
COMPUTER
ARCHITECTURE
AND
PARALLEL
PROCESSING

Kai Hwang

Fayé A. Briggs

COMPUTER ARCHITECTURE AND PARALLEL PROCESSING

Kai Hwang

Purdue University

Fayé A. Briggs

Rice University

McGraw-Hill Book Company

New York St. Louis San Francisco Auckland Bogotá Hamburg
Johannesburg London Madrid Mexico Montreal New Delhi
Panama Paris São Paulo Singapore Sydney Tokyo Toronto

This book was set in Times Roman.
The editors were Eric M. Munson and Jonathan Palace;
the production supervisor was Leroy A. Young.
The drawings were done by ANCO/Boston.
Halliday Lithograph Corporation was printer and binder.

COMPUTER ARCHITECTURE AND PARALLEL PROCESSING

Copyright © 1984 by McGraw-Hill, Inc. All rights reserved.
Printed in the United States of America. Except as permitted under the United States
Copyright Act of 1976, no part of this publication may be reproduced or distributed in
any form or by any means, or stored in a data base or retrieval system, without the prior
written permission of the publisher.

1234567890 HALHAL 8987654

ISBN 0-07-031556-6

Library of Congress Cataloging in Publication Data

Hwang, Kai.

Computer architecture and parallel processing.

(McGraw-Hill series in computer organization and
architecture)

Bibliography: p.

Includes index.

I. Computer architecture. 2. Parallel processing
(Electronic computers) 1. Briggs, Faye A. (Faye Alaye)
II. Title. III. Series.

QA76.9.A73H88 1984 001.64 83-23867
ISBN 0-07-031556-6

ABOUT THE AUTHORS

Kai Hwang is a professor of computer engineering at the School of Electrical Engineering, Purdue University. He received the Ph.D. in EECS from the University of California at Berkeley in 1972. His current research and teaching interest lies mainly in vector supercomputers, multiprocessors, VLSI and dataflow computing structures, and image understanding systems. Dr. Hwang has published about 60 scientific papers and books. He is the author of *Computer Arithmetic Principles, Architecture and Design* (Wiley, 1979) and the editor of *Supercomputer Design and Applications* (Computer Society Press, 1984). Dr. Hwang is the coeditor-in-chief of the *Journal of Parallel and Distributed Computing*. He has been involved in advanced studies on supercomputer architectures and large-scale scientific computations for U.S. National Science Foundation, IBM Fishkill Facilities, Fujitsu in Japan, Academia Sinica, and NATO Advanced Study Institute. He has been a distinguished visitor of IEEE Computer Society since 1981.

Fayé A. Briggs is an associate professor of electrical engineering at Rice University. He received the Ph.D. degree in electrical engineering from the University of Illinois, Urbana. Prior to joining the faculty at Rice University, Dr. Briggs was an assistant professor and the systems manager of the Advanced Automation and Research Laboratory at Purdue University. He has also served as a consultant to IBM Watson Research Center and Exxon Production Research Company. Since 1983 he has been a distinguished visitor of the IEEE Computer Society. Dr. Briggs' current research interests include performance evaluation of advanced computer architectures such as pipeline computers, multiprocessors, and VLSI computing structures. He has published over 30 technical papers and book chapters in the areas of multiprocessing computer architectures, memory organizations, and performance modeling techniques.

To my parents,
Hwang Yuan-Chung and Liu Cheng Fong,
my wife, Pu Fong,
and my sons, Tony and Andy.

Kai Hwang

黃金鏗

To my grandparents,
Rev. P. B. Harry and Mrs. B. P. Harry.

