with Kernels	207
8.3.3	Optical Implementation for Correlation by 2×2 Kernels	210
8.4	Optical Array Logic without Encoding	212
8.4.1	All Logic Operations in Parallel with SLRs	213
8.4.2	Parallel Logic with Liquid Crystal Devices	214
8.4.3	Parallel Logic with Variable Grating LCDs	217
8.5	Exercises	219
9	Optical Logic Circuits	223
9.1	Approaches to Synthesizing Logic Systems	224
9.1.1	Modular Approach	224
9.1.2	Sequential Logic Circuits	227
9.1.3	Programmable Logic Array Approach	227
9.2	Global Interconnection Logic Systems	230
9.2.1	Liquid Crystal and Hologram	230
9.2.2	Matrix-Vector Crossbar Approach	232
9.3	Local Interconnection Logic Circuits	234

9.3.1	Symbolic Substitution	238
9.3.2	Pattern Logic	241
9.4	Regular Interconnection Logic Systems	244
9.4.1	Dual-Rail Logic with S-SEEDs and Fan of Two	244
9.4.2	Dual-Frequency Logic with SLR-OLCD and Fan of Two	249
9.5	Exercises	250
10	Optical Arithmetic Computation	253
10.1	Digital Number Representations	253
10.1.1	Conventional	253
10.1.2	Residue Numbers	254
10.1.3	Modified-Sign-Bit	255
10.1.4	Floating Point	256
10.2	Adder Circuits	257
10.2.1	Half Adder Circuit	258
10.2.2	Full Adder Circuit	258
10.3	Optical Parallel Addition of Words	261
10.3.1	Bit Slice Adder with Liquid Crystal Devices	262
10.3.2	Optical Bit Slice Adder Using PLAs and Minimum Fans	265
10.3.3	Optical Ripple Carry Adder Using Pattern Logic . . .	269
10.3.4	Symbolic Substitution Ripple Carry Adder	271
10.3.5	Separate Carry Adder Using Symbolic Substitution . .	272
10.3.6	Optical Look-Ahead Carry Adders	275
10.4	Optical Parallel Multiplication of Words	277
10.4.1	Optical Symbolic Substitution Multiplication	279
10.4.2	Optical Digital Multiplication by Analog Convolution	280
10.4.3	Optical Wallace Tree Multiplier	281
10.5	Optical Division of Words	284
10.5.1	Optical Divider Using Convergence-Division	284
10.6	Exercises	286
11	Optical Matrix Computation	289
11.1	Optical Matrix-Vector Computation	289
11.1.1	Computation of Quadratics	294
11.2	Systolic Approaches	294
11.2.1	Analog Optical Systolic Matrix-Vector Computation .	297
11.2.2	Digital Optical Systolic Matrix-Vector Multiplication .	297
11.3	Matrix-Matrix Computations	299
11.3.1	Optical Inner Product Computation	301
11.3.2	Optical Middle Product Computation	302
11.3.3	Optical Parallel Middle Product Computation	302
11.3.4	Outer Product Computation	304
11.4	Optical Matrix-Vector Sonar/Radar	305
11.4.1	Description of Processor	305

11.4.2 Optical Implementation	309
11.5 Optical Matrix-Vector Expert System	310
11.5.1 Concept	310
11.5.2 Flow Graph	312
11.5.3 Spatial Light Modulator Implementation	312
11.6 Exercises	315
12 Algorithms for Numerical Computation	319
12.1 Optical Analog to Digital Converters	320
12.1.1 Optical Approach Using Table-Look Up	323
12.2 Signal Processing Algorithms	327
12.2.1 Fast Fourier Transform	327
12.2.2 Nonlinear Spectral Estimation	329
12.3 Finite Approximation Methods	332
12.4 Iterative Methods for Numerical Computation	333
12.4.1 Gradient Methods: Steepest Descent, Newton, and Gauss-Newton	334
12.5 Solving Large Sparse Linear Equations	339
12.5.1 Conjugate Gradient Algorithm Summary	339
12.6 Exercises	341
III Architectural Models of Computation	343
13 Optical Sequential Machines	345
13.1 Evolution of Construction of Sequential Machines	346
13.1.1 The First Programmable computer	346
13.1.2 The First Working Programmable Computer	348
13.1.3 The First Electronic Computers	351
13.1.4 The First Stored Program Computer	351
13.2 An Electronic RISC Machine	352
13.2.1 Levels of Abstraction	352
13.2.2 System Description	353
13.2.3 Description of Critical Parts	354
13.2.4 Instruction Pipelining	357
13.2.5 Stacks	358
13.3 Optical RISC Machine	358
13.3.1 Critical Parts for Optical RISC Design	359
13.3.2 Optical RISC Machine Overall Design	367
13.4 Exercises	371

