COMPUTER AND DIGITAL SYSTEM ARCHITECTURE

William D. Murray

COMPUTER AND DIGITAL SYSTEM ARCHITECTURE

William D. Murray

University of Colorado at Denver

PRENTICE HALL

Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging-in-Publication Data
Murray, William D.
Computer and digital system architecture / William D. Murray.
p. cm.
Includes index.
ISBN 0-13-165721-6
1. Computer architecture. 2. Digital electronics. I. Title.
QA76.9.A73M87 1990
004.2'2—dc20 88-8543
CIP

For Addie

Editorial/production supervision
and interior design: **Cheryl Adelmann**
Manufacturing buyer: **Mary Noonan**

A Division of Simon & Schuster
Englewood Cliffs, New Jersey 07632

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-165721-6

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

Preface

I designed this book as a text for graduate students or seniors in computer engineering, computer science, and electrical engineering. It is also intended for independent study by those interested in computer architecture and digital-system design. The book covers a need-based design methodology and reviews the alternatives available to the computer architect.

Your initial reaction might be, "How is this one different from all the others?" I wrote this book because I was not satisfied with texts available for the study of computer architecture in an advanced undergraduate or introductory graduate course.

Most computer-architecture textbooks follow one of three approaches: an introduction to computer design; a general review of computer architecture; or an investigation of a particular computer type. Those in the first category are aimed at the electrical engineer and devote a large percentage of their content to detailed computer subsystems. They are insufficient for a comprehensive study of computer architectures.

Books that review the organization of computers tend to cover the material in insufficient detail. Detail that is included generally pertains to one or two particular computers. These books generally have limited coverage of design alternatives and of methodologies for computer-architecture development. Subsequently, they must be supplemented with current computer literature.

In the third category are texts or monographs on a single type of computer or a single approach to computer design. While these books contain sufficient detail for serious study, they tend to have a narrow focus.

Closest to my needs were collections of papers on various alternatives to computer architecture, with discussions by the authors of groups of papers covering a specific topic. Although these excellent reference works are very useful, they are not organized as textbooks.

I have attempted to consider the issues faced by those who must design or select new hardware that meets economic as well as functional requirements. I address the composite of function, structure, organization, and performance of a computer or some other digital system. Throughout I focus on the effects of technology on the implementation of digital systems. Little attention is paid to design details of the logic or electronics of the system.

To some, "computer architecture" means the computer as seen by a user or programmer. My view of architecture is broader and is similar to that used by the architects of buildings, communities, and cities. I like the definition of architecture as "the science, art, or profession of designing and constructing . . ." (*Webster's New World Dictionary*, Second College Edition, published by Prentice Hall, 1986). Computer engineering is the profession and computer science is the science I refer to when addressing computer architecture.

The approach I have taken is to start with a discussion of the *methodology* to be used in developing a computer architecture, followed by examination of the *alternatives* that are available to the architect. I do not attempt to arrive at a definition of the optimum computer organization. (If it is possible to describe the "best" computer architecture it is reasonable to conclude that that should be the only computer architecture.)

Examples of architectures and subsystems are included to permit the reader to consider the results of prior undertakings and to evaluate these results. Examples are chosen from the mainstream of general-purpose computer design and from interesting deviations therefrom. These examples show that "good" or "successful" architectures are well matched to the functions for which they were intended, are characterized by a specific set of objectives, and usually feature the "elegance of simplicity" that we find in good solutions to all complex problems.

The material is presented with a top-down design-oriented approach. Computer architecture is a serious subject with many significant issues. A successful computer development does not result from merely assembling a group of electronic subsystems, but must start from a consideration of the requirements for the system with an orderly progression toward the detailed design, including an evaluation of the alternatives at each step of the process. The book is organized, each chapter is written, and examples and problems are selected to emphasize development of computers that are responsive to requirements, are balanced in their features, and are economically as well as technically sound.

The book is organized into 10 chapters, starting with an introduction and a review of computer organization in Chapter 1. In Chapter 2 I develop a design methodology and specify the descriptive tools available to help design a digital system. These tools include the processor-memory-switch (PMS) system for top-level description of the computer and the instruction-set processor (ISP) language for defining the programmer's view of the computer. Both of these tools were introduced by C. G. Bell and A. Newell [BellC71].