Fayé A. Briggs

Fayéfori Alayé Nimifa-ã

PREFACE

High-performance computers are increasingly in demand in the areas of structural analysis, weather forecasting, petroleum exploration, fusion energy research, medical diagnosis, aerodynamics simulations, artificial intelligence, expert systems, industrial automation, remote sensing, military defense, genetic engineering, and socioeconomics, among many other scientific and engineering applications. Without superpower computers, many of these challenges to advance human civilization cannot be made within a reasonable time period. Achieving high performance depends not only on using faster and more reliable hardware devices but also on major improvements in computer architecture and processing techniques. This book is devoted to studying advanced computer architectures, theories of parallel computing, optimal resource allocation, fast algorithms, efficient programming languages, and application requirements of cost-effective computer systems to meet the above demands.

Advanced computer architectures are centered around the concept of parallel processing. State-of-the-art parallel computer systems can be characterized into three structural classes: *pipelined computers*, *array processors*, and *multiprocessor systems*. The development and application of these computer systems require a broad knowledge of the underlying hardware and software structures and close interactions between parallel computing algorithms and the optimal allocation of machine resources. This book provides the readers with the necessary knowledge to design a new computer system; to improve an existing one; to develop fast computing algorithms; and to allocate hardware-software resources in solving large-scale computing problems.

The book is designed to be used by seniors and/or graduate students in computer science, electrical engineering, industrial engineering, and in any other fields demanding the use of high-performance mainframe computers, attached vector processors, scientific supercomputers, and multiprocessors to solve specific applications problems. Parallel processing can be applied at the hardware/software system level or at the algorithmic and programming level. It demands concurrent

execution of many programs in the computer. Parallel-processing computers provide a cost-effective means to achieve high system performance through concurrent activities.

Computer scientists, system designers, advanced programmers, application engineers, computational experimentalists, and computer professionals should find the material useful in their respective lines of work. Results obtained by many researchers, designers, and users of parallel-processing computers have been included in the text. The material being presented is the outgrowth from several courses in Computer Architecture and Advanced Computing taught by both authors, at Purdue University and Rice University. The book is organized into 10 chapters. Chapter 1 introduces the basic concepts of parallel processing and computer structures and prepares readers with an overview of parallelism in computer systems and various application areas. Chapter 2 presents memory hierarchy and input-output subsystems needed in parallel processing computers.

Chapters 3 and 4 are devoted to the design principles and applications of pipelined supercomputers. Systems to be studied include Star-100, TI-ASC, Cray-1, Cyber-205, Fujitsu VP-200, CDC-NASF, and attached scientific processors AP-120B (FPS-164), IBM 3838, and Datawest MATP. Language and compiler requirements are assessed toward optimized vectorization. Pipelined vector processing methods and performance evaluation of pipeline computers are also provided.

Chapters 5 and 6 present the interconnection structures of processor arrays. Such array processors can handle single instruction stream over multiple data streams. Several parallel algorithms are presented for array processors and associative processors. Case study systems include the Illiac-IV, Burroughs Scientific Processor (BSP), the STARAN, and PEPE, and the massively parallel processor (MPP). Performance enhancement methods are also provided for synchronous array processors.

In Chapters 7, 8, and 9, we study hardware system architectures, operating system controls, parallel algorithms, and performance evaluation of multiprocessor systems. Design experiences of three exploratory research multiprocessors, C.mmp, S-1, and Cm*, will be presented. Commercial multiprocessors to be studied include the IBM 370/168 MP, 3033, and 3081, Univac 1100/80 and 1100/90, Cray X-MP, Tandem/16, and Denelcor's HEP. Research issues toward designing tightly coupled multiprocessors are discussed.

Chapter 10 is devoted to studying new computing concepts and their realization issues. Principles of data-driven computations are introduced. Functional languages and existing data flow computer architectures are reviewed. Finally, we study parallel computing algorithms that are suitable for VLSI hardware implementations. Applications of VLSI architectures for image processing are presented.

The prerequisite for reading this book is an introductory undergraduate course in computer organization and programming. We use Fortran and its extensions in Chapters 3 to 6, because most vector supercomputers are Fortran machines. For multiprocessors in Chapters 7 to 9, we use concurrent Pascal as the illustrative

language. Sections marked with asterisk (*) are research-oriented topics. Readers are expected to have some background on discrete mathematics and probability theory in studying these research topics. These difficult sections can be skipped in the first reading without loss of continuity. Homework problems are essential to provide readers with in-depth thinking and hands-on experience in the design, application, and evaluation of parallel computers.