14 Optical Dataflow Computers	375
14.1 Principles of Dataflow	376
14.1.1 Explanation of Dataflow	376
14.1.2 Dataflow Versus Control Flow	377
14.1.3 Writing Dataflow Programs	379
14.2 Electronic Dataflow Architectures	381
14.2.1 Static Dataflow Architecture	381
14.2.2 More Parallelism in Dataflow Machines	382
14.2.3 Dynamic Dataflow Architectures	384
14.3 Optical Static Flow Graph Dataflow Machine	386
14.3.1 System Description	386
14.3.2 Signal Processing Algorithms	387
14.3.3 Optical Matrix-Vector Multiplication	392
14.3.4 Optical Conjugate Gradient Algorithm Implementation	393
14.3.5 Optical Dynamic Dataflow Machines	396
14.4 Optical Prolog Computer	398
14.4.1 Prolog and Its Advantages	398
14.4.2 System Level Architecture	401
14.4.3 Performance Using Simulator	403
14.5 Exercises	405
15 Optical Cellular Automata	409
15.1 Evolution of the Theory of Computing Machines	409
15.1.1 Equivalence of Symbolic and Numeric Computing	410
15.1.2 Limitations on What is Computable	411
15.1.3 The Universal Computer	412
15.2 Theory of Cellular Automata	415
15.2.1 Description of Cellular Automata	415
15.2.2 Game of Life Illustration	416
15.2.3 Implementing Turing Machines as Cellular Automata	417
15.2.4 Multiple Setting Transition Rule for Computing	421
15.3 Optical Computers Based on Cellular Automata	422
15.3.1 Operations Required	422
15.3.2 Optical Binary Cellular Automata using Holograms	423
15.3.3 Spatial Cellular Logic Array Computers	424
15.4 Optical Finite Approximation Machines	428
15.4.1 Operations Required	428
15.4.2 Optical Implementation	429
15.5 Exercises	430
16 Optical Linear Neural Networks	433
16.1 Basic Features of Optical Neural Networks	434
16.1.1 Massively Parallel Computation	434
16.1.2 Learning Simplifies Programming	434

16.2	Linear Heteroassociative Memories	435
16.2.1	Matrix Formulation	435
16.2.2	Under and Overdetermined Learning	437
16.3	Iterative Learning for Linear Heteroassociative Memory	439
16.3.1	Iterative Updating of Weights	440
16.3.2	Optical Orthogonalization using Gram-Schmidt	441
16.4	Orthogonal Associative Memory	442
16.4.1	Principles	442
16.4.2	Optical Implementation using SLRs	444
16.4.3	Experimental Results for Optical Orthogonal Memory	445
16.5	Exercises	449
17	Optical Nonlinear Neural Networks	451
17.1	Nonlinear Feedforward Networks	451
17.1.1	Single Neuron	452
17.1.2	Multilayer Neural Networks	454
17.1.3	Implementation	458
17.2	Optical Learning	460
17.2.1	Photorefractive Device Learning	462
17.2.2	Learning with Spatial Light Rebroadcasters	463
17.3	Optical Steepest Descent Learning for Nonlinear Networks	464
17.3.1	Backpropagation	464
17.3.2	Optical Implementation of Backpropagation	466
17.4	Optical Gauss-Newton Learning for Nonlinear Networks	468
17.4.1	Gauss-Newton Learning Algorithm	468
17.4.2	Split-inversion Learning Algorithm	470
17.4.3	Optical Implementations	471
17.5	Vision Application for Multilayer Neural Network	472
17.5.1	Optical Shift, Rotation, and Size Invariance Computation	473
17.5.2	Neural Network for Aspect Angle Invariance	477
17.6	Exercises	477
18	Optical Autoassociative and Self Organizing Networks	481
18.1	Autoassociative Memories Using Nonlinear Feedback	481
18.1.1	Nonlinear Correlation Feedback	482
18.1.2	Outer Product Nonlinear Feedback	484
18.1.3	Feedback Using Angle Holograms	488
18.2	Self Organizing Polynomial Networks	492
18.2.1	Principles of Polynomial Neural Network	493
18.2.2	Application to Pilot Neural Network Advisor	494
18.2.3	Optical Implementations of Polynomial Neural Networks	496
18.3	Exercises	496

