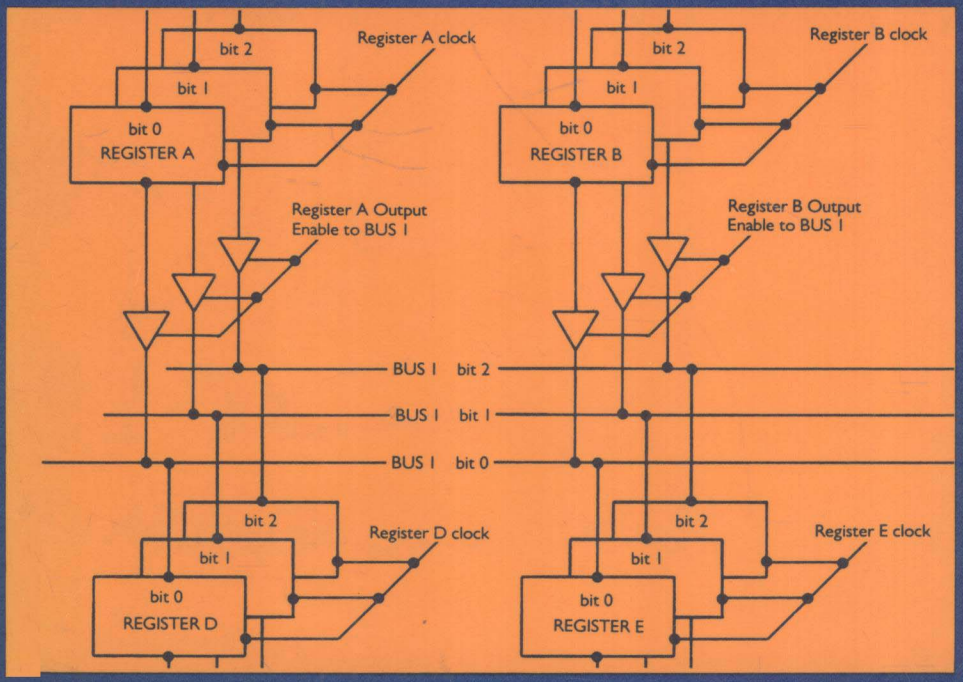
Arthur G. Lippiatt
Graham G. L. Wright

THE ARCHITECTURE OF SMALL COMPUTER SYSTEMS



Second Edition

The Architecture of Small Computer Systems

Second Edition

ARTHUR G. LIPPIATT
(Camelot Software and Systems; formerly The Hatfield Polytechnic)

and

GRAHAM G. L. WRIGHT
(The Polytechnic of Wales)

ENGLEWOOD CLIFFS, NEW JERSEY LONDON NEW DELHI RIO DE JANEIRO
SINGAPORE SYDNEY TOKYO TORONTO WELLINGTON

This edition first published in the United States of America, 1986

British Library Cataloguing in Publication Data

LIPPIATT, Arthur G.
Architecture of small computer systems—2nd ed.
1. Minicomputers
I. Title II. WRIGHT, G. G. L.
001.64'04 QA76.5
ISBN 0-13-044736-6 (p)

Library of Congress Cataloging in Publication Data

LIPPIATT, Arthur G., 1936–
The architecture of small computer systems.

Bibliography: p.
Includes index.
1. Computer architecture. 2. Microcomputers.
3. Minicomputers. I. WRIGHT, Graham G. L., 1936– II. Title.
QA76.9.A73L56 1985 621.3819'582 85–3490
ISBN 0-13-044736-6 (p)

ISBN 0-13-044736-6

PRENTICE-HALL INC, *Englewood Cliffs, New Jersey*
PRENTICE-HALL INTERNATIONAL UK LTD, *London*
PRENTICE-HALL OF AUSTRALIA PTY LTD, *Sydney*
PRENTICE-HALL CANADA INC, *Toronto*
PRENTICE-HALL HISPANO AMERICANA SA, *Mexico*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN INC, *Tokyo*
PRENTICE-HALL OF SOUTHEAST ASIA PTE LTD, *Singapore*
PRENTICE-HALL DO BRASIL LTDA, *Rio de Janeiro*
WHITEHALL BOOKS LIMITED, *Wellington, New Zealand*

Typeset by HBM Typesetting Ltd, Chorley, Lancashire

Printed in Great Britain by A. Wheaton & Co Ltd, Exeter

10 9 8 7 6 5 4 3 2 1

The Architecture of Small Computer Systems

To: *Karen*
Kevin
Sonia
David
Richard
Adrian
Nicola
Esta

A.G.L.

To the memory of my mother – triste ministerium.

G.G.L.W.

Preface to the First Edition

The two major aims of this book are:

(1) To explain to computer science students what the hardware in a computer system does. Many computer scientists find difficulty in distinguishing what the electronic subsystems in a computer can do from what a program can make them do.