Parallel processing and computer architecture are two wide-open areas for research and development. We hope that this book will inspire further advances in these frontier computer areas. Bibliographic notes are attached at the end of each chapter to help interested readers find additional references for extended studies. The authors are fully responsible for any errors or omissions. We apologize to those computer specialists whose original works are not included in this volume. The computer area is changing so rapidly that no book can cover every new progress being made. However, we do welcome inputs and criticisms from our readers. Readers are invited to send their comments directly to the authors, so that improvement can be made in future printings or revisions of the book.

This book can be used as a text when offering a sequence of two courses on computer architecture and parallel processing. Each course contains 45 lectures, each 50 minutes long. We suggest the following materials be covered in the first course of a two-course sequence. The remaining sections are reserved for the second course.

Material suggested for the first course

Chapter	Sections and subsections
1	1.1, 1.2, 1.3.1-3, 1.3.5, 1.4.1, 1.5
2	2.1.1, 2.1.3, 2.2.1-2, 2.4.1-2, 2.5.1
3	3.1.1-2, 3.2.1-2, 3.3.1-2, 3.4.1
4	4.1, 4.2.1, 4.3.1, 4.4.1-3
5	5.1, 5.2.1-2, 5.3.1, 5.4.1
6	6.1, 6.2.1-2, 6.3.1, 6.4.1
7	7.1, 7.2.1-2, 7.4.1, 7.5.1
8	8.1.1, 8.2.1, 8.3.1, 8.4.1
9	9.1, 9.2.1, 9.5.1-2, 9.6.1
10	10.1.1, 10.2.1, 10.3.1

The first course is suitable for senior and first-year graduate students. The second course is mainly for graduate students. The first course is a prerequisite for the second course. If the book is adopted for only one course offering, the instructor can move some sections from the second course to the first one to give more complete coverage of some selected topics which are of special interest to the instructor and students. This may necessitate trading some sections listed above with the added sections from the second course. A Solutions Manual to this book will be available from McGraw-Hill for instructors only. The manual contains solutions to all problems plus a number of design projects suitable for use as term

projects. Instructors are welcome to communicate directly with the authors or with McGraw-Hill representatives for suggestions or sharing their experiences in using this book as either a required text or as a research reference.

The authors are grateful to a number of individuals whose professional encouragement and assistance have made the long writing and production process a very pleasant endeavour. In particular, we wish to thank Harold S. Stone, Jean Loup Baer, C. V. Ramamoorthy, Tse-yun Feng, John P. Hayes, King-Sun Fu, Clarence L. Coates, V. Carl Hamacher, Herschel H. Loomis, Jr., Bart Sinclair, J. Robert Jump, Edward S. Davidson, H. J. Siegel, Tom Mowbray, Wolfgang Händler, Kenichi Miura, Lionel M. Ni, Michel Dubois, and Shun-Piao Su for their valuable comments and suggestions. Many of our students helped us in improving the manuscript. In particular, we thank Chi-Yuan Chin, Zhiwei Xu, and William Carlson for their assistance. Finally, we wish to thank the typing, drafting, and secretarial assistance from Andy Hughes, Wanda Booth, Linda Stovall, Vicki Johnson, Pat Loomis, Mickey Krebs, Sharon Katz, Nancy Lein, and D. Ringe. Last but not least we appreciate the McGraw-Hill editing staffs and production professionals for their excellent work in producing the book. Without the timely efforts of the above individuals, this book might be still in preparation.