Alternatives available at the system and the major subsystem levels of computer architecture are covered in the next seven chapters. In Chapter 3 I examine alternative approaches to the top-level system structure and review the rationale to be used to select from among these alternatives. Then I look at methods for communication among subsystems at the top level and review the considerations that go into design of these subsystems: the memories, the processor(s), and the input/output subsystems.

The various types of computer memory are discussed in Chapter 4. The chapter begins with a review of the levels in the memory hierarchy and of the trade-offs relevant to designs at each level: primary, secondary, and archival memory. Questions related to organization and control of virtual memory and its mapping into physical memory are covered, followed by a discussion of errors in memory operations and their detection and correction.

The discussion of central-processor architectures in Chapter 5 shows that there are many alternatives available at this level of computer design. The requirements imposed on the processor design by the applications, programming languages to be used, and operating systems that control the computer set the framework for processor design. Internal registers for short-term information storage, the relationship of such registers to primary memory, and the alternatives available for organizing the internal registers (also called the processor state) are considered. The processor internal data structure and its instruction set, covered next, require a large part of the computer architect's attention.

Issues and alternatives in implementing processors and computer controls are covered in Chapter 6. Of particular importance are the uses of microprocessors and other highly integrated electronic circuits as well as possibilities and problems of microprogrammed controls. The effect on control structure of the languages, application programs, and operating systems are examined, as are the pragmatic questions of manufacturability and maintainability. The functional design of the arithmetic and logic units entails considerations of the trade-offs between speed and complexity.

If the central processor is not to be bogged down in the detailed control of input/output operations, a separate control for those functions must be provided. This might be a fixed-program controller or it can be a processor with its own instruction set. In many applications there are requirements for specialized processing that suggest that a special-purpose processor be attached to the system to relieve the central processor of some duties. Expansion of the control needs of the memory at various levels might lead to incorporation of an internally programmable memory processor. Processors of all three types are considered in Chapter 7.

When it is necessary to exceed the performance that can be achieved with conventional sequential architectures, parallel organizations can be used to permit concurrent processing activity. In Chapter 8 several different approaches to parallel computer design are introduced. Problems of connection, communication, and related performance issues are discussed.

The general-purpose digital computer does not match all requirements for data-processing systems. In many cases a special-purpose design is more cost effective. The determination of whether this is so and the approach to designing special-purpose digital

systems is covered in Chapter 9. Chapter 10 includes a brief summary and a prognosis for the future for the methodologies and the alternatives covered earlier in the book.

Problems and/or projects for the reader follow each chapter. The problems are designed to reinforce understanding of topics covered. The solution of each problem emphasizes thinking about alternatives and minimizes routine calculation. The projects address design issues, most of which can be adapted to situations that might be faced by a reader currently involved in computer design.

I have covered the material in a graduate electrical engineering and computer-science course, "Advanced Computer Architecture". The material can be covered in a single semester if students enter with a solid background in computer-hardware organization. Students or independent readers should possess a good understanding of the organization of some representative computer, as would be obtained in a course on assembly language programming followed by either a course on computer organization or a course on use of microprocessors in electronic equipment. The reader should have a good understanding of at least one modern higher-level programming language, of boolean algebra and logic design, and a working knowledge of the role and functions of operating systems and compilers in digital computers. If significant time is needed to review computer organization fundamentals, Chapter 9 on special-purpose systems can be omitted.

I want to express my appreciation for the assistance and encouragement offered by my friends and associates: Harry Jordan, Rod Schmidt, and Hans Gethoeffer. Students in course EE 559 have been very helpful in developing the approach and the material used herein. The careful reviews and frank comments by the many external reviewers have permitted major improvements over the original draft manuscript. Tim Bozik and Cheryl Adelmann of Prentice Hall were especially helpful to this new author.

W. D. Murray

Contents

1

Introduction

As the title states, this is a book about the architecture of computers and other digital systems. By architecture we mean the design and implementation at the system level rather than at the component level. We are concerned with the function, structure, performance, and economics of the system being considered. We will examine each of these system aspects and will consider the interactions among them. As we will see, balance and consistency among features are characteristics of a successful computer architecture.