A Conjugate Gradient Method Derivation	501
A.1 Updating Solution and Gradient	501
A.2 Searching in Conjugate Directions	502
A.3 Computing the Optimum Step Size for Linear Searching . . .	505
Bibliography	507
Index	521

Chapter 1

Why Optical Computers?

This chapter discusses the reasons for considering optics for computing and provides incentives for reading many of the subsequent chapters. In section 1.1 we will discuss future requirements for computers, difficulties of meeting them with existing technologies, and suggested future directions. The advantages and possible roles for optics in computing are discussed in section 1.2.

1.1 Electronic Computer Architectures

We will discuss the requirements for future computers, the limitations of currently used sequential models implemented in electronics, and new directions suggested by the human brain.

1.1.1 Requirements for Future Computers

A number of desirable features are identified for future computers. A future computer is expected to have the *flexibility* to run a wide range of algorithms and languages, including ones not yet developed. The user requires *extendability* so that the machine can be extended as his or her needs grow rather than having to purchase a new, larger, and possibly incompatible machine. Similarly, the manufacturer would like *scalability* so that one product line using the same software has machines from small to large. *Reliability* is critical because down-time may have a high cost to the user. *Fault tolerance* permits the system to continue operating at a reduced capability after a part of the system fails. The *software environment* must be efficient and easy to use for the appropriate level of skill of the user: the system software developer, the application software developer, and application user. Also, *cost effectiveness* is clearly critical in the present competitive marketplace.

These desired features influence computer architecture as follows. *Reconfigurability* in parallel machines is required for fault tolerance, flexibility,

and extendability. Such machines suggest complex interconnection networks. These are difficult to construct and expensive with electronic systems due to interference. *Massive parallelism* is required for extendability, scalability, fault tolerance, and high performance. The latter suggests that the interconnection networks have a high bandwidth and are fine grain. *Symbolic* computation capability for string manipulation, reasoning, and expert systems is required to provide good software environments and for some applications. Consequently, there is a move to parallelism and symbolic computation.

1.1.2 Limitations of Current Computer Technologies

Limitations are due to electronics, the architectural model of computation, and software approaches.

Electronic Limitations

Computers have increased their capabilities by approximately an order of magnitude every five years for the same cost since the 1950s. This was accomplished by shrinking the size of active electronic elements such as transistors on semiconductor chips. The cost of manufacturing a chip does not change much, so the cost per element decreases with increasing density. Further, smaller elements have increased speed. It is getting increasingly difficult to shrink the elements further because the switching speed is now similar to the communication speed between elements, and the latter does not decrease with size. This is because the resistance-capacitance (RC) time-constant of the interconnecting conductor remains constant. As the link shrinks, its capacitance decreases but the resistance increases; thinner wire presents greater resistance to current flow. Progress in shrinking electronics further is hampered because the number of pins for accessing the device and the bandwidth per pin are severely limited. Optical interconnections to chips are a method of overcoming these limitations.

Architectural Limitations

Another serious limitation with current technology is the sequential computer architecture in which instructions are implemented in sequence (figure 1.1). The concept involves separating memory, computation, interconnection, and control. The high cost of early electronic arithmetic units favored channeling everything through a single unit. Further, the sequential architecture uses random access memory (RAM) to store information and to save interconnections. Electronic interconnections are expensive in terms of reliability, cost, and power because electrons are charged particles which interfere with one another. In RAM, each item to be remembered is stored in an address represented in binary form. Hence, n bits can be selected to reference 2^n addresses.