*Kai Hwang
Fayé A. Briggs*

CONTENTS

Preface	xv
Chapter 1 Introduction to Parallel Processing	1
1.1 Evolution of Computer Systems	1
1.1.1 Generations of Computer Systems	2
1.1.2 Trends towards Parallel Processing	4
1.2 Parallelism in Uniprocessor Systems	8
1.2.1 Basic Uniprocessor Architecture	8
1.2.2 Parallel Processing Mechanisms	11
1.2.3 Balancing of Subsystem Bandwidth	13
1.2.4 Multiprogramming and Time Sharing	16
1.3 Parallel Computer Structures	20
1.3.1 Pipeline Computers	20
1.3.2 Array Computers	22
1.3.3 Multiprocessor Systems	25
1.3.4 Performance of Parallel Computers	27
1.3.5 Dataflow and New Concepts	29
1.4 Architectural Classification Schemes	32
1.4.1 Multiplicity of Instruction-Data Streams	32
1.4.2 Serial versus Parallel Processing	35
1.4.3 Parallelism versus Pipelining	37
1.5 Parallel Processing Applications	40
1.5.1 Predictive Modeling and Simulations	42
1.5.2 Engineering Design and Automation	44
1.5.3 Energy Resources Exploration	46
1.5.4 Medical, Military, and Basic Research	48
1.6 Bibliographic Notes and Problems	49
Chapter 2 Memory and Input-Output Subsystems	52
2.1 Hierarchical Memory Structure	52
2.1.1 Memory Hierarchy	52
2.1.2 Optimization of Memory Hierarchy	56
2.1.3 Addressing Schemes for Main Memory	58

x. CONTENTS

2.2	Virtual Memory System	60
2.2.1	The Concept of Virtual Memory	61
2.2.2	Paged Memory System	65
2.2.3	Segmented Memory System	71
2.2.4	Memory with Paged Segments	77
2.3	Memory Allocation and Management	80
2.3.1	Classification of Memory Policies	80
2.3.2	Optimal Load Control	86
2.3.3	Memory Management Policies	91
2.4	Cache Memories and Management	98
2.4.1	Characteristics of Cache Memories	98
2.4.2	Cache Memory Organizations	102
2.4.3	Fetch and Main Memory Update Policies	113
2.4.4	Block Replacement Policies	115
2.5	Input-Output Subsystems	118
2.5.1	Characteristics of I/O Subsystem	118
2.5.2	Interrupt Mechanisms and Special Hardware	123
2.5.3	I/O Processors and I/O Channels	128
2.6	Bibliographic Notes and Problems	141
Chapter 3	Principles of Pipelining and Vector Processing	145
3.1	Pipelining: An Overlapped Parallelism	145
3.1.1	Principles of Linear Pipelining	146
3.1.2	Classification of Pipeline Processors	151
3.1.3	General Pipelines and Reservation Tables	154
3.1.4	Interleaved Memory Organizations	156
3.2	Instruction and Arithmetic Pipelines	164
3.2.1	Design of Pipelined Instruction Units	164
3.2.2	Arithmetic Pipelines Design Examples	170
3.2.3	Multifunction and Array Pipelines	181
3.3	Principles of Designing Pipelined Processors	187
3.3.1	Instruction Prefetch and Branch Handling	187
3.3.2	Data Buffering and Busing Structures	193
3.3.3	Internal Forwarding and Register Tagging	196
3.3.4	Hazard Detection and Resolution	200
3.3.5	Job Sequencing and Collision Prevention	203
*3.3.6	Dynamic Pipelines and Reconfigurability	208
3.4	Vector Processing Requirements	212
3.4.1	Characteristics of Vector Processing	213
*3.4.2	Multiple Vector Task Dispatching	218
3.4.3	Pipelined Vector Processing Methods	226
3.5	Bibliographic Notes and Problems	229
Chapter 4	Pipeline Computers and Vectorization Methods	233
4.1	The Space of Pipelined Computers	233
4.1.1	Vector Supercomputers	234
4.1.2	Scientific Attached Processors	235