1.1 WHY STUDY COMPUTER ARCHITECTURE?

In examining computer architecture we will emphasize a design methodology (a rationale for architecture development) and the alternatives available at each level of system detail. We will learn how to construct a system from functional building blocks. Our objective is to meet the requirements with as economical a design as is possible. This is not an easy process that is accomplished by plucking items from shelves of components. The systems we are dealing with are complicated and there are other system architects who will be trying to achieve results better than our own. In our design we must achieve a result that meets the established requirements for performance, reliability, flexibility, and operability. The cost of development, construction, and operation, as well as the performance of our product, will be compared with those of our competitors.

Computer architecture is a scientific and engineering discipline. There is a logical approach to the discipline. There are tools available to assist us. There are needs for analysis, synthesis, and evaluation. There are means to measure whether we have been successful or not. There is a body of knowledge derived from theory and from past experience that must not be ignored. Perhaps most importantly, there are opportunities for creativity, professionalism, excitement, and satisfaction in undertaking the examination, evaluation, and development of an architecture for a computer or another digital system.

1.2 COMPUTERS AND DIGITAL SYSTEMS

It will be useful to define some terminology to establish a common framework for discussion. Starting with the fundamentals, the *computer* is assumed to be an internally programmed electronic digital machine with the associated software required for its operation, maintenance, and efficient use. It is a system of components or subsystems that broadly follows the definition given by von Neumann and his associates [BurkA46]: "Inasmuch as the completed device will be a general-purpose computing machine it should contain certain main organs relating to arithmetic, memory-storage, control and connection to the human operator. It is intended that the machine be fully automatic in character. . . ."

While its name is computer, its role may involve little computing in the usual sense. Its function might be that of data processing, file management, communication control, or some other tasks that require acquisition, storage, retrieval, manipulation, and dissemination of large quantities of information. In any event its major subsystems involve memory (storage and retrieval), input/output (acquisition and dissemination), and processing (data manipulation) functions, and the interconnection and controls to permit the assemblage of components to function as a working system.

To broaden the subject, a *digital system* is an assemblage of components or subsystems designed to perform a set of related tasks under control of one or more digital processors. The digital computer itself is then a digital system since it is an assemblage of subsystems controlled by a digital processor. In fact, the maturing of the discipline of computer engineering and the introduction of a wide variety of standard semiconductor building blocks for computer subsystems allow many computer designs to result from careful selection from inventories of standard subsystems. Thus computer design can be similar to digital system design.

In all of these systems the major subsystem functions are processing, storage, and control of system interfaces. Processors contain their own local storage (registers), an *arithmetic and logic unit* that performs operations on data, paths for transferring information among registers, individual flip-flops for holding temporary control information, and control logic to force the processor to execute instructions as called for by a program. In many cases a part of the control logic itself is a program (a *microprogram*) residing in a fixed or modifiable memory within the processor or in a separate store.

The memory of a computer usually is distributed over several physical modules. As noted, there is memory within the processors. Since the high unit cost of storage in the

processors limits the amount of processor memory that is practical, a main working memory is assembled from lower-cost and lower-speed circuits. This *primary memory* holds active programs and their data. Primary memory is queried by the processor(s) to obtain instructions and data items and is the destination for the results of individual computations. The primary memory is supplemented by lower-unit-cost, but slower, secondary memory that in turn can be supplemented by even-lower-unit-cost archival memory. There might also be specialized memories that are used for buffers between processor registers and primary memory. There must be a mechanism to coordinate the operations of the various memories. For example, we might require a means to transfer "pages" or "segments" of information between primary and secondary memory.

Unless we can provide for communication with the external environment the activities of the computer or digital system will be of little use. This communication is accomplished by input/output controllers. These functional units control the flow of information between the primary memory and external devices, such as keyboards, printers, terminals, and the secondary and archival memory units previously described. Specialized forms of input/output processors might be useful for connection to long distance communications facilities or to real-time analog subsystems. Other specialized processors might be used to relieve the main (or central) processor of certain tasks such as memory management, mathematical operations, or control of special hardware.

This brief description of the major component subsystems of a computer outlines the scope of this book. As a frame of reference for study of the many alternatives available to the digital system architect, and for a consideration of the approach to designing a digital system, the reader should be familiar with some specific conventional digital computer (or even with a very unconventional computer system).