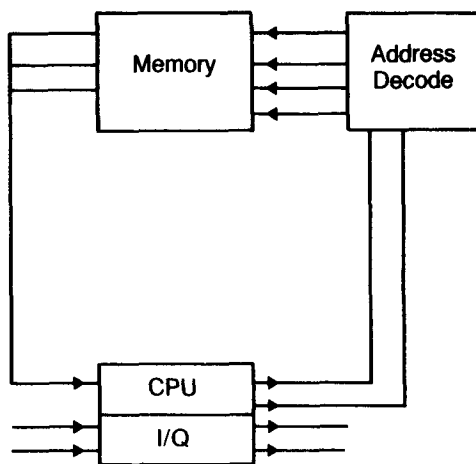


Figure 1.1: Sequential Computer Architecture

A decoder is used to translate the bits to permit selection of the address. This means (figure 1.1) that the total number of connections to memory is only $n + 1$ rather than $2^n + 1$ for addressing 2^n memory cells. In the figure $n = 2$, only one of the 2^n memory cells is accessible at a time. The serial nature of RAM is a disadvantage for parallel computing.

Proposed approaches to parallelism with sequential architectures depend on the application. For example, in transaction systems, such as airline reservation systems, parallel access to memory is critical, but the transactions are otherwise uncoupled. The processors can be loosely coupled to each other. In contrast, tightly coupled processors are needed for a large scientific computation. Unfortunately, if a machine is too specific to an application, the cost of software and hardware development may not be justified by the market size. A massively parallel, tightly coupled machine could handle a wider range of applications, making the development of extensive software and hardware more attractive.

Massively parallel tightly coupled machines are difficult to construct with current electronic sequential architectures. Figure 1.2 shows that even using all the techniques of pipelining and overlap, the improvement in supercomputer performance with time was flattening out until Cray moved to replication of the system to two processors in 1982. The computation rates shown are for a linear equation solving software package LINPAK. Careful vectorizing was used to optimize performance on the Cray to achieve the faster rate. The techniques for increasing performance on early supercomputers, such as instruction pipelining and overlap, are now universal in workstations.

Figure 1.3 [122], shows the difficulty of increasing performance on a single

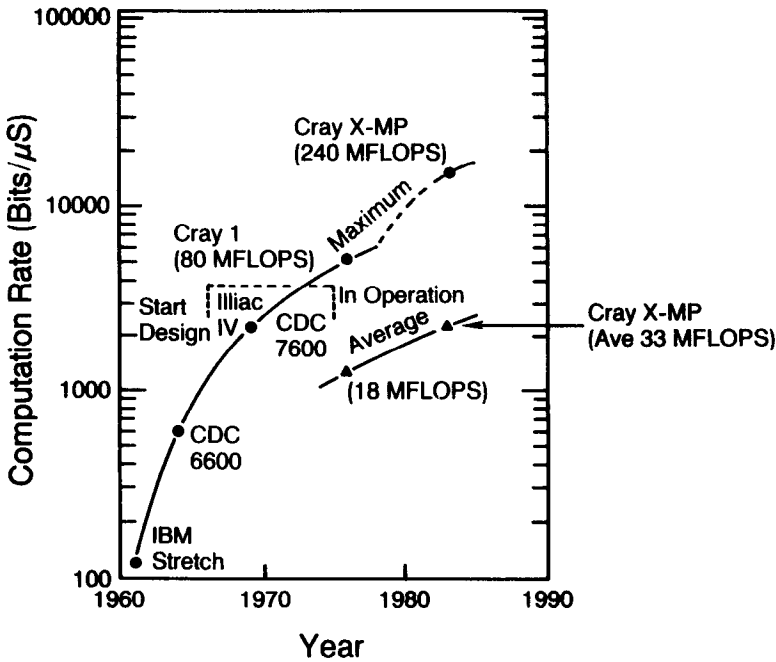


Figure 1.2: Performance of Supercomputers with Time

problem as more processors are used. Speed-up S for a computation (y-axis) is defined as the ratio of the shortest time to perform the computation on a uniprocessor to the shortest time to perform the computation on the parallel processor. Note that different algorithms may be selected to provide the shortest times for the two architectures and that equivalence of accuracy and robustness must be considered. Minsky's conjecture, a maximum speed-up of $\log_2 n$, has proven conservative. For example, for a given computation, a parallel algorithm is often possible that provides greater than eight speed up with 256 processors. The $n/\ln n$ bound is derived in reference [122] and is achievable for some computations. In this case, 32 processors will provide a speed up of nine. Higher performance is rare, because as more processors are used, more time is lost in communication and overhead. Also, long computations, for which high performance machines are needed, often have sections of code that are not very parallelizable because they involve gathering data together from the parts, as in normalization.