4.2	Early Vector Processors	237
4.2.1	Architectures of Star-100 and TI-ASC	237
4.2.2	Vector Processing in Streaming Mode	245
4.3	Scientific Attached Processors	249
4.3.1	The Architecture of AP-120B	249
4.3.2	Back-end Vector Computations	255
4.3.3	FPS-164, IBM 3838 and Datawest MATP	258
4.4	Recent Vector Processors	264
4.4.1	The Architecture of Cray-1	264
4.4.2	Pipeline Chaining and Vector Loops	271
4.4.3	The Architecture of Cyber-205	280
4.4.4	Vector Processing in Cyber-205 and CDC-NASF	285
4.4.5	Fujitsu VP-200 and Special Features	293
4.5	Vectorization and Optimization Methods	301
4.5.1	Language Features in Vector Processing	301
4.5.2	Design of Vectorizing Compilers	305
4.5.3	Optimization of Vector Operations	308
4.5.4	Performance Evaluation of Pipelined Operations Computers	314
4.6	Bibliographic Notes and Problems	320
Chapter 5	Structures and Algorithms for Array Processors	325
5.1	SIMD Array Processors	325
5.1.1	SIMD Computer Organizations	326
5.1.2	Masking and Data Routing Mechanisms	328
5.1.3	Inter-PE Communications	332
5.2	SIMD Interconnection Networks	333
5.2.1	Static versus Dynamic Networks	334
5.2.2	Mesh-Connected Illiac Network	339
5.2.3	Cube Interconnection Networks	342
5.2.4	Barrel Shifter and Data Manipulator	345
5.2.5	Shuffle-Exchange and Omega Networks	350
5.3	Parallel Algorithms for Array Processors	355
5.3.1	SIMD Matrix Multiplication	355
5.3.2	Parallel Sorting on Array Processors	361
5.3.3	SIMD Fast Fourier Transform	367
5.3.4	Connection Issues for SIMD Processing	373
5.4	Associative Array Processing	374
5.4.1	Associative Memory Organizations	375
5.4.2	Associative Processors (PEPE and STARAN)	380
*5.4.3	Associative Search Algorithms	385
5.5	Bibliographic Notes and Problems	388
Chapter 6	SIMD Computers and Performance Enhancement	393
6.1	The Space of SIMD Computers	393
6.1.1	Array and Associative Processors	394
6.1.2	SIMD Computer Perspectives	396

6.2	The Illiac-IV and the BSP Systems	398
6.2.1	The Illiac-IV System Architecture	399
6.2.2	Applications of the Illiac-IV	402
6.2.3	The BSP System Architecture	410
6.2.4	The Prime Memory System	414
6.2.5	The BSP Fortran Vectorizer	417
6.3	The Massively Parallel Processor	422
6.3.1	The MPP System Architecture	423
6.3.2	Processing Array, Memory, and Control	426
6.3.3	Image Processing on the MPP	430
6.4	Performance Enhancement Methods	434
6.4.1	Parallel Memory Allocation	434
6.4.2	Array Processing Languages	440
*6.4.3	Performance Analysis of Array Processors	445
*6.4.4	Multiple-SIMD Computer Organization	448
6.5	Bibliographic Notes and Problems	454
Chapter 7	Multiprocessor Architecture and Programming	459
7.1	Functional Structures	459
7.1.1	Loosely Coupled Multiprocessors	459
7.1.2	Tightly Coupled Multiprocessors	460
7.1.3	Processor Characteristics for Multiprocessing	478
7.2	Interconnection Networks	481
7.2.1	Time Shared or Common Buses	481
7.2.2	Crossbar Switch and Multiport Memories	487
7.2.3	Multistage Networks for Multiprocessors	492
*7.2.4	Performance of Interconnection Networks	502
7.3	Parallel Memory Organizations	508
7.3.1	Interleaved Memory Configurations	508
*7.3.2	Performance Tradeoffs in Memory Organizations	513
7.3.3	Multicache Problems and Solutions	517
7.4	Multiprocessor Operating Systems	525
7.4.1	Classification of Multiprocessor Operating Systems	526
7.4.2	Software Requirements for Multiprocessors	528
7.4.3	Operating System Requirements	531
7.5	Exploiting Concurrency for Multiprocessing	533
7.5.1	Language Features to Exploit Parallelism	533
7.5.2	Detection of Parallelism in Programs	541
*7.5.3	Program and Algorithm Restructuring	545
7.6	Bibliographic Notes and Problems	551
Chapter 8	Multiprocessing Control and Algorithms	557
8.1	Interprocess Communication Mechanisms	557
8.1.1	Process Synchronization Mechanisms	557
8.1.2	Synchronization with Semaphores	565
8.1.3	Conditional Critical Sections and Monitors	572
8.2	System Deadlocks and Protection	577
8.2.1	System Deadlock Problems	577