1.3 A BRIEF LOOK AT HISTORY

As seen in Fig. 1-1, machines for calculation go back at least to the seventeenth century (Blaise Pascal, Samuel Morland, and Gottfried Leibniz). Charles Babbage accomplished his monumental work on automatic computation during the 1800s, a century that also saw the mathematics of Boole and DeMorgan and the automatically controlled loom of Jacquard. Practical adding machines, bookkeeping machines, accumulating cash registers, and tabulating equipment were manufactured by the beginning of the twentieth century. In the 1930s digital computation (as opposed to analog computation where a machine directly reproduces the problem being solved, perhaps at a different speed) was accomplished with machines using electrical relays by Howard Aiken at Harvard University, George Stibitz at Bell Telephone Laboratories, and Konrad Zuse at the Technische Hochschule in Berlin. In the same time period John Atanasoff at Iowa State University developed a digital calculating machine using vacuum-tube circuits and Alan Turing presented his ideas on computing theory.

Following these origins of automatic computation, the ideas for the internally programmed digital computer generally are attributed to John von Neumann and his associates at Princeton University's Institute for Advanced Study, but must have been

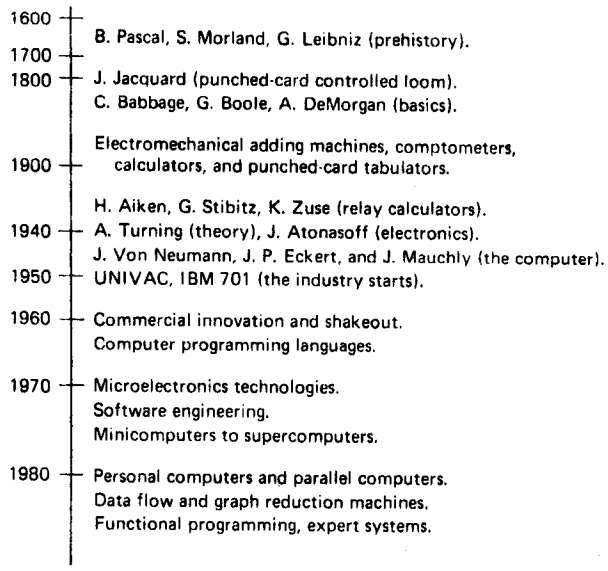

Figure 1-1 Looking back.

influenced considerably by the work of J. Presber Eckert and John Mauchly, who designed and built an electronic digital computer with externally wired programming, the ENIAC, at the University of Pennsylvania. These efforts were accomplished in the mid-1940s and were followed in the late 1940s and early 1950s by an explosion of activity toward development of a variety of digital computers at the institutions already mentioned, at MIT, at the Rand Corporation, at the National Bureau of Standards, at the University of Chicago and the University of Illinois, at the Los Alamos and Argonne National Laboratories, as well as at Cambridge, London, and Manchester universities and the National Physical Laboratory in Great Britain.

In the 1950s and early 1960s many corporations introduced electronic digital computers to follow the UNIVAC and the IBM 701 into the marketplace. In addition to Eckert-Mauchly Computer Corp. (later purchased by Remington Rand) and International Business Machines, these included Bendix (computer division bought by Control Data), Burroughs Adding Machines (later Burroughs Corp. and subsequently Unisys), Consolidated Engineering Corp. (Electrodata division sold to Burroughs), Control Computer Company (acquired by Honeywell), Control Data Corp., Digital Equipment Corp., Engineering Research Associates (with Eckert-Mauchly became Remington Rand UNIVAC, then Sperry Rand UNIVAC, then a part of Sperry Corp. [merged with Burroughs to form Unisys Corp.]), English Electric (merged with I.C.T. to form I.C.L.), Ferranti (merged with English Electric), General Electric (computer operation acquired by Honeywell), Honeywell (computer operations moved to a joint venture with Fujitsu and Cie. Bull), International Computers and Tabulators (became I.C.L.), Librascope (became a division of General Precision Equipment), Philco Corp., Radio Corporation of America (computer division acquired by UNIVAC, RCA later acquired by General Electric), Raytheon, and Scientific Data Systems (acquired by Xerox). The organizations have been as volatile as the products!