(2) To explain to engineers what a computer system does. The advent of the microprocessor and microcomputer has fairly rapidly brought the field of computer systems into the sphere of interest of many engineers who have previously had nothing to do with computers excepting perhaps as a tool in the design process. For them an understanding of the overall system is important. Experience in teaching engineers indicates that they find little difficulty in learning the new digital electronic techniques, many of which they probably have met in any case, but they do have difficulty in finding suitable reading to learn about the wider picture of the overall system.

The aim has been to satisfy these requirements in a volume small enough not to discourage the potential reader by its sheer bulk or its high price. Of necessity much detail has had to be omitted. As a result, however, the system's functions can be described without confusing the reader with details of minor alternatives and the minutiae of implementation techniques. It is hoped that the professional engineer wishing to know something of computer systems will find here a sound first text, and that the student of computer science or of electrical engineering will find sufficient material for an introductory course.

Computer architecture covers that area of knowledge and expertise which is equally of interest to the computer scientist and to the electrical engineer. It is really the common ground which both the hardware specialist and the software specialist should understand. These groups of people are increasingly being required to work together on projects in the design, production, testing and commissioning of equipment. Without a sound understanding of the architecture of computer systems, intelligent interaction between them, and an appreciation by each of the other's problems, becomes impossible.

An examination of the features of some commercially available computer and microprocessor systems leaves the impression that in some cases the engineers who designed them were not as aware as they might have been of how the software designers would wish to use the facilities provided. Consequently those facilities are not very easy or convenient to use. Equally it is clear from conversations with software specialists that it is at times difficult for them to appreciate what the hardware subsystems are really capable of achieving.

It was, of course, difficult to know what topics to include in the text and much more difficult to know which topics to omit. It was decided therefore to keep in mind the needs of the groups of people mentioned above, to aim at an introductory text, to limit the material to that which one may expect a student to cover in a one-year university or college course (when supplemented by suitable practical work), and not to cover material which is easily available from many other books. Numbering Systems have been omitted for that reason.

The principles explained in the text are related to the realities of the world of commercial computer systems in an appendix which contains a summary of the specifications of some popular small computer systems. The terminology used in the appendix is that of the rest of the text and may not correspond with that of the particular computer system manufacturer. It is important to relate principles and practice, to help the reader to understand the nature of the different compromises reached by various design teams in applying those priciples to real system designs.

Technical terms have been *italicized* and defined at their first appearance in the text. It is very easy for a specialist to use terminology which is either not understood by the newcomer to the subject or, much worse, which is misunderstood because the words used are familiar to the reader in another context in which their meaning is rather different. This often seems to be the case in the spheres of hardware and software.

The book is arranged in seven chapters, the first giving a superficial treatment of topics covered in greater detail later on, and introducing the basic terminology of the subject. The second chapter explains the type of instructions which a computer can obey, a topic which is the cause of some considerable difficulty to the newcomer, and gives some idea of the way in which those instructions are used. Chapter 3 tackles another area of computer architecture which provides students with some difficulty—the way in which information is coded into a computer system. A number of topics covered in this chapter are often treated as separate topics. They are covered together to emphasize that they are all to do with information coding and to help to dispel the misconception that they are hardly, if at all, related. Chapter 4 is a section on standard memory addressing techniques. The chapter is designed to help the engineer to understand why the programmer wishes to use these techniques and also to explain to the programmer the functions which the hardware performs to produce the different addressing

modes. Chapters 5 and 6 cover the important topic of the Input/Output subsystem. Chapter 5 is devoted to an explanation of how data transfers are carried out in a computer system. This chapter will present no problem to the electrical engineer, but the computer science student may well find it rather difficult. Chapter 6 presents both the hardware and software aspects of interrupt handling. Chapter 7 considers computer arithmetic (not binary arithmetic) and the effects it has on the architecture of the system, particularly its effect on the instruction set. It includes discussions of floating point and BCD arithmetic. A number of uses of the condition codes (flags) are also considered in this chapter.

Preface to the Second Edition

In writing the second edition of this book, the joint authors have endeavored to adhere to the stated claims of the first edition and to update the text in accord with currently-used systems.

The first three chapters and chapter 7 are substantially unchanged. Chapter 4 has been extended to include additional addressing modes and a discussion of addressing data on the stack.

Chapters 5 and 6 have been rewritten and extended. The I/O examples have been brought up to date by dropping the paper tape and teletype examples. Chapter 5 now includes an explanation of the workings of serial I/O. The approach to interrupts in chapter 6 has been changed to emphasize the need for interrupt-driven systems to implement realtime systems, and the section on DMA has been extended.

A new chapter 8 has been added, dealing with memory management techniques. The new microcomputers can address many megabytes of memory, and the management of that memory is now so fundamental that a discussion of memory management should not be left to more advanced texts.