8.2.2	Deadlock Prevention and Avoidance	580
8.2.3	Deadlock Detection and Recovery	582
8.2.4	Protection Schemes	583
8.3	Multiprocessor Scheduling Strategies	590
8.3.1	Dimensions of Multiple Processor Management	590
8.3.2	Deterministic Scheduling Models	596
*8.3.3	Stochastic Scheduling Models	606
8.4	Parallel Algorithms for Multiprocessors	613
8.4.1	Classification of Parallel Algorithms	614
8.4.2	Synchronized Parallel Algorithms	616
8.4.3	Asynchronous Parallel Algorithms	622
*8.4.4	Performance of Parallel Algorithms	628
8.5	Bibliographic Notes and Problems	637
Chapter 9	Example Multiprocessor Systems	643
9.1	The Space of Multiprocessor Systems	643
9.1.1	Exploratory Systems	643
9.1.2	Commercial Multiprocessors	644
9.2	The C.mmp Multiprocessor System	645
9.2.1	The C.mmp System Architecture	645
9.2.2	The Hydra Operating System	650
*9.2.3	Performance of the C.mmp	654
9.3	The S-1 Multiprocessor	658
9.3.1	The S-1 System Architecture	659
9.3.2	Multiprocessing Uniprocessors	661
9.3.3	The S-1 Software Development	668
9.4	The HEP Multiprocessor	669
9.4.1	The HEP System Architecture	669
9.4.2	Process Execution Modules	674
9.4.3	Parallel Processing on the HEP	680
9.5	Mainframe Multiprocessor Systems	684
9.5.1	*IBM 370/168 MP, 3033, and 3081	684
9.5.2	Operating System for IBM Multiprocessors	693
9.5.3	Univac 1100/80 and 1100/90 Series	694
9.5.4	The Tandem Nonstop System	705
9.6	The Cray X-MP and Cray 2	714
9.6.1	Cray X-MP System Architecture	714
9.6.2	Multitasking on Cray X-MP	717
*9.6.3	Performance of Cray X-MP	721
9.7	Bibliographic Notes and Problems	728
Chapter 10	Data Flow Computers and VLSI Computations	732
10.1	Data-Driven Computing and Languages	732
10.1.1	Control-Flow versus Data Flow Computers	733
10.1.2	Data Flow Graphs and Languages	740
10.1.3	Advantages and Potential Problems	745
10.2	Data Flow Computer Architectures	748

10.2.1	Static Data Flow Computers	748
10.2.2	Dynamic Data Flow Computers	755
10.2.3	Data-Flow Design Alternatives	763
10.3	VLSI Computing Structures	768
10.3.1	The Systolic Array Architecture	769
*10.3.2	Mapping Algorithms into Systolic Arrays	774
10.3.3	Reconfigurable Processor Array	779
10.4	VLSI Matrix Arithmetic Processors	788
10.4.1	VLSI Arithmetic Modules	788
10.4.2	Partitioned Matrix Algorithms	790
10.4.3	Matrix Arithmetic Pipelines	798
*10.4.4	Real-Time Image Processing	803
10.5	Bibliographic Notes and Problems	807
	Bibliography	813
	Index	833

INTRODUCTION TO PARALLEL PROCESSING

Basic concepts of parallel processing on high-performance computers are introduced in this chapter. We will review the architectural evolution, examine various forms of concurrent activities in modern computer systems, and assess advanced applications of parallel processing computers. Parallel computer structures will be characterized as *pipelined computers*, *array processors*, and *multiprocessor systems*. Several new computing concepts, including data flow and VLSI approaches, will be introduced. The material presented in this introductory chapter will provide an overview of the field and pave the way to studying in subsequent chapters the details of theories of parallel computing, machine architectures, system controls, fast algorithms, and programming requirements.