The appendix has been modified in two ways; firstly, it contains architectural information on the more recent microprocessors and superminis, and secondly the systems included are those which are the most popular in their market sector, so that a reader may have the greatest chance of finding a system to which he or she has access described therein. The bibliography has been updated, and specific references to the systems described in the appendix have been added.

Acknowledgments

We wish to thank those readers who have written to point out errors in the first edition. Those errors have been rectified in this edition.

AGL would like to thank Pat Brazier who typed most of the second edition and provided constant encouragement to complete the work involved.

GGLW would like to thank Colin Bowring for his help in suggesting sources and material used in the appendix.

We would both like to record our appreciation of the perseverance of Giles Wright of Prentice-Hall International in encouraging and enabling the production of the second edition as a joint exercise between the two authors.

Contents

CHAPTER 1

Introduction to a Computer System

1.1 INTRODUCTION

To describe a computer system in a few words is impossible. Computer systems are intricate arrangements of sophisticated components, and it takes a lot of time and work with computers to generate a thorough understanding of them. Computer systems come in a large variety of shapes, sizes, capabilities and specialities. They range from small and relatively simple microcomputers to large and very complex mainframes, with a price range from about a hundred to a few million dollars. They have a large number of normal and special uses. This book deals particularly with small computer systems, at the lower end of the price and performance range occupied by the various types of microprocessors, microcomputers and minicomputers. Small computer systems are, by virtue of their lower cost, very widely used and available, particularly in schools and colleges; and they are also correspondingly accessible to the student. Larger computers are, on the other hand, fewer in number and increasingly remote and unapproachable, locked inside securely controlled computer rooms.

This book attempts to explain the fundamental principles of the architecture – the internal organization – of small computer systems, without generating confusion by explaining the many variations possible. Reference is made to real computer systems to show how the principles are applied in practice, and also to show some of the variations encountered commercially. Appendix 1 contains a more detailed description of some common computer systems within the scope of this book, among which the reader is very likely to find a machine, or its processor, to which he or she has access.

A large proportion of the book is also relevant to larger computers, but as might be expected, the range of variations and complexities are correspondingly greater. Also, for reasons which will be explained later in the book, the architecture of large computers that is evident to the student is often not their true architecture, and it is often very difficult to distinguish reality from appearance.

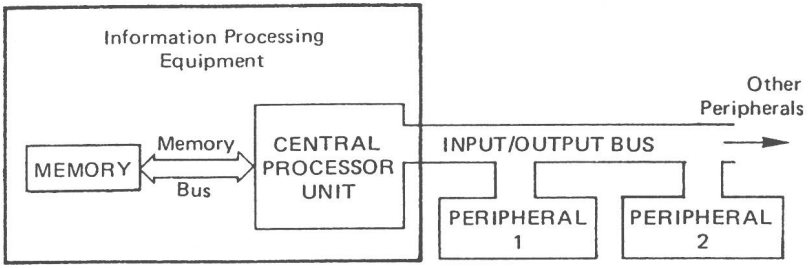


Figure 1.1 *Computer system block diagram*

What then is a computer system? Its major characteristics are:

1. It is an organized set of electronic equipment used to *process* information;
2. It is provided with information for processing by a range of *input* devices, such as keyboards, handsets, punched card readers, magnetic disk and tape drives, thermometers, pressure gauges, voltmeters, counters, weighing machines, speech analyzers, cash registers, telephone lines, timers, character and bar code readers and many others;
3. After it has processed the input information, it transmits the results of processing to one or more *output* devices of which there is again a large variety, such as CRT displays, TV monitors, magnetic tape and disk drives, valves, printing machines such as line printers, matrix printers, typesetters, typewriters and ticket printers, telephone lines, electric motors and other actuators, buzzers, speech amplifiers, traffic lights and many others.

A computer therefore consists of some electronic information processing equipment, a number of information supplying devices—the input devices—and a number of information receiving devices—the output devices. Their arrangement into a computer system is illustrated in Figures 1.1 and 1.2. The input devices and output devices are physically connected to the information processing equipment by a connection known as an

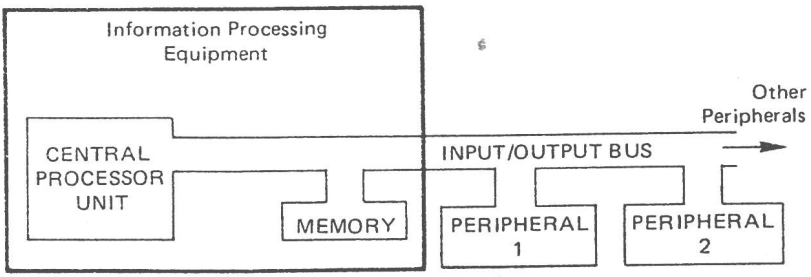


Figure 1.2 *Alternative system block diagram*