1.1 EVOLUTION OF COMPUTER SYSTEMS

Over the past four decades the computer industry has experienced four generations of development, physically marked by the rapid changing of building blocks from relays and vacuum tubes (1940-1950s) to discrete diodes and transistors (1950-1960s), to small- and medium-scale integrated (SSI/MSI) circuits (1960-1970s), and to large- and very-large-scale integrated (LSI/VLSI) devices (1970s and beyond). Increases in device speed and reliability and reductions in hardware cost and physical size have greatly enhanced computer performance. However, better devices are not the sole factor contributing to high performance. Ever since the stored-program concept of von Neumann, the computer has been recognized as more than just a hardware organization problem. A modern computer system is really a composite of such items as processors, memories, functional units, interconnection networks, compilers, operating systems, peripheral devices, communication channels, and database banks.

To design a powerful and cost-effective computer system and to devise efficient programs to solve a computational problem, one must understand the underlying

hardware and software system structures and the computing algorithms to be implemented on the machine with some user-oriented programming languages. These disciplines constitute the technical scope of *computer architecture*. Computer architecture is really a system concept integrating hardware, software, algorithms, and languages to perform large computations. A good computer architect should master all these disciplines. It is the revolutionary advances in integrated circuits and system architecture that have contributed most to the significant improvement of computer performance during the past 40 years. In this section, we review the generations of computer systems and indicate the general trends in the development of high performance computers.

1.1.1 Generations of Computer Systems

The division of computer systems into generations is determined by the device technology, system architecture, processing mode, and languages used. We consider each generation to have a time span of about 10 years. Adjacent generations may overlap in several years as demonstrated in Figure 1.1. The long time span is intended to cover both development and use of the machines in various parts of the world. We are currently in the fourth generation, while the fifth generation is not materialized yet.

The first generation (1938–1953) The introduction of the first electronic analog computer in 1938 and the first electronic digital computer, ENIAC (Electronic Numerical Integrator and Computer), in 1946 marked the beginning of the first generation of computers. Electromechanical relays were used as switching devices

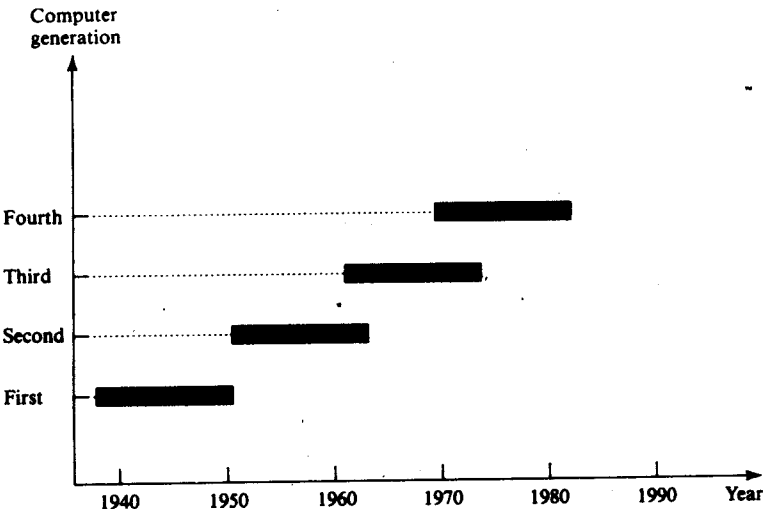


Figure 1.1 The evolution of computer systems.